



# The Development of X-Ray Image Classification System based on Convolutional Neural Network Algorithm

Budi Sugandi<sup>1</sup>, Ridwan Ridwan<sup>1</sup>, Abdullah Sani<sup>1</sup>, Iman Fahruzi<sup>1</sup>, Budiana Budiana<sup>1</sup> and Stevany Stevany<sup>1</sup>

<sup>1</sup> Electrical Engineering Department, Batam State Polytechnic, Batam, Indonesia  
budi\_sugandi@polibatam.ac.id

**Abstract.** This study aims to develop a system to classify the X-ray image based on the Convolution Neural Network (CNN) algorithm. The system is integrated with a Graphical User Interface (GUI) to make it easy to use. The algorithm of the classification process consists of 3 stages. The preprocessing stage to optimize the image quality as the first stage. The second stage is the training process, which aims to train the CNN system to learn and model the dataset for each class. The last stage is the validation process, which aims to evaluate and validate the test data compared to the training data. In the last stage, the training data is used to classify the X-ray image. We used the GUI to display the classification result. We used a 20384 dataset consisting of 5243 COVID cases, 11995 normal cases and 3146 Pneumonia cases. We divided the data into 90% data for training and 10% for test Data. The experimental results are evaluated using a confusion matrix to determine the accuracy, precision, F1 score and recall. The experimental results show the successful rate of the performance of our system in image classification with results as follows average of accuracy is 90 %, precision 92%, recall 90% and F1 score 91%. In addition, the deployed GUI successfully displayed the x-ray image with classification result and the accuracy value. The GUI is also equipped with the report of the classification result in the form of a PDF file.

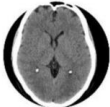
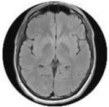
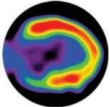
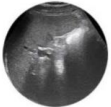

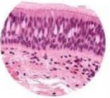
**Keywords:** *X-ray Image, Deep Learning, CNN, X-ray Classification*

## 1 Introduction

Nowadays, deep learning is an algorithm used in many research topics, such as medical image [1], natural language processing [2], education [3], speech recognition [4], cyber security [5, 6], computer vision [7], and so on. This is because deep learning is very suitable for use when the number of data is huge [8].

In the medical field, diagnosis accuracy is a very important thing because it is related to the life of human beings. In order to achieve high accuracy of diagnosis, the medical doctor combined their capability in diagnosis with the use of radiological/medical

image. The diagnosis using medical image can be their second opinion. There are many medical images used by radiologists. They come in many modalities, such as X-ray, Computed Tomography (CT), Ultrasounds, Positron Emission Tomography (PET), Magnetic Resonance Imaging (MRI), histology and as shown in Fig. 1 [9]. X-ray imaging is one of the most common tools used by medical doctors to examine lung conditions. Comparing to other modalities, X-Ray is more simple to obtain and diagnose. However, the X-Ray images still require the accuracy and expertise of doctors in reading and diagnosing the image [9], especially when a large number of images have to be diagnosed. The COVID-19 pandemic is a clear example of the world experiencing a global health crisis. The large number of patients makes the hospital overloaded not only because of the number of patients but also because of the huge number of x-ray images to be diagnosed. These huge numbers could be caused the diagnosis and treatment errors [10]. Some of the COVID cases were diagnosed as pneumonia because of similar symptoms. When the COVID-19 pandemic, we can see that the mortality rate caused by pneumonia is quite high which causing 1.23 million deaths in adults over the age of 70 years [11].

					
Computer Tomography	Magnetic Resonance Imaging (MRI)	Positron Emission Tomography (PET)	Ultrasound	X-Rays	Histology
<ol style="list-style-type: none"> <li>1. Abdomen</li> <li>2. Appendix</li> <li>3. Bladder</li> <li>4. Brain</li> <li>5. Chest</li> <li>6. Kidney</li> <li>7. Cervix</li> <li>8. Breast</li> </ol>	<ol style="list-style-type: none"> <li>1. Neuroimaging</li> <li>2. Cardiovascular</li> <li>3. Liver</li> <li>4. Functional</li> <li>5. Oncology</li> <li>6. Phase contrast</li> </ol>	<ol style="list-style-type: none"> <li>1. Cardiology</li> <li>2. Infected Tissues</li> <li>3. Small Animal Imaging</li> <li>4. Neuroimaging</li> <li>5. Oncology</li> <li>6. Musculoskeletal</li> <li>7. Pharmacokinetics</li> </ol>	<ol style="list-style-type: none"> <li>1. Transrectal</li> <li>2. Breast</li> <li>3. Doppler</li> <li>4. Abdominal</li> <li>5. Transabdominal</li> <li>6. Cranial</li> <li>7. Gallbladder</li> <li>8. Spleen</li> </ol>	<ol style="list-style-type: none"> <li>1. Radiography</li> <li>2. Mammography</li> <li>3. Fluoroscopy</li> <li>4. Contrast Radiography</li> <li>5. Arthrography</li> <li>6. Discography</li> <li>7. Dexa Scan</li> </ol>	<ol style="list-style-type: none"> <li>1. Epithelium</li> <li>2. Endothelium</li> <li>3. Mesenchyme</li> <li>4. Blood Cells</li> <li>5. Neurons</li> <li>6. Germ Cell</li> <li>7. Placenta</li> </ol>

