CLASSIFICATION OF CORONAVIRUS DISEASE (COVID-19) THROUGH CHEST X-RAY IMAGES BASED ON DEEP LEARNING

Erzha Anges Farhandy[1];Sukma Wardhana[2]

Program Studi Teknik Informatika[12]
Universitas Mercu Buana[12]
Jakarta, Indonesia
https://www.mercubuana.ac.id[12]
41518110005@student.mercubuana.ac.id[1],sukma@mercubuana.ac.id[2]
(*) Corresponding Author

Abstract— CoV-2 virus this disease is spreading rapidly throughout the world. Various studies were carried out to control the spread of Covid-19. One way to detect Covid-19 is to study chest X-ray images of patients with Covid-19 symptoms. However, to detect Covid-19 through x-ray images, there are currently few radiology specialists needed. This study researched to detection of Covid-19 disease through chest x-ray images with a deep learning approach based on a convolutional neural network (CNN). Before training the model, data preprocessing is carried out, such as labelling and resizing. This study uses a CNN model with three layers of convolution and max-pooling layers and a fully-connected layer for the output. The results of the training using the CNN method produced a pretty good performance, with the best training accuracy (acc) value obtained in the 31st epoch with a value of 0.9593, training loss (loss) 0.1306, validation accuracy (val_acc) 0.9604, and loss validation (val_loss). 0.1399.

Keywords: Classification, Covid-19, CNN, deep learning, chest x-ray

Abstrak—Penyakit *Coronavirus* (COVID19) adalah penyakit menular yang disebabkan oleh virus *Sars-CoV-2*, penyakit ini menular dengan cepat di seluruh dunia. Berbagai penelitian pun dilakukan untuk mengontrol penyebaran Covid-19. Salah satu cara untuk mendeteksi Covid-19 yaitu dengan mempelajari citra *chest x-ray* pasien dengan gejala Covid-19. Namun untuk mendeteksi Covid-19 melaui citra x-ray dibutuhkan dokter spesialis radiologi yang saat ini jumlahnya masih sedikit. Dalam studi ini, penelitian dilakukan untuk mendeteksi penyakit Covid-19 melalui citra *chest x-ray* dengan pendekatan *deep learning* berbasis *convolutional neural network* (CNN). sebelum melatih model, dilakukan *preprocessing* data seperti pelabelan dan pengubahan ukuran. Penelitian ini menggunakan model CNN dengan 3 lapis *layer convolution* dan *maxpooling* serta *fully-connected layer* untuk *output*. Hasil *training* menggunakan metode CNN menghasilkan performa yang cukup baik, dengan nilai akurasi (acc) pelatihan terbaik diperoleh pada epoch ke-31 dengan nilai 0,9593, training loss (loss) 0,1306, validasi accuracy (val_acc) 0,9604, dan validasi loss (val_loss). 0.1399.

Kata Kunci: *Klasifikasi, Covid-19, CNN, deep learning, chest x-ray*

INTRODUCTION

At the end of 2019, a severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) was discovered. The virus originating from China is the cause of a disease known as Corona Virus Disease 2019 or Covid-19(El-Kenawy, Ibrahim, Mirjalili, Eid, & Hussein, 2020). *The World Health Organization (WHO) declared the disease a pandemic in March 2020. The Covid-19 virus has a fast spread rate, so the World Health Organization (WHO) is designated it a global pandemic* (Ayumi & Nurhaida, 2021). Covid-19 is highly contagious and has the potential to develop into a fatal acute respiratory distress syndrome. Early detection and diagnosis are essential factors in controlling the spread of Covid-19 (Silva et al., 2020). Currently, real-time-reverse-transcriptase-polymerase-chain-reaction (RT-PCR) is widely used as a testing standard to confirm Covid-19 disease. However, RT-PCR testing is too time-consuming and relatively expensive (Wu et al., 2020).

Chest radiography (X-ray) is one of the essential methods to diagnose Covid-19 worldwide. By looking at chest X-rays, Covid-19 can only be diagnosed by a specialist. The number of specialists who can make this diagnosis is less than that of ordinary doctors (Narin, Kaya, & Pamuk, 2021). One alternative testing method that might be used is to utilize artificial intelligence to classify whether or not a person is exposed to COVID-19 based on specific features. One feature that could be used is an x-ray image of the chest(Jaiswal, Gianchandani, Singh, Kumar, & Kaur, 2021).

