

# Automatic Gate for Body Temperature Check and Masks Wearing Compliance Using an Embedded System and Deep Learning

Rivaldo Fernandes <sup>1</sup>, Martin Clinton Tosima Manullang <sup>1</sup>, Rahman Indra Kesuma <sup>2\*</sup>

<sup>1</sup> Department of Informatics Engineering  
Institut Teknologi Sumatera  
Lampung Selatan, Indonesia

<sup>2</sup> Research and Inovation Center of Artificial Intelligence  
Institut Teknologi Sumatera  
Lampung Selatan, Indonesia  
\*rahman.indra@if.itera.ac.id

**Abstract** - A new coronavirus variant known as n-Cov has emerged with a fast transmission rate. The World Health Organization (WHO) has declared the related disease or COVID-19 as a global pandemic that requires special handling. Many parties have shown efforts to reduce virus transmission by implementing health protocols and adapting a new normal lifestyle. Implementation of the health protocol creates new problems, especially in the health check at the main entrance. The officers in charge of measuring body temperature are at risk of getting infected by COVID. Such a measurement is prone to errors. This study proposed a solution to build an automatic gate system that worked based on the new normal health protocol. The system utilizes the MLX90614 contactless temperature sensor to probe body temperature. It applies deep learning implementing the Convolutional Neural Network (CNN) algorithm with the MobileNetV2 architecture as a determinant of the conditions of wearing face masks. The system is equipped with an IoT-based remote controller to control the gate. Experimental results prove that the system works well. Temperature measurement takes a response time of 20 seconds for each user with 99% accuracy for the sensor and masks classification model.

**Keywords:** COVID-19, face mask classification, convolutional neural network, MobileNetV2, MLX90614

*Article info: submitted July 17, 2021, revised August 21, 2021, accepted December 4, 2021.*

## 1. Introduction

Coronavirus Disease 2019 (COVID-19) is an infectious disease caused by a novel corona virus variant discovered in 2019. COVID-19 causes diseases in humans such as the flu, with a very fast transmission through physical contact interactions or with other media within a close range [1]. The World Health Organization (WHO) has determined that the spread of COVID-19 is a global pandemic that requires special handling to suppress its transmission. The COVID-19 virus has spread in several regions of Indonesia with the number of cases reaching 45892 cases, and the ratio of patients recovering 40% and dying by 5% [2]. The Indonesian government to handle COVID-19 cases has implemented several policies such as Large-Scale Social Restrictions (*Pembatasan Sosial Berskala Besar - PSBB*), Enforcement of Community Activity Restrictions (*Pemberlakuan Pembatasan Kegiatan Masyarakat - PPKM*), and adaptation for a new normal

lifestyle. The implementation of the new normal lifestyle is carried out to break the chain of the spread of the COVID-19 virus in line with economic and community activities that must remain active during this pandemic. The ITERA's environment also adapts the implementation of the new normal protocol to deal with the activities of the academic community that cannot be done online. ITERA through the New Life Adaptation Development Agency (*Badan Pembina Adaptasi Kehidupan Baru - BPAKB*) ITERA Campus has compiled a new normal protocol with the standards set by the Ministry of Education and Culture (Kemdikbud). The new normal protocol implemented consists of several main points such as the body temperature and the use of mask checks which are carried out at the ITERA main gate (one gate system) [3]. The gate is used as an intermediary and controls the incoming and outgoing vehicles [4], as well as the academic community. Every ITERA academic community who wants to enter the ITERA's environment need to go

through body temperature and mask-wearing check by ITERA officers. The checking is carried out by officers at the main gate caused various problems, such as a high risk of transmitting the virus to officers and the emergence of queues if the number of visitors was not proportional to the number of existing officers.

Of the various problems that arise due to body temperature and masks wearing checks by officers, it is necessary to have a system that can replace the role of officers and can be integrated automatically through the campus gate. Research related to the development of an intelligent system connected to the gate has been carried out previously, in which the gate can be opened through facial recognition using the HAAR Cascade algorithm [5]. Another similar development has also been carried out previously to deal with influenza by measuring normal body temperature without contact using MLX90614 in monitoring attendance at schools [6]. In addition, research related to the face mask detection model can produce 2-3% better accuracy on public datasets, through the application of the MobileNetV2 architecture [7].

Based on the explanation of previous studies, in this study, an automatic gate system was developed using the MLX90614 sensor to measure non-contact body temperature, which has good accuracy [8], and the MobileNetV2 architecture to categorize the use of masks by considering the low model size, as well as quite capable computing.

## 2. Method

The research methodology used in this study consists of several stages as shown in Figure 1.

