# Classification of Lymphoma, Benign Lesions, and Carcinoma Using Convolutional Neural Network

Hanina Nuralifa Zahra[1(✉)], Isa Anshori[1], Hasna Nadila[1], Hofifa Mulya Utami[1], Joshua Adi Chandra[1], Muhammad Rashid Kurniawan[1], Yunianti Khotimah[1], Widyawardana Adiprawita[1], Hermin Aminah Usman[2], and Okky Husain[1]

[1] Biomedical Engineering, Bandung Institute of Technology, Bandung, Indonesia
haninanz@gmail.com

[2] Pathological Anatomy, Padjadjaran University, Bandung, Indonesia

**Abstract.** Lymphoma, carcinoma, and benign lesions are common diseases that have to go through several stages to be detected due to their structural similarity. A common way to distinguish these diseases is by analysing them manually based on the cell images at a certain magnification. However, this method still has many shortcomings in terms of accuracy as it is vulnerable to possible human error and requires quite a lot of time. Thus, an alternative faster and more accurate method of detection should be developed to increase patients' chances of survival. One solution to overcome this problem is by using a deep learning model which mimics the behaviour of nerve cells in the human brain and has proven to be able to classify certain diseases in many studies. As there are many existing deep learning designs, this paper aims to explore the methods of detecting these diseases and find the best performance (highest accuracy) among them. With the microscope images data (magnifications of 100 and 400 times) provided by the medical faculty of the University of Padjadjaran (UNPAD), we investigated the classification result using different kinds of deep learning designs which were our designed CNNs and transfer learning using the Inception-V3, VGG16, and MobileNet. It was found that the best model used to classify images with a magnification of 100x is MobileNet (accuracy of 53% for benign lesions and 47% for lymphoma) and designed CNN (accuracy of 59% for benign lesions and 41% for lymphoma). While for image classification with a magnification of 400x, Inception-V3 showed the best result (accuracy of 80% for carcinoma and 50% for lymphoma).

**Keywords:** CNN · MobileNet · Inception · VGG16

## 1 Introduction

Cancer, in general, is a term for a large group of diseases that can appear in almost any organ/tissue in the body where the number of abnormal cells grows uncontrollably, exceeds the normal number, and can attack parts that are still connected to the affected organ or tissue or even spread to the organ [1]. According to WHO, cancer is the second


A. Lelono et al. (Eds.): ICOLIB 2021, ABSR 27, pp. 175–192, 2023.
https://doi.org/10.2991/978-94-6463-062-6_18

leading cause of death worldwide. It was reported that in 2018, nearly 9.6 million people died from cancer, and 1 in 6 deaths that occurred in the world were caused by this disease [2]. The type of cancer varies depending on where these abnormal cells are found. In this study, we focus on carcinoma, benign lesions, and non-Hodgkin lymphoma. Lymphoma, for short, is a type of cancer that develops specifically in lymphocyte cells [2]. Based on GLOBOCAN (IARC) data in 2018, lymphoma is one of the ten most common cancers worldwide [3]. From cancer statistics obtained from Jemal et al., as many as 115,000 people worldwide had lymphoma in 2008 [4]. Meanwhile, data from the Ministry of Health of the Republic of Indonesia showed that lymphoma prevalence in 2013 reached 0.06% [5]. Although the 5-year survival rate of lymphoma is generally passable with 72% based on data obtained from the American Cancer Society [7], its rapid development and dangerous symptoms make lymphoma patients need prompt and precise treatment and early detection.

There are common ways to detect lymphoma, carcinoma, and benign lesions, such as biopsy, blood tests, bone marrow aspiration, and scans (X-rays, CT scans, MRI, ultrasound, and PET scans) [8]. Biopsy tests include core needle biopsy (CNB) using a needle to take tissue samples and surgical excisional biopsy (SEB) by surgery to take tissue samples, which are then analysed manually by paying attention to the visual shape of the cells. If needed, genetic testing is also conducted. However, CNB has a lower diagnostic precision and a lower ability to diagnose lymphoma transformation, and in general, visual cell observations can increase errors because it depends on the examiner's condition [9]. From the results of this biopsy, if the results are positive, the diagnosis will be continued with immunohistochemistry (CPI) by utilizing the bond between antigen and antibody to detect more specifically. The next approach is a blood test. A complete blood count (HDL) is taken to help the diagnosis process. Some results from HDL can be used as supporting information for the diagnosis, such as lactic dehydrogenase (LDH), which will increase if you have lymphoma, then there is a sedimentation rate from red blood cells in the test tube to find out how much inflammation is happening and if you have lymphoma this rate will be higher than normal values. The drawback of this blood test is that the results do not specifically determine lymphoma because several other diseases have increased LDH and sedimentation rates [10]. Then there are other diagnoses by scanning using CT, MRI, PET Scan that can be used to determine the stage of the disease spread [11].