**Fig. 1.** Medical Image Modality

In order to help the medical doctor diagnose many patients in a short time and improve the diagnosis result, artificial intelligence (AI) can be one of the solutions. AI can overcome many challenges nowadays [12]. AI can help diagnose diseases and analyze medical image data so that doctors' diagnoses become more accurate and become the initial stage for more effective preventive measures. The use of properly trained and optimal AI models can promise a faster diagnosis process in time [13, 14]. AI is defined as a field of study in computer science that aims to solve cognitive problems related to human intelligence, such as problem-solving, learning and pattern recognition [15]. Recent research has successfully demonstrated the ability of AI to be a second opinion to diagnose medical images because of its high performance and accuracy [16, 17].

There are several methods that can be used to detect, classify and recognize imaging data, one of them using deep learning. In deep learning, the CNN algorithm is one of

the methods used to solve problems related to object detection and image classification [18]. In the classification process, CNN has several layers, which are the convolution, pooling and fully connected layers [19]. CNN is the result of the development of Multi-Layer Perceptron (MLP), which is included in the type of Deep Learning and is designed to process two-dimensional data such as images. However, MLP presents each pixel as an independent feature and does not store spatial information from image data, making it less suitable in terms of image classification [20]. Model design using the CNN algorithm was able to be used to classify nodules on lung X-ray images with a total of 180 training data and an accuracy of 86.67% [21]. Another study in [22], concluded that deep learning algorithms showed strong results in diagnosing COVID-19 and pneumonia.

This research aims to classify the X-ray image using a Convolution Neural Network (CNN). The classification process has three stages. The classification process is started by preprocessing stage to optimize the image quality. The second stage is the training process which aims to train the CNN system to learn and model the dataset for each classification. The last stage is the validation process which aims to evaluate and validate the test data compared to training data. In the last stage, the image will be classified based on the training data. We evaluated the classification result using confusion matrix to obtain accuracy, precision, F1 Score and recall. We also used the Graphical User Interface (GUI) to display the classification result and accuracy on each classified image.

## 2 Research Methods

### 2.1 Preparing X-Ray Image Lungs Dataset

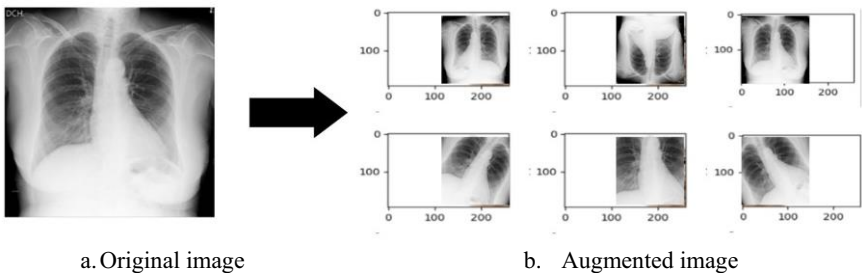
The research worked with a dataset made up of X-ray images obtained from the Kaggle web site (<https://www.kaggle.com>). The website provides a broad range of data sets for research and development purposes in various fields. X-ray datasets consist of images obtained from several different repositories available to the public. This combination images makes data sets more varied. The dataset to be used in this research consists of 5243 images of COVID cases, 3146 pneumonia cases and 11995 healthy or normal cases as shown in Table 1. The acquired source dataset is updated consistently, so the number of datasets available in the repository tends to change in the future. The dataset is divided into two categories which are 90% (18344) of training data and 10% (2040) test or validation data. In order to avoid the differences in ratio and pixels of the x-ray images, the images are rescaled into 150 x 150 pixels. The random distribution of data will be picked up within the training data and the test data. Training data is used for training processes and testing data is used for the validation process of the models.

**Table 1.** Dataset Summary

Classes	Total Data	Training Data	Test Data
Covid	5243	4718	525
Normal	11995	10795	1200
Pneumonia	3146	2831	315
<b>Total data</b>	<b>20384</b>	<b>18344</b>	<b>2040</b>

**2.2 Preprocessing Data**

Preprocessing is performed to optimize the image quality to make the training process easier. Therefore, a large amount of image data is required for intensive training and improving the model performance [23]. In this research, the data augmentation method is used to improve the dataset to be better and varied. In deep learning models, a common problem occurs is overfitting. To prevent overfitting, deep learning requires a lot of training data. However, it is often very difficult to get more training data. To overcome the problem, image augmentation is one of the solution. Augmentation data can prevent overfitting by modifying the data and transforming the original data differently to have a large data characteristic [24]. An overview of the augmentation results compared to the original image is shown in Fig. 2.



