Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 9, No. 3, September 2023, pp. 535-548
ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i3.26374

Automated Detection of COVID-19 Cough Sound using Mel-Spectrogram Images and Convolutional Neural Network

Muhammad Fauzan Nafiz, Dwi Kartini, Mohammad Reza Faisal, Fatma Indriani, Triando Hamonangan Saragih
Computer Science Lambung Mangkurat University, Jalan A.Yani Km. 36, Banjarbaru 70714, Indonesia

ARTICLE INFO

Article history:

Received June 07, 2023
Revised July 06, 2023
Published July 11, 2023

Keywords:

Deep Learning;
Convolutional Neural Network;
COVID-19;
Cough Sound;
Mel-Spectrogram

ABSTRACT

COVID-19 is a new disease caused by the Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) variant. The initial symptoms of the disease commonly include fever (83-98%), fatigue or myalgia, dry cough (76-82%), and shortness of breath (31-55%). Given the prevalence of coughing as a symptom, artificial intelligence has been employed to detect COVID-19 based on cough sounds. This study aims to compare the performance of six different Convolutional Neural Network (CNN) models (VGG-16, VGG-19, LeNet-5, AlexNet, ResNet-50, and ResNet-152) in detecting COVID-19 using mel-spectrogram images derived from cough sounds. The training and validation of these CNN models were conducted using the Virufy dataset, consisting of 121 cough audio recordings with a sample rate of 48,000 and a duration of 1 second for all audio data. Audio data was processed to generate mel-spectrogram images, which were subsequently employed as inputs for the CNN models. This study used accuracy, area under curve (AUC), precision, recall, and F1 score as evaluation metrics. The AlexNet model, utilizing an input size of 227×227, exhibited the best performance with the highest Area Under the Curve (AUC) value of 0.930. This study provides compelling evidence of the efficacy of CNN models in detecting COVID-19 based on cough sounds through mel-spectrogram images. Furthermore, the study underscores the impact of input size on model performance. This research contributes to identifying the CNN model that demonstrates the best performance in COVID-19 detection based on cough sounds. By exploring the effectiveness of CNN models with different mel-spectrogram image sizes, this study offers novel insights into the optimal and fast audio-based method for early detection of COVID-19. Additionally, this study establishes the fundamental groundwork for selecting an appropriate CNN methodology for early detection of COVID-19.

Corresponding Author:

Dwi Kartini, Computer Science Lambung Mangkurat University, Banjarbaru 70714, Indonesia
Email: dwikartini@ulm.ac.id

1. INTRODUCTION

COVID-19, caused by the Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) variant, has emerged as a novel disease. Early-stage symptoms commonly include fever (83-98%), fatigue or myalgia, dry cough (76-82%), and shortness of breath (31-55%) [1]. The limited availability of COVID-19 testing facilities has resulted in delayed detection of many individuals who are positive for COVID-19, leading to inadvertent virus transmission and increased infection rates [2]. Furthermore, delayed detection can contribute to a higher mortality rate as patients may not receive timely or adequate treatment. To address this issue, researchers have explored the use of chest X-ray imaging for COVID-19 detection. However, this method is unreliable for early detection due to time-consuming radiography processes, lasting between 30 to 108 minutes [3], and can only be performed in specific hospitals with radiography facilities. Such limitations hinder the effective use of chest X-rays as an early detection tool.

Therefore, alternative approaches are necessary, such as audio-based COVID-19 detection. Several studies have investigated the use of audio, including the research by Zhou *et al.* [4] who utilized a CNN model for cough recognition on the ESC-50 and Speech Command Dataset with the duration of 1 second then transformed to mel-spectrogram. Nanni *et al.* [5] explored different CNN models and data augmentation techniques to enhance accuracy in classifying cat and bird sounds using the Cat [6], [7] and BIRDZ [8] datasets. Xu *et al.* [9] utilized CRNN for audio tagging and weakly supervised sound event detection on the Google Audioset [10]. Fernandes *et al.* [11] employed CNN models on the Abuzz Project dataset [12], which contains wingbeat sounds from various mosquito species, with the main goal of training the CNN model to detect Aedes aegypti mosquitoes. Another study by Kim *et al.* [13] employed SampleCNN [14], an architecture specifically designed for audio classification, with the MagnaTagATune [15], speech command [16], and audio tagging [17] datasets for music auto-tagging, keyword spotting, and acoustic scene tagging tasks.

While models like SampleCNN utilize raw audio waveforms as input data, other CNN models such as VGG, LeNet-5, AlexNet, and ResNet require image inputs. Therefore, audio files need to be transformed into spectrogram images [18] to visualize the frequency signal of the audio.

Spectrograms have been widely adopted as image representations of sound signals in various research domains, including speech recognition, music genre recognition, and sound recognition [19]–[21]. Transforming the audio files into spectrogram images enables the visualization of the frequency signal patterns present in the cough sounds. Analyzing the spectrogram images makes it possible to identify specific frequency patterns that may be indicative of COVID-19. Spectrograms have several variants, such as chromagrams [22], log-spectrograms [23], scalograms [24], and mel-spectrograms [25]. Mel-spectrograms utilize the mel scale, which closely aligns with human auditory perception of sound by considering the human ear's sensitivity to frequency differences at different levels. Hence, mel-spectrograms are commonly used in sound recognition, speaker identification, and emotion recognition.

Despite the promise of using cough sounds for COVID-19 detection, several limitations must be addressed. Variations in cough patterns among individuals, influenced by age, gender, and underlying health conditions, can introduce challenges in accurately detecting COVID-19. Additionally, high-quality audio recordings are essential to capture the necessary frequency signal patterns for analysis. The influence of other factors, such as background noise or overlapping sounds, may also impact the accuracy of detection algorithms.

This study aims to investigate the effectiveness of Convolutional Neural Network (CNN) models in detecting the COVID-19 status based on cough sounds. Our approach involves transforming cough sounds into mel-spectrogram images, enabling visualization of the frequency signal patterns present in the cough sound. The size of the mel-spectrogram image was resized to 32×32, 128×128, 227×227, and the default input size for each CNN models. The CNN models employed in this study include VGG-16, VGG-19, LeNet-5, AlexNet, ResNet-50, and ResNet-152. To assess the model's performance, we evaluate the accuracy, the area under the curve (AUC), precision, recall, and F1 score, enabling a comprehensive comparison to identify the model with the highest performance. Furthermore, we analyze the impact of different input sizes on the performance of each model, aiming to determine the optimal input size for this particular study.

The primary contribution of this research lies in identifying the CNN model that demonstrates the best performance in COVID-19 detection based on cough sounds. This study makes a unique contribution to the field of audio-based COVID-19 detection by focusing specifically on cough sounds. While previous research has explored various audio and CNN models, the use of different sizes for mel-spectrogram images is relatively unexplored. By exploring the effectiveness of CNN models with different mel-spectrogram image sizes, this study offers novel insights into the optimal and fast audio-based method for early detection of COVID-19. Additionally, this study establishes the fundamental groundwork for selecting an appropriate CNN methodology for early detection of COVID-19.