Vol. 18, No. 2 September 2022 | DOI: 10.33480/pilar.v18i1.3293

In artificial intelligence, there are several algorithms, one of which uses KNN and CNN. Using KNN, the accuracy rate of correct prediction results in recognizing tuberculosis disease through chest x-ray images was 71.81% (Muhathir, Theofil Tri Saputra, & Al-Khowarizmi, 2020). Meanwhile, the use of CNN to predict the type of animal is 97.56%(Dhika, Kurnianda, Irfansyah, & Ananta, 2020).

Neural network-based techniques are often used to solve problems with artificial intelligence approaches because of their excellent performance. In recent years, several studies have detected the disease using CNN. One of the related studies, "Performance of the CNN Method for Pneumonia Classification with Input Image Size Variations," was conducted by Budi Nugroho et al. The classification process uses the flattening, fully connected layer, and dense function and obtains an accuracy value of up to 93.59% (Nugroho & Puspaningrum, 2021).

The following research entitled "Comparative Analysis of Chest X-ray Image Classification Algorithm for Covid-19 Detection" conducted by Mohammad Farid Naufal et al. In this study, a comparison of algorithms was carried out, namely the KNN, SVM, and CNN algorithms. The test results showed CNN got an accuracy value of 0.95%, SVM 0.93%, and KNN 0.92% (Naufal et al., 2021).

Based on this, the authors propose research on the classification of Covid-19 through chest X-ray images using convolutional neural networks with different datasets to see if CNN's accuracy can be stable and perform well. In classifying objects, the CNN algorithm can process the complete information without losing accuracy (Dhika et al., 2020).

The implementation of the research will begin with collecting datasets, then processing the data that has been obtained and testing algorithmic models on the data that has been processed. Then implement the algorithm into the system. With this research, it is hoped that it can help health workers to detect the Covid-19 pandemic.

MATERIALS AND METHODS

The following is a stage of research to determine the performance of the Convolutional Neural Network algorithm in classifying covid-19 patients through chest x-ray images.

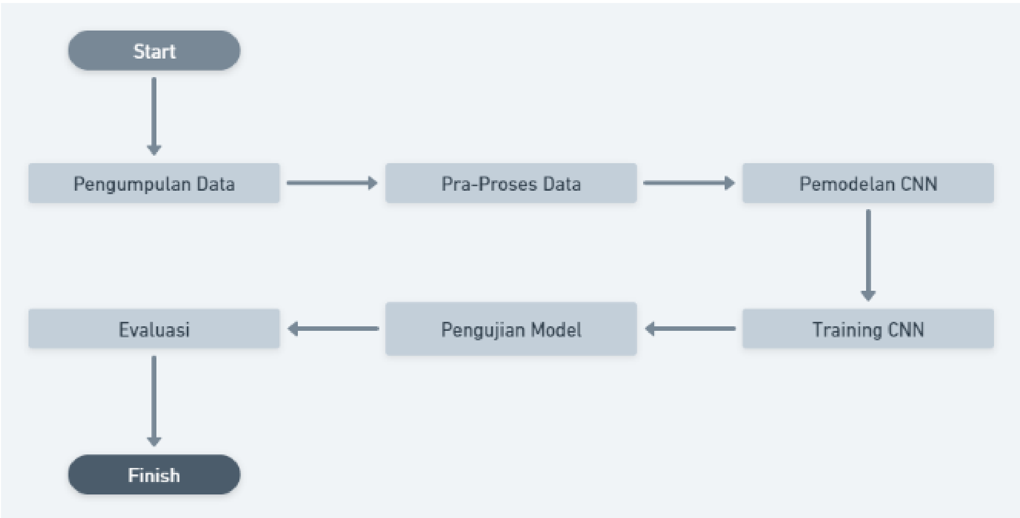

Figure 1. Stages of Research

Below is an explanation of each stage of research in Figure 1 above.

Data Collection

This study's dataset uses chest X-ray images obtained from Kaggle (Rahman, 2022), named Covid-19 Radiography Database. This dataset is the result of a collaboration created by a team of researchers from Qatar University and the University of Dhaka with several medical personnel. The number of images used in this study was 14,508, divided into three classes: the standard class of 10,192 images, the Covid-19 class of 3,616 images, and the TB class of 700 images. PNG image format. Before conducting the training and testing process, researchers first label the data so that the data is easily identifiable—the following overview of the dataset processed for classifying chest x-ray images.