**Fig. 2.** Augmentation Image process

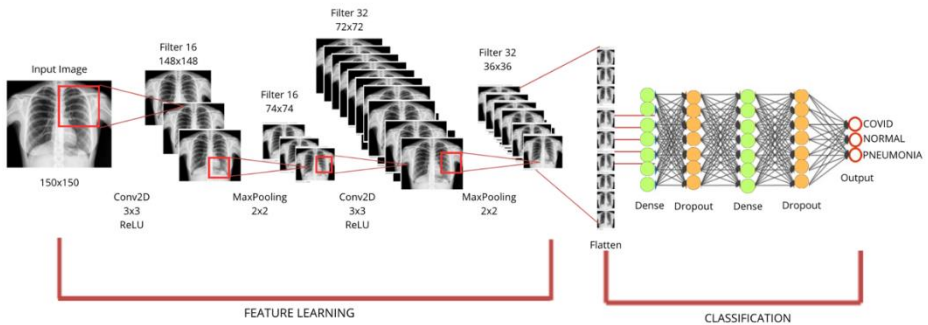
The augmentation process can be described as follows : firstly, the images are rescaled and normalized the pixel value by dividing them by 255, which helps to stabilize the training data. Next, the images are rotated by 30 degrees to the left or right and shifted by 20% of the width and height of the image. The shear or flattening range is a transformation that shifts each point parallel to the *X* and *Y* axes by 30 degrees. The zoom-in and zoom-out ranges allow horizontal rotation and increase or decrease the image size by 10%, respectively. The Nearest setting is used to fill empty image pixels when they are rotated or shifted out of the image boundary. The augmentation image process parameters are summarized in Table 2. The preprocessed images are then used as training data to generate the model.

**Table 2.** Augmentation Data Setting

Method	Setting
rescale	1/255
rotation_range	30
width_shift_range	0.2
height_shift_range	0.2
shear_range	0.3
zoom_range	0.1
horizontal_flip	True
fill_mode	Nearest

### 2.3 Convolutional Neural Networks (CNN) Architecture Design

CNN is widely used for classification, detection and recognition [25]. The ability to predict image classes with great precision makes CNN models often used for image classification. As shown in Fig. 3 in the classification process, CNN has multiple layers such as a convolution, pooling, and fully connected layer [26].

**Fig. 3.** CNN architecture for image classification

In this research, CNN is grouped into two main processes which are feature extraction and classification. The feature extraction process has two layers, the convolution and pooling layer. The convolutional layer learns more complex patterns which increases computational efficiency. The various features of input images are extracted in the convolutional layer. The pooling layer then performs the pooling operation to simplify network calculations, compress feature maps, and remove redundant information [27]. The neurons between two different layers are connected in fully connected layers. This layer is the last layer of CNN. This layer studies every relationship between extracted features and then uses them to make prediction results.

Details of the CNN architecture and parameters used for X-ray image data classification in our research are shown in Table 3. There are three points displayed on the table, which are layer type, output shape and number of parameters. Layer type contains layer information used during the training process which is arranged sequentially starting from the first layer to the last. The first layer is the convolution layer 2D, aimed at detectint image features to obtain a feature maps. Max pooling layer aims to decrease the spatial dimension of the feature map and overfitting. Flatten layer change feature map 2D become a 1D vector to make the input data become one dimension vector. The last layer Dropout do regulation to reduce overfitting and avoid certain neuron becoming dominant. The Output shape is the result of dimensional changes that have been changed by each layer. The output shape consists of width, height and output channel. The width and height come from the previous input dimension. The value of padding and stride uses the default value of 1. The output shape is obtained using (1) – (3).

$$OutputWidth = (InputWidth - KernelWidth + Padding) / Stride \tag{1}$$

$$OutputHeight = (InputHeight - KernelHeight + Padding) / Stride \tag{2}$$

$$OutputChannels = number\ of\ Filter \tag{3}$$

**Table 3.** CNN Architecture

Layer Type	Output Shape	Number of Parameters
conv2d (Conv2D)	148 x 148 x 16	448
max_pooling2d (Max-Pooling2D)	74 x 74 x 16	0
conv2d_1 (Conv2D)	72 x 72 x 32	4640
max_pooling2d_1 (MaxPooling2D)	36 x 36 x 32	0
conv2d_2 (Conv2D)	34 x 34 x 64	18496
max_pooling2d_2 (MaxPooling2D)	17 x 17 x 64	0
flatten (Flatten)	18496	0
dense (Dense)	200	3699400
dropout (Dropout)	200	0
Total Number of Parameters: 3824987		
Total Number of Trainable Parameters: 3824987		
Total Number of Non-trainable Parameters: 0		

The parameters are variables that will be learned during the training process. The number of parameters can be determined by various factors, including kernel size, number of filters, bias or other variables that can be learned by the model. The conv2d layer, the number of parameters is obtained using (4). The dense layer determined by calculating number input, output and bias variables as shown in (5). However, the max pooling, flatten and dropout layers do not have a number of parameter because those layers only change the shape of their dimensions. Those three pieces informations determine how the training process is carried out and can be arranged according to the needs of the desired model. Each layer will store different processes or results according to their respective function.

$$\text{Parameter (conv2D layer)} = \frac{(\text{Number of kernel size} * \text{Number of Class} + \text{Bias})}{\text{Number of Filter}} \quad (4)$$

$$\text{Parameter (dense layer)} = (\text{Number of input} * \text{Number of output}) + \text{number of Bias} \quad (5)$$

There is also activation and loss function in CNN models. The activation function is applied to the convolution and fully connected layer, while the loss function is applied to the output to measure accuracy and precision. The activation function used in the pooling layer is Rectified Linear Units (ReLU), allowing the model to remain mathematically stable [28]. All the layers are stacked in CNN in order to make a full CNN architecture. CNN may also include a batch normalization layer and dropout layer to improve the training time and address the overfitting issue. In its implementation, the CNN method is developed through several stages which are making a model of training, evaluation, and prediction.