2. METHODS

The main objective of this research is to identify the most optimal method for early detection of COVID-19. Segmentation was applied to the virufy dataset to divide the audio recordings into smaller segments, the data was transformed into mel-spectrogram images. These images were resized to standardized dimensions of 32×32, 128×128, 227×227, and the default input size for each CNN models. The CNN models were evaluated using AUC, accuracy, precision, recall, and F1 score. The research workflow in this study can be seen in Fig. 1.

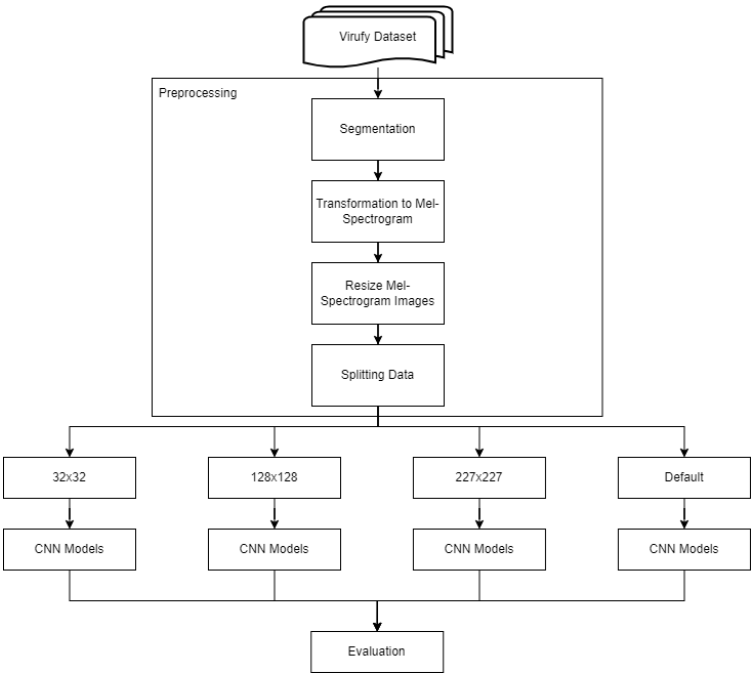


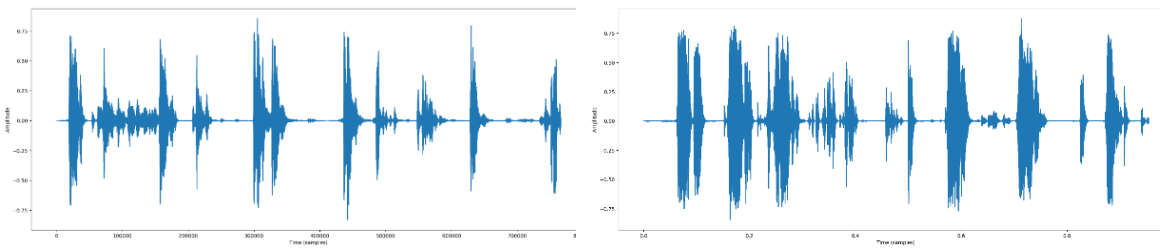
Fig. 1. Research Workflow

2.1 Data Collection

The Virufy dataset [26] was utilized for training and testing in this study. The dataset was collected from a hospital under supervision by physicians following Standard Operating Procedures (SOP) and informed patient consent. The dataset contains a diverse range of cough sounds from different demographics. It comprises 121 cough audio recordings in MP3 format, classified into two classes: "pos" with 48 audio samples and "neg" with 73 audio samples. All audio samples have a sampling rate of 48,000 Hz. Table 1 presents the file names and labels of each audio sample. Waveform representations of samples from the (a) neg and (b) pos classes are depicted in Fig. 2, while their corresponding mel-spectrograms are shown in Fig. 3. The samples had different durations and still contained noise, so preprocessing was necessary to equalize the duration and remove noise from the audio.

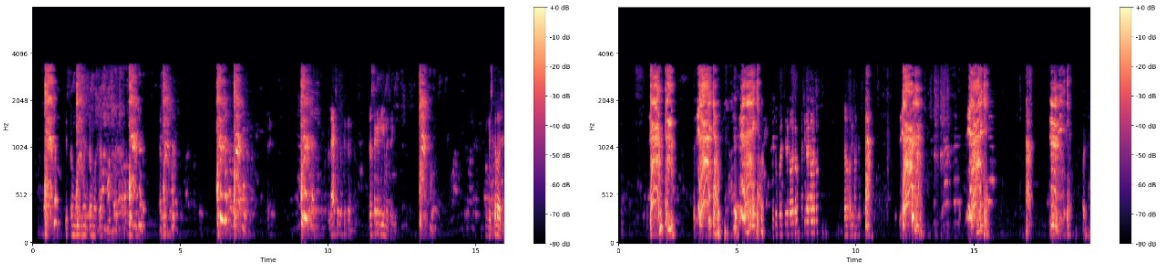
Table 1. Dataset description

No	cough_filename	corona_test
1	neg-0421-083-cough-m-53-0.mp3	neg
2	neg-0421-083-cough-m-53-1.mp3	neg
.....	.....	
72	neg-0421-083-cough-m-53-10.mp3	neg
73	neg-0421-083-cough-m-53-11.mp3	neg
74	pos-0421-094-cough-m-51-4.mp3	pos
75	pos-0421-094-cough-m-51-5.mp3	pos
.....	.....	
120	pos-0422-096-cough-m-31-6.mp3	pos
121	pos-0422-096-cough-m-31-8.mp3	pos



a b

Fig. 2. Visualization of samples in waveform format



a b

Fig. 3. Visualization of cough samples in mel-spectrogram format

2.2 Preprocessing Data

In order to address the difficulty in recognizing sounds caused by very short-duration audio or blending of cough sounds with unrelated sounds in very long-duration audio, the cough audio data was segmented, and noise components were identified and removed manually by audio editing software. Each audio file was divided into one-second segments to ensure consistent duration. The audio segments were then transformed into mel-spectrogram images using the Librosa library [27], to make it compatible with CNN algorithms. The mel-spectrogram images were resized four times, including input sizes of 32×32, 128×128, 227×227, and the default input size of the six models. This process was automated using python programming. The choice of 32×32 input size was based on the smallest default input size among the models, which is 32×32 for LeNet-5 [28]. The 227×227 input size was derived from the model with the largest default input size, which is AlexNet [29]. The value of 128×128 was chosen as the midpoint between 32×32 and 227×227, with a multiple of 32. The VGG-16 model has a default input size of 224×224 [30], as well as the VGG-19, ResNet-50, and ResNet-152 models, which all have the same default input size [31]–[33]. The smaller input size would likely consume less memory, while the larger input size consumes more. The dataset was then split into 70% training data and 30% testing data using a random split.

2.3 Convolutional Neural Network

The mel-spectrogram images were processed through a Convolutional Neural Network (CNN) to determine the presence of COVID-19 based on cough audio. Convolutional Neural Networks are one of the most popular deep learning architectures used in computer vision due to their remarkable ability to detect patterns in images [34]. The CNN models used in this study include AlexNet [35], VGG-16 [36], VGG-19 [37], LeNet-5 [38], ResNet-50 [39], and ResNet-152 [40]. All models were trained using the parameters listed in the Table 2.