Preprocess Data

The preprocessing stage of image data is carried out by labelling images and resizing pixels. The process of labelling data serves to give names to the data to be recognizable. Researchers created three main folders: the standard folder for image data indicated usually, the covid folder for image data indicated by covid, and the tuberculosis folder for image data indicated by tuberculosis. Then during the coding process, the dataset folder is changed to healthy for the standard folder, Covid for the Covid folder, and tuberculosis for the tuberculosis folder, as shown in figure 2.

	image_file	corona_result	path	image
0	Normal/Normal-1.png	Healthy	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
1	Normal/Normal-10.png	Healthy	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[9, 9, 9, 8, 8, 7, 6, 4, 3, 2, 1, 1, 1, 0, 0,...
2	Normal/Normal-100.png	Healthy	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[0, 0, 0, 0, 12, 57, 75, 95, 111, 117, 119, 1...
3	Normal/Normal-1000.png	Healthy	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[0, 6, 23, 51, 66, 69, 71, 67, 97, 96, 101, 1...
4	Normal/Normal-10000.png	Healthy	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[136, 118, 106, 100, 96, 93, 91, 88, 87, 87, ...
...	...	...	...	...
14503	Tuberculosis/Tuberculosis-95.png	TBC	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[0, 1, 4, 0, 0, 0, 0, 0, 0, 5, 7, 8, 10, 13, ...
14504	Tuberculosis/Tuberculosis-96.png	TBC	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[0, 0, 72, 169, 185, 183, 183, 182, 185, 190,...
14505	Tuberculosis/Tuberculosis-97.png	TBC	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[210, 215, 190, 133, 120, 141, 154, 165, 176,...
14506	Tuberculosis/Tuberculosis-98.png	TBC	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[207, 212, 174, 122, 108, 107, 108, 115, 131,...
14507	Tuberculosis/Tuberculosis-99.png	TBC	F:/UMB/MPTI/X-ray/COVID-19_Radiography_Dataset...	[[219, 220, 222, 221, 221, 221, 222, 222, 223,...

Figure 2. Data that has been labelled

After labelling, before proceeding to the classification stage, the image data used is quite a lot, so it needs to be adjusted to the image pixels so that it does not take up much storage, making it easier to carry out the data processing process. The image dataset in the form of pixel image data is converted into 75 x 75 pixels using the source code as in Figure 3.

```
data['image'] = data['path'].map(lambda x: np.asarray(Image.open(x).resize((75,75))))
# resize image dengan resolusi 75x75
```

Figure 3. Source Code resize an image

The number of X-ray datasets with three labels of healthy, covid-19, and tuberculosis can be seen in figure 4. X-ray dataset display with healthy polarity, covid-19, and tuberculosis is shown in Figure 5.

