Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
14/1 (2021), 9-16. DOI: http://dx.doi.org/10.21609/jiki.v14i1.898

Pleural Effusion Classification Based on Chest X-Ray Images using Convolutional Neural Network

Ahmad Rafiansyah Fauzan, Mohammad Iwan Wahyuddin, and Sari Ningsih

Faculty of Communication and Information Technology, National University,
Jl. Sawo Manila, South Jakarta, 12520, Indonesia

E-mail: ahmadrafiansyahfauzan@gmail.com, iwan_wyd@yahoo.com, sariningsih.lectures@yahoo.com

Abstract

Pleural effusion is an abnormal lung condition characterized by a buildup of fluid between the two layers of the pleura, which causes specific symptoms such as chest pain and shortness of breath. In Indonesia, pleural effusion cases alone account for 2.7% of other respiratory diseases, with an estimated number of sufferers in general at more than 3000 people per 1 million population annually. Pleural effusion is a severe case and can cause death if not treated immediately. Based on a study, as many as 15% of 104 patients diagnosed with pleural effusion died within 30 days. This paper proposes a model that automatically detects pleural effusion based on chest x-ray images using a Machine Learning algorithm. The machine learning algorithm used is Convolutional Neural Network (CNN), with the dataset used from ChestX-ray14. The number of data used was 2500 in x-ray images, based on two different classes, x-ray with pleural effusion and x-ray with normal condition. The evaluation result shows that the CNN model can classify data with an accuracy of 95% of the test set data; thus, we hope it can be an alternative to assist medical diagnosis in pleural effusion detection.

Keywords: *Computer Vision, Image Classification, Convolutional Neural Network, Pleural Effusion*

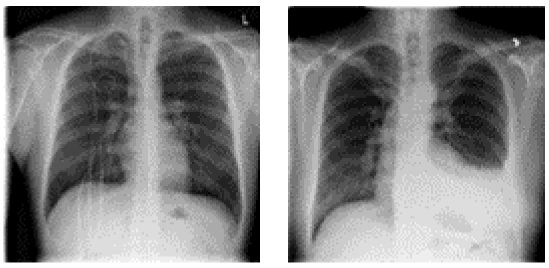
1. Introduction

The lungs are a vital organ of the respiratory system that functions as an oxygen-exchanging organ from the air and carbon dioxide from the blood. Therefore, they are an essential organ for the human respiratory system. If the lungs are impaired, it will directly affect the respiratory system, which can be life-threatening.

A pleural membrane covers the lungs, and between the membrane and the lungs, there is a pleural cavity that usually contains about 10-20 ml of fluid that serves as a lubricant so that the lungs can move freely while breathing [1]. However, if the fluid is excess and accumulates, it can pressure the lungs, causing chest pain and difficulty breathing; this condition is called Pleural Effusion.

Pleural effusion is an abnormal lung condition characterized by a buildup of fluid between the two layers of the pleura, which is generally caused by the formation of pleural fluid faster than the resorption process. With normal resorption, pleural effusion can occur if pleural fluid formation continues to increase to 30 times. On the other hand, decreased resorption of pleural fluid alone will not produce a significant buildup of fluid in the pleural cavity, given the normal rate of pleural fluid formation is very slow. Other diseases can cause pleural effusion, either originating from the lungs (such as tuberculosis, pneumonia), the pleural membrane, or extrapulmonary (such as congestive heart failure) [1].

Pleural effusion is a common health condition. In Indonesia, pleural effusion cases alone account for 2.7% of other respiratory diseases, and with an estimated number of sufferers in general (international) at more than 3000 people per 1 million population annually. Sufferers also come from any age group, although it generally occurs



(a) (b)

Figure 1. Chest x-ray samples (a) normal (b) pleural effusion

in women than men [2]. Pleural effusion is a severe case and can cause death if not treated immediately. According to a study [3], as many as 15% of 104 patients diagnosed with pleural effusion died within 30 days.