The use of gradient descent is widely used in neural network-based methods, including CNN, to minimize errors during the training process and adjust internal parameters. One effective technique in deep learning to speed up model optimization is Adam (Adaptive Momentum) [29]. Adam is a combination of two optimization techniques, RMSprop and Momentum. It considers smooth gradient variants and includes a bias correction mechanism. Adam minimizes the computational cost as it requires less execution memory. Moreover, it is not affected by any change in the diagonal scale change of the gradient. Adam is a simple and efficient computational technique that provides adaptive learning rates for all parameters involved in gradient training [30].

## 2.4 Training

The training process is carried out after preparing the dataset and designing the architecture to be used. Data training is a learning process carried out using image datasets that have been preprocessed which produce models that are able to make predictions. The training process will determine how the model will be formed. Basically, a data training workflow using the CNN method involves data preparation, initialization of the CNN model, model training using practice data, evaluation of the model using validation data, tuning the model if needed, final evaluation of the model using test data, and use of the model to perform predictions on new images.

It needs to configure parameters before the model is trained, with the aim of improving learning optimization, minimizing errors and maximizing accuracy. Our research uses Adam as an optimization algorithm in the TensorFlow module. The model will be compiled by setting the optimizer, loss function, metrics and number of steps per epoch accordingly. Because this study was conducted with multiclass training, the Adam optimizer used in this research had a learning rate of 0.001 and the loss function was set as categorical cross-entropy. The parameters used can vary depending on training needs. With the use of appropriate optimizers and metrics, the model will be trained and evaluated efficiently and accurately. In the training process, the number of epochs

also needs to be considered, because the number of epochs will affect the performance of the model. This study used 25 as parameter epochs based on several considerations of the dataset used.

### 2.5 Evaluation Model

In order to evaluate the result, our research used confusion matrix, which measures the performance of a model by comparing the predicted class with the true and false class. The confusion matrix will be used to calculate the accuracy, precision, recall and f1 score metrics [22], as shown in Fig 5.

		Predicted Class	
		Positive (1)	Negative (0)
Actual Class	Positive (1)	<b>TP</b> (True Positive)	<b>FN</b> (False Positive)
	Negative (0)	<b>FP</b> (False Positive)	<b>TN</b> (True Negative)

**Fig. 4.** Confusion *Matrix*

Based on Fig. 4, the formula of accuracy, precision, recall and F1 score is determined by (6) – (9).

$$Accuracy = (TP + TN) / (TP + FP + TN + FN) \quad (6)$$

$$Precision = (TP) / (TP + FP) \quad (7)$$

$$Recall = (TP) / (TP + FN) \quad (8)$$

$$F1\ Score = 2(Precision \cdot Recall) / (Precision + Recall) \quad (9)$$

These metrics are important for evaluating the effectiveness of a classification model and can guide further improvements or adjustments to enhance its performance. By calculating accuracy, precision, F1 score and recall values for each class, we can gain a comprehensive understanding of how well the model is performing across different classifications. This information can be crucial to evaluating the overall performance of a classification model and adjustments or fine-tuning model in making decisions.