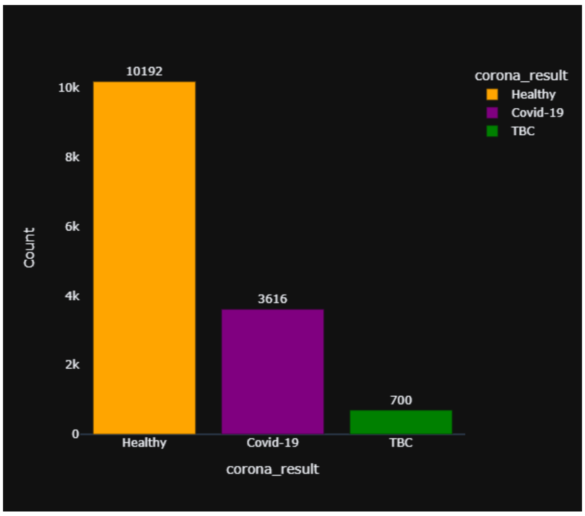

Figure 4. Dataset Composition

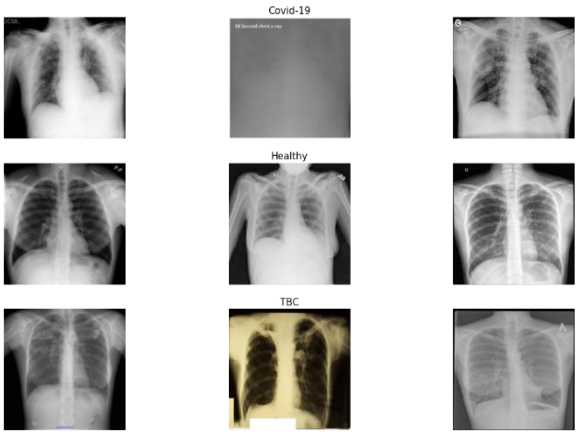

Figure 5. Data that has been labelled

CNN Modelling

This classification stage will use the Convolutional Neural Network algorithm model using the Adam optimization function. Adam Optimization can improve the accuracy and performance of neural network algorithms(Jais, Ismail, & Nisa, 2019). CNN is a neural network commonly used in data images (Allaouzi & Ben Ahmed, 2019). At this stage, training testing data will be processed, and then the accuracy level will be calculated

Training CNN

Objective: Improve the accuracy of chest X-ray picture classification using object identification techniques. The algorithm must first be taught with some training data. The purpose of training this algorithm is to find the characteristics of each image and then mark which neurons will be activated when the image is classified. Therefore, it is

Vol. 18, No. 2 September 2022 | DOI: 10.33480/pilar.v18i1.3293

necessary to create a scheme or model to conduct training on the algorithm so that it has been trained when object recognition testing is carried out. The algorithm used in this research, Convolutional Neural Network (CNN), is an architecture of deep learning. CNN includes many layers of representation. CNN structure consists of feature extraction, consisting of a convolutional layer usually followed by a pooling layer and a softmax classifier. The convolutional layer extracts features from images, while the pooling layer reduces dimensions and computation time. This architecture can achieve this form of regularization by itself. The extracted features are then fed into the upper softmax layer for classification (Marques, Agarwal, & de la Torre Díez, 2020).

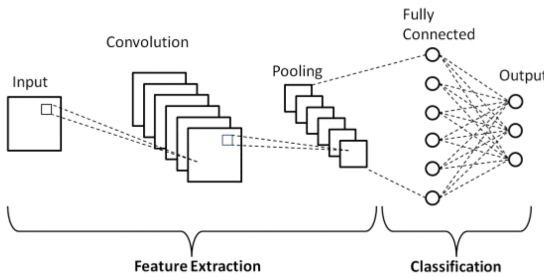

Figure 6. CNN Architecture

The process on CNN is shown in Figure 6. starting from the input, in this process, data in the form of image imagery is entered, data taken from each pixel of ×wide×1 image for black and white imagery (grayscale) and length×width×3 for imagery with colour (RGB). The next stage is the extraction of features from the image. In this section, "encoding" from an image into features in the form of numbers representing the image. Feature extraction has two main parts: the convolutional layer and the pooling layer. In the Feature extraction process, the number of convolutional layers and pooling layers can be adjusted to the needs. The more the number, the deeper the architecture, thus increasing the accuracy of classification (Jogin et al., 2018). The classification process has several hidden layers, an activation function, and a loss function. The process in the softmax classifier converts the log alias number into a probability of one, and the softmax classifier generates a vector representing the probability of a list of possible results (labels).

Model Testing and Evaluation

After conducting data training on the Convolutional Neural Network algorithm, results were obtained from image classification capabilities, which were assessed from the accuracy, precision, recall, and f1-score obtained based on data training of chest x-ray. Accuracy is comparing the correctly predicted amount of data to the total amount. The greater the accuracy value, the better the performance of the algorithm. Here is the similarity:

$$\text{Accuracy} = \frac{TP+TN}{(TP+FP+TN+FN)} \quad \ldots\ldots (2)$$

Precision compares true positive (tp) to total data predicted to be positive. Precision has the following equation:

$$\text{Precision} = \frac{TP}{TP+FN} \quad \ldots\ldots (3)$$

Recall, didefinisikan sebagai perbandingan *true positive* (tp) terhadap total data positif. Recall memiliki rumusan sebagai berikut :

$$\text{Recall} = \frac{TP}{TP+FP} \quad \ldots\ldots (4)$$

F1-Score, which is the average value obtained from weighted Recall and Precision. F1-Score has the following formulation:

$$\text{F1} - \text{Score} = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \quad \ldots\ldots (5)$$

Evaluation of the model is carried out by reviewing the accuracy, precision, recall, and f1-score values. If the value is insufficient, then an evaluation is carried out to improve its accuracy. It is essential to get the best accuracy value so that the chest x-ray image classification has accurate results in determining the classification of positive covid-19 and negative covid-19.