Currently, research has been carried out to assist experts in detecting lymphoma using machine learning. The development of whole slide imaging (WSI) enables deep learning, a branch of machine learning, to assist the histopathological process [6, 12]. Deep learning consists of several layers called 'neurons' and is connected to form an artificial neural network. The number of layers that make up deep learning architecture affects how many features are extracted from an image and how complex the pattern is learned by machines [6]. The convolutional neural network (CNN) is a branch of deep learning that is commonly used in the medical world is the convolutional neural network (CNN). The implementation of CNN for pathology computation purposes includes tumor classification and segmentation, mutation classification, and disease diagnosis [13]. However,

building a CNN model from scratch requires a long process and goes through a large set of hyper-parameters. Transfer learning, a method using a trained model for new data, was also tested for lymphoma detection. However, the limited dataset to validate the algorithms is an obstacle to using transfer learning and deep learning in general in the lymphoma detection process [13, 14].

In this study, we will investigate how different machine learning model can artificially improve the performance of deep learning (DL) in detecting lymphoma, benign lesion, and carcinoma. We use the "Data Augmentation" method by rotating the image, cropping the image, and reversing the image to increase and variate the number of images. Furthermore, the Convolutional Neural Network (CNN) model that we built from scratch will be applied to classify images. We identify which machine learning model will best classify lymphoma, benign lesion, and carcinoma.

## 2 Materials and Methods

### 2.1 Materials

The dataset used in this study is provided by Universitas Padjadjaran's Faculty of Medicine (2020). It consisted of 19 cases with 468 images (1376 × 1038 pixels) and are divided into three classes; carcinoma (98 images), benign lesions (97 images), and lymphoma (235 images). Based on the magnification used when the images were taken, they are further divided into 100x and 400x magnification images. Figure 1 shows the sample images for each class in 400x magnification.

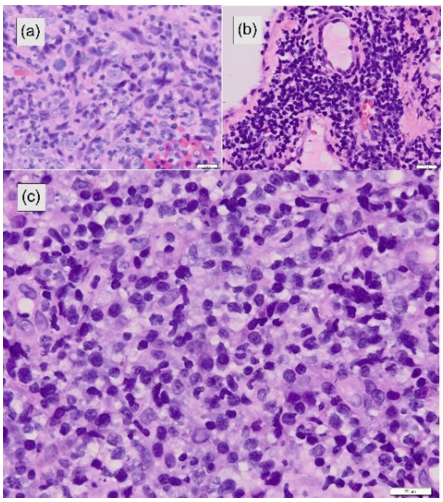


**Fig. 1.** Sample Images of (a) carcinoma, (b) benign lesions, (c) lymphoma.

### 2.2 Methods

In this study, we adopted methods proposed by Achi [6] and Upeka [14]. For Achi, the method used to classify lymphoma cells is CNN that is built from scratch. Meanwhile, for Upeka, the method used is transfer learning using the Alexnet model. Upeka's proposed method also involved pre-processing and data augmentation before classification. This study compared several pre-processing, data augmentation, and classification methods to obtain various deep learning performances on detecting the diseases in different setups (Fig. 2).

### 2.3 Data Balancing

As shown in Table 1, the data amount to each class differed quite significantly. Before model training could take place, we need to balance the number of images for each class from all the available data so that the model results would not be bias to a certain class.

In this data balancing process, we created "new data" to balance the number of images so that each class had the same amount of data. These new images were created by cropping existing images randomly while making sure that they did not lose the necessary information. For the data at 100x magnification, we cropped the images to 700 $\times$ 700 pixels beforehand to increase the number of images, mainly for carcinoma and benign lesions. With this method, there would be some of the original images of carcinoma and benign lesions labels that were cropped twice. Such an approach results in an increasing number of carcinoma images from 48 to 70, benign lesions images from 37 to 70, and lymphoma images from 69 to 70. Then, we separate the data into train data (53 images) and test data (17 images) (Table 2).

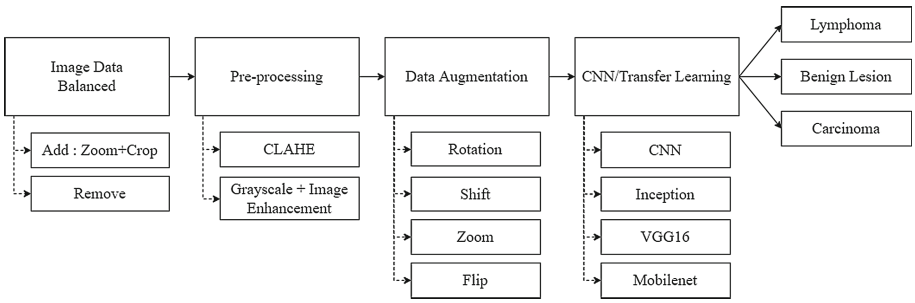


**Fig. 2.** Methodology diagram.

**Table 1.** The amount of data before being balanced

| Data | Carcinoma (images) | Benign lesion (images) | Lymphoma (images) |
|---|---|---|---|
| Total | 98 | 97 | 235 |
| 100x | 48 | 37 | 69 |
| 400x | 50 | 60 | 166 |

**Table 2.** The amount of data at 100x magnification after balancing

| **Data** | 100x | | | 400x | | |
|---|---|---|---|---|---|---|
| | **A** | **B** | **C** | **A** | **B** | **C** |
| Train | 53 | 53 | 53 | 40 | 40 | 40 |
| Test | 17 | 17 | 17 | 10 | 10 | 0 |
| Total | 70 | 70 | 70 | 50 | 50 | 50 |