### a. Problem Identification

The initial stage carried out in this research is the identification of the problem, (especially to reduce interaction of officers) at the ITERA campus gate with an automatic gate system that can check the temperature and detect the use of masks.

### b. Literature Studies and Literature Studies

The literature study was conducted to obtain relevant references related to the research. The references used are sourced from various domestic and international journals regarding the methods and objects studied to avoid repeated research. The literature study was conducted to obtain the basic and conceptual theory that underlies this research. Some of the basic theories in this research are:

#### a) Computer Vision

The technology which is development part of artificial intelligence, has an important role in processing images and videos by an electronic machine. The application of computer vision in various systems has the main goal of being able to imitate the ability of the human eye so that machines can recognize objects through images or videos provided. In addition to imitating the human ability to see, several

important things need to be considered in the field of Computer Vision, specifically systems with low power and resources but still running powerful [9]. Currently, the use of computer vision has expanded and is widely used in the development of intelligent devices capable of detecting and recognizing various objects such as faces (Face Recognition) and other objects [10].

#### b) Body temperature

Temperature is the level of hot or cold in a situation that can be measured using an instrument called a thermometer. Body temperature is generally divided into two types, namely skin/surface temperature and core temperature. The temperature on the inside of the body is called core temperature, which has a constant level in the non-febrile condition, which is around  $\pm 1^{\circ}\text{F}$  ( $\pm 0.6^{\circ}\text{C}$ ). Furthermore, skin temperature is the level of heat found on the surface of the skin, with fluctuating values because it can be influenced by the state of the ambient temperature. If excessive heat is formed in the body, then the skin temperature will rise, and vice versa. In general, the normal human body temperature ranges from  $97^{\circ}\text{F}$  ( $36^{\circ}\text{C}$ ) –  $99.5^{\circ}\text{F}$  ( $37.5^{\circ}\text{C}$ ). Body temperature can also be affected by the activities carried out, for example, the temperature will rise when doing sports or other physical activities. In addition, extreme environmental temperatures can also affect body temperature [11].

#### c) MobileNetV2

MobileNetV2 is a neural network architecture that was developed to overcome the problem of limited mobile device resources with several improvements to support efficiency [12]. MobileNetV2 development is done by reducing the number of operations and memory usage while maintaining the accuracy of the model. MobileNetV2 can be implemented efficiently on a variety of standard operations and modern frameworks. In addition, MobileNetV2 is very suitable for the implementation of embedded systems because it allows memory reduction during the use [13].

#### d) OpenCV

OpenCV is an open-source library that can be used for computer vision applications. Initially, this library was developed to accelerate the performance and capabilities of computing machines in the fields of computer vision and artificial intelligence by providing robust infrastructure. OpenCV, which was developed using the C and C++ programming languages, can operate well on several operating systems such as Linux, Windows, and macOS [14]. In addition, development is also carried out in several other programming languages and interfaces such as Python, Java, Matlab, and others. OpenCV provides an infrastructure that makes it easier for users to apply computer vision concepts (e.g., image

processing). Currently, OpenCV contains more than 500 functions that can be applied to various problem areas such as health, security, and others [15].

e) Tensorflow

Tensorflow is a widely used library for deploying and developing deep neural networks which is developed by the Google Brain team. Tensorflow is an open source machine learning system that can operate in heterogeneous environments. Tensorflow uses dataflow to represent computations, shared states, and other operations. It maps each node of the dataflow graph on multiple machines within a cluster, and on each machine across multiple computing devices such as multicore CPUs, general-purpose GPUs, and custom-designed ASICs known as Tensor Processing Units (TPUs). This architecture gives developers flexibility in designing parameters and managing shared states on the system [16].

c. System planning

The design of the system aims to compile component and system requirements. In addition, at this stage, the circuit and workflow of the system are also made. In the development of this automatic gate system, there are several main requirements needed to build a smart device that is integrated with the gate. The main requirements needed for this automatic gate system can be observed in Table 1.

Table 1. Hardware requirements

No	Name	Function
1	Raspberry Pi 4 Model B	A microcontroller that has a role as a data processing center, which can work in three stages, namely Input, Process, and Output.
2	MLX90614	A non-contact temperature sensor that functions to measure body temperature by utilizing a passive infrared mechanism.
3	Raspberry Pi Camera	Retrieve image data in image or video file types to be used as input data.
4	Raspberry Pi Display / Monitor	Graphical data viewer generated from microcontroller processing.
5	Relay	A switch that serves as a controller and a large electric current with a small electric current control.
6	HC-SR04	The ultrasonic sensor serves to measure the distance of the object relative to the sensor.
7	MicroSD	A tool that is used as storage space for the operating system and other files on the Raspberry Pi.