RESULTS AND DISCUSSION

The study's outcomes employing the CNN algorithm to classify chest x-ray images will be discussed at this point. The main libraries include TensorFlow, Keras, pandas, NumPy, and Matplotlib. After the dataset is prepared, retrieving features from each image and resizing with a size of 75x75 is carried out.

```
all_data = []

# Storing images and their labels into a list for further Train Test split

for i in range(len(data)):
    image = cv2.imread(data['path'][i])
    image = cv2.resize(image, (70, 70)) / 255.0
    label = 1 if data['corona_result'][i] == "Covid-19" else 2 if data['corona_result'][i] == "TBC" else 0
    all_data.append([image, label])
```

Figure 7. Source Code Retrieval Features

After taking the image feature, splitting the dataset is carried out so that it will be split into data train, test data, and validation data.

```
x = []
y = []

for image, label in all_data:
    x.append(image)
    y.append(label)

# konvesi x dan y ke dalam array numpy
x = np.array(x)
y = np.array(y)

# mengsplit data train dan data test, mengambil data test sebanyak 0.2 artinya 20% dari keseluruhan data secara random
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)
# mengsplit data train dan data validasi, mengambil data validasi sebanyak 0.1 artinya 10% dari keseluruhan data secara random
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size = 0.1, random_state = 42)

# menampilkan sebaran data training, data testing dan data validasi
print(x_train.shape, x_test.shape, x_val.shape, y_train.shape, y_test.shape, y_val.shape)
```

Figure 8. Splitting Datasets

The next stage is the formation of a CNN model using the Hard Library and Tensorflow as follows:

```
cnn_model = models.Sequential()
cnn_model.add(layers.Conv2D(filters = 128, kernel_size = (3, 3), activation = 'relu', input_shape = (70, 70, 3)))
cnn_model.add(layers.MaxPooling2D((2, 2)))
cnn_model.add(layers.Dropout(0.3))

cnn_model.add(layers.Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
cnn_model.add(layers.MaxPooling2D((2, 2)))
cnn_model.add(layers.Dropout(0.5))

cnn_model.add(layers.Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
cnn_model.add(layers.Flatten())
cnn_model.add(layers.Dense(units = 16, activation = 'relu'))
cnn_model.add(layers.Dropout(0.2))

cnn_model.add(layers.Dense(units = 2))

cnn_model.compile(optimizer = 'adam',
          loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits = True),
          metrics = ['accuracy'])

# Menampilkan Summary (Ringkasan) Model
cnn_model.summary()
```

Figure 9. Cnn Model Formation

After making the model, the training process with epoch 50 is carried out. It takes 130 minutes,

PILAR Nusa Mandiri: Journal of Computing and Information System
Vol. 18, No. 2 September 2022 | DOI: 10.33480/pilar.v18i1.3293

```
In [16]: es = tf.keras.callbacks.EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 1, patience = 4)

         #tf.random.set_seed(42)
         history = cnn_model.fit(x_train, y_train,
                                 epochs = 50, batch_size = 256,
                                 validation_data = (x_val, y_val),
                                 callbacks = [es])

         Epoch 1/50
         41/41 [==============================] - 49s 1s/step - loss: 0.7563 - accuracy: 0.6885 - val_loss: 0.6616 - val_accuracy: 0.686
         5
         Epoch 2/50
         41/41 [==============================] - 48s 1s/step - loss: 0.6069 - accuracy: 0.7342 - val_loss: 0.5330 - val_accuracy: 0.750
         2
         Epoch 3/50
         41/41 [==============================] - 48s 1s/step - loss: 0.5045 - accuracy: 0.7733 - val_loss: 0.4197 - val_accuracy: 0.837
         2
         Epoch 4/50
         41/41 [==============================] - 47s 1s/step - loss: 0.4348 - accuracy: 0.8223 - val_loss: 0.3808 - val_accuracy: 0.853
         6
         Epoch 5/50
         41/41 [==============================] - 47s 1s/step - loss: 0.3873 - accuracy: 0.8463 - val_loss: 0.3503 - val_accuracy: 0.875
         1
         Epoch 6/50
         41/41 [==============================] - 48s 1s/step - loss: 0.3680 - accuracy: 0.8576 - val_loss: 0.3454 - val_accuracy: 0.866
         5
         Epoch 7/50
         41/41 [==============================] - 48s 1s/step - loss: 0.3527 - accuracy: 0.8630 - val_loss: 0.3405 - val_accuracy: 0.879
         4
         Epoch 8/50
         41/41 [==============================] - 48s 1s/step - loss: 0.3375 - accuracy: 0.8737 - val_loss: 0.2669 - val_accuracy: 0.905
         3
         Epoch 9/50
         41/41 [==============================] - 48s 1s/step - loss: 0.3118 - accuracy: 0.8877 - val_loss: 0.2793 - val_accuracy: 0.912
         1
         Epoch 10/50
         41/41 [==============================] - 49s 1s/step - loss: 0.3039 - accuracy: 0.8878 - val_loss: 0.2606 - val_accuracy: 0.908
```

Figure 10. Training Process

In this training, the author uses a callback function that automatically stops when for 4x consecutive accuracy values are fixed. After the training process is complete, the model is saved into a cnn_covid19.h5 file

```
# menyimpan model ke dalam file cnn_covid19.h5
cnn_model.save('cnn_covid19.h5')
```

Figure 11. Save Model CNN

From the training, it is known that performance values of this training are in figure 12 following:

Classification Report for Train Data

	precision	recall	f1-score	support
0	0.99	0.97	0.98	7338
1	0.92	0.98	0.95	2601
2	0.99	0.92	0.96	506
accuracy			0.97	10445
macro avg	0.97	0.96	0.96	10445
weighted avg	0.97	0.97	0.97	10445

Figure 12. Classification report train data

The following figure 13 is a classification report for validation data.

Classification Report for Validation Data

	precision	recall	f1-score	support
0	0.98	0.97	0.98	797
1	0.91	0.96	0.93	319
2	0.97	0.80	0.88	45
accuracy			0.96	1161
macro avg	0.95	0.91	0.93	1161
weighted avg	0.96	0.96	0.96	1161

Figure 13. Classification report validation data

The following figure 14 is a classification report for test data.

Classification Report for Test Data

	precision	recall	f1-score	support
0	0.98	0.96	0.97	2057
1	0.86	0.94	0.90	696
2	0.99	0.79	0.88	149
accuracy			0.95	2902
macro avg	0.94	0.90	0.91	2902
weighted avg	0.95	0.95	0.95	2902

Figure 14. Classification report test data

It is known in the test data that precision is 98% of Normal, 86% for Covid, and 99% for tuberculosis. Recall 0.96, 0.94, and 0.79. And F1-Score is 0.97, 0.91, and 0.88. From the results of the training, it can be known confussion_matrix for train data is as figure 15 follows:

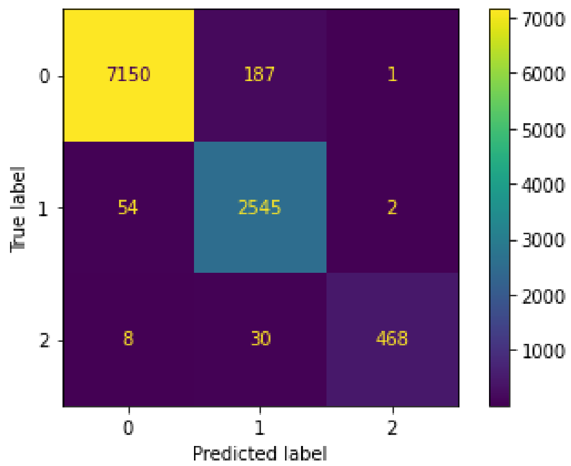

Figure 15. Confusion Matrix Train Data

Figure 16 below is a confusion matrix for data validation.

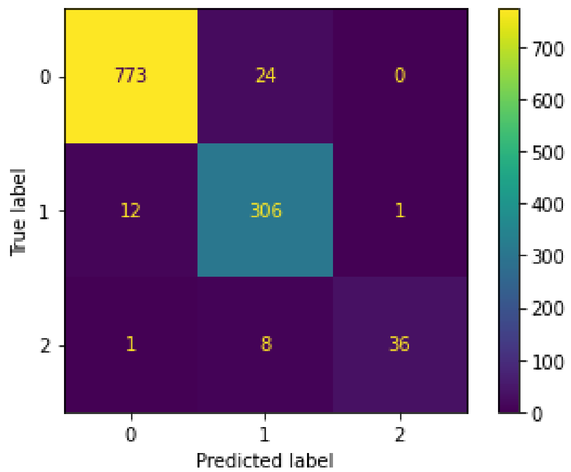

Figure 16. Confusion Matrix Data Validation

Figure 17 below is a confusion matrix for test data.

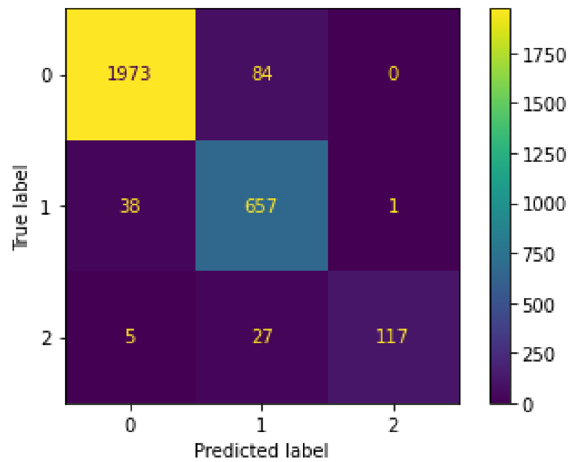

Figure 17. Confusion Matrix Test Data

While accuracy is seen in the following figure 18:

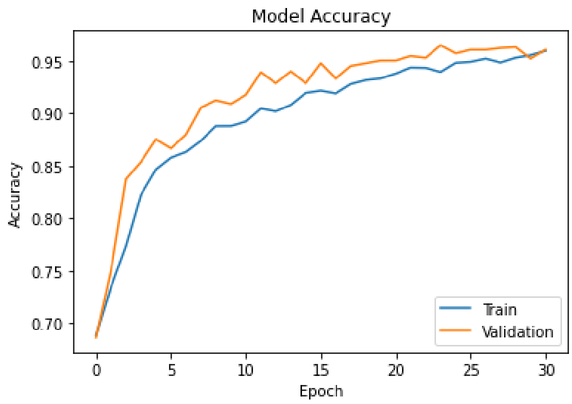

Figure 18. Model Accuracy

Figure 19 is an image visualization using gradient-weighted class activation mapping (Grad-CAM). Grad-CAM can visualize all connected neural networks. Figure 19 illustrates heat maps on chest x-ray imagery.

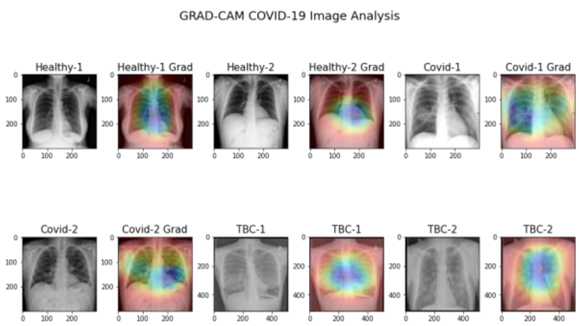

Figure 19. Representation of infected lung areas in chest x-ray imagery using Grad-CAM

Figure 20 shows a trial of a new chest x-ray image with an average target.

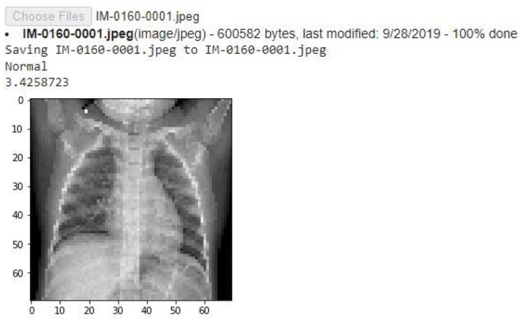

Figure 20. Trial Results

CONCLUSION

Based on the results of the trials conducted, it can be concluded that using the CNN method to classify covid-19 disease through chest x-ray images on the python platform produces quite a reasonable accuracy. With the number of data sets of 14,508 images at the time of training, accuracy was obtained by 95%. The model evaluation results with test data also showed a relatively high number: precision of 98% from Normal, 86% for Covid, and 99% for TB. Recall 0.96, 0.94, and 0.79. Moreover, F1-Score is 0.97, 0.91, and 0.88. Based on the amount of accuracy value in the CNN algorithm research, it is considered capable of obtaining accurate results in classifying chest x-ray images between normal lungs, tuberculosis, and people with Covid-19.

REFERENCE

Allaouzi, I., & Ben Ahmed, M. (2019). A Novel Approach for Multi-Label Chest X-Ray Classification of Common Thorax Diseases. *IEEE Access*, *7*, 64279–64288. https://doi.org/10.1109/ACCESS.2019.2916849

Ayumi, V., & Nurhaida, I. (2021). Klasifikasi Chest X-Ray Images Berdasarkan Kriteria Gejala Covid-19 Menggunakan Convolutional Neural Network. *JSAI (Journal Scientific and Applied Informatics)*, *4*(2), 147–153. https://doi.org/10.36085/jsai.v4i2.1513

Dhika, H., Kurnianda, N. R., Irfansyah, P., & Ananta, W. (2020). *Model Prediksi Jenis Hewan dengan Metode Convolution Neural Network*. *9*, 31–40.

El-Kenawy, E. S. M., Ibrahim, A., Mirjalili, S., Eid, M. M., & Hussein, S. E. (2020). Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2020.3028012

Jais, I. K. M., Ismail, A. R., & Nisa, S. Q. (2019). Adam Optimization Algorithm for Wide and Deep Neural Network. *Knowledge Engineering and Data Science*, *2*(1), 41. https://doi.org/10.17977/um018v2i12019p41-46

Jaiswal, A., Gianchandani, N., Singh, D., Kumar, V., & Kaur, M. (2021). Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. *Journal of Biomolecular Structure and Dynamics*, *39*(15), 5682–5689. https://doi.org/10.1080/07391102.2020.1788642

Jogin, M., Mohana, Madhulika, M. S., Divya, G. D., Meghana, R. K., & Apoorva, S. (2018). Feature extraction using convolution neural networks (CNN) and deep learning. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2018 - Proceedings*, (May 2018), 2319–2323. https://doi.org/10.1109/RTEICT42901.2018.9012507

Marques, G., Agarwal, D., & de la Torre Díez, I. (2020). Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network. *Applied Soft Computing Journal*, *96*, 106691. https://doi.org/10.1016/j.asoc.2020.106691

Muhathir, M., Theofil Tri Saputra, S., & Al-Khowarizmi, A.-K. (2020). Analysis K-Nearest Neighbors (KNN) in Identifying Tuberculosis Disease (Tb) By Utilizing Hog Feature Extraction. *Al'adzkiya International of Computer Science and Information Technology (AIoCSIT) Journal*, *1*(1), 33–38.

Narin, A., Kaya, C., & Pamuk, Z. (2021). Automatic detection of coronavirus disease (COVID-19) using X-Ray. *Pattern Analysis and Applications*, *24*(3), 1207–1220. Retrieved from https://link.springer.com/article/10.1007/s10044-021-00984-y

Naufal, M. F., Kusuma, S. F., Tanus, K. C., Sukiwun, R. V., Kristiano, J., Lieyanto, J. O., & R., D. C. (2021). Analisis Perbandingan Algoritma Klasifikasi Citra Chest X-ray Untuk Deteksi Covid-19. *Teknika*, *10*(2), 96–103. https://doi.org/10.34148/teknika.v10i2.331

Nugroho, B., & Puspaningrum, E. Y. (2021). Kinerja Metode CNN untuk Klasifikasi Pneumonia dengan Variasi Ukuran Citra Input. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *8*(3), 533. https://doi.org/10.25126/jtiik.2021834515

Rahman, T. (2022). COVID-19 Radiography Database. Retrieved from Kaggle website: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database

Silva, P., Luz, E., Silva, G., Moreira, G., Silva, R., Lucio, D., & Menotti, D. (2020). COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis. *Informatics in Medicine Unlocked*, *20*, 100427. https://doi.org/10.1016/j.imu.2020.100427

Wu, X., Hui, H., Niu, M., Li, L., Wang, L., He, B., … Zha, Y. (2020). Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: A multicentre study. *European Journal of Radiology*, *128*(March), 1–9. https://doi.org/10.1016/j.ejrad.2020.109041