A: Carcinoma (images)
B: Benign Lesion (images)
C: Lymphoma (images)

For the data at 400x magnification, as the images show more detailed features than in 100x magnification, we only cropped the image to 1000 × 1000 resolution to avoid losing important information. To balance the number of images, we reduce the number of images for benign lesions and lymphoma labels as it is known that the images were taken multiple times for one case (one patient). With this method, we tried to cut down the data amount in benign lesions and lymphoma labels to make sure they have the same number of carcinoma-labeled images, which is 50 images; as seen in Table 4, the data decreased from 60 to 50 for benign lesions and from 166 to 50 for lymphoma. Then, we separate the data into train data (40 images) and test data (10 images).

### 2.4 Pre-processing

Histogram equalization is a method to adjust image intensity by distributing the histogram value to the optimal range [15]. CLAHE is a part of adaptive histogram equalization in which the contrast amplification is limited by clipping histogram at a predefined value called 'clip limit' [16]. This study used 0.2 for the clip limit value, and grid size of 20 × 20 as the common parameters used to enhance the contrast image. At this pre-processing stage, an increase in contrast and linewidth is carried out in the image. The purpose of increasing the contrast is to make the data in the form of the cell image more clearly visible, and we also increase the line width to make the cell shape clearer. We increase the contrast value with the maximum and minimum value comparison system in the image then multiply by 255 as the total value as shown in the Eq. (1) to increase the line width, we use a 3x3 kernel and iterate one time.

$$\text{Contrast} = \frac{(\text{image} - \min)}{(\max - \min)} \times 255 \tag{1}$$

### 2.5 Data Augmentation

Data augmentation can be defined as producing a larger set of images than an existing train image set. Data augmentation is widely used to solve overfitting problems caused by the small amount of data for training or data amount imbalance in each class [16]. Data augmentation consists of simple transformations such as flipping, colour space

**Table 3.** Data augmentation parameter values

| Augmentation Method | Parameter of Augmentation |
|---|---|
| Rotation | 45, 90, 180 |
| Width Shift | 0.1, 0.2, 0.3 |
| Height Shift | 0.1, 0.2, 0.3 |
| Zoom | 0.1, 0.2, 0.3 |
| Horizontal Flip | True |
| Vertical Flip | True |
| Fill mode | Reflect |

augmentations, and cropping [17]. In this study, several augmentation methods are utilized all at once to obtain more data for training as much as possible. The methods used for augmentation and the values can be seen in Table 3.

We used the module ImageDataGenerator from Keras to implement the data augmentation. The values written in Table 3 are the possible values for the most optimal augmentation parameter combination for future training, chosen randomly. To determine the most optimal combination, we decided to use grid search method. This was done by trying all the possible combinations and observed both the training and validation accuracy of the model for each combination. Since there are four parameters to be optimized and each has three values, the total training time required is 162, which is quite an arduous task. To fasten the process and ease the computation, we implement the callback EarlyStopping from Keras to stop the training once it has reached a certain point.

The evaluation of the parameter search was executed by using our designed CNN model as in the next subsection. The model was set up with almost the same settings as in the subsection II.D; with the addition of callback during training, a bigger batch size of 32, a smaller dropout of 0.2, and fewer epochs used. We used 200 epochs for this experiment since from our initial tests, 200 epochs are the minimum epochs to obtain excellent accuracy for several setups whereas a bigger batch size could help make the training much faster. As for the callback, we set it to monitor the training accuracy with min_delta of 0.01 and patience of 10, which means the training will be stopped once there is no training accuracy improvement of a minimal 0.01 after 10 epochs.

### 2.6 Designed Convolutional Neural Network (CNN)

The next step is to classify the data using the Convolutional Neural Network (CNN). CNN in general is a class of Artificial Neural Network (ANN) designed to introduce hierarchical and generally consist of a convolutional layer, a pooling layer, and a fully

**Table 4.** Designed CNN setup

| Image Magnification | Hyperparameters | Value |
|---|---|---|
| 100x | Input Shape | (224, 224, 3) |
| | Optimizer | Adam |
| | Learning Rate | 0.0001 |
| | Batch Size | 16 |
| | Epochs | 400 |
| | Dropout | 0.3 |
| 400x | Input Shape | (224, 224, 3) |
| | Optimizer | Adam |
| | Learning Rate | 0.0005 |
| | Batch Size | 16 |
| | Epochs | 400 |
| | Dropout | 0.3 |