The automatic gate system was developed using the Raspberry Pi4 microcontroller as a center for processing data obtained from sensors and cameras. In addition, actuators in the form of relays are also controlled by a

microcontroller based on data validation that has been obtained previously. The interaction between each component can be represented through a schematic diagram which can be seen in Figure 2.

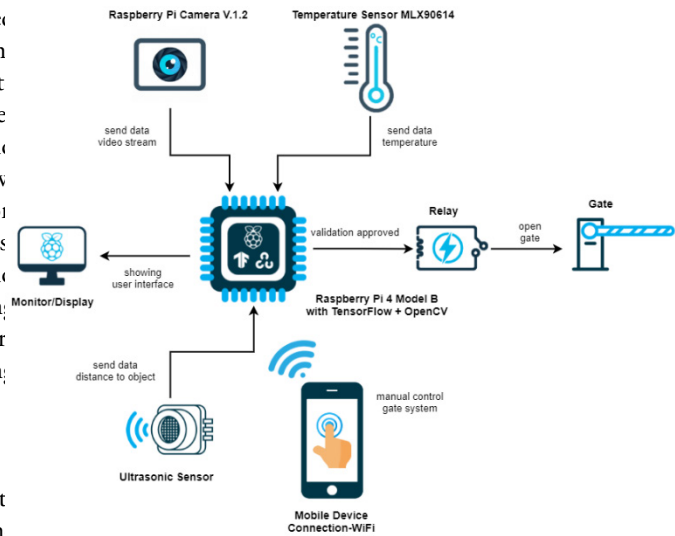


Figure 2. System schematic diagram

The workflow of the automatic gate system is designed according to the defined new normal protocol. To open the gate, two methods can be done, the system must check the user's compliance with the new normal protocol through the body temperature checks and the use of masks detection. The workflow of the gate system is represented through a block diagram which can be observed in Figure 3.

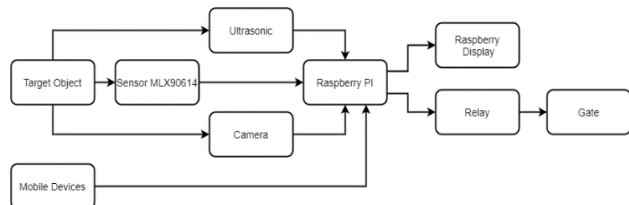


Figure 3. System block diagram

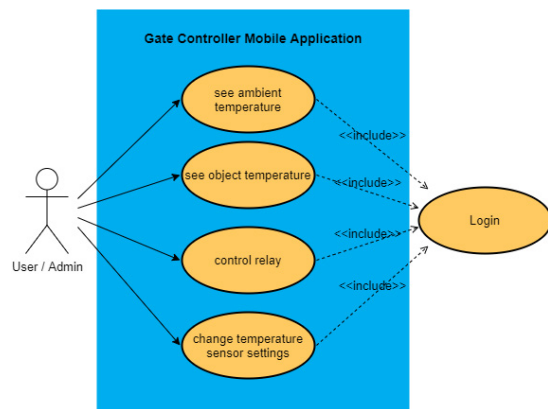


Figure 4. Mobile applications use case diagrams

In addition, the gateway system can be controlled remotely by utilizing an Internet connection and a mobile

device. The software for controlling the gate system generally consists of several functions which can be seen through the use case diagram in Figure 4.

#### d. Dataset Preparation

The dataset is a collection of data that is used to develop an existing face mask classification model. In this study, there are three sources of the dataset used, which are then used to form the most optimal model to be implemented in the system. The source of the dataset is obtained through the Kaggle data scientist platform. Dataset specifications can be seen in Table 2.

The dataset used contains a collection of images with the extension \*.jpg or \*.png. The image is divided into two labels/classes, 'with\_mask' and 'without\_mask' labels which are arranged into two different folders with the same composition so that the dataset is in a balanced state.

Table 2. Dataset specifications

Dataset	Number of Class	File Type	Data with mask	Data Without Mask	Number of data	Source
1	2	*.jpg	3725	3828	7553	[17]
2	2	*.png	220	220	440	[18]
3	2	*.jpg	208	131	339	[19]