### 3 Result and Discussion

The experiment was performed to evaluate the effectiveness of our method to classify the X-ray image. We used three classes data, which are COVID, pneumonia and normal cases. The total number of images was 20384 consisting of 18344 training data and 2040 test data. The training process works with an image dataset of size 150 x 150 pixels. To improve the classification result, we used a data augmentation technique by applying random transformations such as rotation, rescale and zoom to the input images. The process generated varied data. A rectified Adam optimizer is used to perform model training with 25 epochs and a learning rate value set at 0.0001, as shown in Fig. 5. We ran our program on Google Collaboratory which is designed to implement machine learning solutions using the Python programming language. It can be used without any hardware requirements and installation and provides free GPU access. The validation is done after the training process to find out how good the model is in making predictions. This can be done by displaying graphs for train accuracy, train loss, train accuracy and validation accuracy. As shown in Fig. 6, there are four important points that will illustrate changes in model performance during the training process which are train loss, validation loss, train accuracy and validation accuracy. Train loss measures the value of the loss function during the training process of the model against the training data. Validation loss measures the loss value during the training process on validation data. Train accuracy measures the model's performance in making correct predictions on the training data. Validation accuracy measures the model's performance in making correct predictions on data that is not in the training. The training and validation accuracy values have a very small range around 0.2 indicating that the model has learned the pattern well. The train and validation accuracy values also show the improvement during training. However, there is instability in both graphs. Some of the factors causing the fluctuations in the graphs are that the model is too specialized and complex but still within normal limits.

```

Epoch 1/25
1834/1834 [=====] - 73s 39ms/step - loss: 0.6994 - accuracy: 0.6746 - precision: 0.7003 - recall: 0.6112 - auc: 0.8589 - categorical_crossentropy: 0.6994 --
Epoch 2/25
1834/1834 [=====] - 76s 41ms/step - loss: 0.5471 - accuracy: 0.7553 - precision: 0.7701 - recall: 0.7337 - auc: 0.9140 - categorical_crossentropy: 0.5471 --
Epoch 3/25
1834/1834 [=====] - 75s 41ms/step - loss: 0.4823 - accuracy: 0.7914 - precision: 0.7990 - recall: 0.7704 - auc: 0.9339 - categorical_crossentropy: 0.4823 --
Epoch 4/25
1834/1834 [=====] - 240s 131ms/step - loss: 0.4473 - accuracy: 0.8115 - precision: 0.8179 - recall: 0.8019 - auc: 0.9433 - categorical_crossentropy: 0.4473 --
Epoch 5/25
1834/1834 [=====] - 75s 41ms/step - loss: 0.4141 - accuracy: 0.8271 - precision: 0.8326 - recall: 0.8198 - auc: 0.9507 - categorical_crossentropy: 0.4141 --
Epoch 6/25
1834/1834 [=====] - 939s 512ms/step - loss: 0.3973 - accuracy: 0.8383 - precision: 0.8438 - recall: 0.8312 - auc: 0.9552 - categorical_crossentropy: 0.3973 --
Epoch 7/25
1834/1834 [=====] - 73s 40ms/step - loss: 0.3699 - accuracy: 0.8509 - precision: 0.8558 - recall: 0.8454 - auc: 0.9610 - categorical_crossentropy: 0.3699 --
Epoch 8/25
1834/1834 [=====] - 208s 113ms/step - loss: 0.3553 - accuracy: 0.8590 - precision: 0.8626 - recall: 0.8531 - auc: 0.9638 - categorical_crossentropy: 0.3553 --
Epoch 9/25
1834/1834 [=====] - 73s 40ms/step - loss: 0.3487 - accuracy: 0.8692 - precision: 0.8642 - recall: 0.8527 - auc: 0.9654 - categorical_crossentropy: 0.3487 --
Epoch 10/25
1834/1834 [=====] - 89s 48ms/step - loss: 0.3380 - accuracy: 0.8692 - precision: 0.8736 - recall: 0.8638 - auc: 0.9675 - categorical_crossentropy: 0.3380 --
Epoch 11/25
1834/1834 [=====] - 72s 39ms/step - loss: 0.3315 - accuracy: 0.8728 - precision: 0.8763 - recall: 0.8687 - auc: 0.9685 - categorical_crossentropy: 0.3315 --
Epoch 12/25
1834/1834 [=====] - 89s 48ms/step - loss: 0.3207 - accuracy: 0.8764 - precision: 0.8862 - recall: 0.8713 - auc: 0.9704 - categorical_crossentropy: 0.3207 --
Epoch 13/25
...
Epoch 24/25
1834/1834 [=====] - 72s 39ms/step - loss: 0.2843 - accuracy: 0.8960 - precision: 0.9000 - recall: 0.8911 - auc: 0.9769 - categorical_crossentropy: 0.2843 --
Epoch 25/25
1834/1834 [=====] - 71s 39ms/step - loss: 0.2732 - accuracy: 0.8971 - precision: 0.9003 - recall: 0.8929 - auc: 0.9785 - categorical_crossentropy: 0.2732 --

```

Fig. 5. The output of the model training program with 25 epochs



**Fig. 6.** Training loss, validation loss, train accuracy and validation accuracy of model CNN

The confusion matrix displays the number of correct and incorrect predictions made by the model against the validation data. Confusion matrix evaluates the performance of the model by comparing the prediction results with the actual value based on the observed or validation data. Predictions will be made for each class namely COVID, normal and pneumonia. Fig. 7 shows the classification results in the form of a confusion matrix for each prediction of each class with batch size 10 and epoch 25 parameters.

According to Fig.7, we can find that within 2040 validation data, there are 1872 true predictions (91,76 %) and 168 prediction errors (8.24 %). Analysis of the confusion matrix results also shows the prediction errors for each class. The prediction error in the COVID class was 16.95 %, the pneumonia class was 9.52% and normal class was 4.08 %. The prediction error in the normal class is the smallest compared to other classes. This is because the dataset in the normal class has a larger number. The diversity of images and the large number of datasets will affect the training results on model performance. That is why CNN architecture will produce a good model on a large dataset. The larger the dataset, the better the performance of the model in classifying. However, if we observe the number of COVID data is more than the pneumonia class. When viewed from the amount of data available, the COVID class has more data than pneumonia. The COVID class has 525 and the pneumonia class has 315 validation data. The prediction errors of COVID class is more than the pneumonia class. It may be caused by the similar features of those classes. It can be a challenge for the future research to identify COVID symptoms from X-ray images.