connected layer using a backpropagation algorithm [17]. Convolutional and pooling layers are tasked with extracting features while fully connected layers map the final result, which will provide the expected data classification [18]. CNN is well-known for its good performance in image classification and has been widely used for that purpose; hence it is reasonable to utilize it in this study. Generally, CNN consists of several convolutional layers and pooling layers, followed by one flatten layer and some fully connected layers [19]. We used two kinds of CNN: our designed CNN and transfer learning CNN. Our designed CNN consists of 12 layers in general, with 4 convolutional and pooling layers, followed by a flatten layer and two dense layers with a dropout layer sandwiched between them. We adopt a similar CNN model to Achi et al. [6] with changes in the number of layers, kernel size, and filters of the convolutional layers. Adding more convolutional layers could help maintain spatial information of the images while keeping dense layers minimum is advised to reduce training parameters. Aside from that, adding a dropout layer to the model might help prevent overfitting [20]. We designed our CNN with these ideas in mind. There are slight differences in setting up the model for images with 100x and 400x magnification. The details regarding the model setup will be provided in Table 4. We discovered that the CNN model's performance for both image magnifications was quite different from each other, with the model for 100x image magnification performed significantly better than the other. This discrepancy was due to the 400x magnification model not learning well with the learning rate of 0.0001; hence we used a slightly larger learning rate in this experiment for the model. As for the input shape, we adjusted it accordingly to VGG16 since the transfer learning model could only accept input of (224, 224, 3).

**Table 5.** Inception, MobileNet, VGG setup

| Image Magnification | Hyperparameters | Value |
|---|---|---|
| 100x | Input Shape | (224, 224, 3) |
| | Optimizer | Adam |
| | Learning Rate | 0.0001 |
| | Batch Size | 16 |
| | Epochs | 35 |
| | Dropout | 0.3 |
| 400x | Input Shape | (224, 224, 3) |
| | Optimizer | Adam |
| | Learning Rate | 0.0005 |
| | Batch Size | 16 |
| | Epochs | 35 |
| | Dropout | 0.3 |

### 2.7 Transfer Learning

In the transfer learning section, we divided the results into Inception, VGG16, MobileNet. These three transfer learning models were chosen for their excellent performance so far in recognizing images. For Inception, we used the third version Inception-V3 which is one of the transfer learning models that is used for image recognition developed by TensorFlow. All transfer learning models in this study were implemented in a similar way; we took the feature extraction (convolution and pooling) layers provided by Keras then appended two dense and one dropout layers to them, similar to our designed CNN's architecture in general. The details regarding the models' setup will be provided in the Table 5. We set it similar to the CNN model that we have built (except for the number of epochs) as we will compare and analyze all the results.

## 3 Results and Discussion

### 3.1 Pre-processing

The pre-processing experiment was evaluated using the CNN methods that we proposed by observing which image format would help the CNN in achieving best result. In this initial experiment, we used RMSprop for optimizer and set the training to take place for 200 epochs. As shown in Table 6, the best accuracy for 100x magnification images is achieved by using the RGB method, with training and validation accuracy 95.6% and 78.43% respectively. Meanwhile, for 400x magnification images, RGB with CLAHE provides the 5.72 cm best result. However, we could see that both of them experienced overfitting (Fig. 3).

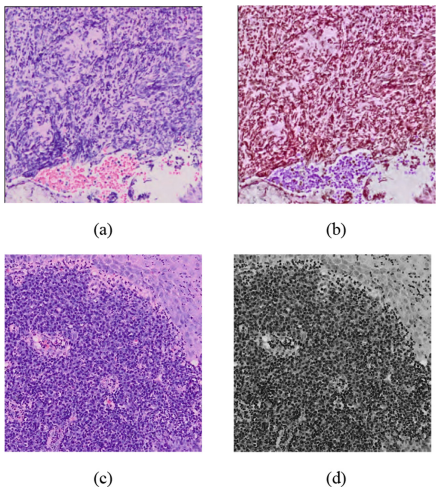


**Fig. 3.** (a) Carcinoma image before we applied CLAHE, (b) after CLAHE, (c) Carcinoma image before grayscale and image enhancement, and (d) after grayscale and image enhancement pre-processing.

**Table 6.** Pre-processing performances

| Magnification | Condition | Accuracy | Validation Accuracy |
|---|---|---|---|
| 100x | RGB | 0.9560 | 0.7843 |
| | RGB + CLAHE | 0.7987 | 0.7647 |
| | Grayscale + image enhancement | 0,7248 | 0,6863 |
| 400x | RGB | 0.9250 | 0.7667 |
| | RGB + CLAHE | 0.9583 | 0.8333 |
| | Grayscale + image enhancement | 0,7583 | 0,7000 |

### 3.2 Data Augmentation

The results for the experiment of data augmentation can be seen in Tables 7 and 8. We differentiate the experiment into two different trials: grid search for the 100x magnification images and the 400x magnification images. The data provided in the tables are only the best 10 data we obtained from grid search.

The best parameter combinations from this experiment were then used for the next step of classification. We chose these combinations based on the obtained training and validation accuracy. We examined how well the model performed during training and validation by averaging the accuracies obtained during training and checking the highest accuracy it managed to achieve.