This study was made to detect pleural effusion based on chest x-ray images, which is one of the most common media for detecting abnormalities in the lungs [4], such as pleural effusion, which usually appears as a whitish area at the lung base. They may occur on only one side of the lung (unilateral) or both sides (bilateral) [5]. The primary purpose is to help improve medical diagnosis performance in identifying x-ray images with pleural effusion. Figure 1 (b) shows a chest x-ray sample with pleural effusion, and for comparison, Figure 1 (a) shows a chest x-ray with a normal lung condition.

The algorithm used in this study is Convolutional Neural Network (CNN), by creating a model to classify an x-ray image into one of two classes, pleural effusion or normal condition.

The rest of the paper presented in the following order, section 2 is an overview of some previous similar studies, followed by a brief review about CNN, section 3 describes the research methods, section 4 presents the results and discussion, and section 5 is the conclusion of the study and suggestions for further research.

2. Related Work

This section is an overview of some previous similar studies in completing the task in the form of classification of chest x-ray images using a similar algorithm and a brief review of the algorithm used in this study, CNN, and its components.

CNN in Classifying Chest X-ray Images

The ability of machine learning algorithms, especially deep learning in the medical field, particularly in detecting abnormalities based on x-ray images, has become popular in recent years. One of the studies is the diagnosis of 14 diseases based on chest x-ray images using CNN conducted by [6], which obtained accuracy in 10 conditions (including edema, pleural effusion, pneumonia, and others), which claimed to be equivalent to the radiological level (after specialized testing). Similar studies were also conducted by [7] and [8], which also obtained reliable accuracy on the 14 diseases.

Convolutional Neural Network

CNN is a type of Artificial Neural Network (ANN) or commonly called Neural Network (NN). It is a machine learning algorithm (also called deep learning because it generally has more than one layer) inspired by the human brain's neural network. CNN is not much different from ordinary NN. The thing that distinguishes CNN from ordinary NN at the initial layer of CNN is a special layer called the Convolution Layer, which functions to extract features in data, especially images. That is why CNN is commonly used to complete tasks such as image classification, object detection, image segmentation, and other computer vision tasks. This algorithm is designed to work well on unstructured data such as images [9]. CNN is generally divided into the convolution layer, Pooling Layer, and Fully Connected Layer [10].

Convolution Layer

The convolution layer is the main part of CNN. This layer has a component called kernel (also called filter) that functions as a feature detector to detect specific features in all parts of the image by doing an elementwise-product of every value in the kernel with every pixel value in the image, followed by summing the results into one value. This value will then be stored in the matrix called feature map [11]. Hyperparameter stride determines how far the kernel will move; if the stride used is 1, it means the kernel will move one pixel to the right until the end of the image, then move 1 pixel down and back to the beginning of the line. This process is repeated until the kernel reaches the lower right part of the image, as the illustration shown in Figure 2.

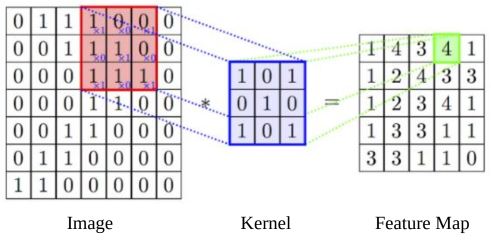


Figure 2. Illustration of the convolution process

A non-linear activation function can be applied in this layer for the non-linear transformation of the feature map [12]. The non-linear activation function commonly used is ReLU (Rectified Linear Unit), shown in equation (1).

$$\boldsymbol{f(x) = max(0, x)} \tag{1}$$

Where x is the output of a neuron (or value in feature map), if $x \leq 0$, then $x = 0$ and if $x > 0$, then $x = x$.

Pooling Layer

The pooling layer (or subsampling layer) is a component of the CNN, which functions to reduce the dimensions of the feature map by selecting pixel values based on specific rules. Pooling layer operations that are commonly used include max pooling and average pooling [13]. Max pooling works by maintaining the highest value of a feature map (as Illustrated in Figure 3). In contrast, average pooling looks for the average value of a feature map (if only return one value per feature map is called global pooling or can also return multiple values using configured kernel).