The evaluation is also carried out by measuring the model’s performance by displaying performance metrics including accuracy, recall, and F1 score, as listed in Table 4. According to Table 4, the average accuracy is 90%, the precision is 92%, the recall is 90%, and the F1 score is 91%. These evaluation results show that the model has been trained using our CNN architecture and has a good ability to perform classification

tasks with high accuracy and precision values. The recall value also shows that the model has the ability to recall most of the relevant data to minimize the existence of undetected data. Then, the F1 score value shows that the model has a good balance between precision and recall.

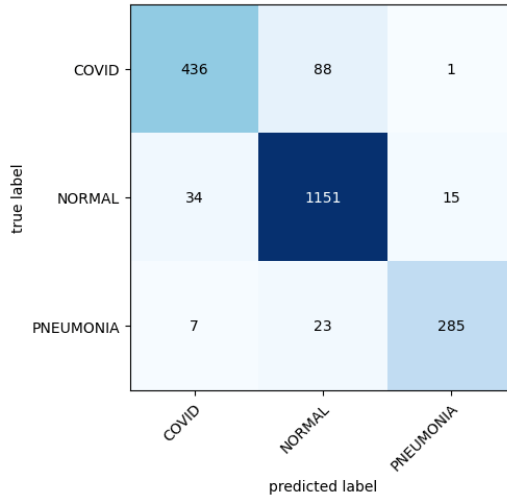
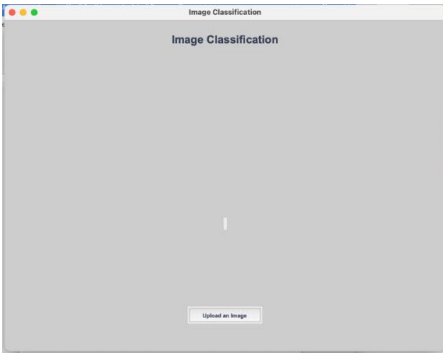


Fig. 7. Experimental result on the confusion matrix

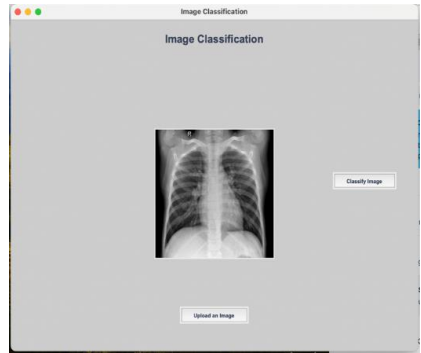
Table 4. Performance Evaluation

Class	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
<b>Covid</b>	83	91	83	87
<b>Normal</b>	96	91	96	94
<b>Pneumonia</b>	90	95	90	93
<b>Average</b>	90	92	90	91

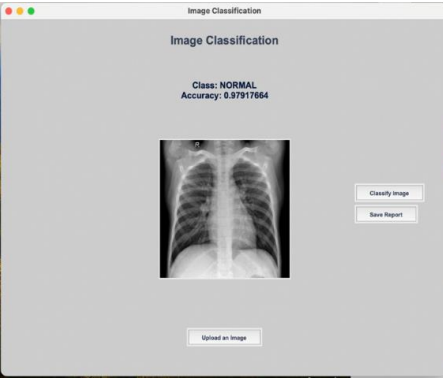
We developed our system with integration into a GUI in order to ease for users to use the system, as shown in Fig. 8. The GUI consists of Upload Image, Classification and Save Report buttons. The initial view of GUI is shown in Fig. 8.a. Once the Upload button is clicked, the image will be displayed on the picture box and the Classification button become active (Fig. 8.b.). When the Classification button is clicked, the classification class and the accuracy result are displayed(Fig. 8.c.). As shown in the figure, the classification result is COVID class and the accuracy rate is 0.979 (97.9 %). Then the classification result is saved in the pdf file (Fig. 8.d). Some of classification result of each class is shown in Fig. 9, while the file report sample is shown in Fig. 10.