**Table 7.** Best 10 data augmentation parameter combinations for 100x magnification images

| No | Augmentation Parameter | Average Accuracy | | Max Accuracy | |
|---|---|---|---|---|---|
| | | Training | Validation | Training | Validation |
| 1 | (90, 0.2, 0.3, 0.2) | 0,503 | 0,532 | 0,704 | 0,705 |
| 2 | (45, 0.1, 0.2, 0.1) | 0,494 | 0,538 | 0,667 | 0,706 |
| 3 | (90, 0.1, 0.3, 0.2) | 0,479 | 0,521 | 0,635 | 0,686 |
| 4 | (90, 0.3, 0.1, 0.3) | 0,472 | 0,519 | 0,622 | 0,686 |
| 5 | (180, 0.1, 0.1, 0.1) | 0,486 | 0,509 | 0,597 | 0,725 |
| 6 | (180, 0.2, 0.3, 0.3) | 0,449 | 0,498 | 0,591 | 0,647 |
| 7 | (90, 0.3, 0.2, 0.1) | 0,471 | 0,497 | 0,616 | 0,667 |
| 8 | (45, 0.2, 0.3, 0.1) | 0,378 | 0,492 | 0,635 | 0,627 |
| 9 | (180, 0.1, 0.2, 0.1) | 0,463 | 0,492 | 0,603 | 0,686 |
| 10 | (180, 0.3, 0.2, 0.2) | 0,460 | 0,483 | 0,610 | 0,647 |

**Table 8.** Best 10 data augmentation parameter combinations for 400x magnification images

| No | Augmentation Parameter | Average Accuracy | | Max Accuracy | |
|---|---|---|---|---|---|
| | | Training | Validation | Training | Validation |
| 1 | (90, 0.1, 0.1, 0.2) | 0,556 | 0,538 | 0,708 | 0,633 |
| 2 | (45, 0.2, 0.1, 0.1) | 0,544 | 0,537 | 0,692 | 0,633 |
| 3 | (180, 0.1, 0.1, 0.3) | 0,554 | 0,537 | 0,683 | 0,633 |
| 4 | (45, 0.3, 0.1, 0.1) | 0,545 | 0,533 | 0,700 | 0,633 |
| 5 | (180, 0.3, 0.1, 0.1) | 0,529 | 0,517 | 0,675 | 0,633 |
| 6 | (45, 0.3, 0.2, 0.1) | 0,514 | 0,515 | 0,683 | 0,633 |
| 7 | (180, 0.2, 0.1, 0.2) | 0,525 | 0,511 | 0,675 | 0,600 |
| 8 | (180, 0.2, 0.1, 0.3) | 0,523 | 0,508 | 0,700 | 0,633 |
| 9 | (180, 0.1, 0.1, 0.1) | 0,500 | 0,500 | 0,642 | 0,633 |
| 10 | (180, 0.2, 0.1, 0.1) | 0,490 | 0,500 | 0,642 | 0,633 |

### 3.3 Classification

The result for classification using our designed CNN is shown in Fig. 4 and 5. As we can see in the Figs. 4 and 5, both the training accuracy and loss graphs provide considerably satisfactory results. On the other hand, the validation graphs did not show the exact same behaviour; both accuracy and loss increased and decreased respectively within around 100 epochs but further than that there is not much improvement. We can infer from this that the model suffered from overfitting as the validation accuracy and loss showed a stagnant trend in contrast to the increasing/decreasing training accuracy/loss

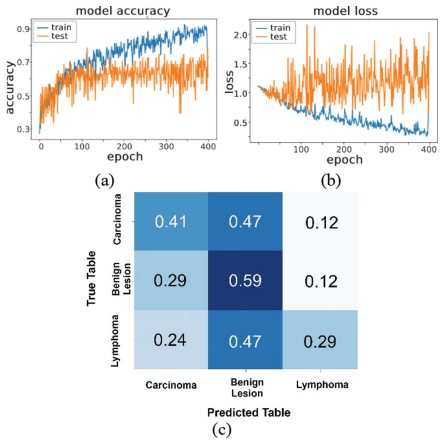


**Fig. 4.** Model performances for designed CNN on 100x images; (a) accuracy over epoch graph, (b) loss over epoch graph, and (c) confusion matrix.

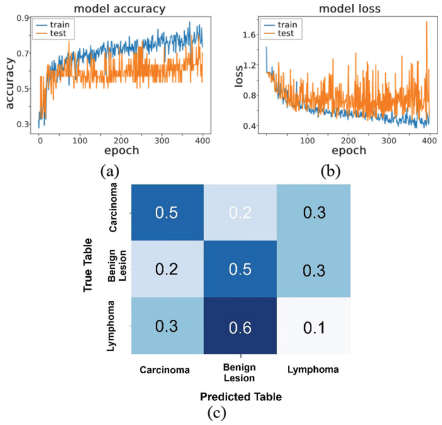


**Fig. 5.** Model performances for designed CNN on 400x images; (a) accuracy over epoch graph, (b) loss over epoch graph, and (c) confusion matrix.

respectively. The model's performance as shown in both confusion matrix also signifies that the model was bias to the benign lesion class to a certain extent. Although the correct prediction of carcinoma class was only a slightly bit lower compared to the benign lesion, the CNN model failed to classify images belonging to the lymphoma class.