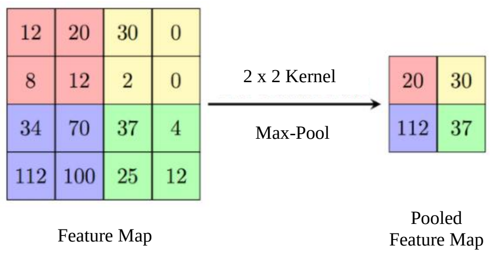


Figure 3. Illustration of the pooling process (max-pool)

Fully Connected Layer

The image through the convolution and pooling layers for feature extraction, then each feature map, needs to be transformed into a one-dimensional vector (called the flattening process) before entering the fully connected layer (FC) [14]. FC layer is the part of the CNN that functions like ordinary NN, which is to perform non-linear transformations of the data (extracted image features) to obtain the output value [15].

The output is then converted into a probability (that sum to one) by applying the Softmax function [16] as a classification method based on the existing class. The softmax function is shown in equation (2).

$$y_j = \frac{e^{y(j)}}{\sum_{i=1}^{n} e^{y(i)}} \tag{2}$$

Where y is the output of neuron j, divided by the summation of outputs of all neurons ($constant\ e = 2.71828$).

In this study, we built a CNN model to detect pleural effusion by receiving an input of chest x-ray image (frontal-view) and produced a predicted label as an output. To interpret the model's result, the model also generates a heatmap to visualize the area that the model mostly focuses on when making the prediction; the goal is to build appropriate trust in users who use the model.

3. Methods

There are several stages carried out in this section. First, we describe the data used in this study with some preprocessing techniques, followed by the proposed CNN architecture and the hyperparameter. We also describe the performance metrics, model interpretation technique, and how we implement the model to an application at the end of this section.

Dataset

The data used in this study came from the ChestX-ray14 dataset [7], one of the most massive frontal-view chest x-ray image datasets with a total of 112,120 images acquired from 30,805 patients with many advanced lung diseases. This dataset is divided into 14 classes on disease type and normal condition. The images were saved in PNG format and rescaled to a size of 1024 x 1024 pixels, with one or multiple labels (overlap with other diagnoses). They were acquired from the radiologist report using a natural language processing technique.

This study used 2500 data samples of x-ray images, consisting of two different classes, x-ray images with pleural effusion and normal condition, with each data per class containing 1250 images.

The data then goes through several preprocessing stages. First, we divide the data into three parts, including the training set used to train the CNN model, the validation set used to estimate the model's performance of given hyperparameters, which then used to determine the best combination of hyperparameters, and the test set used to test/evaluate the performance of the CNN model in classifying new data. The comparison of the training set, validation set, and test set are 80:10:10, as shown in Table 1.

Table 1. Data splitting

	Pleural Effusion	Normal
Training Set	1000	1000
Validation set	125	125
Tes set	125	125

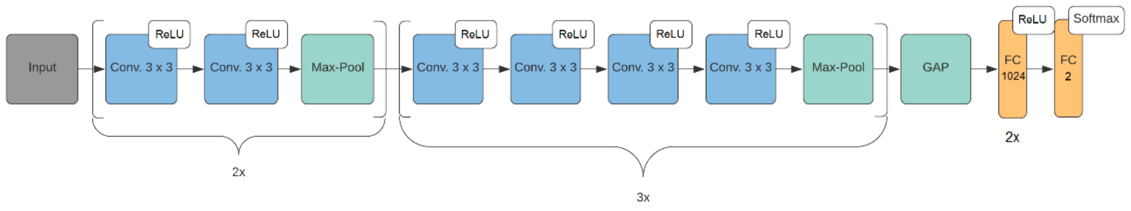


Figure 4. CNN architecture, VGG19 with modifications on the top layer

We also downscale the images' resolution to 224 x 224 pixels (adjusted to the VGG19 architecture) and rescale the pixel value to between 0-1 (by dividing every pixel with 255) as a normalization method. We also applied data augmentation techniques to reduce the possibility of model overfitting; the data augmentation used includes width shift range and height shift range.

The amount of data used in this study (2500 images) is adjusted to the system's specifications to avoid too long training time.