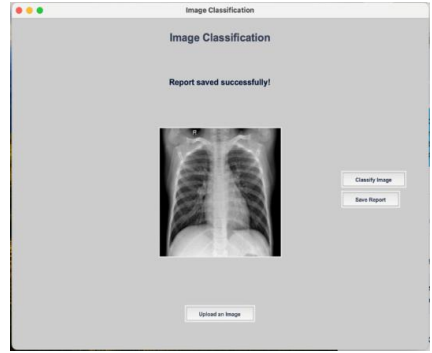
a. The initial view of GUI



b. After upload image



c. After classification

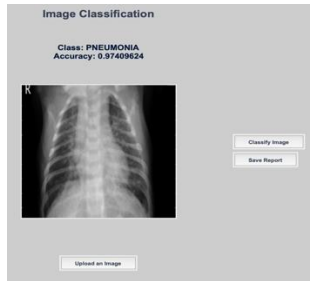


d. After report is successfully saved

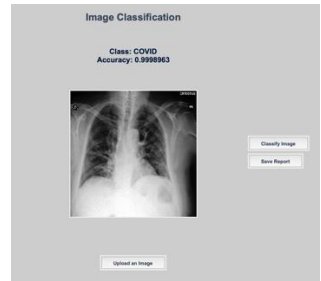
**Fig. 8.** System GUI



a. Normal



b. Pneumonia



c. Covid

**Fig. 9.** Sample of the classification result

Classification Report

Class: NORMAL  
 Accuracy: 0.98609436  
 File Path: /Users/teknikelektro/Documents/WOI/Validasi/NORMAL/NORMAL\_42.png  
 Kondisi pleural: Normal  
 Kondisi diafragma: Normal  
 Lung Opacity: Normal



**Fig. 10.** Sample of report file of normal class

A comparison of our method to classify the lung disease using CNN s to the previous studies is shown in table 5. It shows that our method has high accuracy compare to previous method. This is possibly because of the huge number of data used in our experiment. We used 20384 X-Ray images for 3 classes of classification.

**Table 5.** Comparison result

Application	Method	Dataset	Number images/classes	Accuracy
Lung Texture Classification and Airway Detection	Convolutional Restricted Boltzmann Machine [32]	ILD (interstitial lung diseases ) CT scans	73 CT scans 5 classes	89%
Lung Pattern Classification	Convolutional Neural Network [33]	ILD (interstitial lung disease) CT scans	109 high-resolution CT scans 7 classes	85.5%
Detection and Classification of Nuceli	Two architectures of Convolutional Neural Network [34]	histology images of colorectal adenocarcinomas	100 histology images 4 classes	80.2%
Our Method: Lung Diseases Classification	Convolutional Neural Network	X-ray dataset	20384 X-Ray Images 3 classes	92%

## 4 Conclusion and Future Work

This study implemented one of the AI technologies using CNN to classify the X-ray image as a second opinion for a medical doctor's diagnosis. The study has implemented three stages of the classification process, which are preprocessing, training and validation process. The system is integrated into GUI to display the classification result. We used 20384 dataset consisting of 5243 COVID cases, 11995 normal cases and 3146 pneumonia cases. We divide the data into 90% data for training and 10% for validation Data. The experimental results are evaluated using a confusion matrix and showed the performance evaluation of image classification as follows: average accuracy is 92 %, precision 92%, recall 90% and F1-score 91 %. In addition, the deployed GUI successfully displayed the X-ray image with classification result and the accuracy value. The GUI is also equipped with the report of the classification result in the pdf file. However, the system still has limitations that need further improvement. With the collaboration of medical doctors, the system can be developed to real medical images and to other modalities like CT or MRI. These are remaining for future work.

## References

- [1] S. Hossain, S. Umer, V. Asari, and R. K. Rout, "A Unified Framework of Deep Learning-Based Facial Expression Recognition System for Diversified Applications," *Appl. Sci.*, vol. 11, no. 19, p. 9174, 2021.
- [2] A. Alsobhani, H. M. A. ALabboodi, and H. Mahdi, "Speech Recognition using Convolution Deep Neural Networks," *J. Phys. Conf. Ser.*, vol. 1973, no. 1, p. 012166, 2021.
- [3] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," *J. Am. Med. Informatics Assoc.*, vol. 18, no. 5, pp. 544–551, 2011.
- [4] L. Aguiar-Castillo, A. Clavijo-Rodriguez, L. Hernández-López, P. De Saa-Pérez, and R. PérezJiménez, "Gamification and deep learning approaches in higher education," *J. Hosp. Leis. Sport Tour. Educ.*, vol. 29, p. 100290, Nov. 2021
- [5] N. Q. Do, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita, "Phishing Webpage Classification via Deep Learning-Based Algorithms: An Empirical Study," *Appl. Sci.*, vol. 11, no. 19, p. 9210, 2021.
- [6] A. Anand, S. Rani, D. Anand, H. M. Aljahdali, and D. Kerr, "An Efficient CNN-Based Deep Learning Model to Detect Malware Attacks (CNN-DMA) in 5G-IoT Healthcare Applications," *Sensors*, vol. 21, no. 19, p. 6346, 2021.
- [7] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.
- [8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [9] E. Çallı, E. Sogancioglu, B. van Ginneken, K. van Leeuwen, and K. Murphy, "Deep learning for chest X-ray analysis: A survey", *Medical Image Analysis*, vol 72, 2021.
- [10] H. Haneya, D. AlKaf, F. Bajammal, and T. Brahimi, "A Meta-Analysis of Artificial Intelligence Applications for Tracking COVID-19: The Case of the U.A.E ". *Procedia Computer Science*, vol. 194, pp. 180–189. 2021.
- [11] S. Anai, J. Hisasue, Y. Takaki, and N. Hara, "Deep Learning Models to Predict Fatal Pneumonia Using Chest X-Ray Images". *Canadian Respiratory Journal*, pp. 1–12, 2022
- [12] P. Lakhani, A. Prater, R. Hutson, K. Andriole, K. K. Dreyer, J. Morey, L. Prevedello, et al, "Machine Learning in Radiology: Applications Beyond Image Interpretation," *Journal of the American College of Radiology*, Vol. 15, No. 2, pp. 350–359, 2018.
- [13] H. Ali and Z. Shah, Z, "Combating COVID-19 Using Generative Adversarial Networks and Artificial Intelligence for Medical Images: Scoping Review ", *JMIR Med Inform*, Vol. 10, No. 6, 2022