Table 2. The parameters used in the models

Parameter	Value
Batch size	8
Epochs	50
Learning Rate	0.0001
Optimizer	Adam
Activation	Relu
Dropout Rate	0.5

The CNN architecture is built with three main types of layers: convolutional layers, pooling layers, and fully connected layers [41]. The convolutional layer [42] is the key layer that extracts features from the input data. It utilizes a mathematical operation called convolution to perform processing. The convolution operation involves using a filter (kernel) to traverse the input data gradually. This filter shifts (performs shifting) across the input, and at each shift, it multiplies the covered input values with the corresponding weights in the filter. After multiplication, all the resulting products are summed up into a single value. This process is performed for each part of the input data, resulting in a new feature map. In Fig. 4, it can be observed that the filter matrix (middle) is multiplied with the focus area (left matrix), which is marked with blue and red colors as the center. The resulting products are then stored in the corresponding locations centered on the next layer.

The pooling layer [44] is used to reduce the dimensionality of the feature maps generated by the convolutional layer. The pooling layer helps reduce the network's complexity and improves resilience to spatial shifts in the input data. The pooling layer also employs a shifting filter to perform operations such as taking

the maximum value (max pooling) or the average value (average pooling) from the area covered by the filter. The visualization of the pooling layer operation can be seen in Fig. 5.

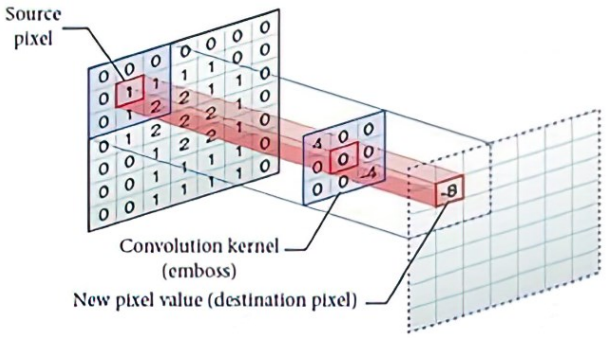


Fig. 4. The operation of the convolutional layer [43]

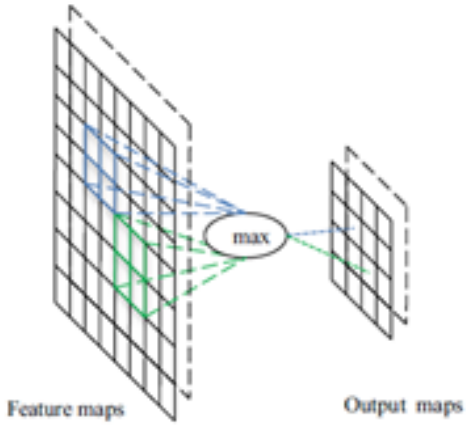


Fig. 5. The operation of the pooling layer [45]

After passing through the convolutional and pooling layers, the resulting features are flattened into a vector and connected to the fully connected layer. The fully connected layer [46] is a layer where each neuron is connected to all units in the previous layer. The purpose of the fully connected layer is to learn the linear and non-linear relationships between the input and output. This layer acts as the classification or regression output layer, transforming the previously extracted features into the final predictions based on the given task. The visualization of the fully connected layer operation can be seen in Fig. 6.

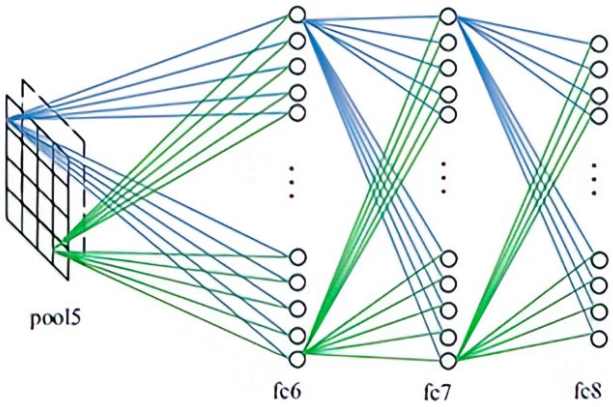


Fig. 6. The operation of fully connected layer [45]

2.4 Assessment Indices

This study uses accuracy, the area under curve (AUC), precision, recall, and F1 score as evaluation metrics for classifying COVID-19 based on cough sounds.

Accuracy [47] is a metric that measures how well a model or classification system can correctly classify data overall. Accuracy indicates how well the model can correctly classify cough sounds as indicative or non-indicative of COVID-19. A higher accuracy value suggests a better overall performance of the model in correctly identifying COVID-19 cases. The formula for accuracy is shown in (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Recall [48], also known as sensitivity or true positive rate (TPR), is one of the evaluation metrics used in classification problems. Recall represents the ability of the model to identify cough sounds from COVID-19 positive individuals correctly. A higher recall value indicates a lower rate of false negatives, meaning the model is correctly capturing a higher number of actual COVID-19 cases. The formula for the recall is shown in (2).

AUC [49] measures the area under the receiver operating characteristic (ROC) curve. The ROC curve is a plot that shows the relationship between the true positive rate (also known as recall) as the Y-axis and the false positive rate (FPR) as the X-axis. AUC represents the model's performance in distinguishing between cough sounds from COVID-19 positive individuals and those from COVID-19 negative individuals. A higher AUC value indicates a better ability of the model to classify COVID-19 cases while minimizing false positives correctly. The formulas for TPR and FPR are shown in (2) and (3).

$$TPR/Recall = \frac{TP}{TP + FN} \tag{2}$$

$$FPR = \frac{FP}{FP + TN} \tag{3}$$

Precision [50] measures the extent to which positive predictions made by a model or classification system are correct. Precision indicates the proportion of correctly identified COVID-19 cases among all the cough sounds classified as positive. A higher precision value suggests a lower rate of false positives, which means the model is making fewer incorrect predictions of COVID-19. The formula for precision can be seen in (4).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

F1 Score [51] is a metric that combines precision and recall into a single value that reflects the balance between the two. This metric is useful when there is an imbalance in the class distribution or when we want to consider both metrics in a balanced way. A higher F1 score indicates a good balance between correctly identifying COVID-19 cases and minimizing false positives and false negatives. The formula for F1 Score is shown in (5).

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

These five evaluation metrics are calculated using a confusion matrix, which includes the following four measures:

- True Positive (TP): The number of positive cases correctly classified as positive by the model.
- True Negative (TN): The number of negative cases correctly classified as negative by the model.
- False Positive (FP): The number of negative cases incorrectly classified as positive by the model.
- False Negative (FN): The number of positive cases incorrectly classified as negative by the model.

3. RESULTS AND DISCUSSION

3.1 Result

After the preprocessing process, all audio samples underwent noise removal, and their duration was standardized to one second. The effectiveness of the preprocessing can be observed in Fig. 7.

Subsequently, the data was divided into 70% training data and 30% testing data for the classification model's training. Four experiments were conducted, employing different input sizes, namely 32×32, 128×128, 227×227, and the default input size for each model.

