Implementation of CNN for Mobile-based COVID-19 Chest X-Ray Images

Indo Intan[1], Suryani*[2], ST Aminah Dinayati Ghani[3], Moh. Rifkan[4], Syamsul Bahri[5]
[1,2]Universitas Dipa Makassar; Jl. P. Kemerdekaan Km.9 90245, Makassar, 0411587194
[3]Department of Informatics Engineering, Makassar
e-mail: [1]indo.intan@undipa.ac.id, *[2]**suryani187@undipa.ac.id**, [3]dinayati.amy@undipa.ac.id, [4]mohrfkn@gmail.com, [5]bahrisyam777@gmail.com

Abstract

The COVID-19 pandemic outbreak is the most significant event from 2019 until 2021. A medical examination of radiological images is carried out to check the condition of the patient's lungs. The limitations of this examination need alternative computer-assisted applications for patient CXR. This research aims to implement a back-end and front-end-based Convolutional Neural Network (CNN) model. Its advantage is that it can detect CXR images in real-time and non-real-time using multi-classification, namely normal, pneumonia, and COVID-19. The CNN model carries out the process of convolutional feature extraction and multi-layer perceptron classification at the back-end stage. In contrast, it uses an Android mobile-based application at the front-end stage. The research results show that the non-real-time condition has an accuracy of 98%, while the real-time is 95% lower. This research produces model and application performance that is flexible for user needs. The results can be recommended for developing applications for more comprehensive users.

Keywords— COVID-19, CNN, Image Analysis, Application, Android

1. INTRODUCTION

An infectious disease caused by the severe acute Respiratory Syndrome Coronavirus Coronavirus-19 (COVID-19) [1]. COVID-19 is a virus that was first identified in Wuhan, China, in December 2019. This virus is very easy to infect people [2], which then spreads throughout China [3]. The WHO designated it as a pandemic because it's spreading all over the world. In Indonesia, on 2 March 2020, the first case of COVID-19 was identified through tourism. The symptoms of this virus are headache, cough, fatigue, fever, body aches and pains, and shortness of breath [4]. Many contactless imaging techniques [5] have been developed during the COVID-19 virus outbreak [6], [7] including one of them is radiological examination or imaging [8].

The protocol for testing whether a subject is infected with COVID-19 must be based on clinical and epidemiological factors and assessing the possibility of infection. One of the tests carried out in the laboratory as a clinical factor is a radiological or imaging examination. Examination efforts have also concerned the Ministry of Health and related agencies. One of the examinations that play an essential role in chest x-ray radiological examination. This examination is carried out before the PCR examination is carried out. The two results of this examination are carried out sequentially because they mutually reinforce one another [9]. Imaging examination results have an accuracy of more than 90[8], [10], [11] as good as PCR [12].

Clinical and epidemiological data, along with an evaluation of the likelihood of infection, should inform the technique for determining if an individual has COVID-19 infection. A radiological or imaging examination is one of the clinical examinations conducted in the laboratory. The Ministry of Health and allied authorities have also concentrated their attention on examination activities. A chest x-ray radiological examination is a crucial diagnostic test. The PCR examination comes first. We conduct the two outcomes of this evaluation sequentially, as they mutually reinforce each other [9]. The imaging examination results have a precision of over 90%[8], [10], [11], comparable to PCR [12].

PCR is a very specific way to diagnose COVID-19, but it takes a long time. Of course, during the pandemic, there is a huge demand for this type of testing. This is the initial imaging test [13] associated with pneumonia symptoms, followed by a chest X-ray (CXR) [14]. A radiological test is crucial for numerous reasons, including [15]:. a) It saves time and money [11], and it's easy to use because the results come out quickly and can work around equipment problems. In contrast to the RT-PCR test results, the set of tests takes a long time [16]; b) radiological tests can find viral infections early on instead of tests that come back negative [17].

Moreover, radiologists have performed radiological tests in several hospitals. Radiologists continue to rely on their experience and visual aids; they frequently conduct various studies. When the number of patients rises, the limited quantity of health workers, particularly radiologists, in terms of number, staff, and skill is a constraining element. It inevitably leads to tension, boredom, anxiety, and exhaustion [18]. This means that a reexamination may be necessary because the analytical results may be more accurate. This course raises concerns about the legitimacy and trustworthiness of a diagnostic result issued by the hospital, as well as concerns about the family's protestations and the correctness of the examination results. In comparison to PCR, the primary justification for radiological evaluation is its low cost and simplicity of release. Continuous chest imaging is required while under surveillance or receiving treatment [19]. The benefit of the CXR test is its accessibility in many clinics and hospitals, along with its speed, low cost, and affordability for all demographics; yet, its accuracy is not as sensitive [10][20]. Thus, rather than relying solely on eye visualization, radiologists can benefit from automated image recognition systems to help them perform examinations and produce more accurate results.

There are two main types of methods used in picture recognition: machine learning (ML) and deep learning (DL). Each of these methods has its benefits, depending on their time. Historically, researchers have employed various machine learning techniques to automatically categorize digital CXR pictures into distinct groups [21][22]. We calculate three statistical features from the texture of the lungs in Bharathi using an SVM classifier to distinguish between normal and malignant tumors in lung lumps. We use backpropagation [23] and the grey level-co-occurrence matrix together to sort standard cancer photos into groups. Studies [13] [24] have demonstrated the superiority of DL-based approaches over ML-based ones. The most common deep-learning method used in medical imaging is CNN [25]. CNNs' major strength is their ability to automatically recognize features from certain image domains. Unlike ML classification, the strategy in CNN architecture training transfers the trained information from a network already trained for one task to another. As a result, experts frequently use this method, especially for medical imaging tests [13].

The DL method is a technique for learning representations that use many levels of presentation. We create these levels by arranging basic non-linear modules, where each transformation captures a low-level representation and gradually builds up to a high-level representation or elementary abstract level, mirroring the functions of the human brain [25][26]. The acceptance of DL models as a significant advancement in artificial intelligence is growing in society, particularly for their diverse applications in image recognition [27]. The use of DL as a method for ML and pattern recognition is crucial in the field of medical image analysis [28]. It is important to study its substantial impact in the domain of medical imaging [20]. Abbas et al. [13] state that the main applications of medical image analysis are segmentation, classification, and abnormality detection. These applications use images from various clinical imaging modalities. We specifically developed CNN, a deep learning model, to handle data with a grid pattern. Its goal is to autonomously and flexibly acquire knowledge about the spatial arrangement of information, progressing from simple to more complex patterns [29].

The multilayer perceptron (MLP) is the source of the CNN classification technique. CNNs consist of several input, convolution, and output layers [30]. The input layer's role is to gather information about the image's dimensions. The convolution layer performs feature extraction and selection by identifying meaningful image pixels using max pooling and average pooling techniques. The large pixels will select characteristics with smaller measurements. Furthermore, the selected features will function as both the output of the feature extraction procedure and the input to the neural network, specifically the fully connected layer [30]. This neural network performs hyperparameter tweaking. The optimal settings will dictate the training procedure's conclusion for testing and evaluation. Target labels, specifically normal, pneumonia, and COVID-19, represent the outcome.

Prior researchers have conducted a review of the research on COVID-19 detection. These options involve utilizing a front end, which can be either web-based [31], [32], or Android-based. Alternatively, one can employ a back-end approach employing machine learning (ML) or deep learning (DL) on the Python and TensorFlow frameworks [33]. Saiful conducted research on a Python and TensorFlow framework-based back-end platform. Saiful assessed the effectiveness of the implementation of the Convolutional Neural Network (CNN). The dataset used consists of 436 data points, with 358 allocated for training and 78 reserved for testing. This model offers annotations for the COVID-19, viral pneumonia, and normal categories, with accuracy levels ranging from 60% to 99%. Building on the work of [32], [33] uses a dataset consisting of 1,125 data points, including 125 cases of COVID-19, 500 cases of normal conditions, and 500 cases of pneumonia. The acquired results exhibit an accuracy of 86.93% together with the swiftest detection time of 0.0027 seconds. The findings suggest that the approach is suitable for use in mobile applications. To conduct research [32], [33], it is necessary to use a back-end programming method without a flexible user interface. Additionally, a mobile device cannot scan the research in real-time and still relies on file input.