- [14] V. Magni, A. Cozzi, S. Schiaffino, A. Colarieti, and F. Sardanelli, "Artificial intelligence for digital breast tomosynthesis: Impact on diagnostic performance, reading times, and workload in the era of personalized screening". *European Journal of Radiology*, Vol. 158, 2023
- [15] L. Chen, P. Chen, and Z. Lin, "Artificial Intelligence in Education: A Review. *IEEE Access*, Vol. 8, pp. 75264–75278 2020
- [16] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, et al, "International evaluation of an AI system for breast cancer screening", *Nature*, Vol. 577, No. 7788, pp. 89–94. 2020
- [17] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence", *Nature Medicine*, Vol. 25, No. 1, pp. 44–56, 2021
- [18] A. Anton, N. Nissa, A. Janiati, N. Cahya, and P. Astuti, "Application of Deep Learning Using Convolutional Neural Network (CNN) Method for Women Skin Classification", *Scientific Journal of Informatics*, Vol. 8, No. 1, pp. 144-153, 2021
- [19] M. Farayola, and A. Dureja, " A Proposed Framework: Face Recognition With Deep Learning", *International journal of scientific & technology research*, Vol. 9, No. 7, 2020
- [20] Y. Mulyani, D. Septiangraini, M. A. Muhammad, and G. Forda, "Comparison Study of Convolutional Neural Network Architecture in Aglaonema Classification", *Int. J. Electron. Commun. Syst*, Vol. 2, No. 2, pp. 75–83, 2020
- [21] V. Thamilarasi, and R. Roselin, "Automatic Classification and Accuracy by Deep Learning Using CNN Methods in Lung Chest X-Ray Images", *IOP Conference Series: Materials Science and Engineering*, Vol. 1055, No. 1, 2021
- [22] M. Fontanellaz, I. Ebner, A. Huber, A. Peters, L. Löbelenz, C. Hourscht, J. Klaus, J. Munz, et al, "A Deep-Learning Diagnostic Support System for the Detection of COVID-19 Using Chest Radiographs: A Multireader Validation Study". *Investigative Radiology*, Vol.56, No. 6, 2021
- [23] S. H. Khan, A. Sohail, A. Khan, M. Hassan, Y. S. Lee, J. Alam, A. Basit, and S. Zubair, "COVID-19 detection in chest X-ray images using deep boosted hybrid learning". *Computers in Biology and Medicine*, Vol. 137, 2021
- [24] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", *Journal of Big Data*, Vol.6, No. 1, 2019.
- [25] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects,"*IEEE transactions on neural networks and learning systems*, 2021.
- [26] Goumiri, D. Benboudjema, and W. Pieczynski, "A new hybrid model of convolutional neural networks and hidden Markov chains for image classification". *Neural Computing and Applications*, Vol. 35, No. 24, pp. 17987–18002, 2023
- [27] C. Yan, X. Wang, X. Liu, W. Liu and J. Liu, "Research on the UBI Car Insurance Rate Determination Model Based on the CNN-HVSVM Algorithm", *IEEE Access*, Vol. 8, pp. 160762–160773, 2020
- [28] D. Krstinić, M. Braović, L. Šerić, and D. Božić, "Multi-label Classifier Performance Evaluation with Confusion Matrix", *Computer Science and Information Technology*, pp. 01–14, 2020
- [29] M. Yaqub, F. Jinchao, M. S. Zia, K. Arshid, K. Jia, Z. U. Rehman and A Mehmood, "State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images", *Brain Sciences*, Vol. 10, No. 7, 2020.
- [30] H. Saad and A. M. Abdulazeez, "Comparison of optimization techniques based on gradient descent algorithm: a review". *PalArch's Journal of Archaeology of Egypt*, Vol. 18, No. 4, pp. 2715–2743, 2021
- [31] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical ImageAnalysis using Convolutional Neural Networks: A Review," *J. Med. Syst.*, vol. 42, no. 11, p. 226, 2018.
- [32] G. van Tulder, M. de Bruijne, Combining generative and discriminative representation learning for lung ct analysis with convolutional restricted boltzmann machines, *IEEE transactions on medical imaging* 35 (5) (2016) 1262-1272.
- [33] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, S. Mougiakakou, Lung pattern classification for interstitial lung diseases using a deep convolutional neural network, *IEEE transactions on medical imaging* 35 (5) (2016) 1207-1216
- [34] A. Qayyum, S. M. Anwar, M. Awais, M. Majid, Medical image retrieval using deep convolutional neural network, *Neurocomputing* 266 (2017) 8-20.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