Figures 6 and 7 show the results of the training performance (model accuracy and loss) and the confusion matrix of the Inception model for magnification of 100x and 400x. The model performance in Fig. 6(a, b) and Fig. 7(a, b) show that the training accuracy and loss are significantly higher than the validation performance. As we have mentioned above, this indicates that the model suffered from overfitting. To further analyse the model's performance, we can see the confusion matrices of both models (100x and 400x) in Figs. 6(c) and 7(c). The Inception model for 100x magnification images performed

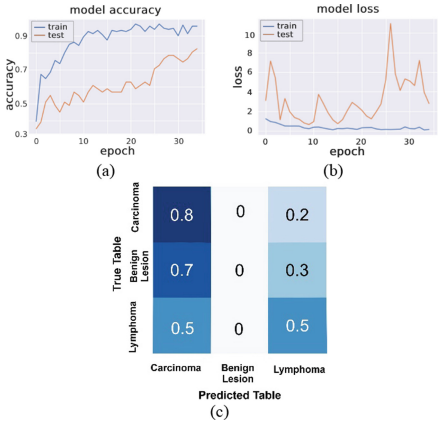


**Fig. 6.** Model performances for Inception on 100x images; (a) accuracy over epoch graph, (b) loss over epoch graph, and (c) confusion matrix.

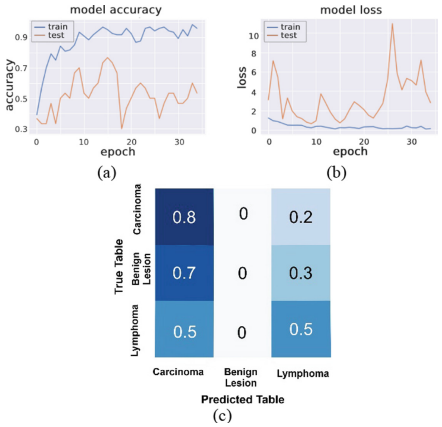


**Fig. 7.** Model performances for Inception on 400x images; (a) accuracy over epoch graph, (b) loss over epoch graph, and (c) confusion matrix.

quite well in classifying carcinoma images with 41% accuracy, although it did not show the same performance in the other classes. On the other hand, the model for 400x magnification images managed to obtain 80% and 50% accuracy in classifying carcinoma and lymphoma images, respectively, in which 80% accuracy for one class is the highest we have so far. However, despite its good performance in classifying the other two classes, the model failed to classify any benign lesion images, as seen in the confusion matrix.

The performance of VGG16 models for both 100x and 400x magnification images can be seen in Figs. 8 and 9, respectively. As we can see from Fig. 8(a, b) and Fig. 9(a, b), VGG16 did not show significant overfitting compared to the previous two models. Judging from the confusion matrices of both VGG16 models, we can say that the 400x

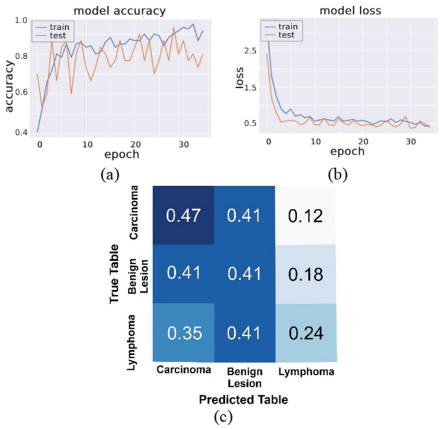


**Fig. 8.** Model performances for VGG16 on 100x images; (a) accuracy over epoch graph, (b) loss over epoch graph, and (c) confusion matrix.

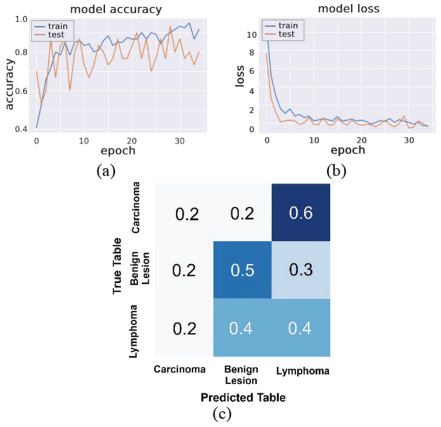


**Fig. 9.** Model performances for VGG16 on 400x images; (a) accuracy over epoch graph, (b) loss over epoch graph, and (c) confusion matrix.

model performed slightly better than the other. As we can see, the 100x model obtained 47% accuracy on classifying carcinoma images, which can be considered quite good, but in the other two classes, its performance was subpar. Although the 400x model failed in classifying carcinoma images, it obtained better results in classifying both lymphoma and benign lesions images.

Based on previous studies that also used the VGG16 model [22], reducing data imbalance does not necessarily improve the classification performance. It also needs proper fine tuning. The VGG16 model in this study however used the same setup as the other models, so that it did not specifically tune after the data balancing. Furthermore, in other studies [22]-[23] the VGG16 model showed better accuracy using relatively large training data compared to the data used in this study which could cause overfitting.

The two Figs. 10 and 11 show the overall performance of the MobileNet models for both magnifications 100x and 400x. From the training performance, we can safely say that in our case, MobileNet performed quite similarly to VGG16; the graphs did not show any significant sign of overfitting. As for the performance in each class, we can see from the confusion matrix in Fig. 10(c) that the model performed quite well in classifying both benign lesion and lymphoma images, as it obtained 53% and 47% accuracy on each class, respectively. On the other hand, the model for 400x images did not manage to classify carcinoma and benign lesion images well, though the accuracy obtained for classifying the lymphoma images was perfect, as the other models performed rather poorly on this class compared to this model.

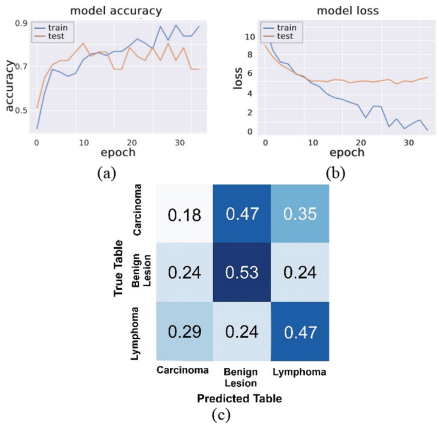


**Fig. 10.** Model performances for MobileNet on 100x images; (a) accuracy over epoch graph, (b) loss over epoch graph, and (c) confusion matrix.

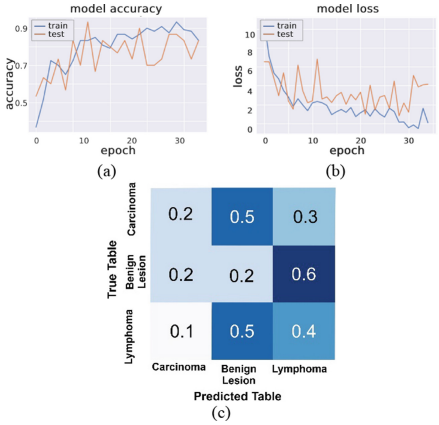


**Fig. 11.** Model performances for MobileNet on 400x images; (a) accuracy over epoch graph, (b) loss over epoch graph, and (c) confusion matrix.

The classification of lymphoma, carcinoma and benign lesion images has yielded various results using various models as well; our own CNN, Inception-V3, VGG16, and MobileNet. Interestingly enough, as shown in all confusion matrices, the results didn't seem to share much similarities to each other despite the same setup all models used except for the number of epochs. This discrepancy might be caused by the feature extraction layers of each model that quite differed to each other.

Inception-V3, VGG16, and MobileNet all have much more complicated architecture that caused them to process all the images much longer compared to the designed CNN, although as compensation they didn't require too many epochs to reach the similar performance as the designed CNN. All transfer learning models' feature extraction layers here were set to non-trainable to see how well the pre-trained weights from ImageNet, a large image database built upon the backbone of WordNet structure [24]. From the results, we could see that the models didn't perform as well as expected. For future reference, setting the transfer learning models' weights to trainable might yield better results as we could update them through gradient descent and make them adjust better to the images.

Setting that aside, compared to other works we referred to in this study, the proposed models had not accomplished the task excellently. In their study, Achi et al. [6] managed to achieve an impressive result (95% accuracy) with their own designed CNN. The CNN was evaluated on their own dataset as well. The dataset covered more cases compared to the one we used in this study; it consists of 128 cases with a total of 2560 images for both training and validating purposes, roughly 5 times the size of our dataset. Considering how crucial data is to deep learning approaches, this might be our main issue on the subpar deep learning performances we obtained. As mentioned in the previous section, our dataset contained only 468 images from 19 cases with only 70 images of each class for 100x magnification and 50 images of each class for 400x magnification used in the classification. Although data augmentation could remedy the lack of data problem [25], the results show otherwise. This led to quite a few other possibilities as to what caused it. Firstly, there might be one of few couple of augmentation parameter values that work better than the ones we used currently, but unfortunately the narrow search space we used didn't cover those values. As seen in Table 3, we didn't use much varying values for the data augmentation parameters. This could be fixed by using a wider search space but since we used a brute force approach to search for the optimal parameters, it is impossible to widen the search space computationally. However, using other methods for parameter optimization, such as random search or genetic algorithm, which are better than grid search in terms of runtime [21], we may use a much wider search space to acquire a better parameter combination. Another possible cause was due to the generalized use of the augmentation parameters. The supposedly optimal parameters were obtained from

the grid search implemented using the designed CNN only instead of all the models used. This could be a consideration for future works.

**Acknowledgments.** We are thankful for our supervisors Mr. Anshori and Mr. Adiprawita for their guidance through the process of the experiment and journal writing. We also thank our colleagues from Bandung Institute of Technology who conducted the experiment that greatly assisted the research. Finally, we thank our colleagues from Padjajaran University Mrs. Usman and Mr. Husain who helped us provided the dataset which has been used for this experiment.

**Authors' Contributions.** ZHN, NH, UHM, CJA, KMR, and KY conceived the original idea and executed all the experiments, AI and AW acted as supervisor and advisor, while AH and HO provided the dataset used in this study. The main text was written by ZHN, NH, UHM, CJA, KMR, and KY. All authors read and approved the final manuscript.

## Bibliography

1. Cancer. (n.d.). Retrieved November 01, 2020, from https://www.who.int/health-topics/cancer
2. Cancer. (n.d.). Retrieved November 01, 2020, from https://www.who.int/news-room/fact-sheets/detail/cancer
3. Global Cancer Observatory. (n.d.). Retrieved November 01, 2020, from https://gco.iarc.fr/
4. Jemal A, Siegel R, Ward E, Hao Y, Xu J, Murray T, Thun MJ. Cancer statistics, 2008. CA Cancer J Clin Mar-Apr;2008 58:71–96. [PubMed: 18287387]
5. Infodatin. (2015). Data dan kondisi penyakit limfoma di Indonesia. Pusat Data dan Informasi Kementerian Kesehatan RI. ISSN. 2015; 2442–7659.
6. Achi HE, Belousova T, Chen L, Wahed A, Wang I, Hu Z, Kanaan Z, Rios A, Nguyen AND. Automated Diagnosis of Lymphoma with Digital Pathology Images Using Deep Learning. Ann Clin Lab Sci. 2019 Mar;49(2):153-160. PMID: 31028058.
7. Survival Rates and Factors That Affect Prognosis (Outlook) for Non-Hodgkin Lymphoma. (n.d.). Retrieved November 05, 2020, from https://www.cancer.org/cancer/non-hodgkin-lymphoma/detection-diagnosis-staging/factors-prognosis.html
8. Ansell, S. M. (2018). Hodgkin lymphoma: 2018 update on diagnosis, risk-stratification, and management. American Journal of Hematology, 93(5), 704–715. doi:https://doi.org/10.1002/ajh.25071
9. Johl, A., Lengfelder, E., Hiddemann, W., & Klapper, W. (2016). Core needle biopsies and surgical excision biopsies in the diagnosis of lymphoma—experience at the Lymph Node Registry Kiel. Annals of Hematology, 95(8), 1281–1286. DOI https://doi.org/10.1007/s00277-016-2704-0
10. Board, P. (2020, April 03). Adult Hodgkin Lymphoma Treatment (PDQ®). Retrieved November 01, 2020, from https://www.ncbi.nlm.nih.gov/books/NBK65804/
11. Mccarten, K. M., Nadel, H. R., Shulkin, B. L., & Cho, S. Y. (2019). Imaging for diagnosis, staging and response assessment of Hodgkin lymphoma and non-Hodgkin lymphoma. Pediatric Radiology, 49(11), 1545-1564. doi:https://doi.org/10.1007/s00247-019-04529-8
12. Syrykh, C., Abreu, A., Amara, N., Siegfried, A., Maisongrosse, V., Frenois, F. X., Brousset, P. (2020). Accurate diagnosis of lymphoma on whole-slide histopathology images using deep learning. Npj Digital Medicine, 3(1). doi:https://doi.org/10.1038/s41746-020-0272-0

13. Iizuka, O., Kanavati, F., Kato, K., Rambeau, M., Arihiro, K., & Tsuneki, M. (2020, January 30). Deep Learning Models for Histopathological Classification of Gastric and Colonic Epithelial Tumours. Retrieved November 01, 2020, from https://www.nature.com/articles/s41598-020-58467-9
14. Somaratne, U. V., Wong, K. W., Parry, J., Sohel, F., Wang, X., & Laga, H. (2019). Improving Follicular Lymphoma Identification using the Class of Interest for Transfer Learning. 2019 Digital Image Computing: Techniques and Applications (DICTA). doi:https://doi.org/10.1109/dicta47822.2019.8946075
15. Yadav, G., Maheshwari, S., & Agarwal, A. (2014, September). Contrast limited adaptive histogram equalization based enhancement for real time video system. 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). https://doi.org/10.1109/icacci.2014.6968381
16. Adaptive histogram equalization. (2020, January 21). In Wikipedia. https://en.wikipedia.org/wiki/Adaptive_histogram_equalization
17. Mikołajczyk and M. Grochowski. (2018). Data augmentation for improving deep learning in image classification problem, 2018 International Interdisciplinary PhD Workshop (IIPhDW), Swinoujście, 2018., doi: https://doi.org/10.1109/IIPHDW.2018.8388338
18. Shorten, C., & Khoshgoftaar, T. (2019, July 06). A survey on Image Data Augmentation for Deep Learning. Retrieved November 24, 2020, from https://doi.org/10.1186/s40537-019-0197-0/tables/2
19. Wu, J. (2017). Introduction of convolutional neural network.
20. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. In Y. Bengio (Ed.), The Journal of Machine Learning Research, 15 (pp. 1929–1958). Retrieved from https://dl.acm.org/doi/10.5555/2627435.2670313.
21. Liashchynskyi, P. & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS, arXiv, abs/1912.06059v1.
22. Kahia, M., Echtioui, A., Fathi, K., Hamida, A. (2022). Skin Cancer Classification using Deep Learning Models. https://doi.org/10.5220/0010976400003116
23. Agustina, E., Magdalena, R., Kumalasari, N., & Pratiwi, C. (2022). Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16. Ekomika (2022), 10(2), DOI : https://doi.org/10.26760/elkomika.v10i2.446
24. Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. 248-255. doi: https://doi.org/10.1109/CVPR.2009.5206848.
25. Perez, L. & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. ArXiv 1712.04621.