The researchers [31] created a backend using the Python and TensorFlow frameworks, and a web-based frontend using HTML and JavaScript. The dataset has 329 data points, including 90 cases of COVID-19, 89 cases of viral pneumonia, and 90 cases of normal conditions. The Mean Squared Error (MSE) loss function yielded a value of 0.0210, while the validation loss was 0.0409, both achieved by the Adams optimizer. The obtained accuracy value ranged from approximately 89.6% to 90.89%, while the F1 score was 87.9%. Prior scholars [31][32] have conducted a thorough examination of studies of COVID-19 identification. These options include using a user interface, such as a web-based or Android front end, or employing machine learning (ML) or deep learning (DL) on the Python and TensorFlow frameworks as a backend. Saiful conducted research on a back-end platform using the Python and TensorFlow frameworks. We evaluated the performance of the CNN implementation. The dataset used consists of 436 data points, with 358 allocated for training and 78 for testing. This model offers annotations for the COVID-19, viral pneumonia, and normal categories, with an accuracy rate ranging from 60% to 99%.

The following chapters describe this study. The introduction covers the benefits of radiological examination and its development in front-end and back-end techniques. The method delineates the phases of training, testing, and evaluation of the model and design utilized. In the results and discussion section, we discuss the outcomes of the model simulation and the user's (radiologist) front-end implementation. The research successfully achieved its ultimate goal, which was to improve back-end performance over front-end performance.

2. RESEARCH METHOD

The researcher viewed the approach as both manual visualizations by a radiologist and an application-based Android mobile (Figure 1). Radiologists observe CXR images using a tool. Their outputs are viewed on real-time display and shifted to the left, right, output, and bottom through horizontal and vertical rotation via inline and outline position. They look for cancer cell areas and annotate the number of expressed cancer cells. They are classified based on the number of thresholds determined by the WHO.

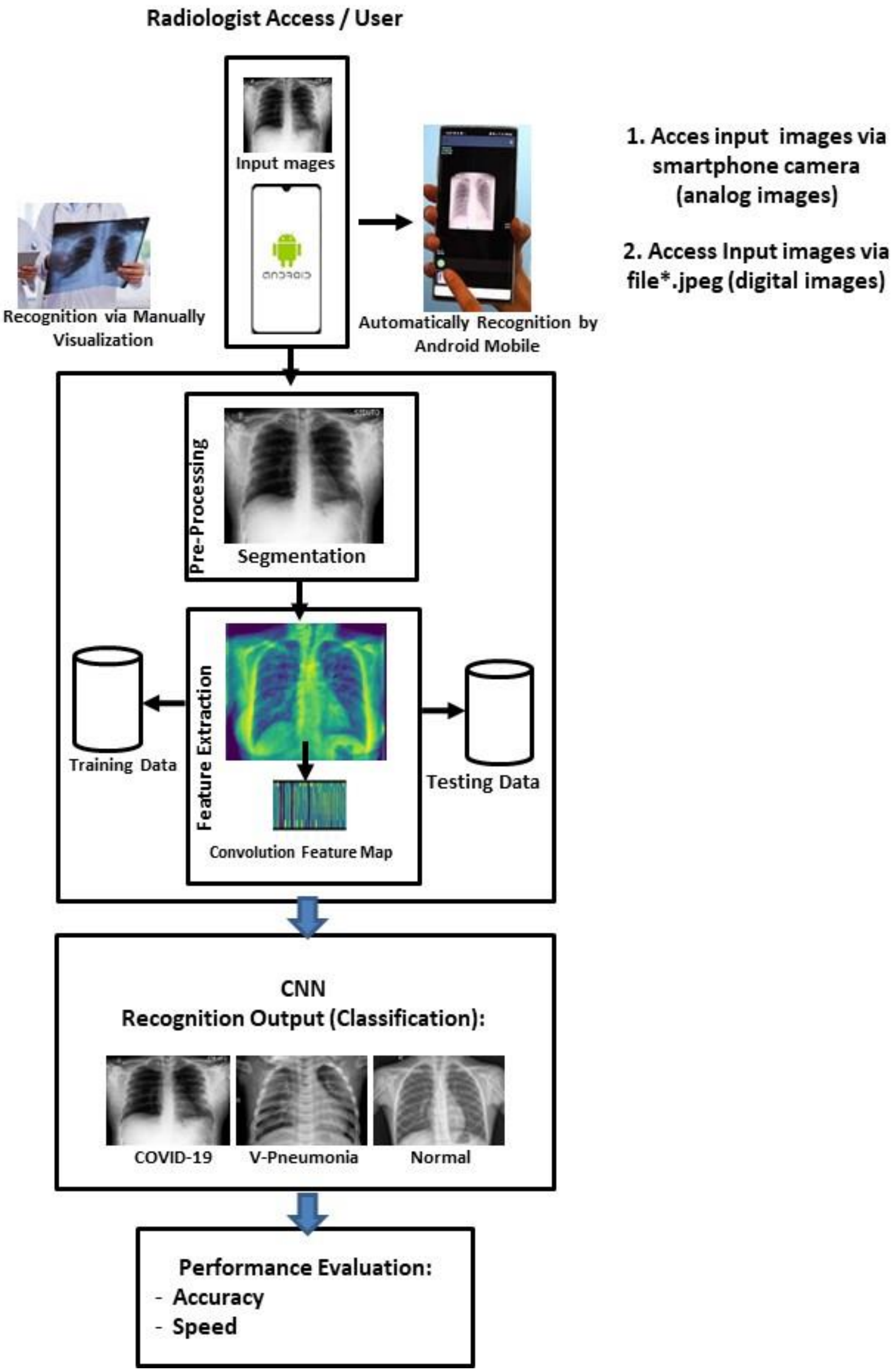

Radiologist Access / User
Input mages
android
Recognition via Manually Visualization
Automatically Recognition by Android Mobile
1. Acces input images via smartphone camera (analog images)
2. Access Input images via file*.jpeg (digital images)
Pre-Processing
Segmentation
Feature Extraction
Training Data
Testing Data
Convolution Feature Map
CNN
Recognition Output (Classification):
COVID-19
V-Pneumonia
Normal
Performance Evaluation:
- Accuracy
- Speed

Figure 1. Proposed Stages

Light illumination provides a bright effect on objects in the image. The radiologist's eye visualization will differentiate each area and color in the CXR section. Radiologists carry out diagnoses according to their knowledge and expertise. The team determining this diagnosis involves a quorum of the same results. If it is not achieved, repeat. The automatic application based on Android mobile captures CXR images and goes to pre-processing to segment CXR images in sample form. The feature extraction process extracts the outputs using convolution, max pooling, and average pooling processes to arrange feature maps. These processes use datasets CXR images such as those described in the section Research Type.

The application includes (Figure 2): (1) User as a user; (2) Front-end as a graphical application interface; (3) Back-End, the coding process of the Convolutional Neural Network (CNN) algorithm; and 4) Rest API (Application Programming Interface), a link between two different platforms to communicate with each other through user access to resources on the server (Google Colab).