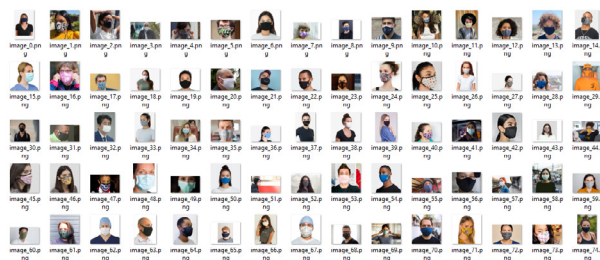


Figure 5 Example of an image on a dataset

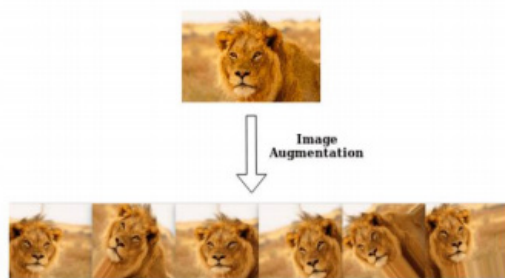


Figure 6. Illustration of image augmentation

#### e. Data Pre-Processing

This stage is useful for processing and manipulating datasets before they are included in the model or training process. Data pre-processing includes label transformation, data sharing, and data augmentation. The label transformation is done to change the data label ('with\_mask', 'without\_mask') into a NumPy array [1,0]

so that it can be used in the model creation. Data sharing is done to divide the entire dataset into two parts, 75% of training data and 25% of testing data. Data Augmentation is a step taken to enrich data by manipulating a set of images. Augmentation is useful for assisting the model in generalizing features and increasing accuracy, especially during training [20]. In the field of computer vision, augmented data can be generated by performing geometric transformations on the original image, so that another image is obtained. The geometry transformation process carried out generally includes several processes such as translations, rotations, scaling, shearing, zoom, and flip. An illustration of image augmentation can be seen in Figure 5.

#### f. Model Development

The model creation stage is done by designing the model architecture and training the model based on the dataset that has been prepared before the data pre-processing stage. The face mask classification model created uses the MobileNetV2 architecture as the base model, while the fully connected layer is arranged according to the dataset and needs. In general, MobileNetV2 still has the same architectural form as MobileNetV1, but this architecture has two additions facility, namely the existence of linear bottlenecks and shortcut connections. MobileNetV2 consists of a convolution layer using 32 filters, followed by 19 residual bottleneck layers. In addition, MobileNetV2 uses ReLU6 for non-linearity transformation [13]. A simple visualization of the working architecture of MobileNetV2 can be seen in Figure 6.

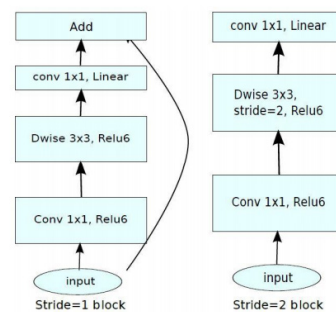


Figure 7. MobileNetV2 Architecture

MobileNetV2 architecture generally consists of layers that are arranged in such a way. MobileNetV2 has layers with specifications as shown in Table 3.

Table 3. The basic architecture of MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2

Input	Operator	t	c	n	s
14 <sup>2</sup> x 64	bottleneck	6	96	3	1
14 <sup>2</sup> x 96	bottleneck	6	160	3	2
7 <sup>2</sup> x 160	bottleneck	6	320	1	1
7 <sup>2</sup> x 320	conv2d 1x1	-	1280	1	1
7 <sup>2</sup> x 1280	avgpool 7x7	-	-	1	-
1 x 1 x 1280	conv2d 1x1	-	k	-	-

In this study, the default fully connected layer on MobileNetV2 was not used and was redesigned to match the face mask classification and dataset used. The fully connected layer used can be seen in Table 4.

The fully connected layer consists of 5 layers connected to the MobileNetV2 base model, in which there is a 2D average pooling layer with a pooling size of 7 x 7, followed by a flatten layer, relu with 128 dense, 0.5 dropout, and softmax activation function with size 2.

**Table 4. Customized fully connected layer components**

Input	Operator	t	c	n	s
7 <sup>2</sup> x 1280	avgpool2D 7x7	-	-	1	-
1 x 1 x 1280	flatten	-	-	1	-
1280	relu	-	-	1	-
128	dropout	-	-	1	-
128	softmax	-	2	1	-

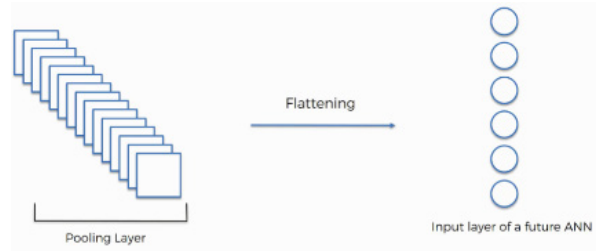
Layer pooling is generally used to reduce the size of the matrix. Layer pooling is a filter that has a certain size and will alternately shift in the feature area. A simple application of the processing applied to the pooling layer can be seen in Figure 7.

4	3	1	5
1	3	4	8
4	5	4	3
6	5	9	4

$$\text{Avg}([4, 3, 1, 3]) = 2.75$$

**Figure 8. Illustration of average pooling process**

Average pooling is one type of layer pooling that calculates each block of values through the average function [21][22]. The next stage is flattening, which is a layer with the function of “flattening” the matrix into a form that is easily spreadable into neural networks. Figure 8 provides a visualization of the simple process of flattening, with a focus on allowing data in the form of a matrix to be entered into the input layer of the neural network.



**Figure 9. Flattening process illustration**

Furthermore, on the neural network layer components, Relu is used as an activation function to convert each linear component into non-linear ones. Formula (1) can be used to calculate the relu activation function [23][24].

$$f_{Relu}(h_i, x) = \max(0, h_i, x) \tag{1}$$

Relu in general only makes a limit on the number zero which means if the value of x is 0 then x = 0 and if x > 0 then the value of x remains.

At the end of processing the neural network layer, the Softmax function is used which is an activation function in the output layer. Softmax can be used to classify multiple classes [25]. This function generally calculates each probability of some cases, which in this case calculates the probability of the label to be classified by the model. The output of this function is a range of probability values from zero to one. Softmax activation function can be calculated by equation (2) [26][27].

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \tag{2}$$

Information:

- $\vec{z}$  = vector input in the softmax function, consisting of (z0...zK)
- $z_i$  = the input vector element is a real number
- $e^{z_i}$  = fungsi eksponensial yang diterapkan pada setiap  $z_i$
- $\sum_{j=1}^K e^{z_j}$  = normalization term to ensure that the output is a valid probability distribution
- K = number of classes

After designing the model, the next step is training process, where there is a hyperparameter configuration that is used to optimize model training to form a good model [28]. The hyperparameter consists of batch size, learning rate, and the number of epochs used. The hyperparameters used in the mask classification model training process can be seen in Table 5.

**Table 5. Component hyperparameter used**

Dataset to-	Epochs	Learning Rate	Batch Size
1	10	0.0001	64
2	20	0.0001	32
3	20	0.0001	32

The training process that has been carried out will produce a model which will later be implemented in the system. In addition, the resulting model can be seen from the evaluation results through the Classification Report which consists of Precision, Recall, and F1-Score. Precision (Positive Predictive Value) [29][28] is an evaluation result that describes how accurate the requested data is with the classification results or predictions given. Precision can be calculated by applying equation (3).

$$\text{precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}} \quad (3)$$

Recall/Sensitivity (True Positive Rate) is an evaluation result that describes the success of a model in finding data in a certain class/label. Equation (4) can be applied to obtain the recall value.

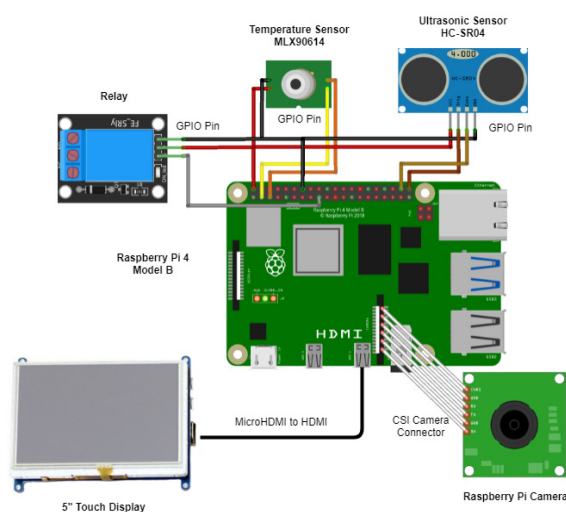
$$\text{recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}} \quad (4)$$

In the end, the classification quality calculation ends with a comparison of the F1-Score values, which compares the average value of precision with weighted recall. F1-Score, through equation (5), can describe the performance if the number of FP and FN is not balanced (non-symmetric)

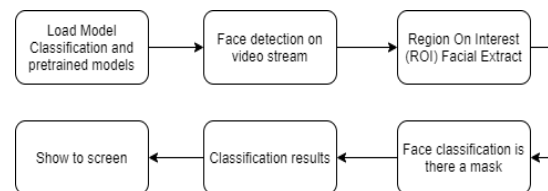
$$F1 \text{ Score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} = \frac{2TP}{2TP + FP + FN} \quad (5)$$

#### g. Implementation

Implementation is the stage carried out to assemble the entire system components. The hardware components are then assembled so that they can be integrated into the flow as shown in Figure 9.

**Figure 10. Hardware circuit diagram**

Furthermore, after the model training process was carried out, the research produced a model with the extension \*.model with the h5 storage format. The model is then stored in the internal storage space on the microcontroller. Based on the experiments in the previous training, the model that has the most optimal performance is the model using the 2nd dataset. This model will be used by the microcontroller during the mask classification process for users. The working process of the system in classifying masks can be seen in the following picture

**Figure 11. The working process of classifying masks on the system**

Several processes are carried out to classify the use of masks through video data captured by the camera. The model file that has been created in the training process along with the pre-trained model provided by OpenCV is entered into the system, which then the system will get a data stream in the form of image frames through the camera. From the image frame obtained, the model will extract the Region of Interest (ROI) to get user face data to be classified. This ROI data will later be included in the mask classification model. The results of the detection of the user's facial area will be displayed in the form of a boundary box with a percentage of classification accuracy. The results of the classification with the model can be seen in Figure 11.

### 3. Result

Automated gate system development results in embedded devices that are fully integrated between every hardware and software component. In addition, the gate system has a user interface consisting of several features such as home and sensor calibration which can be observed in Figure 12.

**Figure 12. Example of model classification results**

In addition, there is the development of mobile applications that are used to control certain conditions. The mobile application was developed using a mobile multiplatform framework, Flutter [30]. The selection of Flutter is considered more effective and efficient for developing mobile applications on several platforms at once. Flutter uses the Dart programming language as the main programming language. The development of a mobile application on this gate system has several features such as calibration and direct gate control through relay activation, as can be seen in Figure 13. The results of the gate system development that have been carried out are then evaluated through several test components, namely sensor testing, classification model testing, and system performance testing. In addition, there are also model training results that are considered to obtain the best model to be implemented in the gate system.

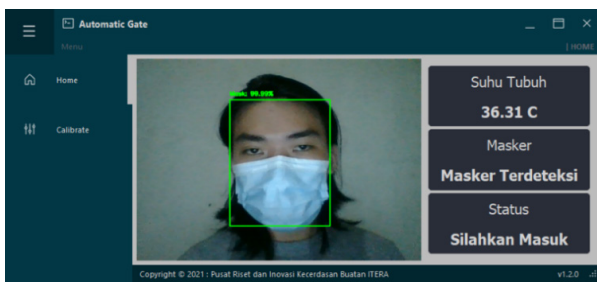


Figure 13. System application user interface

Sensor testing used in this study consisted of the MLX90614 temperature sensor and the HC-SR04 ultrasonic sensor. Sensor testing is done by comparing the results of temperature measurements with standard measuring instruments, in this case, the temperature sensor will be compared with a thermogenic and ultrasonic sensor compared with a ruler. The test experiment was carried out 10 times and the sensor accuracy results were shown in Table 6.

Based on the test data of the MLX90614 temperature sensor and the HC-SR04 ultrasonic sensor, the sensor accuracy rate is very good, reaching 99%. The use of temperature sensors MLX90614 and HC-SR04 can also generally work well with minimum standards equivalent to standard measuring instruments.

Table 6. Sensor test results

Test	MLX90614	HC-SR04
1	98.87%	98.90%
2	99.77%	99.82%
3	99.74%	99.50%
4	99.57%	99.85%
5	99.95%	100.00%
6	98.54%	97.53%
7	97.72%	100.00%
8	99.46%	99.94%
9	99.40%	99.50%
10	98.54%	99.53%
$\bar{x}$	99.16%	99.45%

Testing the mask classification model is done by looking at the results of the confusion matrix obtained after training the classification model. The confusion matrix obtained in the training process can be observed in Figure 15.

In the confusion matrix shown in Figure 15, a classification report can be calculated during model training and testing the model's ability to classify different types of masks. The results of the classification report for the mask classification model showed the highest results in the first and second datasets. The results of performance measurement on the classification can be observed in Table 7.

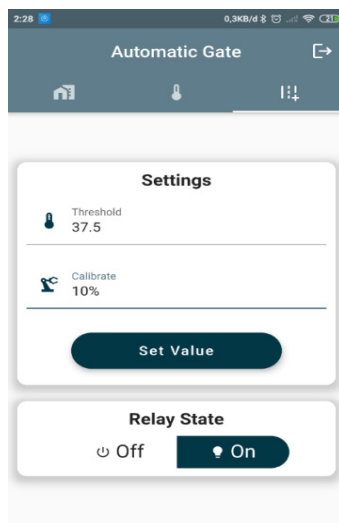


Figure 14. Mobile application user interface

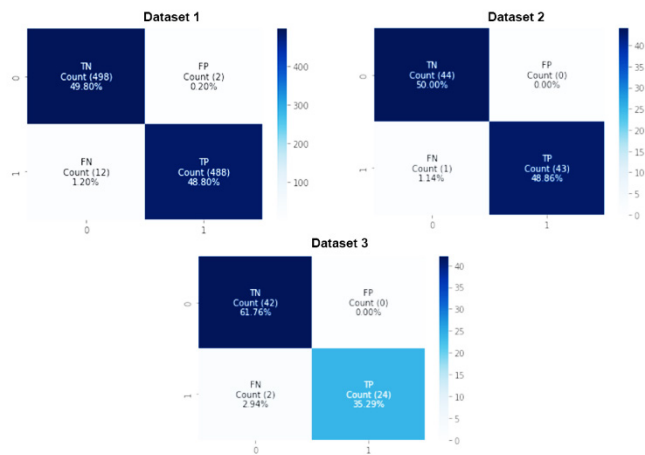


Figure 15. Confusion matrix training

**Table 7. Classification Report model training**

Dataset to-	Class	Precision	Recall	F1-Score
1	with_mask	0.98	1.00	0.99
	without_mask	1.00	0.98	0.99
2	with_mask	0.98	1.00	0.99
	without_mask	1.00	0.98	0.99
3	with_mask	0.95	1.00	0.99
	without_mask	1.00	0.92	0.99

To choose the best face mask classification model, it is necessary to test the success of the model in classifying the variance of masks. The percentage of success of the classification model shows that the model with the second dataset excels with a value of 80% compared to the first dataset by 40% and the third dataset by 20%. Thus, the model that will be used is a model that is trained using the second dataset. In addition, the selected model will also be tested on various types of masks circulating in the community. The test results for the variation of masks can be seen in Table 8.

Tests were carried out to see the model's performance in recognizing mask variations, carried out on 15 variations of masks, and obtained an average accuracy of 99.63%. This shows that in general, the classification model can recognize masks that are generally used by the public

Evaluation of response time or system operating time also needs to be considered to find out how fast the system works to take into account the possibility of queues. Evaluation of response time is done by testing 10 times. The results of the response time test can be observed in Table 9.

Based on the results of the response time test, the classification of masks has the largest response time, this is because the classification process requires a fairly high resource. The total time in general for the system to work is 19.8 seconds, which is quite fast compared to manual testing by officers.

**Table 8. The results of model testing on variations of masks**

No	Mask Type	Motive	Accuracy	Status
1	Medical Mask	Blue color	99.98%	Succeed
2	Medical Mask	Grey color	99.81%	Succeed
3	Medical Mask	Green color	99.83%	Succeed
4	N95 Mask	White color	99.89%	Succeed
5	Cloth Mask	Dark green color	99.75%	Succeed
6	Cloth Mask	Light green color	99.96%	Succeed
7	Cloth Mask	Brown color	98.90%	Succeed

No	Mask Type	Motive	Accuracy	Status
8	Cloth Mask	Grey color	99.91%	Succeed
9	Cloth Mask	Black color	99.31%	Succeed
10	Cloth Mask	Purple color	99,90%	Succeed
11	Cloth Mask	Landscape painting pictures	99.70%	Succeed
12	Cloth Mask	Bear pictures	98.47%	Succeed
13	Cloth Mask	Batik pictures	99.99%	Succeed
14	Cloth Mask	Smile pictures	90,78%	Succeed
15	Cloth Mask	Smile pictures	99,55%	Succeed
16	No Mask	-	99.86%	Succeed
Average ( $\bar{x}$ )			99.63%	Succeed

**Table 9. Response time test results**

Feature	Average time
Temperature	4 second
Face mask	11.4 second
Object	2.7 second
Gate Activation	5 second
<b>Total</b>	<b>19.8 second</b>

#### 4. Conclusion

Research results suggest that the automatic gate entrance system works well as expected. It checks body temperature accurately and quickly. It enforces the compliance of wearing masks. The gate system that uses the MLX90614 sensor is accurate enough to measure the body temperature without contact. The classification model using MobileNetV2 can classify the wearing of masks with an accuracy of 99.63%. In addition, the average working time of the system is 19.8 seconds per person which is fast enough to overcome the possibility of queues at the gate.

#### Acknowledgment

This research was funded by the Institute for Research, Community Service and Education Quality Assurance, Institut Teknologi Sumatera (through the 2020 ITERA Research Grant Program), contract number B/418d/IT9. C1/PT.01.03/2020. The author and the team would like to thank all colleagues at Institut Teknologi Sumatera, especially the Department of Informatics and the Artificial Intelligence Research and Innovation Center, for their scientific assistance and cooperation in this research.



## Reference

- [1] R. Q. Cron and W. W. Chatham, "The Rheumatologist's Role in COVID-19," *J. Rheumatol.*, vol. 47, no. 5, pp. 639–642, 2020, doi: 10.3899/jrheum.200334.
- [2] Kementrian Kesehatan RI, "Info Infeksi Emerging Kementrian Kesehatan RI," *infeksiemerging.kemkes.go.id*, 2020. <https://infeksiemerging.kemkes.go.id/> (accessed Jun. 21, 2020).
- [3] Institut Teknologi Sumatera, "ITERA Mulai Terapkan New Normal Kehidupan Kampus," *itera.ac.id*, 2020. <https://www.itera.ac.id/itera-mulai-terapkan-new-normal-kehidupan-kampus/> (accessed Jun. 21, 2020).
- [4] A. S. Oluwole, T. Adefarati, K. Olusuyi, A. Babarinde, and E. Hilary, "Design of Automatic Gate Control Using Infrared Remote With Password Protected," *Int. J. Res. Dev. Technol.*, vol. 2, no. 5, pp. 6–12, 2014.
- [5] N. Dileep and S. Shanthi, "Automatic Gate using Face Recognition Technique using HAAR Cascade Algorithm," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 3, pp. 1302–1305, 2020, doi: 10.35940/ijeat.c5195.029320.
- [6] H. F. Tang and K. Hung, "Design of a non-contact body temperature measurement system for smart campus," *2016 IEEE Int. Conf. Consum. Electron. ICCE-China 2016*, pp. 0–3, 2017, doi: 10.1109/ICCE-China.2016.7849773.
- [7] M. Jiang, X. Fan, and H. Yan, "RetinaMask: A Face Mask detector," 2020, [Online]. Available: <http://arxiv.org/abs/2005.03950>.
- [8] N. H. Wijaya, Z. Oktaviahandani, K. Kunal, E. T. Helmy, and P. T. Nguyen, "Tympani thermometer design using passive infrared sensor," *J. Robot. Control*, vol. 1, no. 1, pp. 27–30, 2020, doi: 10.18196/jrc.1106.
- [9] S. Alyamkin *et al.*, "Low-Power Computer Vision: Status, Challenges, and Opportunities," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 9, no. 2, pp. 411–421, 2019, doi: 10.1109/JETCAS.2019.2911899.
- [10] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, Analysis, and Machine Vision-Cengage Learning*. 2014.
- [11] J. E. Hall and A. C. Guyton, *Guyton and Hall Textbook of Medical Physiology*, 12th ed. Jackson, 2011.
- [12] S. Bi, Y. Zhang, M. Dong, and H. Min, "An embedded inference framework for convolutional neural network applications," *IEEE Access*, vol. 7, pp. 171084–171094, 2019, doi: 10.1109/ACCESS.2019.2956080.
- [13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- [14] J. Sigut, M. Castro, R. Arnay, and M. Sigut, "OpenCV Basics: A Mobile Application to Support the Teaching of Computer Vision Concepts," *IEEE Trans. Educ.*, vol. 63, no. 4, pp. 328–335, 2020, doi: 10.1109/TE.2020.2993013.
- [15] A. Kaehler and G. Bradski, *OpenCV 3*. 2016.
- [16] M. Abadi *et al.*, "TensorFlow: A System for Large-Scale Machine Learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, Nov. 2016, pp. 265–283, [Online]. Available: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>.
- [17] O. Gurav, "Face Mask Detection Dataset," 2020. <https://www.kaggle.com/omkargurav/face-mask-dataset> (accessed Jun. 26, 2020).
- [18] D. Makwana, "Face Mask Classification," *www.kaggle.com*, 2020. <https://www.kaggle.com/dhruvmak/face-mask-detection> (accessed Jun. 21, 2020).
- [19] Sumansid, "FaceMask Dataset," 2020. <https://www.kaggle.com/sumansid/facemask-dataset> (accessed Jun. 21, 2020).
- [20] E. Ayan and H. M. Ünver, "Data augmentation importance for classification of skin lesions via deep learning," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 2018, pp. 1–4, doi: 10.1109/EBBT.2018.8391469.
- [21] A. Kasagi, T. Tabaru, and H. Tamura, "Fast algorithm using summed area tables with unified layer performing convolution and average pooling," in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017, pp. 1–6, doi: 10.1109/MLSP.2017.8168154.
- [22] T. Y. Hsiao, Y. C. Chang, H. H. Chou, and C. Te Chiu, "Filter-based deep-compression with global average pooling for convolutional networks," *J. Syst. Archit.*, vol. 95, no. June 2018, pp. 9–18, 2019, doi: 10.1016/j.sysarc.2019.02.008.
- [23] S. Dittmer, E. J. King, and P. