Facial recognition and body temperature measurements based on thermal images using a deep-learning algorithm

Suci Dwijayanti, Muhammad Ridho Ramadhan, Bhakti Yudho Suprapto

Department of Electrical Engineering, Universitas Sriwijaya, Palembang, Indonesia

Article Info

ABSTRACT

Recognizing the early symptoms of the SARS-CoV-2 virus (COVID-19) is essential for minimizing its spread. One of the typical symptoms of a person infected with COVID-19 is increased body temperature beyond the normal range. Facial recognition can be used to separate healthy people from those with high body temperatures based on thermal images of the faces. In this study, the XEAST XE-27 thermal imager modes 2, 3, and 4 comprising 1500 thermal images each were compared. The facial recognition was performed using a convolutional neural network. Additionally, body temperatures were extracted from thermal images using matrix laboratory (MATLAB) by considering the minimum and maximum temperatures of each mode and class. The network training results indicate that the accuracies achieved by the proposed facial recognition system in modes 2, 3, and 4 are 87.33%, 92.33%, and 91.66%, respectively. Furthermore, the accuracies of body temperature extraction in modes 2, 3, and 4 are 70%, 60%, and 40%, respectively. Thus, the proposed system serves as a contactless technique for the early detection of COVID-19 symptoms by combining facial recognition and body temperature measurements.

Corresponding Author:

Suci Dwijayanti
Department of Electrical Engineering, Universitas Sriwijaya
Indralaya, Ogan Ilir, Sumatera Selatan 30662, Indonesia
Email: sucidwijayanti@ft.unsri.ac.id

1. INTRODUCTION

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2; COVID-19) caused the deaths of millions of people worldwide in 2020 [1]. The rapid transmission of this virus has increased the number of people affected by it in various countries. Song *et al*. [2] reported that all persons affected by COVID-19 exhibited symptoms of fever, followed by cough and fatigue. The fever symptoms of COVID-19 are mainly characterized by body temperatures exceeding the normal range. Thus, body temperature can be used as an early indicator for detecting COVID-19 infection.

Infrared thermometers are often used to measure body temperatures. When employing this method, data corresponding to an individual, such as the name of the person and other details, need to be recorded manually. This can result in inaccurate data collection. A person's identity can be obtained through facial recognition technology; therefore, a system that combines facial recognition and body temperature measurements is essential.

Facial recognition is a prevailing technology in the field of biometrics and has one of the highest levels of acceptance from users among biometric features, such as fingerprints or retina scans [3]. The use of infrared thermal images for recognizing faces is a commonly used approach to obtain data in terms of the subject's body temperature. The human body emits thermal radiation continuously. The higher the body

temperature, the greater is the intensity of the emitted infrared energy [4]. Infrared thermography is a contactless, adaptable, and noncalamitous technique used to measure body temperatures [5].

Some studies have been conducted to apply recognition to thermal images. Seal *et al*. designed a thermal infrared face recognition scheme using the gappy principal component analysis (PCA) method [6]. In this study, linear regression was used as the classifier. Joardar *et al*. discussed the impact of face pose to recognize thermal infrared face images. This study used feature extraction from raw images using patch-wise self-similarity [7]. Kantarci and Ekenel focused on the matching of the thermal-to-visible cross-domain face [8]. Litvin *et al*. also discussed the reconstruction of facial images in the visible spectrum using thermal images [9]. However, thermal images are very important for measuring temperature as they contain such information. Thus, this study focuses on the recognition of thermal images that can also be used to measure temperature.

The subject's face must be detected before performing facial recognition. The Haar cascade is an algorithm used to perform facial detection owing to its fast and highly efficient computations in terms of recognition of facial patterns [10]. However, this algorithm has several drawbacks, e.g., it can only detect faces. Therefore, the facial detection process must be supported by deep learning in the facial recognition phase.

A convolutional neural network (CNN) is a deep-learning method with the most significant results for image recognition [11]. An additional CNN study was conducted for thermal face recognition [12]. However, this study used secondary data of thermal images obtained from a specific database. In this study, the dataset contained primary data obtained from the thermal imager camera. Another study related to this work was conducted by Tan and Liu [13]. However, these authors utilized a combination of two cameras and used the thermal images to measure the temperature and a visible-light red–green–blue (RGB) camera to detect and recognize faces. The contribution of this study pertains to the use of thermal images to measure temperature using a pixel-based approach and recognize thermal images in three different modes using a combination of Haar cascade and CNN. In this study, the Haar-cascade method was combined with a CNN to test the trained data in the form of infrared images for facial recognition and obtaining body temperatures from subjects. Considering that there is a lack of literature on comparisons between facial recognition using infrared images and visible light, this study presents a thermal-image-based facial recognition system.

2. CNN

A CNN is a deep-learning method with a high degree of tissue depth. Therefore, it is efficient for classifying image data [14] to obtain the best representation. CNNs were developed from multilayer perceptrons (MLPs), which exhibit certain drawbacks, such as the inabilities to store spatial information from image data and to consider each pixel as an independent feature [15]. Figure 1 depicts the workflow of the CNN method. The CNN is composed of various layers, such as convolutional, activation, pooling, fully connected, and batch normalization layers in its architecture [14]. However, the typical CNN architecture involves three layers of neurons, namely convolutional, pooling, and fully connected layers.

The convolutional layer performs convolution operations on the output of the previous layer. This layer underlies the CNN architecture generated by the kernels or convolution filters. It comprises a small filter of weights that convolves the one-dimensional input as an image represented in the form of a matrix [10]. The convolution operation on this layer can be calculated using the following equation,

$$h(t) = \sum_{a=-\infty}^{\infty} I(a).K(t-a) \tag{1}$$

where $h(t)$ denotes the convolution result, $I(a)$ indicates the input image, and $K(a)$ represents the kernel.
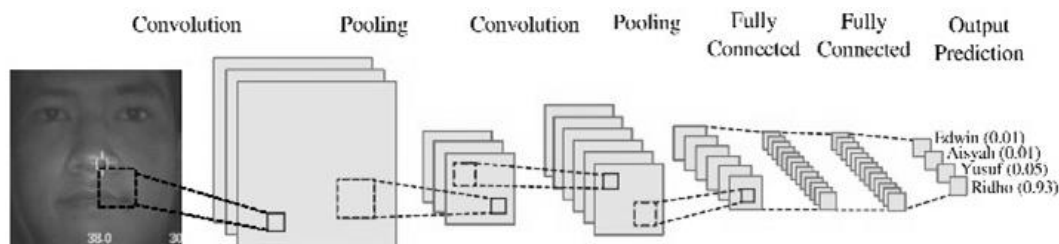