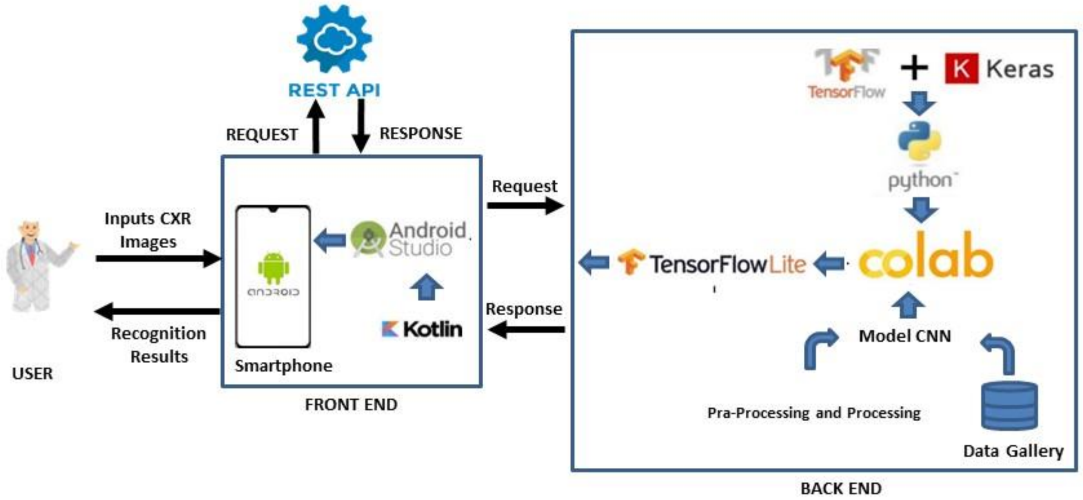


Figure 2 Software Design Architecture

2.1. Type of Research

This research is an experimental type of research, namely to produce a product in the form of an Android mobile-based application that provides information on the results of recognition of radiographic images of COVID-19 patients equipped with the distribution of COVID-19 in Indonesia. The technique is through training and performance testing, which tests the system's accuracy and speed of recognition.

The research carried out has several stages in image processing, namely at the first stage, preparing and pre-processing the dataset, then loading the training dataset and testing the dataset after the dataset is ready, then storing the dataset labels, are normal, viral pneumonia, and COVID-19, after that, it is(Fig 1) carried out. Designing the CNN architecture, then the training dataset is trained using the CNN architecture that has been designed previously to produce a model. Then, the model is evaluated using the test dataset prepared to produce accuracy at the last stage.

The dataset used consisted of 4,411 training data samples (368 COVID-19; 2,734 viral pneumonia; and 1,012 normal), 1,018 validation data samples (81 COVID-19; 684 viral pneumonia; and 253 normal), and 1,288 testing data (116 COVID -19; 855 viral pneumonia; and 317 normal). The total dataset is 6,410 data samples. This dataset has a larger composition compared to [9][34][32][33].

2.2. System Architecture

The pre-processing and processing process uses the Google Collaboratory framework's CNN model, which uses Python and is equipped with the Tensorflow and Keras libraries. The coding process in Colab is translated into TensorFlow-Lite as a framework that Kotlin can read in Android Studio. Android Studio functions as a smartphone user device. When the user accesses the application menu feature, a request signal will be sent to the REST API, and the REST API will connect the request to the destination resource (several frameworks and libraries). The Rest API also sends responses as output. The final output received by the user is the result of identifying the type of disease (ID) based on the input image for the user. The detailed description is as follows.

The application includes (Figure 2): (1) User as a user; (2) Front end as a graphical application interface; (3) Back End, the coding process of the Convolutional Neural Network (CNN) algorithm; and 4) Rest API (Application Programming Interface), a link between two different platforms to communicate with each other through user access to resources on the server (Google Collaboratory).

2.2.1 *Back End Design in Google Collaboratory*

The first step is to import all the necessary libraries based on the needs of the TensorFlow and Keras frameworks. *The second step*, retrieve datasets from Google Drive and coordinate reading with Google Colab and Google Drive. *The third step* completes the data definition process, which defines the data type and dimensions, making mathematical manipulation easier. *The fourth step*, data splitting, divides data into training, validation, and testing sets. *The fifth step* of pre-processing involves scaling, converting, and normalizing feature data. *The sixth step* converts COVID-19 image data into a convolutional neural network (CNN) via feature extraction, then convolution and max pooling from input to output. After reaching the output layer, *the seventh step* classifies data into three classes. *The sixth and seventh steps* are critical for training and testing. *The eighth step* calculates the model's score, accuracy, or precision using the Confusion Matrix technique.

The CNN architecture used to handle the chest ray image dataset details the Google Collaboratory modeling approach. CNN architecture has one input layer, six convolution layers, and a flattened layer. After building the architecture, train it using the dataset. We train, validate, and test the architecture using the training, validation, and testing data. COVID-19, viral pneumonia, and normal imaging data classification are shown in Fig 3. The current image is grayscale. Three different types of photos feed the back and front ends. CNN analyzes images. Normal lungs are wide and covered in a white haze, while viral pneumonia has white masses in the lung cavity. COVID-19-affected lungs feature white nodules that cover one-third, half, or virtually the whole lung volume. This indicates each patient's COVID-19 severity, which varies.

The next step is feature extraction, which is done automatically by the model (which is linked to CNN); in this case, CNN is both the feature extractor and the classifier. Eq (2–9) illustrates the employed method.

The method of extracting features from start to finish [30] Equation 1 describes how to apply convolution and a non-linear function in the network to determine the output on the layer at position (i, j).

$$a_{ij} = \sigma((W * X)_{ij} + b) \tag{1}$$

$$S_q^l(i,j) = \frac{1}{4}\sum_{u=0}^{z}\sum_{v=0}^{z} C_q^l(2i-u, 2j-v) \tag{2}$$

$$Output = \sigma(W\ x\ f + b) \tag{3}$$

$$\hat{y}(i) = \frac{e^{\widehat{output}}}{\sum_1^{labels} e^{\widehat{output}}} \tag{4}$$

$$L = \frac{1}{2}\sum_{i=1}^{n}(\hat{y}(i) - y(i))^2 \tag{5}$$

$$\Delta W(i,j) = \frac{\partial(\frac{1}{2}\sum_{i=1}^{p*}(\hat{y}(i)-y(i))^2)}{\partial\hat{y}} \cdot \frac{\partial\hat{y}}{\partial W(i,j)} \tag{6}$$

$$p *= \hat{y}(i) - y(i) \times \frac{\partial}{\partial W(i,j)} \left(\sigma \left(\sum_{j=1}^{\substack{number\\node}} W(i,j) \times f(j) + b(i) \right) \right) \tag{7}$$

$$\Delta W(i,j) = \Delta \hat{y}(i) \times f(j) \tag{8}$$

$$\Delta b = \frac{\partial L}{\partial b(i)} \tag{9}$$

Equation 2 implements pooling to introduce shifts after the convolution operation, next, we generate Equation 3 by applying pooling to the last layer, followed by a fully linked classification and output calculation. Classification is the process of assigning labels to each category class using a neural network. We perform this additional step to extract features for the CNN model. Since $\hat{y}(i)$ represents the desired output and $y(i)$ is the obtained output (Equation 4). The comparison of training and testing data obtained the accuracy or error threshold values (Equation 5). Equation 6 for the first derivative, the first derivative of equation 6 continues with the sequential derivative on each of its nodes, as shown in equations 7–9.

The feature map visualization shows the output of the feature extractor. Following CNN architecture, the first convolution layer processes the 224 × 224 pixel input picture with three filters. A feature map, not logic, marks the patient's chest limitations in image two from the left, using the background picture in the first condition. After receiving a 224x224 image, three filters and max pooling give the convolution-1 layer 112x112 and 224x224 input-output dimensions. The second convolution max uses three input filters, eight output filters, and 112x112 and 56x56 input-output dimensions. Thus, we form 1152 features up to convolution layer 6, using 7x7 input-output pixels and max pooling of 3x3 pixels with 128 filters.

The next step involves flattening, which utilizes all 1152 features in the input-output to consolidate all features into a row vector or column vector for dense layer and dropout scenarios. After it has passed through the dropout, the activation function receives it and generates three output features, which serve as the basis for the CNN model's initial classification of the COVID-19 virus into up to three classes. The process becomes more sophisticated and yields a higher number of features and parameters with additional convolution layers, resulting in a unique categorization.

2.2.2 Front End Design and Android Studio

The use case diagram (Figure 3) for the COVID-19 detector allows users to access real-time information, daily graphs, news, and COVID-19 referral hospitals. The REST API retrieves this data. Users have the option to input chest radiograph images into the system and categorize them, allowing users to view the outcomes of the categorization. Additionally, users can upload chest radiograph images to the database.

The CNN modeling program's main material, "*.tflite," dominates Google Colab's back-end design stage. We design Android app interfaces in Kotlin. To take pictures, the app may use the camera and photo gallery. To import TensorFlow Lite, you must import photos from the gallery and store "*.tflite" and "*.txt" files in the assets folder. In the front-end design, a "*.xml" layout displays each class's COVID-19 classification application interface. We will develop the interface using Android. The "AndroidManifest.xml" file controls camera and gallery access.

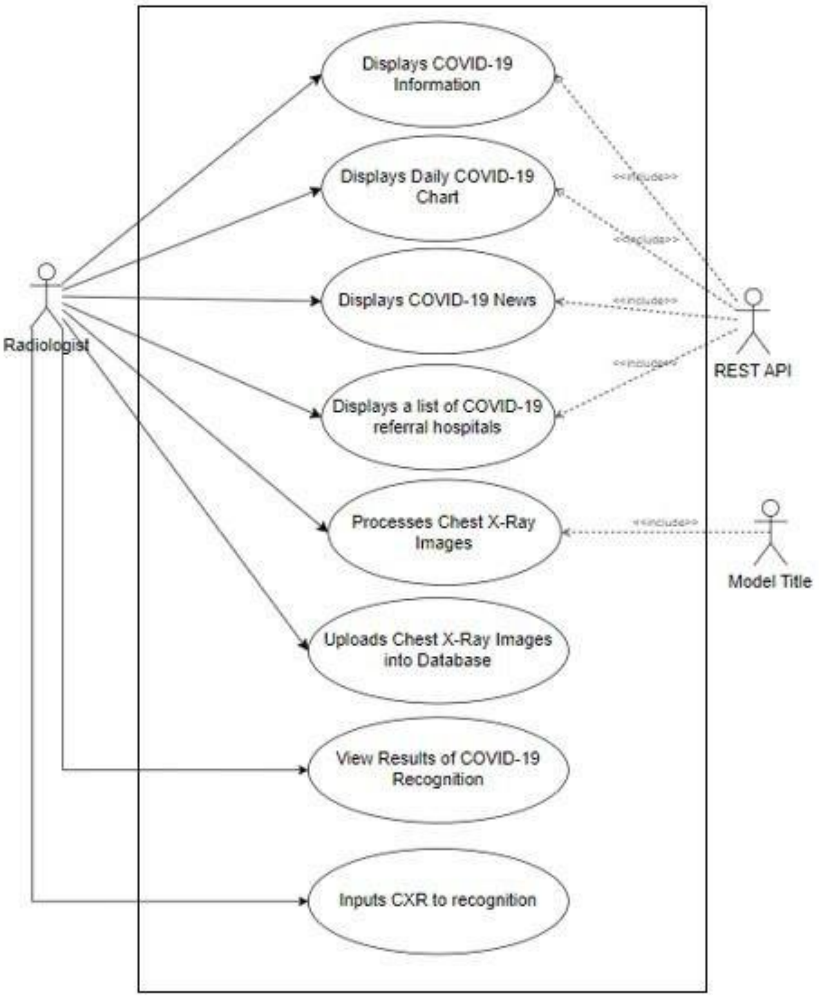

Displays COVID-19 Information
Displays Daily COVID-19 Chart
Displays COVID-19 News
Displays a list of COVID-19 referral hospitals
Processes Chest X-Ray Images
Uploads Chest X-Ray Images into Database
View Results of COVID-19 Recognition
Inputs CXR to recognition
Radiologist
REST API
Model Title
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>

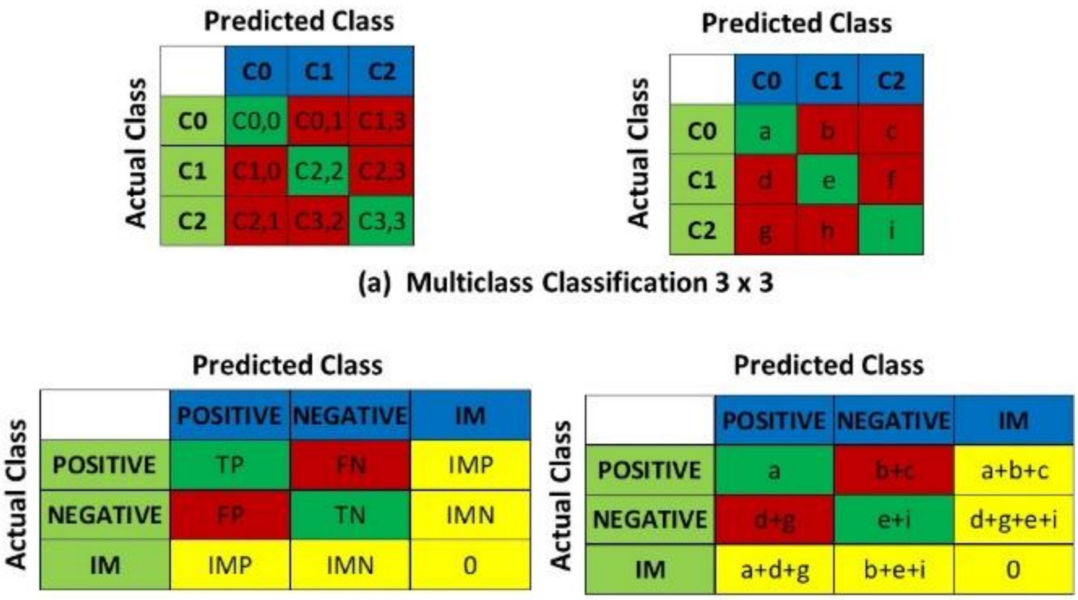
Figure 3 Use Case Diagram of COVID-19 Application

***2.2.3** Testing*

1) Confusion Matrix

A confusion matrix [35] is a metric used to assess the performance of machine learning models and visualize the properties of the model in the context of supervised learning [36], such as a convolutional neural network (CNN) on the back end. The parameters considered are directly related to precision.

The matrix elements [37][38] are defined based on the predicted label (positive, negative) and the outcome of comparing the predicted label with the actual class label (true, false): true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), true positive rate (TPR) or recall, false positive rate (FPR), intragroup mismatch for positive class (IMP), intragroup mismatch for negative class (IMN), and Matthew coefficient correlation (MCC).



Predicted Class

Actual Class

	C0	C1	C2
C0	C0,0	C0,1	C1,3
C1	C1,0	C2,2	C2,3
C2	C2,1	C3,2	C3,3

Predicted Class

Actual Class

	C0	C1	C2
C0	a	b	c
C1	d	e	f
C2	g	h	i

(a) Multiclass Classification 3 x 3

Predicted Class

Actual Class

	POSITIVE	NEGATIVE	IM
POSITIVE	TP	FN	IMP
NEGATIVE	FP	TN	IMN
IM	IMP	IMN	0

Predicted Class

Actual Class

	POSITIVE	NEGATIVE	IM
POSITIVE	a	b+c	a+b+c
NEGATIVE	d+g	e+i	d+g+e+i
IM	a+d+g	b+e+i	0

(b) Summation of Matrix Elements "2 x 2 + IM" Form

Figure 4 Multi Classification of Confusion Matrix

The multi-class confusion matrix has dimensions N x N, where N is labeled class C0, C1, and C2. C0 is COVID-19, C1 is normal, and C2 is pneumonia (Figure 4a). To obtain actual parameters or predicted labels, the matrix elements are summarized so that they experience reduced multi-classes (Figure 4b). CNN model metrics are shown in equations 10–17.

$$\text{Actual Positive, } P_{actual} = TP + FN + IMP \tag{10}$$

$$\text{Actual Negative, } N_{actual} = TN + FP + IMN \tag{11}$$

$$\text{Predicted Positive, } P_{predicted} = TP + FP + IMP \tag{12}$$

$$\text{Predicted Negative, } N_{predicted} = TN + FN + IMN \tag{13}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN+IMP+IMN} \tag{14}$$

$$True\ Positive\ rate\ (Recall) = \frac{TP}{(TP+FN+IMP)} \tag{15}$$

$$True\ Negative\ rate\ (Specificity) = \frac{TN}{(TN+FP+IMP)} \tag{16}$$

$$MCC = \frac{IMP.\ IMN+(TP+IMP)TN-FN\ .\ FP}{(TP+FN+IMP)(TP+FP+IMP)+(FN+TN+IMN)(FP+TN+IMN)} \tag{17}$$

2) Espresso

Espresso is a testing framework for Android to make it easy to write robust user interface tests[39][40]. The testing method used in this research is the Instrumental Test using the Espresso Library. The espresso library tests access to instrumentation information, such as applications. Espresso automatically synchronizes Test actions with the user interface application. Tasking in reading sequential programs and time tasking programs so that they can fix problems when checking logic and procedures; for example, testing will be carried out if the main thread is off so that time is used more efficiently.

Test steps in Espresso are the creation of a test scenario for both the back and the front end; writing code for testing or validation is carried out in the form of suitability between the expected test cases and the actual conditions, ensuring that the procedures and logic are valid; performing of testing automatically.

3. RESULT AND DISCUSSION

In this section, the researcher will explain the research results obtained. Researchers can also use images, tables, and curves to explain the study results. These results should present the raw data or the results after applying the techniques outlined in the methods section. The results are simply results; they do not conclude. The design results are discussed and compared with those from the previous work.

3.1. Results of Back End

The test results on the back end of the application include testing on Google Collaboratory (Figure 5), showing that all processes are according to the design stages on Google Collaboratory, executed validly, and no error messages appear. This process starts from the first stage to the eighth stage.

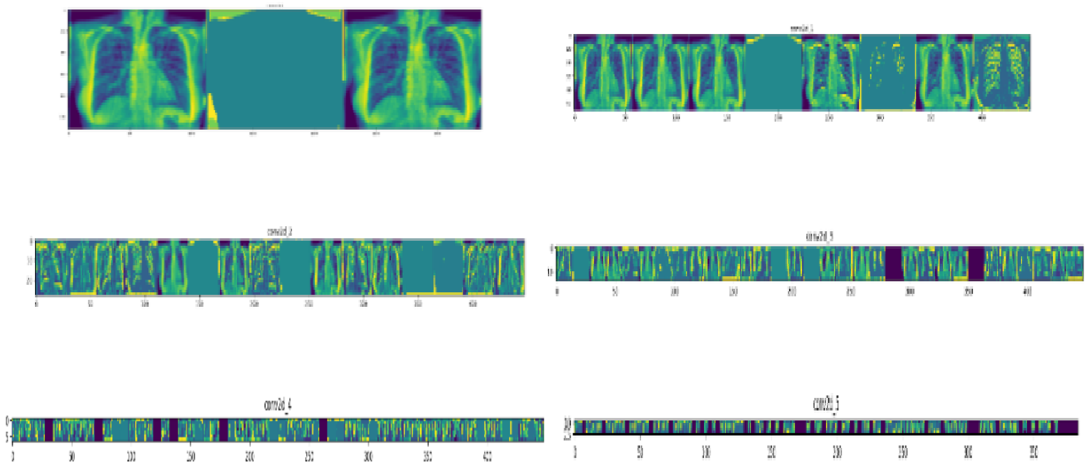
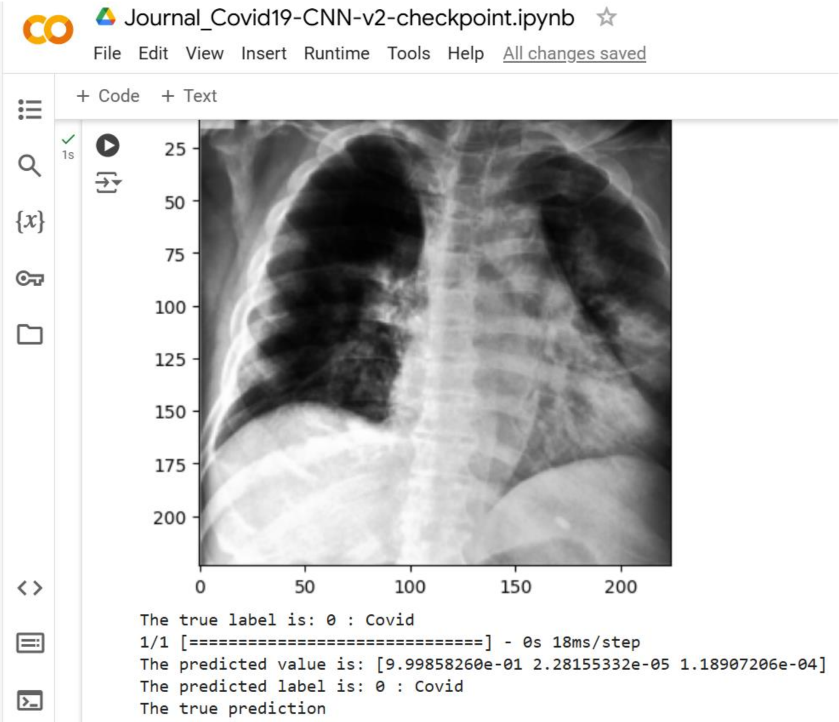


Figure 5 Feature Maps Process on 1st Convolution Layer until Feature Maps Process on 6th Convolution Layer

3.1.1. Testing Results

Making predictions of image recognition of testing data when entering normal, viral pneumonia, and COVID-19 classes using the Confusion Matrix test. Fig shows a comparison between testing data and actual data (actual test data). If the data from the data testing results are the same as the values in the dataset, then they fall into the same class category based on the classification carried out by CNN. The predicted output value is the value that will be automatically classified and shown on the image output display (Figure 6).

Sample-1, entered in Google Colab, has an index of n = 6, the 6th image in the test data (Figure 6). The image is a COVID-19 chest ray, and the probabilities from the test results obtained are COVID-19 at 9.97%, Normal at 7.36%, and Viral Pneumonia at 2.89%. The resulting prediction is correct because the probability of COVID-19 is higher than other chest ray images.


Journal_Covid19-CNN-v2-checkpoint.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
1s
25
50
75
100
125
150
175
200
0
50
100
150
200

```
The true label is: 0 : Covid
1/1 [==============================] - 0s 18ms/step
The predicted value is: [9.99858260e-01 2.28155332e-05 1.18907206e-04]
The predicted label is: 0 : Covid
The true prediction
```

Figure 6 Testing Results of 1st Sample Using Sequential CNN Architecture on Back End

3.1.2. Evaluation Results

They are validating the testing data against the training data. The result is the performance of the model on the back end. The evaluation results use a training comparison chart with validation that the test result curve and its validation (Figure 7) increase closer to the value of 1.0 as epochs (worksheets) increase. In the last epoch, the accuracy value of the test reached 1.0000, and the accuracy value of the validation test reached 0.9811. The blue and orange test validation accuracy curves are the accuracy values for each epoch. If the value gets closer to 1 during the increasing epoch, the test and validation process is declared successful in the classification. If the test's accuracy value and validation do not increase or decrease, then the CNN network, which is trained due to it, can not be appropriately classified.

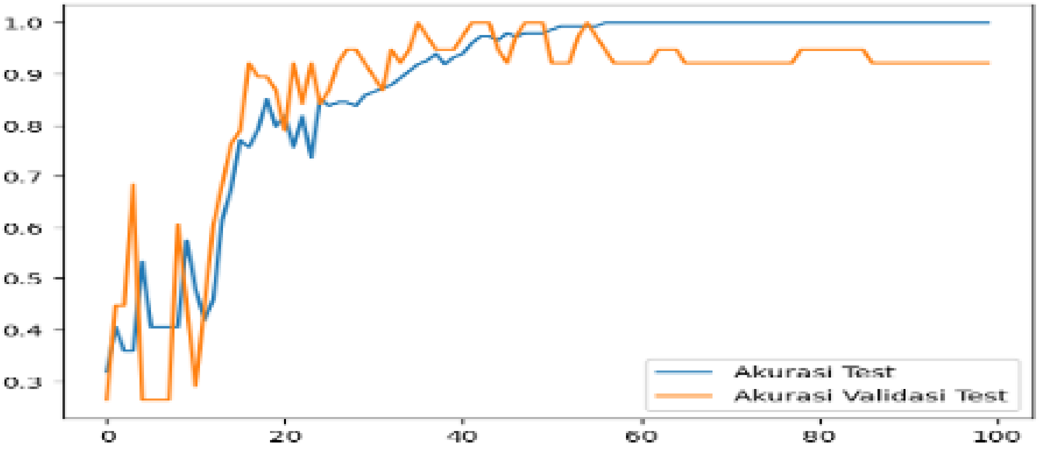


Figure 7 Accuracy Curve of Testing and Validation Results

The curve shows that the test accuracy curve has almost the same increase as the test accuracy validation curve. It shows the training results with good fit conditions, meaning that the training and validation results achieve the expected value at the best performance.

The confusion matrix is used as a metric, an evaluation tool to determine the results of validation checks between the results of model recognition and actual data in real time. In this confusion matrix, correct and predictive labels are displayed with each chest ray image label, namely, COVID-19, Normal, and Viral Pneumonia. The prediction label is the label of the test results on the CNN architecture, while the correct label is from the test dataset, which has been tagged with a label for each chest ray image.

$$Accuracy = \frac{1228}{1288} x\ 100\ \% \ = \ 95{,}5\ \%$$

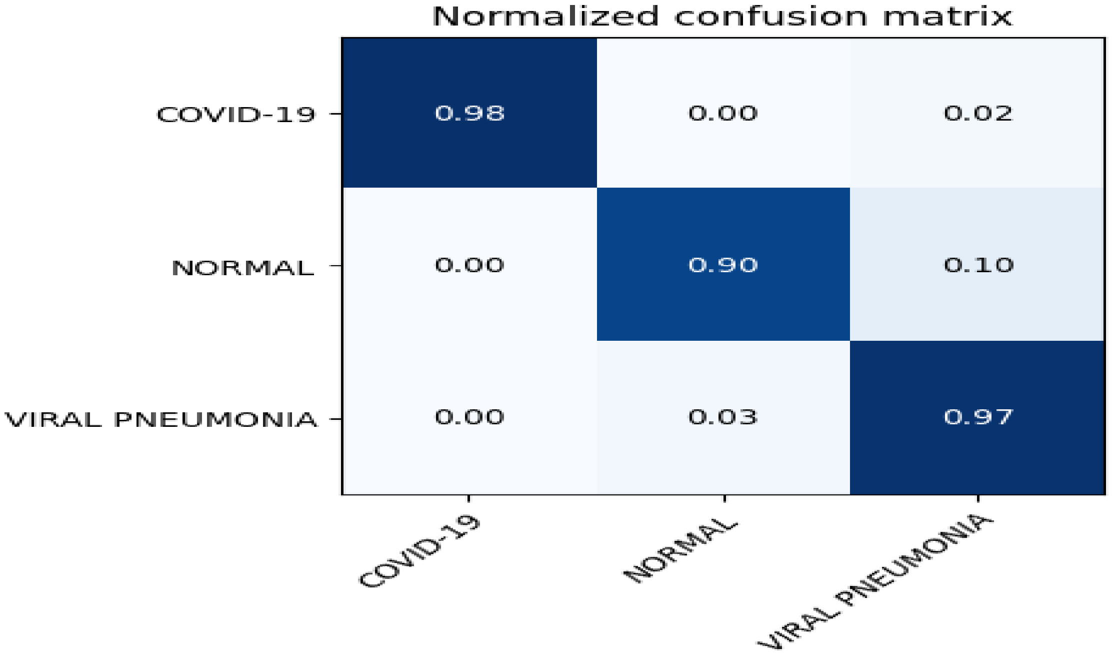


Figure 8 Evaluation Results of Confusion Matrix

Based on the confusion matrix (Figure 8), the prediction results' percentage accuracy and error rate can also be calculated using eq (2). The results provide an accuracy of 95.5%. The results of this accuracy are the results of testing in real-time to the user. This result also outperformed Fadli's [31] accuracy of 89.6%. Based on press 1, the resulting accuracy value is as follows. Based on these results, some of the significance of this study compared to the previous one is that the resulting accuracy results outperform the results given by [13][41][9][42][16][17] which produces an accuracy of around 93.2 - 98%. However, the results are still below [43].

3.2. Results of Front End

The results of the front-end design in the form of an interface that connects users in general and users who want to detect Chest X-ray results (radiologists) are shown in Figure 9. The page consists of four parts: the front page display, the display of COVID-19 information, the display of the COVID-19 detector, and the display of the referral hospital.

Information on the page becomes a menu in the user's application. This menu accesses camera activation to capture user X-Ray images in real-time. To prove that this application is free from logic and procedural errors and to test the functionality of the buttons and page menus, tests are carried out.

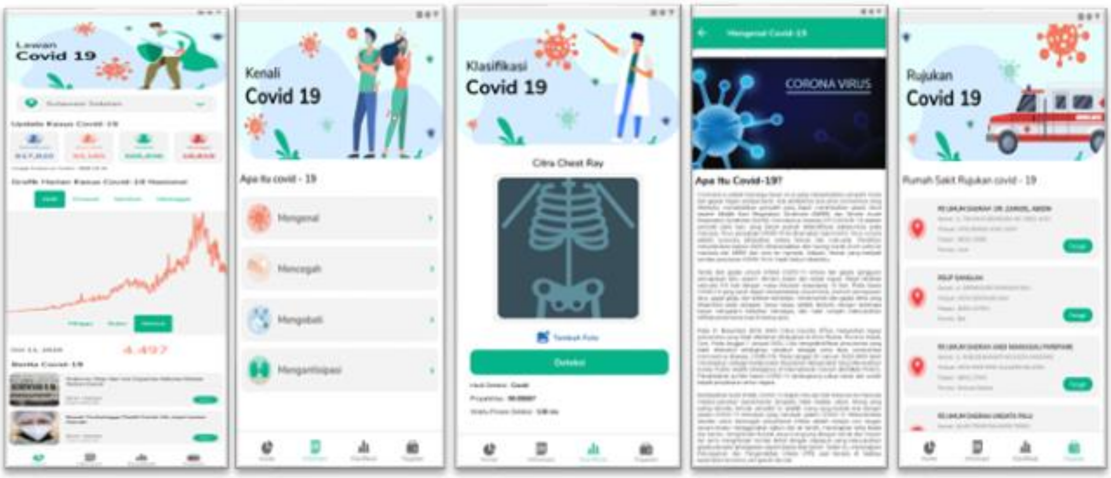


Figure 9 Application Interface for Users on the Front End

There are four front-end form interfaces: the home fragment, the information fragment, the classification fragment, and the referral fragment. The information fragment provides detailed information about how to prevent COVID-19, how to treat it, and how to anticipate it. The classification fragment is a very important form about the implementation of the classification algorithm for COVID-19. The form displays a captured image, both in real-time and in non-real-time, as an object for detection. If the image is detected, it displays the classification and accuracy results, which are the diagnostic results of the used model. Finally, a referral fragment presents hospital reference information for COVID-19 patients connected to the hospital contact number in question. All buttons on the four forms performed their functions.

The author's findings contribute to the advancement of deep learning in both back-end and front-end applications. These results, naturally, hold their respective standings among other researchers. In prior research [44], the author utilized the CNN-based MobileNet model and achieved an accuracy score of 97%, surpassing that of the CNN model. The previous findings were generated solely through back-end processes, without utilizing mobile devices as a user interface.

Ayumi and Nurhaida, both aged [45], employed the Convolutional Neural Network (CNN) model to achieve a training accuracy of 94% and a validation accuracy of 96%. The results showed overfitting because the validation accuracy exceeded the training accuracy. Similarly, in the study conducted by Pratama et al. [46], it was observed that the training accuracy was lower than the testing accuracy, indicating the occurrence of overfitting. Unlike prior studies that solely employed CNN, Naufal et al. [47] conducted a comparative analysis of machine learning techniques including KNN, SVM, and CNN. The CNN model achieved superior performance with an accuracy rate of 95.9% compared to other models. In contrast to the author, this result is robust in terms of comparing back-end models, as it presents multiple findings from three models, whereas the author only employs a single CNN model.

Using either max pooling or average pooling layers with convolutional layers is a good way to get features and characteristics out of CXR images of COVID-19 patients. It has been shown that the contrast-limited adaptive histogram equalization (CLAHE) feature extraction method can achieve maximum accuracy of 85%, even when multiple uniform, Rayleigh, and exponential techniques are used [48]. This approach is highly effective in clearly defining the boundaries of lung objects and overcoming the limits of CXR images. Nevertheless, effective feature extraction must be complemented by a classifier that exhibits exceptional performance. Ghozali & Sumarti [49] presented findings on the Sobel edge detection model's capability to differentiate the progression of COVID-19 dissemination in the lungs.

4. CONCLUSION

CNN is a fundamental framework for image processing in deep learning models. Optimal performance can be achieved by configuring it with other models during its development. To achieve a more precise model implementation, it is important to build a model that exhibits superior performance and stability at the convergence point, devoid of significant oscillations and overfitting. In this example, the author's contributions include the successful implementation of CNN, which exhibits excellent performance. Additionally, users can conveniently obtain this information through mobile-based applications.

5. FUTURE WORKS

This research can be applied to users with even larger datasets, patients and doctors or radiologists at the hospital, and continue to test user satisfaction, especially radiologists. This research can be applied to users with even larger datasets, patients and doctors or radiologists at the hospital, and continue to test user satisfaction, especially radiologists.

This research has the potential to be implemented in scenarios involving users with significantly larger datasets, such as patients, doctors, and radiologists in a hospital setting. Furthermore, it can be used to further evaluate user satisfaction, particularly among radiologists. This research has the potential to be implemented for users with significantly larger datasets, including patients, doctors, and radiologists at the hospital. Furthermore, it may be used to further evaluate user satisfaction, particularly among radiologists.

ACKNOWLEDGEMENT

Many thanks to all parties of Universitas Dipa Makassar who contributed funding for research publications so that the results of this research could reach readers.

REFERENCES

[1] A. P. Agustin, A. C. Fauzan, and Harliana, "Implementasi K-Nearest Neighbor Dengan Jarak Minkowski Untuk Deteksi Dini Covid-19 Pada Citra Ct-Scan Paru - Paru," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 4, no. 1, pp. 23–30, 2022.

[2] Q. Li *et al.*, "Early Transmission Dynamics in Wuhan, China, of Novel Coronavirus–Infected Pneumonia," *N. Engl. J. Med.*, vol. 382, no. 13, pp. 1199–1207, 2020, doi: 10.1056/nejmoa2001316.

[3] Y. M. Arabi, S. Murthy, and S. Webb, "COVID-19: a novel coronavirus and a novel challenge for critical care," *Intensive Care Med.*, vol. 46, no. 5, pp. 833–836, 2020, doi: 10.1007/s00134-020-05955-1.

[4] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1207–1216, 2016, doi: 10.1109/TMI.2016.2535865.

[5] Y. Wang *et al.*, "Precise pulmonary scanning and reducing medical radiation exposure by developing a clinically applicable intelligent CT system: Toward improving patient care," *EBioMedicine*, vol. 54, 2020, doi: 10.1016/j.ebiom.2020.102724.

[6] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru, "An adaptable system for RGB-D based human body detection and pose estimation," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 39–52, 2014, doi: 10.1016/j.jvcir.2013.03.011.

[7] D. Selvaraj, A. Venkatesan, V. G. V. Mahesh, and A. N. Joseph Raj, "An integrated feature frame work for automated segmentation of COVID-19 infection from lung CT

images," *Int. J. Imaging Syst. Technol.*, vol. 31, no. 1, pp. 28–46, 2021, doi: 10.1002/ima.22525.

[8] G. Jia, H. Lam, and Y. Xu, "Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19 . The COVID-19 resource centre is hosted on Elsevier Connect , the company ' s public news and information ," no. January, 2020.

[9] B. Abraham and M. S. Nair, "Computer-aided detection of COVID-19 from X-ray images using multi-CNN and Bayesnet classifier," *Biocybern. Biomed. Eng.*, vol. 40, no. 4, pp. 1436–1445, 2020, doi: 10.1016/j.bbe.2020.08.005.

[10] R. Karthik, R. Menaka, and M. Hariharan, "Learning distinctive filters for COVID-19 detection from chest X-ray using shuffled residual CNN," *Appl. Soft Comput.*, vol. 99, p. 106744, 2021, doi: 10.1016/j.asoc.2020.106744.

[11] A. Rehman, T. Sadad, T. Saba, A. Hussain, and U. Tariq, "Real-Time Diagnosis System of COVID-19 Using X-Ray Images and Deep Learning," *IT Prof.*, vol. 23, no. 4, pp. 57–62, 2021, doi: 10.1109/MITP.2020.3042379.

[12] R. Shrestha and L. Shrestha, "Coronavirus disease 2019 (Covid-19): A pediatric perspective," *J. Nepal Med. Assoc.*, vol. 58, no. 227, pp. 525–532, 2020, doi: 10.31729/jnma.4977.

[13] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Appl. Intell.*, vol. 51, no. 2, pp. 854–864, 2021, doi: 10.1007/s10489-020-01829-7.

[14] H. Shi *et al.*, "Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study," *Lancet Infect. Dis.*, vol. 20, no. 4, pp. 425–434, 2020, doi: 10.1016/S1473-3099(20)30086-4.

[15] B. Yanti and U. Hayatun, "Peran pemeriksaan radiologis pada diagnosis Coronavirus disease 2019," *J. Kedokt. Syiah Kuala*, vol. 20, no. 1, pp. 53–57, 2020, doi: 10.24815/jks.v20i1.18300.

[16] S. Hassantabar, M. Ahmadi, and A. Sharifi, "Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network approaches," *Chaos, Solitons & Fractals*, vol. 140, p. 110170, Nov. 2020, doi: 10.1016/J.CHAOS.2020.110170.

[17] S. Pathan, P. C. Siddalingaswamy, and T. Ali, "Automated Detection of Covid-19 from Chest X-ray scans using an optimized CNN architecture," *Appl. Soft Comput.*, vol. 104, p. 107238, 2021, doi: 10.1016/j.asoc.2021.107238.

[18] I. Sulistyowati and L. R. W. Utami, "Tingkat Kecemasan Radiografer dalam Memberikan Pelayanan Radiologi pada Masa Pandemi Covid-19 di Rumah Sakit Baitul Hikmah Kendal," *J. Ilmu dan Teknol. Kesehat. STIKES Widya Husada*, vol. 12, no. 2, pp. 55–61, 2021.

[19] S. Ahmad, "A Review of COVID-19 (Coronavirus Disease-2019) Diagnosis, Treatments and Prevention," *Eurasian J. Med. Oncol.*, vol. 2019, 2020, doi: 10.14744/ejmo.2020.90853.

[20] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, pp. 8914–8924, 2016, doi: 10.1109/ACCESS.2016.2624938.

[21] E. Dandil, M. Cakiroglu, Z. Eksi, M. Ozkan, O. K. Kurt, and A. Canan, "Artificial neural network-based classification system for lung nodules on computed tomography scans," *6th Int. Conf. Soft Comput. Pattern Recognition, SoCPaR 2014*, pp. 382–386, 2014, doi: 10.1109/SOCPAR.2014.7008037.

[22] J. Kuruvilla and K. Gunavathi, "Lung cancer classification using neural networks for CT images," *Comput. Methods Programs Biomed.*, vol. 113, no. 1, pp. 202–209, 2014, doi: 10.1016/j.cmpb.2013.10.011.

[23] P. B. Sangamithraa and S. Govindaraju, "Lung tumour detection and classification using EK-Mean clustering," *Proc. 2016 IEEE Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2016*, pp. 2201–2206, 2016, doi:

10.1109/WiSPNET.2016.7566533.

[24] W. Sun, B. Zheng, and W. Qian, "Computer aided lung cancer diagnosis with deep learning algorithms," *Med. Imaging 2016 Comput. Diagnosis*, vol. 9785, p. 97850Z, 2016, doi: 10.1117/12.2216307.

[25] L. Deng, "Deep Learning: Methods and Applications," *Found. Trends Signal Process.*, vol. 7, no. June 2014, pp. 197–387, 2014, doi: 10.1561/2000000039.

[26] J. Wan *et al.*, "Institutional Knowledge at Singapore Management University Deep learning for content-based image retrieval : A comprehensive study Chinese Academy of Sciences," 2014.

[27] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[28] 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, doi: 10.1109/TMI.2016.2553401.

[29] R. Yamashita, M. Nishio, R. K. G. D, and K. Togashi, "Convolutional Neural Networks: an Overview and applications in radiology," *Insights Imaging*, vol. 195, pp. 611–629, 2021, doi: 10.1007/978-981-15-7078-0_3.

[30] S. Indolia, A. Kumar, S. P. Mishra, and P. Asopa, "ScienceDirect Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

[31] A. Fadli, Y. Ramadhani, and M. S. Aliim, "Purwarupa Sistem Deteksi COVID-19 Berbasis Website Menggunakan," vol. 5, no. 10, pp. 876–883, 2021.

[32] M. Saiful and L. M. Samsu, "Sistem Deteksi Infeksi COVID-19 Pada Hasil X-Ray Rontgen menggunakan Algoritma Convolutional Neural Network (CNN) COVID-19 (corona virus desease 2019) atau dikenal juga dengan virus corona adalah virus yang menyerang sistem pernapasan . Penyakit karen," vol. 4, no. 2, pp. 217–227, 2021.

[33] N. Yudistira and A. W. Widodo, "Deteksi Covid-19 pada Citra Sinar-X Dada Menggunakan Deep Learning yang Efisien," no. December, 2020, doi: 10.25126/jtiik.202073651.

[34] M. M. Rahman, M. S. I. Khan, and H. M. H. Babu, "BreastMultiNet: A multi-scale feature fusion method using deep neural network to detect breast cancer," *Array*, vol. 16. 2022. doi: 10.1016/j.array.2022.100256.

[35] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput. Oper. Res.*, vol. 152, no. December 2022, p. 106131, 2023, doi: 10.1016/j.cor.2022.106131.

[36] K. Parang, L. Wiebe, and E. Knaus, *Novel Approaches for Designing 5-O-Ester Prodrugs of 3-Azido-2,3-dideoxythymidine (AZT).*, vol. 7, no. 10. 2012. doi: 10.2174/0929867003374372.

[37] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies*, vol. 9, no. 4, 2021, doi: 10.3390/technologies9040081.

[38] D. Chicco, N. Tötsch, and G. Jurman, "The matthews correlation coefficient (Mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Min.*, vol. 14, pp. 1–22, 2021, doi: 10.1186/s13040-021-00244-z.

[39] L. Cruz and R. Abreu, "On the Energy Footprint of Mobile Testing Frameworks," *IEEE Trans. Softw. Eng.*, vol. 47, no. 10, pp. 2260–2271, 2021, doi: 10.1109/TSE.2019.2946163.

[40] S. Negara, N. Esfahani, and R. Buse, "Practical Android Test Recording with Espresso Test Recorder," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2019, pp. 193–202. doi:

10.1109/ICSE-SEIP.2019.00029.

[41] M. M. A. Asmaa Abbas, “Learning Trnasformations for Automated Classification of Manifestation of Tuberculosis using Convolutional Neural Network,” in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, IEEE, 2018, pp. 122–126. doi: 10.1109/ICCES.2018.8639200.

[42] G. Caseneuve, I. Valova, N. LeBlanc, and M. Thibodeau, “Chest X-Ray image preprocessing for disease classification,” *Procedia Comput. Sci.*, vol. 192, pp. 658–665, 2021, doi: 10.1016/j.procs.2021.08.068.

[43] G. Jain, D. Mittal, D. Thakur, and M. K. Mittal, “A deep learning approach to detect Covid-19 coronavirus with X-Ray images,” *Biocybern. Biomed. Eng.*, vol. 40, no. 4, pp. 1391–1405, 2020, doi: 10.1016/j.bbe.2020.08.008.

[44] S. A. D. Ghani, I. Intan, and M. Rizal, “MobileNet Classifier for Detecting Chest X-Ray Images of COVID-19 based on Convolutional Neural Network,” ILKOM Jurnal Ilmiah.

[45] V. Ayumi and I. Nurhaida, “Klasifikasi Chest X-Ray Images Berdasarkan Kriteria Gejala Covid-19 Menggunakan Convolutional Neural Network,” *JSAI (Journal Sci. Appl. Informatics)*, vol. 4, no. 2, pp. 147–153, 2021, doi: 10.36085/jsai.v4i2.1513.

[46] P. A. H. Pratama, R. Teguh, A. S. Sahay, and V. Wilentine, “Deteksi COVID-19 Berdasarkan Hasil Rontgen Dada (Chest Xray) Menggunakan Python,” *J. Inf. Technol. Comput. Sci.*, vol. 1, no. 1, pp. 58–67, 2021, doi: 10.47111/jointecoms.v1i1.2956.

[47] M. F. Naufal *et al.*, “Analisis Perbandingan Algoritma Klasifikasi Citra Chest X-ray Untuk Deteksi Covid-19,” *Teknika*, vol. 10, no. 2, pp. 96–103, 2021, doi: 10.34148/teknika.v10i2.331.

[48] D. A. Putra, J. Na` am, and Yuhandri, “Identifikasi Objek pada Citra Thorax X-Ray Pasien COVID-19 dengan Metode Contrast Limited Adaptive Histogram Equalization (CLAHE),” *J. Inf. dan Teknol.*, vol. 4, pp. 33–38, 2022, doi: 10.37034/jidt.v4i1.184.

[49] M. Ghozali and H. Sumarti, “Jurnal Imejing Diagnostik,” vol. 10, pp. 63–70, 2024.