The models were tested using the testing data. The results for the different input sizes (32×32, 128×128, 227×227, and default) can be found in Table 3, Table 4, Table 5, and Table 6, respectively. The detailed performance of each test can be observed in the confusion matrices presented in Table 7 to Table 10.

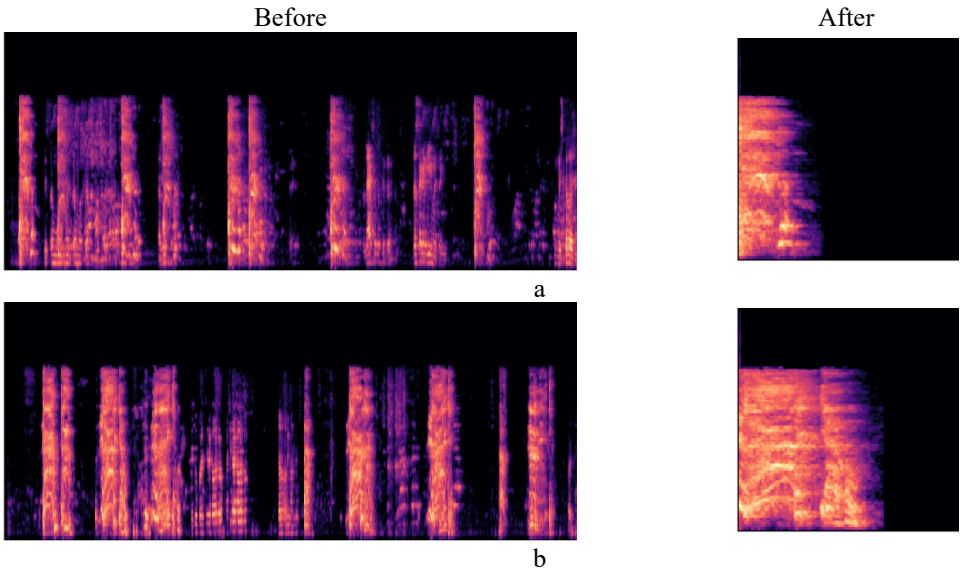


Fig. 7. The results of preprocessing, (a) negative and (b) positive

Table 3. Experimental results for 32×32 input size

Model	Training Time (s)	AUC	Accuracy	Precision	Recall	F1 Score	Standard Deviation
VGG-16	18.94	0.812	0.757	0.727	0.842	0.780	0.045
VGG-19	22.37	0.864	0.730	0.818	0.750	0.783	0.054
AlexNet	12.71	0.852	0.784	0.818	0.818	0.818	0.024
LeNet-5 (Greyscale)	11.01	0.824	0.838	0.818	0.900	0.857	0.033
LeNet-5 (RGB)	11.00	0.834	0.811	0.818	0.857	0.837	0.018
ResNet-50	34.01	0.794	0.757	0.818	0.783	0.800	0.023
ResNet-152	71.38	0.618	0.595	0.773	0.630	0.694	0.072

Table 4. Experimental results for 128×128 input size

Model	Training Time (s)	AUC	Accuracy	Precision	Recall	F1 Score	Standard Deviation
VGG-16	31.72	0.752	0.676	0.591	0.813	0.684	0.084
VGG-19	35.49	0.724	0.784	0.818	0.818	0.818	0.041
AlexNet	21.48	0.9	0.838	0.818	0.900	0.857	0.037
LeNet-5 (Greyscale)	10.66	0.870	0.838	0.864	0.864	0.864	0.012
LeNet-5 (RGB)	11.26	0.894	0.784	0.909	0.769	0.833	0.063
ResNet-50	48.06	0.830	0.703	0.909	0.690	0.784	0.091
ResNet-152	100.11	0.828	0.730	0.636	0.875	0.737	0.093

Table 5. Experimental results for 227×227 input size

Model	Training Time (s)	AUC	Accuracy	Precision	Recall	F1 Score	Standard Deviation
VGG-16	62.48	0.758	0.703	0.727	0.762	0.744	0.024
VGG-19	83.41	0.727	0.676	0.636	0.778	0.700	0.053
AlexNet	21.58	0.930	0.865	0.909	0.870	0.889	0.027
LeNet-5 (Greyscale)	11.98	0.906	0.811	0.818	0.857	0.837	0.038
LeNet-5 (RGB)	14.08	0.894	0.838	0.870	0.870	0.870	0.020
ResNet-50	88.72	0.795	0.703	0.818	0.720	0.766	0.049
ResNet-152	160.08	0.764	0.703	0.955	0.677	0.792	0.109

Table 6. Experimentel result for default input size of each models

Model	Input Size	Training Time (s)	AUC	Accuracy	Precision	Recall	F1 Score	Standard Deviation
VGG-16	224×224×3	59.62	0.773	0.703	0.773	0.739	0.756	0.029
VGG-19	224×224×3	83.41	0.767	0.730	0.773	0.773	0.773	0.019
AlexNet	227×227×3	21.58	0.930	0.865	0.909	0.870	0.889	0.027
LeNet-5 (Greyscale)	32×32×1	11.01	0.824	0.838	0.818	0.900	0.857	0.033
LeNet-5 (RGB)	32×32×3	11	0.833	0.811	0.818	0.857	0.837	0.018
ResNet-50	224×224×3	89.01	0.798	0.784	0.955	0.750	0.840	0.079
ResNet-152	224×224×3	160.61	0.753	0.649	0.409	1.000	0.581	0.219

Table 7. Confusion matrix performance for 32×32 input size

Model	TP	TN	FP	FN
VGG-16	16	12	6	3
VGG-19	18	9	4	6
AlexNet	18	11	4	4
LeNet-5 (Greyscale)	18	13	4	2
LeNet-5 (RGB)	18	12	4	3
ResNet-50	18	10	4	5
ResNet-152	17	5	5	10

Table 8. Confusion matrix performance for 128×128 input size

Model	TP	TN	FP	FN
VGG-16	13	12	9	3
VGG-19	18	11	4	4
AlexNet	18	13	4	2
LeNet-5 (Greyscale)	19	12	3	3
LeNet-5 (RGB)	20	9	2	6
ResNet-50	20	6	2	9
ResNet-152	14	13	8	2

Table 9. Confusion matrix performance for 227×227 input size

Model	TP	TN	FP	FN
VGG-16	16	10	6	5
VGG-19	14	11	8	4
AlexNet	20	12	2	3
LeNet-5 (Greyscale)	18	12	4	3
LeNet-5 (RGB)	20	11	3	3
ResNet-50	18	8	4	7
ResNet-152	21	5	1	10

Table 10. Confusion matrix performance for default input size of each models

Model	TP	TN	FP	FN
VGG-16	17	9	5	6
VGG-19	17	10	5	5
AlexNet	20	12	2	3
LeNet-5 (Greyscale)	18	13	4	2
LeNet-5 (RGB)	18	12	4	3
ResNet-50	21	8	1	7
ResNet-152	9	15	13	0

3.2 Discussion

As shown in Table 3, Table 4, Table 5, and Table 6, the input size of 32×32 yielded the highest AUC score of 0.864 for VGG-19, followed closely by AlexNet with a score of 0.851. These models demonstrated strong accuracy, precision, recall, and F1 score. On the other hand, LeNet-5 (Greyscale) and LeNet-5 (RGB) also performed well, with AUC scores exceeding 0.82. This suggests that for smaller input sizes, these models were able to capture relevant features effectively and achieve good discriminative ability.

However, as the input size increased to 128×128, AlexNet emerged as the top performer with an AUC score of 0.9. This indicates that AlexNet had excellent discriminative ability when presented with larger input sizes. Similarly, LeNet-5 (Greyscale) and LeNet-5 (RGB) demonstrated high AUC scores above 0.86, showcasing their effectiveness in capturing relevant features even with larger input sizes. In contrast, VGG-16 and VGG-19 exhibited relatively lower AUC scores in this configuration.

For the largest input size of 227×227, AlexNet maintained its strong performance with a high AUC score of 0.930. LeNet-5 (Greyscale) and LeNet-5 (RGB) also achieved AUC scores above 0.89, indicating their robust discriminative ability. However, VGG-16 and VGG-19 displayed lower AUC scores compared to smaller input sizes. This suggests that as the input size increased further, these models struggled to maintain their discriminative performance.

The decrease in performance for certain models, such as VGG-16 and VGG-19, with larger input sizes can be attributed to several factors. One reason could be the increased computational complexity associated with larger input sizes, which might have posed challenges in capturing intricate details and patterns. Additionally, overfitting becomes a concern when dealing with larger input sizes, as the models may struggle to generalize well to unseen data. Furthermore, architectural limitations of the VGG models could have played a role in their decreased performance with larger input sizes.

Across all four testing sets, all models performed well across different input size settings, and parameter choices also played a role. AlexNet consistently demonstrated outstanding AUC performance, attaining the highest score of 0.930. LeNet-5 (Greyscale) and LeNet-5 (RGB) also exhibited commendable performance, particularly with larger input sizes. VGG-16 and VGG-19 achieved satisfactory AUC scores but displayed a potential decrease when the input size increased, while ResNet-50 showed slightly lower scores. Despite ResNet-152's high recall, it obtained a lower AUC score due to an imbalance between precision and recall. Comparative performance graphs for each experimental set are presented in Fig. 8.

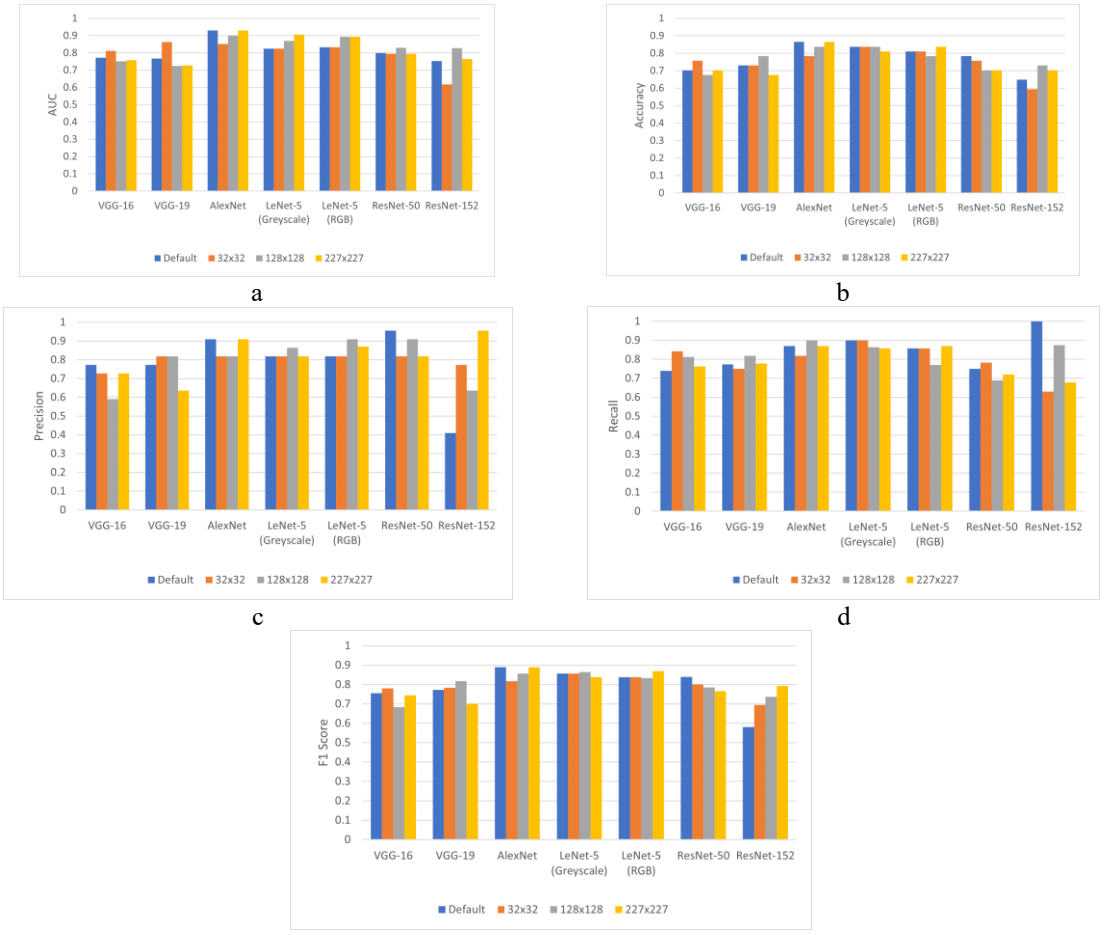


Fig. 8. Performance charts of each input size settings (a) AUC, (b) Accuracy, (c) Precision, (d) Recall, (e) F1 Score

Table 4 shows that for the input size of 32×32, AlexNet exhibited the shortest training time of 12.71 seconds, followed by LeNet-5 (Greyscale) with a time of 11.01 seconds and LeNet-5 (RGB) with a time of 11.00 seconds. VGG-16 and VGG-19 required training times of 18.94 seconds and 22.37 seconds, respectively. The ResNet models, ResNet-50 and ResNet-152, had the longest training times of 34.01 seconds and 71.38 seconds, respectively.

Moving to Table 5, which employed a larger input size of 128×128, an increase in training times can be observed for most models. VGG-16 and VGG-19 displayed training times of 31.72 seconds and 35.49 seconds, respectively. AlexNet took 21.48 seconds, while LeNet-5 (Greyscale) and LeNet-5 (RGB) exhibited similar training times of 10.66 seconds and 11.26 seconds. ResNet-50 and ResNet-152 demonstrated longer training times of 48.06 seconds and 100.11 seconds.

In Table 6 training times continued to rise by utilizing an increased input size of 227×227. VGG-16 and VGG-19 had training times of 62.48 seconds and 83.41 seconds, respectively. AlexNet maintained a relatively shorter training time of 21.58 seconds. LeNet-5 (Greyscale) and LeNet-5 (RGB) showcased similar training times as the previous table, with 11.98 seconds and 14.08 seconds, respectively. ResNet-50 and ResNet-152 exhibited the longest training times among all input sizes, with times of 88.72 seconds and 160.08 seconds, respectively.

Lastly, Table 7 provides the training time for the default input size of each model. VGG-16, VGG-19, and ResNet-50, with a default input size of 224×224, displayed training times of 59.62 seconds, 83.41 seconds, and 89.01 seconds, respectively. AlexNet, with a default input size of 227×227, still utilized the performance results from the 227×227 experiment, which was 21.58 seconds. Similarly, LeNet-5 (Greyscale) and LeNet-5 (RGB) with a default input size of 32×32 employed the performance results from the 32×32 experiment, amounting to 11.01 seconds and 11.00 seconds, respectively. ResNet-152, with a default input size of 224x224, had the longest training time among all models, totalling 160.61 seconds.

Based on the results, it is evident that training times increased as the input size grew larger. This can be attributed to the increased numbers of parameters and computations required to process larger images. Models like VGG and ResNet, which have deeper and more complex architectures, tend to have longer training times compared to simpler models like LeNet-5 and AlexNet. It is also worth noting that longer training times do not necessarily guarantee significantly better performance. Models like AlexNet and LeNet consistently showed good performance across different input sizes, even with relatively shorter training times. A visualization comparing the training times for each input size can be observed in Fig. 9.

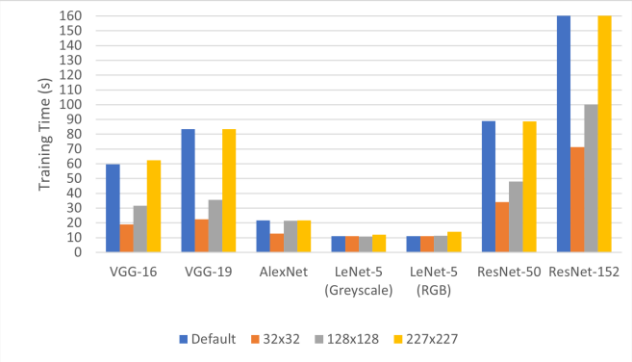


Fig. 9. Training time of each input size settings

Based on the performance results obtained from the confusion matrix, several conclusions can be drawn regarding the model's performance in this research task. One notable model that consistently performed well across all four testing sets was AlexNet, exhibiting the ability to accurately identify positive cases (True Positive) with high accuracy. Additionally, both LeNet-5 (Greyscale) and LeNet-5 (RGB) models demonstrated solid performance by correctly identifying both positive and negative cases. However, some models such as VGG-16 and VGG-19 tended to produce relatively higher numbers of false positives, particularly with increased input sizes. This suggests that these models may have a tendency to classify objects as the actual object with slight differences. On the other hand, the ResNet-152 model exhibited a high number of false negatives in certain tests, especially with larger input sizes. This indicates that the model tends to miss some positive object cases that should have been correctly identified. This was likely caused by differences in the architecture of the CNN model.

Furthermore, it is evident that the model's performance can be influenced by the input size. Some models displayed a decline in performance as the input size increased, while others demonstrated stable or even improved performance. Hence, choosing an appropriate input size can significantly impact the model's performance and final outcomes. Table 11 presents a comparison of input size, AUC, accuracy, precision, recall, and ROC values for several models previously used in similar cases, alongside the models employed in this research.

Table 11. Comparison of evaluation matrices with other models

Models	Input Size	AUC	Accuracy	Precision	Recall	F1 Score
CIdeR [52]	-	0.846	-	-	-	-
ResNet-18 [53]	224×224	-	0.76	-	-	-
VGG-16	32×32	0.812	0.76	0.66	0.8	0.73
VGG-19	32×32	0.863	0.73	0.70	0.6	0.64
AlexNet	227×227	0.930	0.86	0.86	0.8	0.83
LeNet-5	227×227	0.906	0.81	0.75	0.8	0.77
ResNet-50	128×128	0.830	0.70	0.75	0.4	0.52
ResNet152	128×128	0.828	0.73	0.62	0.87	0.72

Based on the comparison between the models used in this research and models from previous studies with similar cases, it can be concluded from Table 11 that VGG-19, AlexNet, and LeNet-5 models showcased superior performance based on the AUC values compared to the models used in previous research. Nevertheless, it is essential to note that the choice of dataset significantly influences the model's performance. Additionally, parameters and model architecture also play a crucial role in influencing the performance achieved.

In the given study, it is important to consider certain limitations that could have influenced the results. Firstly, the study only focused on a specific dataset, and the findings may not generalize to other datasets with different characteristics. The performance of the models could vary when applied to different domains or datasets with distinct features, such as variations in image quality, object types, or class imbalances. Another limitation is that the study evaluated the models using a fixed set of parameters. The selection of hyperparameters, such as learning rate, regularization techniques, and optimization algorithms, can significantly impact the model's performance. It is possible that different parameter settings could have led to different results. Performing a hyperparameter tuning or optimization could further enhance the performance of the models. Lastly, it is worth mentioning that the study primarily focused on training times as a measure of computational efficiency. However, training time alone may not be the sole factor determining the practicality of a model. In real-world applications, factors like inference time, model size, and hardware requirements also play a significant role. Considering these aspects would provide a more holistic understanding of the models' efficiency and practicality.

Future research could address these limitations by exploring different datasets, exploring a wider range of CNN models, optimizing hyperparameters, examining interpretability, and considering other practical aspects beyond training time.

4. CONCLUSION

This study investigated the impact of different input sizes on the performance of six CNN models in classifying COVID-19 positive or negative status based on cough sounds. Through four testing scenarios, it was observed that varying input sizes had an influence on the model's performance, with some models demonstrating improved performance as the input size increased, while others exhibited a decrease.

Among the models tested, AlexNet delivered the best performance, achieving the highest AUC of 0.930 at an input size of 227×227. The other models also exhibited relatively good performance, with an average AUC above 0.80. The ResNet-152 model showed the poorest performance at an input size of 32×32, with an AUC of 0.618.

The complexity of the models, including the number of layers and input size, had a notable impact on the training time. Larger input sizes and more complex models required longer training times.

It is crucial to emphasize that the dataset, model architecture, and parameters used play significant roles in determining the model's performance. For future research, it is recommended to employ larger and more diverse datasets, including samples from different populations, age groups, and disease severity levels, incorporate a more comprehensive set of features, explore different CNN models, and incorporate hyperparameter tuning techniques such as grid search or bayesian optimization to further enhance the performance of CNN models in this domain.

Acknowledgments

This study is the result of a final project conducted in the Computer Science Program at the Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University. The research was supported by the Program Dosen Wajib Meneliti (PDWM), funded by the Lambung Mangkurat University.

REFERENCES

[1] Y.-C. Wu, C.-S. Chen, and Y.-J. Chan, "The outbreak of COVID-19: An overview," *Journal of the Chinese Medical Association*, vol. 83, no. 3, pp. 217–220, Mar. 2020, https://doi.org/10.1097/JCMA.0000000000000270.

[2] A.-A. Seidu, J. E. Hagan, E. K. Ameyaw, B. O. Ahinkorah, and T. Schack, "The role of testing in the fight against COVID-19: Current happenings in Africa and the way forward," *International Journal of Infectious Diseases*, vol. 98, pp. 237–240, Sep. 2020, https://doi.org/10.1016/j.ijid.2020.06.089.

[3] W. Qiu *et al.*, "Machine Learning for Detecting Early Infarction in Acute Stroke with Non–Contrast-enhanced CT," *Radiology*, vol. 294, no. 3, pp. 638–644, Mar. 2020, https://doi.org/10.1148/radiol.2020191193.

[4] Q. Zhou *et al.*, "Cough Recognition Based on Mel-Spectrogram and Convolutional Neural Network," *Front Robot AI*, vol. 8, p. 580080, May 2021, https://doi.org/10.3389/frobt.2021.580080.

[5] L. Nanni, G. Maguolo, and M. Paci, "Data augmentation approaches for improving animal audio classification," *Ecol. Inform.*, vol. 57, p. 101084, May 2020, https://doi.org/10.1016/j.ecoinf.2020.101084.

[6] Y. R. Pandeya and J. Lee, "Domestic Cat Sound Classification Using Transfer Learning," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 2, pp. 154–160, Jun. 2018, https://doi.org/10.5391/IJFIS.2018.18.2.154.

[7] Y. R. Pandeya, D. Kim, and J. Lee, "Domestic Cat Sound Classification Using Learned Features from Deep Neural Nets," *Applied Sciences*, vol. 8, no. 10, p. 1949, Oct. 2018, https://doi.org/10.3390/app8101949.

[8] Z. Zhao *et al.*, "Automated bird acoustic event detection and robust species classification," *Ecol. Inform.*, vol. 39, pp. 99–108, May 2017, https://doi.org/10.1016/j.ecoinf.2017.04.003.

[9] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-Scale Weakly Supervised Audio Classification Using Gated Convolutional Neural Network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 121–125, Apr. 2018, https://doi.org/10.1109/ICASSP.2018.8461975.

[10] J. F. Gemmeke *et al.*, "Audio Set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, Mar. 2017, https://doi.org/10.1109/ICASSP.2017.7952261.

[11] M. S. Fernandes, W. Cordeiro, and M. Recamonde-Mendoza, "Detecting Aedes aegypti mosquitoes through audio classification with convolutional neural networks," *Comput. Biol. Med.*, vol. 129, p. 104152, Feb. 2021, https://doi.org/10.1016/j.compbiomed.2020.104152.

[12] H. Mukundarajan, F. J. H. Hol, E. A. Castillo, C. Newby, and M. Prakash, "Using mobile phones as acoustic sensors for high-throughput mosquito surveillance," *Elife*, vol. 6, p. e27854, Oct. 2017, https://doi.org/10.7554/eLife.27854.

[13] T. Kim, J. Lee, and J. Nam, "Comparison and Analysis of SampleCNN Architectures for Audio Classification," *IEEE J. Sel. Top. Signal Process*, vol. 13, no. 2, pp. 285–297, May 2019, https://doi.org/10.1109/JSTSP.2019.2909479.

[14] J. Lee, J. Park, K. Kim, and J. Nam, "SampleCNN: End-to-End Deep Convolutional Neural Networks Using Very Small Filters for Music Classification," *Applied Sciences*, vol. 8, no. 1, p. 150, Jan. 2018, https://doi.org/10.3390/app8010150.

[15] S. Bengani, S. Vadivel, and J. A. Arul Jothi, "Efficient Music Auto-Tagging with Convolutional Neural Networks," *Journal of Computer Science*, vol. 15, no. 8, pp. 1203–1208, Aug. 2019, https://doi.org/10.3844/jcssp.2019.1203.1208.

[16] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018, https://doi.org/10.48550/arXiv.1804.03209.

[17] A. Mesaros *et al.*, "Sound Event Detection in the DCASE 2017 Challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 992–1006, Jun. 2019, https://doi.org/10.1109/TASLP.2019.2907016.

[18] S. O. Arik, H. Jun, and G. Diamos, "Fast Spectrogram Inversion Using Multi-Head Convolutional Neural Networks," *IEEE Signal Process Lett*, vol. 26, no. 1, pp. 94–98, Jan. 2019, https://doi.org/10.1109/LSP.2018.2880284.

[19] R. Doshi *et al.*, "Extending Parrotron: An End-to-End, Speech Conversion and Speech Recognition Model for Atypical Speech," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6988–6992, Jun. 2021, https://doi.org/10.1109/ICASSP39728.2021.9414644.

[20] N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 3, pp. 170–173, Jun. 2020, https://doi.org/10.1109/CJECE.2020.2970144.

[21] D. Polap and M. Wozniak, "Voice recognition by neuro-heuristic method," *Tsinghua Sci Technol*, vol. 24, no. 1, pp. 9–17, Feb. 2019, https://doi.org/10.26599/TST.2018.9010066.

[22] Y. Singh, R. Kumar, and A. Biswas, "Swaragram: Shruti-Based Chromagram for Indian Classical Music," in *Advances in Speech and Music Technology*, pp. 109–118, 2021, https://doi.org/10.1007/978-981-33-6881-1_10.

[23] Y.-H. Byeon and K.-C. Kwak, "Pre-Configured Deep Convolutional Neural Networks with Various Time-Frequency Representations for Biometrics from ECG Signals," *Applied Sciences*, vol. 9, no. 22, p. 4810, Nov. 2019, https://doi.org/10.3390/app9224810.

[24] M. Loey and S. Mirjalili, "COVID-19 cough sound symptoms classification from scalogram image representation using deep learning models," *Comput. Biol. Med.*, vol. 139, p. 105020, Dec. 2021, https://doi.org/10.1016/j.compbiomed.2021.105020.
[25] J. Shen *et al.*, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4779–4783, Apr. 2018, https://doi.org/10.1109/ICASSP.2018.8461368.
[26] G. Chaudhari *et al.*, "Virufy: Global Applicability of Crowdsourced and Clinical Datasets for AI Detection of COVID-19 from Cough," *arXiv preprint arXiv:2011.13320*, 2021, https://doi.org/10.48550/arXiv.2011.13320.
[27] B. McFee *et al.*, "librosa/librosa: 0.10.0.post2," *Zenodo*, Mar. 2023, https://doi.org/10.5281/ZENODO.7746972.
[28] S. Mahmoud, M. Gaber, G. Farouk, and A. Keshk, "Heart Disease Prediction Using Modified Version of LeNet-5 Model," *International Journal of Intelligent Systems and Applications*, vol. 14, no. 6, pp. 1–12, Dec. 2022, https://doi.org/10.5815/ijisa.2022.06.01.
[29] W. Setiawan, Moh. I. Utoyo, and R. Rulaningtyas, "Classification of neovascularization using convolutional neural network model," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 1, p. 463, Feb. 2019, https://doi.org/10.12928/telkomnika.v17i1.11604.
[30] S. S. Patil, S. H. Patil, A. M. Pawar, N. S. Patil, and G. R. Rao, "Automatic Classification of Medicinal Plants Using State-Of-The-Art Pre-Trained Neural Networks," *Journal of Advanced Zoology*, vol. 43, no. 1, pp. 80–88, Oct. 2022, http://jazindia.com/index.php/jaz/article/view/116.
[31] A. Gaikwad, V. V Gohokar, R. Kute, and B. Paranjape, "Stair Detection and Classification Using Deep Neural Network for the Visually Impaired," *NVEO-Natural Volatiles & Essential Oils Journal NVEO*, pp. 8312–8321, 2021.
[32] H. Ismail, A. F. Mohamad Ayob, A. M. S. M Muslim, and M. F. R. Zulkifli, "Convolutional Neural Network Architectures Performance Evaluation for Fish Species Classification," *Journal of Sustainability Science and Management*, vol. 16, no. 5, pp. 124–139, Jul. 2021, https://doi.org/10.46754/jssm.2021.07.010.
[33] J. Ribeiro, S. Nóbrega, and A. Cunha, "Polyps Detection in Colonoscopies," *Procedia Computer Science*, vol. 196, pp. 477–484, 2022, https://doi.org/10.1016/j.procs.2021.12.039.
[34] D. K. Jain, P. Shamsolmoali, and P. Sehdev, "Extended deep neural network for facial emotion recognition," *Pattern Recognition Letters*, vol. 120, pp. 69–74, Apr. 2019, https://doi.org/10.1016/j.patrec.2019.01.008.
[35] R. A. Minhas, A. Javed, A. Irtaza, M. T. Mahmood, and Y. B. Joo, "Shot Classification of Field Sports Videos Using AlexNet Convolutional Neural Network," *Applied Sciences*, vol. 9, no. 3, p. 483, Jan. 2019, https://doi.org/10.3390/app9030483.
[36] Q. Guan *et al.*, "Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study," *J. Cancer*, vol. 10, no. 20, pp. 4876–4882, 2019, https://doi.org/10.7150/jca.28769.
[37] Z. Hu, Z. Yang, K. J. Lafata, F. Yin, and C. Wang, "A radiomics-boosted deep-learning model for COVID-19 and non-COVID-19 pneumonia classification using chest x-ray images," *Med. Phys.*, vol. 49, no. 5, pp. 3213–3222, May 2022, https://doi.org/10.1002/mp.15582.
[38] G. Wei, G. Li, J. Zhao, and A. He, "Development of a LeNet-5 Gas Identification CNN Structure for Electronic Noses," *Sensors*, vol. 19, no. 1, p. 217, Jan. 2019, https://doi.org/10.3390/s19010217.
[39] A. Victor Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 375–381, Nov. 2021, https://doi.org/10.1016/j.gltp.2021.08.027.
[40] S. Athisayamani, R. S. Antonyswamy, V. Sarveshwaran, M. Almeshari, Y. Alzamil, and V. Ravi, "Feature Extraction Using a Residual Deep Convolutional Neural Network (ResNet-152) and Optimized Feature Dimension Reduction for MRI Brain Tumor Classification," *Diagnostics*, vol. 13, no. 4, p. 668, Feb. 2023, https://doi.org/10.3390/diagnostics13040668.
[41] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, https://doi.org/10.1007/s13244-018-0639-9.
[42] Y. H. Liu, "Feature Extraction and Image Recognition with Convolutional Neural Networks," *J. Phys. Conf. Ser.*, vol. 1087, no. 6, p. 062032, Sep. 2018, https://doi.org/10.1088/1742-6596/1087/6/062032.
[43] S. Albawi, O. Bayat, S. Al-Azawi, and O. N. Ucan, "Social Touch Gesture Recognition Using Convolutional Neural Network," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–10, Oct. 2018, https://doi.org/10.1155/2018/6973103.
[44] A. Zafar *et al.*, "A Comparison of Pooling Methods for Convolutional Neural Networks," *Applied Sciences*, vol. 12, no. 17, p. 8643, Aug. 2022, https://doi.org/10.3390/app12178643.
[45] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016, https://doi.org/10.3390/app12178643.
[46] J. Shen, A. Aboutaleb, K. Sivakumar, B. J. Belzer, K. S. Chan, and A. James, "Deep Neural Network *a Posteriori* Probability Detector for Two-Dimensional Magnetic Recording," *IEEE Transactions on Magnetics*, vol. 56, no. 6, pp. 1–12, Jun. 2020, https://doi.org/10.1109/TMAG.2020.2985636.
[47] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, Mar. 2021, https://doi.org/10.1016/j.isprsjprs.2020.12.010.

[48] J. S. Kim, S. H. Kim, and S. B. Pan, "Personal recognition using convolutional neural network with ECG coupling image," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 5, pp. 1923–1932, 2020, https://doi.org/10.1007/s12652-019-01401-3.
[49] S. Naseer *et al.*, "Enhanced Network Anomaly Detection Based on Deep Neural Networks," *IEEE Access*, vol. 6, pp. 48231–48246, Jun. 2018, https://doi.org/10.1109/ACCESS.2018.2863036.
[50] O. Ghorbanzadeh, T. Blaschke, K. Gholamnia, S. Meena, D. Tiede, and J. Aryal, "Evaluation of Different Machine Learning Methods and Deep-Learning Convolutional Neural Networks for Landslide Detection," *Remote Sens (Basel)*, vol. 11, no. 2, p. 196, Jan. 2019, https://doi.org/10.3390/rs11020196.
[51] A. Kamilaris and F. X. Prenafeta-Boldú, "A review of the use of convolutional neural networks in agriculture," *The Journal of Agricultural Science*, vol. 156, no. 3, pp. 312–322, Apr. 2018, https://doi.org/10.1017/S0021859618000436.
[52] H. Coppock, A. Gaskell, P. Tzirakis, A. Baird, L. Jones, and B. Schuller, "End-to-end convolutional neural network enables COVID-19 detection from breath and cough audio: a pilot study," *BMJ Innovations*, vol. 7, no. 2, pp. 356–362, Apr. 2021, https://doi.org/10.1136/bmjinnov-2021-000668.
[53] M. Effati and G. Nejat, "A Performance Study of CNN Architectures for the Autonomous Detection of COVID-19 Symptoms Using Cough and Breathing," *Computers*, vol. 12, no. 2, p. 44, Feb. 2023, https://doi.org/10.3390/computers12020044.

BIOGRAPHY OF AUTHORS

Muhammad Fauzan Nafiz is an undergraduate student in the Department of Computer Science, Lambung Mangkurat University. His research interest is centered on Data Mining. Email: Muhammadnafiz643@gmail.com.

Dwi Kartini is a lecturer in Department of Computer Science, Lambung Mangkurat University. Her research interest is centered on Data Science. Email: dwikartini@ulm.ac.id.

Mohammad Reza Faisal is a lecturer in Department of Computer Science, Lambung Mangkurat University. His research interest is centered on Data Science. Email: reza.faisal@ulm.ac.id.

Fatma Indriani is a lecturer in Department of Computer Science, Lambung Mangkurat University. Her research interest is centered on Data Science. Email: f.indriani@ulm.ac.id.

Triando Hamonangan Saragih is a lecturer in Department of Computer Science, Lambung Mangkurat University. His research interest is centered on Data Science. Email: triando.saragih@ulm.ac.id.