Proposed CNN Architecture

The CNN model used in this study is VGG19 from the Visual Geometry Group [17]. VGG19 is a deep CNN model with 19 layers (16 convolution layers and 3 FC layers), a kernel size of 3x3 with a stride of 1 on all convolution layers, along with a 2x2 kernel and a stride of 2 on all max-pooling layers. The number of feature maps in this model starts at 64 in the first block of the convolution layer, doubling after every max-pooling layer until it reaches 512 at the last convolution layer. The number of neurons in the FC layers are 4096, 4096, 1000, respectively.

The VGG19 model used in this study is a model with parameters that have been initialized to the ImageNet dataset (also called pre-trained model) in the convolution layers, with modification at the top layer of the model architecture (fully connected layer), to match the used dataset.

We also add the Global Average Pooling (GAP) layer before the FC layer to reduce the model's parameters. The number of neurons used in the first two FC layers is 1024 (equipped with ReLU for non-linearity), and the last FC layer is 2 (based on the existing class), as illustrated in Figure 4. Then the softmax function is applied to convert the output into probability.

A CNN model that built to do a classification task behaves as object detectors despite without any supervision on the object's location was provided; however, this ability is lost when the features are flattened for the classification stage. Therefore, the GAP layer acts as a regularizer (reducing the model's parameter to help prevent overfitting) also improves the model's localization ability since the GAP encourages the model to identify the extent of the object [18].

With our model, we can visualize an image's important features using the Grad-CAM technique (discussed in the model interpretation section) by generating a heatmap to highlight the most indicative area of effusion or highlight the most contributed area of chest x-ray for normal condition.

The model has 11,016,194 learnable parameters with the total number of parameters of 21,601,346, built using Keras framework that runs on Tensorflow and trained using a system with 8 GB of RAM and an NVIDIA GeForce GTX 960M 4GB GPU.

The reason for choosing the pre-trained VGG19 is because this model has been trained on the ImageNet dataset, which has more than 1 million images. Therefore, this model's convolution layers have good enough at detecting features in images, ranging from low-level features to high-level features. So it can have higher accuracy with shorter training time (compared to training a custom deep CNN from scratch).

Learning Algorithm and Other Hyperparameters

Gradient Descent (GD) is generally used as an algorithm to update parameters (such as weight and bias) based on the loss value [16]. The parameters update is done to minimize the loss value to a minimum. A special function is needed to obtain the value of the loss, such as Cross-Entropy, shown in equation (3).

$$L_{cross-entropy}(\hat{y}, y) = -\sum_{i} y_i \log(\hat{y}_i) \quad (3)$$

Where y_i is the desired output (actual value), and $\hat{y}_i$ is the output produced by the model (predicted value).

There are several types of gradient descent, one of which is Stochastic Gradient Descent (SGD), which is the type of GD that works by calculating the value of the gradient (the value that will be used for updating the parameter) by one training sample determined randomly [16]. In this study, instead of using one training sample, a set sample (or mini-batch) is used. Therefore, fewer parameters update will be performed, reducing training time.

There are several learning algorithms are based on GD (with specific optimizations), Adaptive Moment Estimation (Adam) is one of them. In this study, adam is used as a learning algorithm (also called an optimizer) with learning rate initialization of 0.0001 and default parameters (beta1 = 0.9, beta2 = 0.999). Other hyperparameters used are shown in Table 2.

Table 2. Learning algorithm and other hyperparameters

Optimizer	Learning rate	Mini-batch	Epoch
Adam	0.0001	16	100

The hyperparameters used include a mini-batch of 16, which shows the number of samples used to calculate the value of the gradient (in one time), which means that in one epoch, the parameters will be updated 125 times (number of data in training set divided by the number of mini-batch). Lastly, we trained the model for 100 epochs.

Performance Metrics for Evaluation

Confusion Matrix is used as a tool to measure the model's performance on the test set, with several metrics including accuracy, sensitivity, and specificity. The illustration of the confusion matrix is shown in Figure 5.

		Predicted Label	
		Negative	Positive
Actual Label	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 5. Illustration of the confusion matrix

Where *True Negative* (TN) shows the amount of negative data that is predicted correctly, *False Positive* (FP) shows negative data that is predicted as positive, *True Positive* (TP) shows positive data that is predicted correctly, and *False Negative* (FN) shows positive data that is predicted as negative.

Accuracy

Accuracy is used to measure the model's ability to classify samples (positive and negative) correctly of all test data [19]. The equation to calculate accuracy is shown in equation (4).

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \tag{4}$$

Sensitivity

Sensitivity (or true positive rate) is used to measure the model's ability to classify positive samples correctly [19]. The equation for calculating sensitivity is shown in equation (5).

$$Sensitivity = \frac{TP}{FN + TP} \tag{5}$$

Specificity

Specificity (or true negative rate) is used to measure the model's ability to classify negative samples correctly [19]. The equation for calculating specificity is shown in equation (6).

$$Specificity = \frac{TN}{TN + FP} \tag{6}$$

Model Interpretation

Gradient-weighted Class Activation Mapping (Grad-CAM) [20] is used to interpret the model's prediction by generating a heatmap that shows the area/region on a chest x-ray image that contributes the most to the model's prediction.

Grad-CAM works by computing the weight for each final convolution layer's feature maps and then using it to weight the associated feature map then summing all the weighted feature maps to create a coarse heatmap. Formula to compute the weight shown in equation 7, and equation 8 is the formula to generate a heatmap using weight obtained from equation 7.

$$a_k = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\substack{Applied\ GAP \\ to\ the\ computed \\ gradients}} \underbrace{\frac{\partial y}{\partial A^k_{i,j}}}_{\substack{Compute\ the \\ gradient\ of\ each \\ value\ in\ A^k}} \tag{7}$$

Where a_k represents the weight for k feature map (with k represent the k^{th} feature map) of the final convolutional layer. y represents the score of the predicted class (before softmax), $A^k_{i,j}$ represents each value of k feature map of the final convolution layer, and Z represents the number of values in k feature map. GAP: Global Average Pooling.

$$L = ReLU\left(\sum_k a_k A^k\right) \tag{8}$$

ReLU is applied to the coarse heatmap (indexed by L) to maintain only the features that positively influence the predicted class. This coarse heatmap has the same dimensions as the final convolution layer’s feature maps; therefore, it needs to be upscaled based on the input dimensions to overlay the given input.

This technique is used to provide more explainable results and to ensure the model sees the correct part of the image when making its prediction.

Application Implementation

The model that has been designed, trained, and evaluated will then be implemented into a web-based application. The implementation carried out using Flask Framework (Python) as a back-end programming language with Javascript, HTML, and CSS as front-end.

4. Results and Discussion

This section discusses several things, including the model's performances in the training and validation process, the evaluation of the model's performance in conducting classification on the test set, briefly discuss the interpretation of the model's prediction, and the interface of the application implementation. Note, because this study was made to detect single pathology (pleural effusion), with a portion of data (1250 images of each class) of the entire dataset, the results presented below cannot be compared directly to the previous studies [6]-[8].

Training and Validation

The accuracy and loss graphs of the training and the validation are shown in Figure 6.

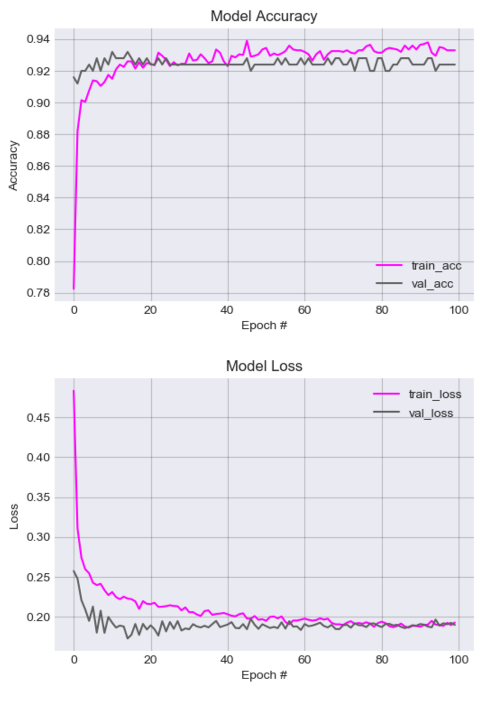


Figure 6. The graphs of the model’s accuracy and loss

Based on the graphs above, the model gets good accuracy in the training process. It is showed by the loss value of the training set that continues to decreases (as the accuracy increases) with the increasing epoch; this implies that the model fits the training data well. On the other hand, the validation loss indicates that the model suffered from overfitting with the loss on the validation set that continuously increases (as the accuracy decreases) after the 16th epoch. Thus, the checkpoint technique is used by saving the model with the lowest loss on the validation set and then using it for the evaluation phase. With the pre-determined hyperparameter (Table 2), the training process takes 141 minutes. The lowest loss achieved by the model (on the 16th epoch) is 0.1730, with an accuracy of 93% in the validation set. This model will be used for the evaluation phase.

Model Evaluation

As discussed earlier, the tool used in evaluating the model’s performance on the test set is the confusion matrix. The confusion matrix of the test set is shown in Figure 7.

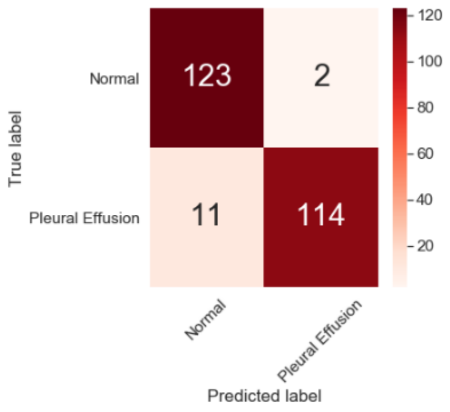


Figure 7. Confusion matrix of the test set

Based on the confusion matrix above, the accuracy, sensitivity, and specificity can be calculated using equations (4), (5), and (6). The results are shown in Table 3.

Table 3. Accuracy, sensitivity, and specificity (%)

Accuracy	Sensitivity	Specificity
95	91	98

The model obtained a result of 95% on the accuracy metric, which indicates the model’s ability to classify data (positive and negative) correctly from all test data (test set). On the sensitivity metric, the model obtained 91%, which indicates the model’s ability to classify positive data (pleural effusion) as positive. The last metric, specificity, achieved 98% results, which indicates the model’s ability to classify negative data (normal condition) as negative.

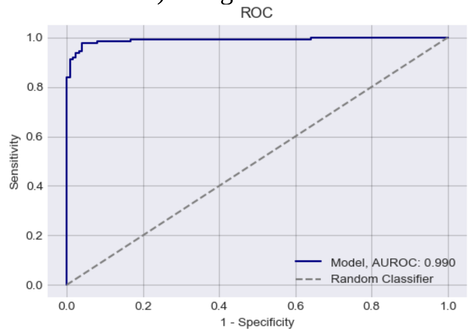
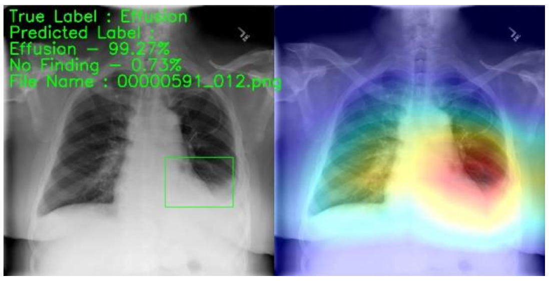


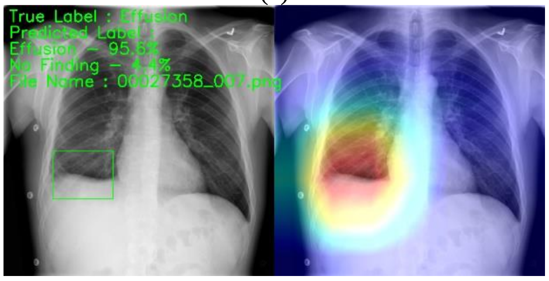
Figure 8. ROC curve with AUROC = 0.99

The graph above is the ROC (Receiver Operating Characteristics) curve, which summarizes the trade-off between sensitivity and specificity (at all possible classification thresholds) with an AUROC (Area Under ROC) of 0.99, which indicates the model's ability to classify positive and negative data correctly.

Model Interpretation



(a)



(b)

Figure 9. heatmap's samples for pleural effusion

Figure 9 shows samples of two chest x-ray images with effusion on the left lung (a) and right lung (b). Based on the model's prediction, the model successfully classified the images correctly with a high confidence level (99% and 95% on the image a and b). The model also produced heatmaps that focused on the lower left lungs (a) and lower right lungs (b), which indicate the model sees at the correct part of the image when predicting pleural effusion. We also generated a bounding box for localization purposes using a simple thresholding technique based on the generated heatmap.

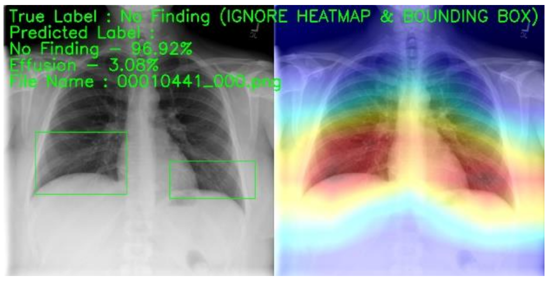
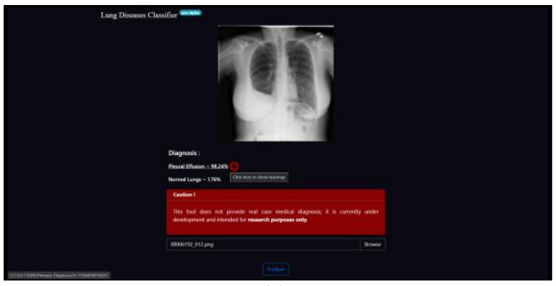

True Label : No Finding (IGNORE HEATMAP & BOUNDING BOX)
Predicted Label :
No Finding – 96.92%
Effusion – 3.08%
File Name : 00010441_000.png

Figure 10. heatmap's sample for normal condition

In contrast, figure 10 shows that the model predicted an x-ray correctly as a normal lung also with high confidence, with a heatmap that focuses mostly on the black region of both sides of the lungs, which can be interpreted as the most contributing features for the normal condition class.

From the heatmaps produced by the model, it can be concluded that the result predicted by the model can be interpreted by visualizing the heatmap of the predicted class.

Application Implementation



Diagnosis :

Pleural Effusion – 98.24%

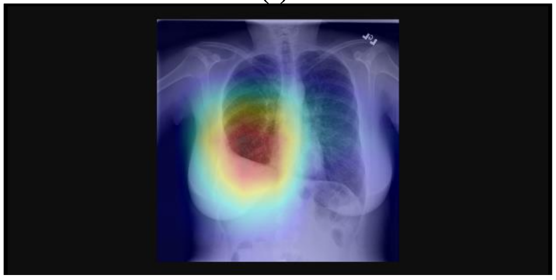
Normal Lungs – 1.76%

Caution !

This tool does not provide real case medical diagnosis; it is currently under development and intended for **research purposes only.**

00006192_012.png

(a)



(b)

Figure 11. The interface of pleural effusion classifier application

Figure 11 (a) shows the application interface showing the prediction result of an x-ray image with the confidence level in the form of a percentage. The application also provides a button (provided on the label of the classification result) to display a heatmap that shows the area that the model focuses on in its prediction, as shown in Figure 11 (b).

5. Conclusion

The results of the evaluation and analysis show that the model built to perform a task in the form of pleural effusion classification based on x-ray images (frontal-view) shows promising results on the tested metrics, with 95%, 91%, and 98% on the accuracy, sensitivity, and specificity respectively. To provide a more explainable result, the model also produces a heatmap to show the area it focuses on when making its prediction. This heatmap can also be used to locate the area most indicative of pleural effusion; thus, we hope it can be an alternative to improve the medical diagnosis performance in identifying the presence of pleural effusion.

CNN algorithm will have better performance using large amounts of data for the model training. Therefore, further study is expected to use a more significant amount of image data to produce a better performance model in image classification and object localization. Furthermore, further study will also be expected to use x-ray image data with more classes of various diseases to be more helpful and applicable.

References

[1] S. A. Nasution, "Skrining Makroskopis Cairan Pleura dari Efusi Pleura di Unit Laboratorium Patologi Anatomi Rumah Sakit Umum Pendidikan Haji Adam Malik Medan," *J. AnLabMed Vo.1 No.1 Desember*, 2019.

[2] I. Puspita, T. Umiana Soleha, and G. Berta, "Penyebab Efusi Pleura di Kota Metro pada tahun 2015," *J AgromedUnila* , vol. 4, p. 25, 2017.

[3] J. T Puchalski, "Mortality of Hospitalized Patients with Pleural Effusions," *J. Pulm. Respir. Med.*, vol. 04, no. 03, 2014, doi: 10.4172/2161-105x.1000184.

[4] P. Rajpurkar *et al.*, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," Nov. 2017.

[5] "9 Pleural Effusion Causes, Symptoms, Treatment, Prognosis." https://www.medicinenet.com/pleural_effusion_fluid_in_the_chest_or_on_lung/article.htm (accessed Jul. 02, 2020).

[6] P. Rajpurkar *et al.*, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists," *PLoS Med.*, vol. 15, no. 11, Nov. 2018, doi: 10.1371/journal.pmed.1002686.

[7] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, May 2017, vol. 2017-Janua, pp. 3462–3471, doi: 10.1109/CVPR.2017.369.

[8] H. Wang and Y. Xia, "ChestNet: A Deep Neural Network for Classification of Thoracic Diseases on Chest Radiography," 2018.

[9] Ž. Knok, K. Pap, and M. Hrnčić, "Implementation of intelligent model for pneumonia detection," *Teh. Glas.*, vol. 13, no. 4, pp. 315–322, 2019, doi: 10.31803/tg-20191023102807.

[10] I. B. L. M. Suta, R. S. Hartati, and Y. Divayana, "Diagnosa Tumor Otak Berdasarkan Citra MRI (Magnetic Resonance Imaging)," *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 2, Jun. 2019, doi: 10.24843/mite.2019.v18i02.p01.

[11] L. Devnath, S. Luo, P. Summons, and D. Wang, "Tuberculosis (TB) Classification in Chest Radiographs using Deep Convolutional Neural Networks," *Int. J. Adv. Sci. Eng. Technol.*, vol. ISSN, no. 3, pp. 2321–9009, 2018.

[12] R. H. Abiyev and M. K. S. Ma'aitah, "Deep Convolutional Neural Networks for Chest Diseases Detection," *J. Healthc. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/4168538.

[13] L. A. Andika, H. Pratiwi, and S. S. Handajani, "Klasifikasi Penyakit Pneumonia Menggunakan Metode Convolutional Neural Network Dengan Optimasi Adaptive Momentum," *Indones. J. Stat. Its Appl.*, vol. 3, no. 3, pp. 331–340, 2019.

[14] O. Stephen, M. Sain, U. J. Maduh, and D. U. Jeong, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare," *J. Healthc. Eng.*, vol. 2019, 2019, doi: 10.1155/2019/4180949.

[15] R. Rokhana *et al.*, "Convolutional Neural Network untuk Pendeteksian Patah Tulang Femur pada Citra Ultrasonik B-Mode," *JNTETI*, vol. 8, no. 1, 2019.

[16] E. Ebermam and R. A. Krohling, "A Comprehensive Introduction to Convolutional Neural Networks: A Case Study for Character Recognition," pp. 49–59, 2018.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2015.

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 2921–2929, 2016, doi: 10.1109/CVPR.2016.319.

[19] J. Novakovic, A. Veljovi, S. Iiic, Z. Papic, and M. Tomovic, "Evaluation of Classification Models in Machine Learning," *Theory Appl. Math. Comput. Sci.*, vol. 7, no. 1, pp. 39–46, 2017.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," Dec. 2019.