Figure 1. Illustration of the workflow of a convolutional neural network (CNN)

The pooling layer comprises hyperparameter stride and pooling size [16], which are shifted alternately throughout the feature map area. Average and max pooling are the commonly used pooling layers [17]. The data in the pooling layer are downsampled to decrease the feature map size and extract features to achieve an efficient training process [17]. The max pooling has a feature pattern output (z) that can be obtained by determining the maximum value [18] for each spatial block at each shift. The max pooling equation is expressed,

$$u_{i,j}^{(k)} = \max z_{p,q}^{(k)} \tag{2}$$

Meanwhile, the average pooling generates the output of the feature pattern by reporting the average values over blocks of each input feature [19]. The average pooling equation is given,

$$u_{i,j}^{(k)} = \frac{1}{H^2}\sum_{(p,q)\in Pij} z_{p,q}^{(k)} \tag{3}$$

The fully connected layer is completely connected to all the previous layers and produces the final prediction output [20]. This layer is used in the application of MLPs to transform data dimensions for linear data classification [21]. The feature map obtained in the previous layer undergoes a flattening process to produce a vector that can be input from the fully connected layer. To generate feature patterns in the network architecture, the activation function is placed either after convolution, after pooling calculations, or in the final calculation of the feature map output; softmax is a commonly used activation function. The softmax activation function is used to convert the actual values generated by the CNN layers into probabilities [22]. This activation function is used for more than two classes and is based on consideration of the probability of the target class in the final layer of the neural network. The greater the softmax activation value, the higher is the probability of the data being part of a class. The softmax activation function is given,

$$f_i(Z) = \frac{e^{zJ}}{\sum_k e^{zk}}, \tag{4}$$

where the notation f_i is the result of the function for each jth element in the class output vector, argument z denotes an arbitrary vector with actual values generated at the ith CNN layer, and k indicates the vector size [22]. Softmax provides a better probability interpretation than other classification algorithms because of its ability to calculate the probabilities of all labels and predict the probability of the input image for each category [23].

During CNN training, a learning parameter exists in the form of a loss function. Two types of loss functions, namely the mean-squared error (MSE) and categorical cross-entropy [24], are used often. The MSE loss function can be calculated using the following equation,

$$L(y,\hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(y - \hat{y_i})^2 \tag{5}$$

where y is the actual value and $\hat{y}$ is the predicted value.

Conversely, the categorical cross-entropy loss function can be calculated mathematically,

$$L = -\sum_{i=1}^{output\ size} y_i.log\hat{y_i} \tag{6}$$

3. RESEARCH METHODOLOGY

3.1. Data collection

Datasets were obtained in the form of thermal images using the thermal imager XEAST XE-27, which has a tolerance of $\pm2\%$ for temperatures > 0 °C. The thermal imager XEAST XE-27 has five types of thermal imager modes, where the higher modes are more difficult to recognize using naked eyes. The second, third, and fourth thermal imager modes were thus used because these modes can determine whether the images can be recognized. The resulting images were of size 320×240 pixels. The face was imaged at a distance of approximately 35–40 cm from the camera. Twenty students from the Faculty of Electrical Engineering at Sriwijaya University were the subjects in this research; all participants provided informed consent regarding the use of their data. Images of the subjects' faces with expressions were captured from different angles. The intensity of the ambient light was constant in each sample. The thermal images included 20 classes, with each class comprising 75 images. In each class, 60 and 15 thermal images were used as the training and test data, respectively. Figures 2(a) to 2(e) depict an image captured in first, second, third, fourth, and fifth modes of the thermal imager, respectively.
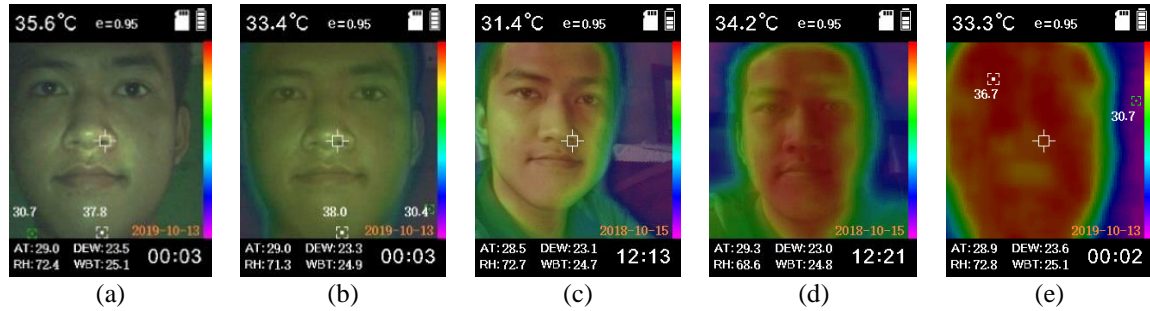
Figure 2. Thermal images captured from the thermal imager in the (a) first, (b) second, (c) third, (d) fourth, and (e) fifth modes

3.2. Body temperature measurements

Temperatures were extracted from thermal images of the participants' faces using matrix laboratory (MATLAB) (R2018a, MathWorks, Natick, MA, and USA). The limitation of the XEAST XE-27 thermal imager is that it can only obtain the final temperature data of the detected object. However, the temperature from thermal images that are not preprocessed can be extracted only if the minimum and maximum temperatures can be detected by the camera. Additionally, the minimum and maximum temperatures of each image exhibit a probability of variation based on variations in the ambient temperature. Thus, the coefficients associated with the minimum and maximum temperature limits must be determined for each frame; accordingly, a fixed value cannot be used for all frames.

To read the facial image files from which the temperature is extracted, the system reads the file path of the facial image as an input. The thermal image is considered as a grayscale image (0–255) to ease the temperature interpolation. Subsequently, the pixel values in the image were interpolated into temperatures by assuming that the minimum and maximum temperature limits indicate the lowest and highest intensities, respectively. In this study, the temperature limits for the minimum and maximum values were set to 32 °C and 42 °C, respectively. These values were chosen as they are the range of values of the thermo gun used to measure body temperatures. The final temperature of the input image was obtained by determining the average of all pixels that were converted into temperatures using interpolation.

3.3. CNN architecture

The reliability of a neural network can be determined using the CNN architecture. In this study, the best architecture used in the fourth mode of XEAST XE-27 was tested. The CNN architecture used in each mode comprised seven layers, namely convolutional layer 1, pooling layer 1, convolutional layer 2, pooling layer 2, flatten layer, and two fully connected layers. Figure 3 illustrates the CNN architecture.
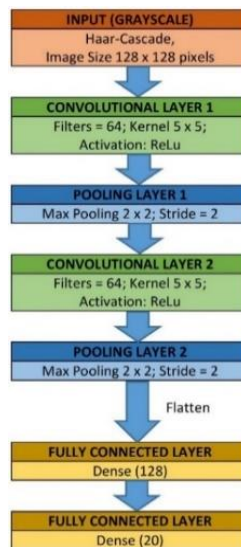

Figure 3. Architecture of the convolutional neural network model

3.4. Testing phase

System testing was performed using the confusion matrix method, which measures the classifier in terms of predicting various classes. The confusion matrix contains information on the classification system in the form of actual classes and prediction classes, represented by rows and columns, respectively [25]. Table 1 lists the parameters used in the confusion matrix model. These parameters are the basis for calculating the accuracy, precision, and recall values.

Table 1. Parameters of the confusion matrix

Confusion matrix		Actual values	
		True	False
Predicted values	True	True positive	False positive
	False	False negative	True negative

4. RESULTS AND DISCUSSION

4.1. Preprocessing dataset

The thermal images were preprocessed into some steps. It aims to ease the training of the CNN algorithm. This phase involved three steps, namely scaling, grayscaling, and augmentation.

4.1.1. Scaling

The scaling process was performed to obtain thermal images of equal sizes without affecting image quality. In this study, the Haar-cascade (Viola–Jones) algorithm was used for face detection. Thereafter, the faces in the images were localized to ensure identical dimensions of 128×128 pixels across all datasets. Figure 4(a) depicts the results of face detection using the Haar-cascade algorithm and Figure 4(b) is the scaling result, which were then used as the inputs to the grayscaling process.

4.1.2. Grayscaling

Grayscaling was achieved by converting the scaled infrared image to a grayscale image. This process resulted in more efficient storage of the image data, and the algorithm obtained the image characteristics conveniently. Figure 5 depicts the results of the grayscaling process.
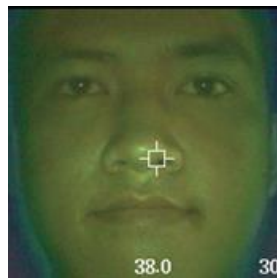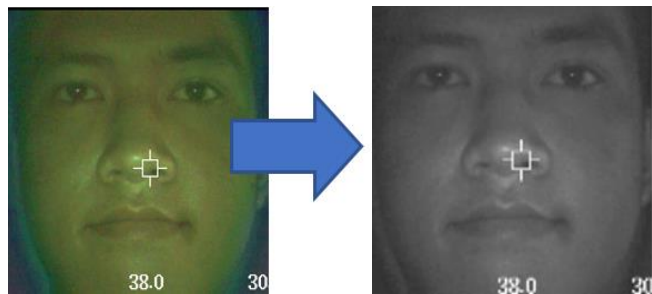

Figure 4. Scaling preprocessing: (a) facial detection and (b) 128×128-pixel image

Figure 5. Grayscaling preprocessing

4.1.3. Augmentation

Data augmentation was performed by shifting, brightening, darkening, magnifying, and flipping each image that had undergone grayscaling. This augmentation process aims to increase the variety of data without losing the primary characteristics of each image. Table 2 lists several types of facial image augmentation methods.

4.2. Temperature readings

The temperature was extracted by initializing the minimum and maximum temperature limits of the tested image. In the second mode, the minimum and maximum temperatures used were 33.05 °C and 40.1 °C, respectively. The third mode used minimum and maximum temperatures of 33.05 °C and 40.35 °C, respectively. In the fourth mode, the minimum and maximum temperatures used were 32.93 °C and 42 °C, respectively. To validate the accuracy of the extraction process, trials of each mode were performed using 10

thermal images from an individual sample. Table 3 summarizes the test results for the temperature readings in modes 2 to 4.

Table 2. Augmentation for the preprocessing phase

Augmentation method	Image results
Height shifting Shift the image pixels by 10% of the height of the image dimensions	
Width shifting Shift the image pixels by 10% of the height of the image dimensions	
Horizontal flip Flip the image horizontally	
Zoom Enlarge the image by 110% of its previous size	
Brightness Brighten the image with a value of 1.3 (right) and darken the image with a value of 0.8 (left)	

Table 3. Temperature extraction from modes 2 to 4 of the thermal imager XEAST XE-27

Mode 2			Mode 3			Mode 4		
Actual temperature (°C)	Measured temperature (°C)	Results	Actual temperature (°C)	Measured temperature (°C)	Results	Actual temperature (°C)	Measured temperature (°C)	Results
35.8	35.8	Correct	35.6	35.6	Correct	35.4	35.4	Correct
35.9	35.8	Correct	35.7	35.6	Wrong	35.5	35.5	Correct
35.7	35.7	Correct	35.8	35.6	Wrong	35.5	35.5	Correct
35.6	35.7	Wrong	35.8	35.6	Wrong	35.5	35.5	Correct
35.7	35.7	Correct	35.6	35.6	Correct	35.6	35.5	Wrong
35.6	36.0	Correct	35.7	35.7	Correct	35.5	35.3	Wrong
35.6	36.0	Correct	35.7	35.7	Correct	35.5	35.3	Wrong
35.6	36.1	Correct	35.8	35.8	Correct	35.6	35.2	Wrong
35.8	36.0	Wrong	35.4	35.5	Wrong	35.6	35.2	Wrong
35.8	35.9	Wrong	35.6	35.6	Correct	35.6	35.5	Wrong

As shown in Table 3, extraction in mode 2 indicated that 7 of the 10 test images yielded accurate extraction temperatures, whereas the third and fourth modes yielded six and four correct predictions, respectively. However, these results indicate that the measured temperatures using the proposed approach of body temperature measurement were close to the actual temperatures measured directly from the thermal imager. From the table, it can be seen that the minimum and maximum values can be used to extract the temperatures from the images. This indicates that the colors in the thermal image represent the spread of human body temperatures.

4.3. Facial recognition

Facial recognition was performed by loading the designed CNN architecture model of the preprocessed thermal facial image data. The training process in this study used 80% of the total data, which comprised 1,200 images. This training process was implemented for 20 types of architecture model scenarios based on various learning parameters. The architecture model tested herein was created by changing the learning parameters, such as the number of epochs, filter size, kernel size, optimizer type, loss function type, and learning rate. The training process aimed to identify the CNN architecture model that produced the best classification performance. Furthermore, the architecture model was stored and used in the testing process for local thermal facial images. The faces were recognized using the three modes of the thermal imager XEAST XE-27, namely modes 2 to 4.

4.3.1. Effects of the number of epochs

The first parameter considered in the training process is the number of epochs. In this study, five different numbers of epochs were used, namely 50, 100, 150, 200, and 250. As indicated in Figure 6, the highest accuracy achieved was 86.67% at 100 epochs in the case of the second mode shows in Figure 6(a), 91.33% at 200 epochs in the case of the third mode shows in Figure 6(b), and 94.33% at 100 epochs in the case of the fourth mode shows in Figure 6(c). However, the accuracy decreased when the number of epochs increased, indicating that higher numbers of epochs can result in overfitting that may render the network incapable of generalizing the existing parameters. Thus, the best number of epochs in this study is 100 for thermal images in modes 2 and 3.
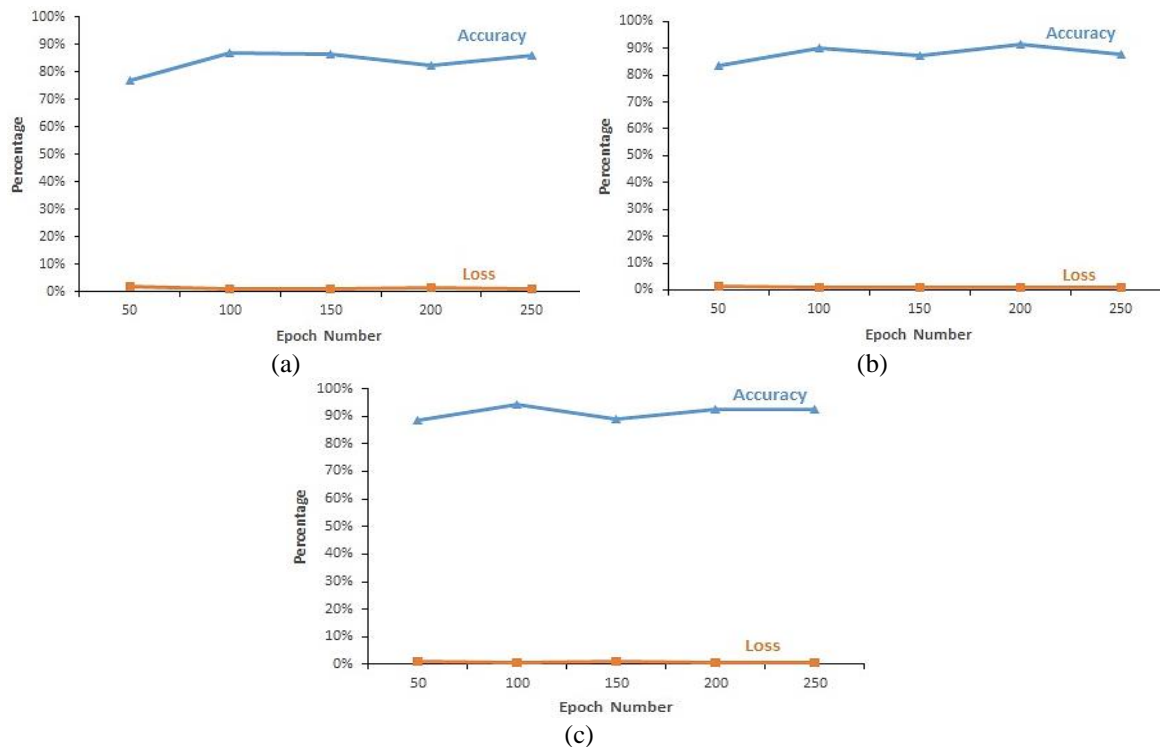

Figure 6. Effects of the epoch number on modes (a) 2, (b) 3, and (c) 4

4.3.2. Effects of filter number and kernel size

The convolution layer has a filter with small dimensions in the form of the kernel, which is represented as a matrix to perform convolution with the thermal face images. Thus, the filter number and kernel size are important in the training process. To evaluate their effects, various filter numbers and kernel sizes were examined, where the loss parameter used in the test was the MSE and optimizer was RMSprop with a learning rate of 0.001. As indicated in Figure 7, the highest accuracies were achieved for each case using a 5×5 kernel size and 64 filters; the accuracies obtained were 86.67%, 91.33%, and 94.33% for the second in Figure 7(a), third in Figure 7(b), and fourth modes in Figure 7(c), respectively. If the number of filters was reduced to 32 and kernel size was reduced to 3×3, the accuracy decreased slightly, and the resulting losses tended to increase. This can be attributed to the input image size used, which is 128×128 pixels. Specifically, more image details can be obtained from a kernel size of 5×5 compared with those extracted with a kernel size of 3×3.

4.3.3. Effects of optimizer and loss function type

The optimizer aims to maximize the accuracy and minimize the loss value so that optimal weights can be obtained from the training process. Thus, it is essential to find a suitable optimizer and loss function type. Stochastic gradient descent (SGD), root mean-squared propagation (RMSprop), and Adam were evaluated as optimizers in this study. Meanwhile, the loss functions used were MSE and cross entropy. As indicated in Figures 8(a) to 8(c), the highest accuracies achieved were 81.67%, 89.99%, and 94.33% for the second, third, and fourth modes, respectively, with the MSE as the loss function and RMSprop as the optimizer. The MSE obtained a loss of < 5% for all modes. The MSE loss function tended to yield a lower loss percentage in

comparison with the categorical cross-entropy loss function. The optimizers were compared in terms of the highest accuracy results generated by the RMSprop optimizer with the MSE loss function and the Adam optimizer with the categorical cross-entropy loss function. Conversely, the SGD optimizer yielded the lowest accuracy for both loss functions in comparison with the other two optimizers. Thus, the results indicate that RMSprop (as an optimizer) and MSE (as a loss function) are ideal for training the CNN using thermal image data.

(a)   (b)

(c)
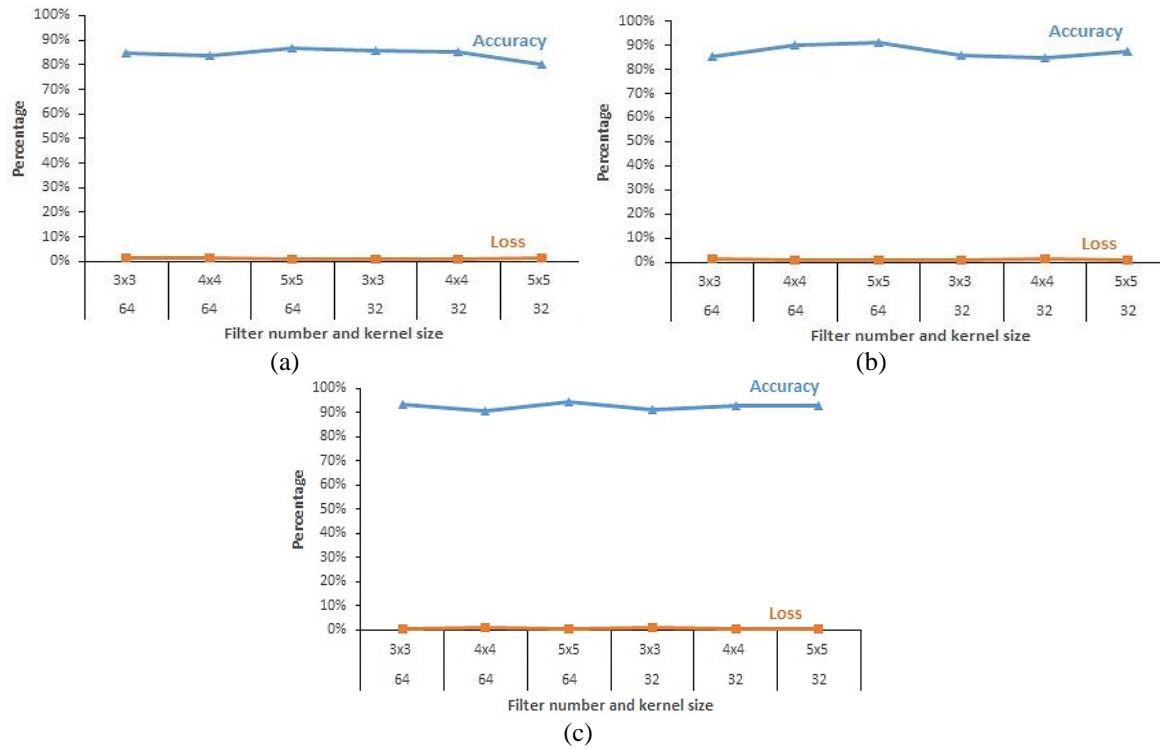
Figure 7. Effects of filter number and kernel size in modes (a) 2, (b) 3, and (c) 4

4.3.4. Effects of learning rate

This study also involved evaluation of the learning rate in the training process of the thermal images using CNN. The learning rate has a significant role in the performance of the training model because it renews the weights in the training process. The effects of the learning rate are shown in Figure 9. As indicated in the Figures 9(a) to 9(c), the highest accuracies were achieved when the learning rate was 0.001; these were 86.67%, 91.33%, and 94.33% for the second, third, and fourth modes, respectively. If the learning rates were increased to 0.005 and 0.01 or decreased to 0.0001 and 0.00001, the accuracies were substantially lower and the losses tended to increase. Therefore, a learning rate of 0.001 was considered optimal for the training process in this study.

Based on the training process by comparing the effects of various parameters, the best architecture model exhibited the highest level of accuracy and lowest loss over other architecture models. The feature map was obtained from the trained image by performing parameter iterations for 100 epochs. The process was repeated 100 times, where each iteration used a batch size of 20. The filter size used was 64, and the kernel size was 5×5. MSE was used as the loss parameter in this architecture model, and RMSprop with a learning rate of 0.001 was selected as the optimizer for the best architecture model to update the weights. The accuracy and model loss were 94.33% and 0.53%, respectively. This model was later used with the test data that were not included in the training process.

4.4. Confusion matrix for testing data

The best model obtained from training was used with the test data. The results of recognition on the test data are presented in the confusion matrix shown in Figure 10. Based on the results of the confusion matrix shows in Figure 10, the best CNN architecture model used 64 filters, a kernel size of 5×5, the MSE as the loss function, RMSprop as the optimizer, a learning rate of 0.001, and 100 epochs. The model exhibited accuracies greater than or equal to 94%, indicating that the CNN-based architecture can effectively recognize faces from thermal images even in mode 4, which entails more difficult recognition than the other two modes. Thus, the proposed model was able to recognize thermal images well despite image distortions from the applied thermal filter and feature distortions in the thermal images. This was further validated by the success of the local image testing.
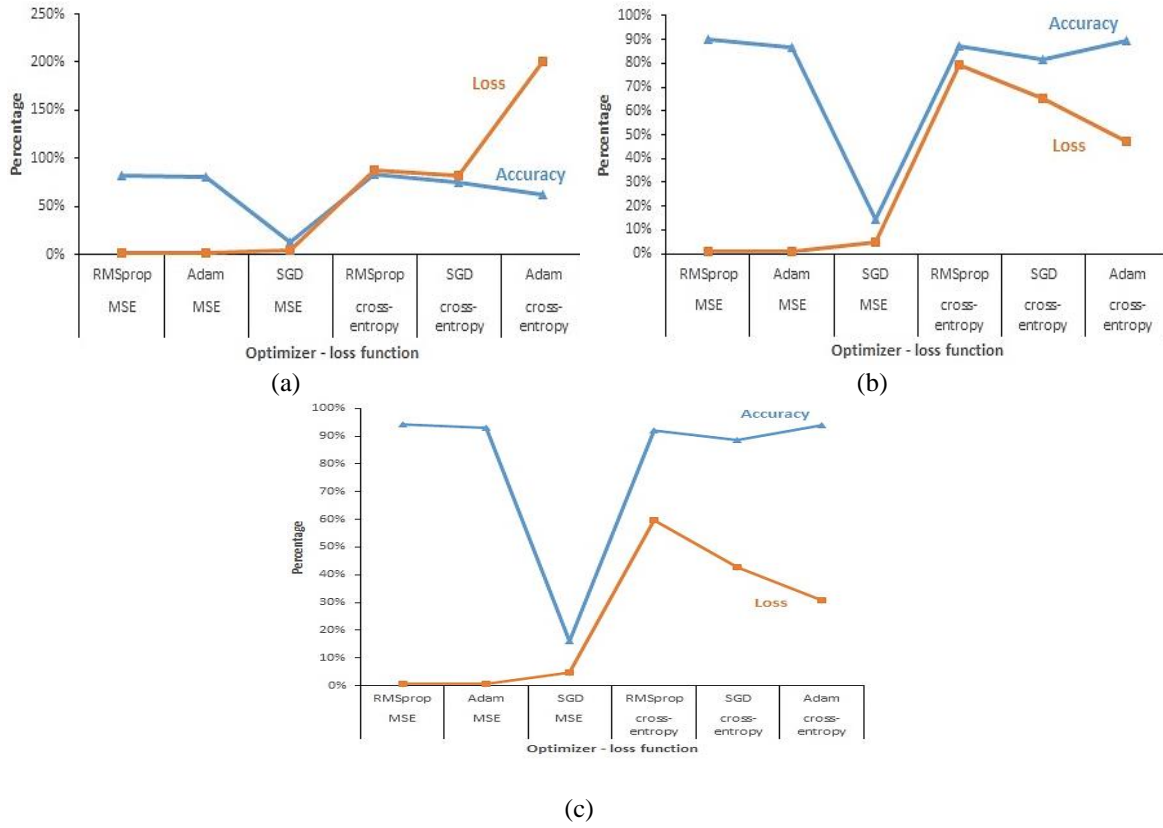
(a)

(b)

(c)

Figure 8. Effects of the optimizer and loss function types in modes (a) 2, (b) 3, and (c) 4
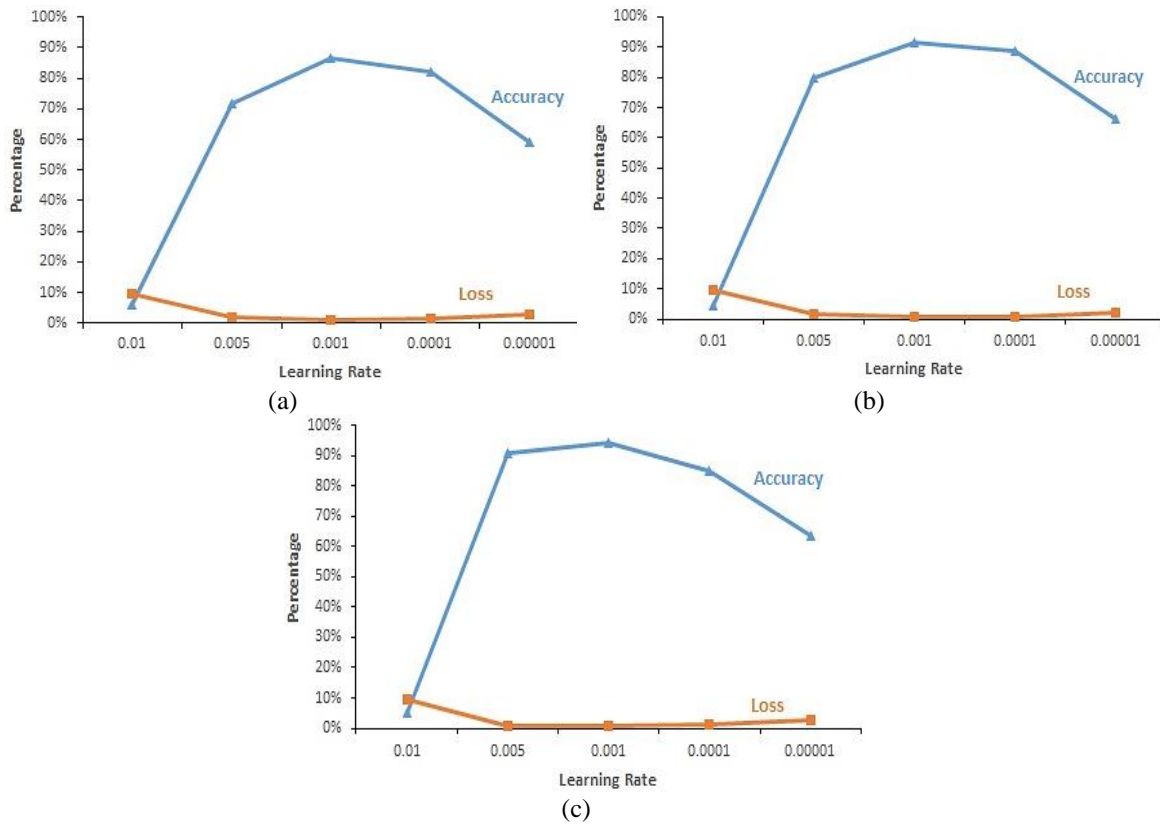

(a)

(b)

(c)

Figure 9. Effects of the learning rate in modes (a) 2, (b) 3, and (c) 4

4.5. Recognition of thermal local images

The CNN model was next used to recognize the thermal images. The tested local images contained raw (unprocessed) data obtained directly from the XEAST XE-27 thermal imager. The testing process was performed by detecting faces in the local images using the Haar-cascade algorithm. The best CNN architecture model with 64 filters, kernel size of 5×5, MSE as the loss function, RMSprop as the optimizer, learning rate of 0.001, and 100 epochs was loaded for each mode and applied for testing the local images. The predicted subject names and recognition probabilities are displayed and applied to these tested local images.

The recognition of local images was applied to the faces of all twenty participants for each mode (modes 2 to 4). Table 4 lists the sample results of the local images tested with the best CNN architecture model. As shown in the table, the local image testing verified that the names of 15 subjects were predicted correctly. For the testing performed in the three different modes, 16 images of mode 2, 16 images of mode 3, and 15 images of mode 4 were recognized well. However, four images each in modes 2 and 3 and five images in mode 4 were predicted incorrectly; this indicates that as the thickness of the thermal filter increases, a smaller minNeighbors parameter value must be applied because it is more difficult to detect faces with thick thermal filters. Table 4 also shows that the faces and temperatures are detected well from the thermal images. Hence, this combination can be used to detect the early symptoms of COVID-19 through body temperature measurements and identify the individual via face recognition.
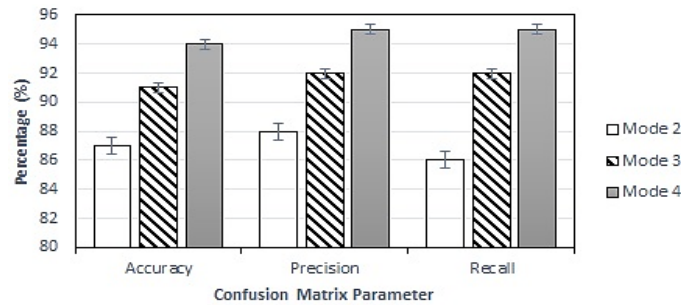

Figure 10. Confusion matrix test results

Table 4. Some local image test results for modes 2 to 4 of the thermal imagers

	Mode 2			Mode 3			Mode 4	
Tested image	Tested results		Tested image	Tested results		Tested Image	Tested results	
	Prediction	Accuracy		Prediction	Accuracy		Prediction	Accuracy
	Sample 1	88.17%		Sample 6	92.30%		Sample 11	99.10%
	Sample 2	62.85%		Sample 7	100%		Sample 12	100%
	Sample 3	91.56%		Sample 8	99.99%		Sample 13	100%
	Sample 4	99.99%		Sample 9	100%		Sample 14	100%
	Sample 5	99.71%		Sample 10	99.27%		Sample 15	92.29%

5. CONCLUSION

In this study, the application of a deep-learning algorithm was validated for facial recognition and body temperature measurements based on thermal images. The experimental results confirm that the thermal image facial recognition process can be performed successfully using a CNN model with seven layers, namely convolution layer 1, pooling layer 1, convolution layer 2, pooling layer 2, flatten layer, and two fully connected layers. It was determined that the maximum accuracy could be achieved using the MSE as the loss function, RMSprop as the optimizer, 100 epochs, 64 filters, a kernel size of 5×5, and a learning rate of 0.001. The training process produced validation accuracies of 87.33%, 92.33%, and 91.66% for modes 2, 3, and 4, respectively. Furthermore, the temperature was successfully extracted from the thermal images using specific minimum and maximum temperature limits with accuracies of 70%, 60%, and 40% in the second, third, and fourth modes, respectively. Although the facial recognition and body temperature detection were successful, additional improvements are required to obtain more optimal results. As facial recognition was performed using local images, real-time thermal facial recognition needs to be explored using a high-quality, forward-looking infrared camera. Additionally, other CNN architectures can be investigated to achieve higher and more reliable architecture models, such as the visual geometry group network (VGGnet), AlexNet, and so on. The number of datasets used in this study could also be increased to enhance the training and testing accuracies.

ACKNOWLEDGEMENTS

REFERENCES
[1] Y. Han, K. M. Millar, and M. J. Bayly, "COVID-19 as a mass death event," *Ethics and International Affairs*, vol. 35, no. 1, pp. 5–17, 2021, doi: 10.1017/S0892679421000022.
[2] F. Song *et al.*, "Emerging 2019 novel coronavirus (2019-NCoV) pneumonia," *Radiology*, vol. 295, no. 1, pp. 210–217, 2020, doi: 10.1148/radiol.2020200274.
[3] V. Alizadeh, S. Rayatdoost, and E. Arbabi, "Effect of different partitioning strategies of face imprint on thermal face recognition," *22nd Iranian Conference on Electrical Engineering, ICEE 2014*, pp. 1108–1112, 2014, doi: 10.1109/IranianCEE.2014.6999701.
[4] X. Dérobert, J. P. Balayssac, Z. M. Sbartaï, and J. Dumoulin, "Electromagnetic methods," *Non-destructive Testing and Evaluation of Civil Engineering Structures*, pp. 87–137, 2017, doi: 10.1016/B978-1-78548-229-8.50003-0.
[5] J. Crisóstomo and R. Pitarma, "The importance of emissivity on monitoring and conservation of wooden structures using infrared thermography," *Advances in Structural Health Monitoring*, 2019, doi: 10.5772/intechopen.82847.
[6] A. Seal, D. Bhattacharjee, M. Nasipuri, and D. K. Basu, "Thermal human face recognition based on GappyPCA," *2013 IEEE 2nd International Conference on Image Information Processing, IEEE ICIIP 2013*, pp. 597–600, 2013, doi: 10.1109/ICIIP.2013.6707662.
[7] S. Joardar, D. Sen, D. Sen, A. Sanyal, and A. Chatterjee, "Pose invariant thermal face recognition using patch-wise self-similarity features," *Proceedings - 2017 3rd IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2017*, vol. 2017-December, pp. 203–207, 2017, doi: 10.1109/ICRCICN.2017.8234507.
[8] A. Kantarci and H. K. Ekenel, "Thermal to visible face recognition using deep autoencoders," *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*, vol. P-296, pp. 213–220, 2019.
[9] A. Litvin, K. Nasrollahi, S. Escalera, C. Ozcinar, T. B. Moeslund, and G. Anbarjafari, "A novel deep network architecture for reconstructing RGB facial images from thermal for face recognition," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 25259–25271, 2019, doi: 10.1007/s11042-019-7667-4.
[10] C. Li, Z. Qi, N. Jia, and J. Wu, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," *ICEMI 2017 - Proceedings of IEEE 13th International Conference on Electronic Measurement and Instruments*, vol. 2018-January, pp. 483–487, 2017, doi: 10.1109/ICEMI.2017.8265863.
[11] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *Journal of Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0276-2.
[12] Z. Wu, M. Peng, and T. Chen, "Thermal face recognition using convolutional neural network," *Proceedings - 2016 International Conference on Optoelectronics and Image Processing, ICOIP 2016*, pp. 6–9, 2016, doi: 10.1109/OPTIP.2016.7528489.
[13] W. Tan and J. Liu, "Application of face recognition in tracing COVID-19 fever patients and close contacts," *Proceedings - 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020*, pp. 1112–1116, 2020, doi: 10.1109/ICMLA51294.2020.00179.
[14] Z. Xiang, R. Zhang, and P. Seeling, "Machine learning for object detection," *Computing in Communication Networks: From Theory to Practice*, pp. 325–338, 2020, doi: 10.1016/B978-0-12-820488-7.00034-7.
[15] I. A. Anjani, Y. R. Pratiwi, and S. Norfa Bagas Nurhuda, "Implementation of deep learning using convolutional neural network algorithm for classification rose flower," *Journal of Physics: Conference Series*, vol. 1842, no. 1, 2021, doi: 10.1088/1742-6596/1842/1/012002.
[16] T. Zebin, P. J. Scully, N. Peek, A. J. Casson, and K. B. Ozanyan, "Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition," *IEEE Access*, vol. 7, pp. 133509–133520, 2019, doi: 10.1109/ACCESS.2019.2941836.
[17] H. J. Jie and P. Wanda, "Runpool: A dynamic pooling layer for convolution neural network," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 66–76, 2020, doi: 10.2991/ijcis.d.200120.002.
[18] F. I. Chou, Y. K. Tsai, Y. M. Chen, J. T. Tsai, and C. C. Kuo, "Optimizing parameters of multi-layer convolutional neural network by modeling and optimization method," *IEEE Access*, vol. 7, pp. 68316–68330, 2019, doi: 10.1109/ACCESS.2019.2918563.
[19] J. Teuwen and N. Moriakov, "Convolutional neural networks," *Handbook of Medical Image Computing and Computer Assisted Intervention*, pp. 481–501, 2019, doi: 10.1016/B978-0-12-816176-0.00025-9.

[20] C. H. Yeh, M. H. Lin, P. C. Chang, and L. W. Kang, "Enhanced visual attention-guided deep neural networks for image classification," *IEEE Access*, vol. 8, pp. 163447–163457, 2020, doi: 10.1109/ACCESS.2020.3021729.

[21] J. M. Valls, R. Aler, I. M. Galván, and D. Camacho, "Supervised data transformation and dimensionality reduction with a 3-layer multi-layer perceptron for classification problems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 12, pp. 10515–10527, 2021, doi: 10.1007/s12652-020-02841-y.

[22] I. Kouretas and V. Paliouras, "Simplified hardware implementation of the softmax activation function," *2019 8th International Conference on Modern Circuits and Systems Technologies, MOCAST 2019*, 2019, doi: 10.1109/MOCAST.2019.8741677.

[23] Q. Zhu, Z. He, T. Zhang, and W. Cui, "Improving classification performance of softmax loss function based on scalable batch-normalization," *Applied Sciences (Switzerland)*, vol. 10, no. 8, 2020, doi: 10.3390/APP10082950.

[24] E. Dufourq and B. A. Bassett, "Automated problem identification: Regression vs classification via evolutionary deep networks," *ACM International Conference Proceeding Series*, vol. Part F130806, 2017, doi: 10.1145/3129416.3129429.

[25] R. O. Duda, D. G. Stork, and P. E.Hart, "Pattern classification, 2nd edition," *Handbook of Neural Computation*, 2004, doi: 10.1887/0750303123/b365c83.

BIOGRAPHIES OF AUTHORS

Suci Dwijayanti received her M.S. in Electrical and Computer Engineering from Oklahoma State University, Stillwater, OK, USA, in 2013. She received a Fulbright scholarship for her Master's degree. In 2018, she received her Doctoral degree from the Graduate School of Natural Science and Technology, Kanazawa University, Japan. Her research interests include signal processing and machine learning. She also worked as an engineer at ConocoPhillips Indonesia Inc. Ltd. from 2007 to 2008. Since 2008, she has been with the Department of Electrical Engineering at Universitas Sriwijaya, Indonesia. She became an IEEE member in 2014. She can be contacted at email: sucidwijayanti@ft.unsri.ac.id.

Muhammad Ridho Ramadhan was born in Palembang, South Sumatra, Indonesia, in 1999. He received his B E. (S.T.) degree with cum laude honors in Electrical Engineering from the Universitas Sriwijaya, Indralaya, in 2021. From 2018 to 2019, he was a Laboratory Assistant at the Electronic Basic and Electrical Circuit Laboratory of Universitas Sriwijaya. In 2020, he was appointed as the President Director of Engineering Science Community of the Engineering Faculty, Universitas Sriwijaya. He was ranked as an outstanding Engineering faculty member at the Universitas Sriwijaya in 2020, with over 40 achievements, more than 30 event speakers, and two published papers. His research interests include deep learning, image processing, internet-of-things (IoT), data visualization, renewable energy, and innovation health monitoring devices. He was also an awardee of the Bright Scholarship from Yayasan Baitul Maal Bank Rakyat Indonesia, and the Bidikmisi Scholarship from the Ministry of Research, Technology, and Higher Education along his studies in Universitas Sriwijaya. He can be contacted at email: m.ridhoramadhan29@gmail.com.

Bhakti Yudho Suprapto was born on February 11, 1975 in Palembang (South Sumatra, Indonesia). He is a graduate of Universitas Sriwijaya with a major in Electrical Engineering. His Master's and Doctoral degrees were in Electrical Engineering from Universitas Indonesia (UI). He is an academic staff of Electrical Engineering of Sriwijaya University of Palembang. His research interests are control and intelligent systems. He became an IEEE member in 2014. He can be contacted at email: bhakti@ft.unsri.ac.id.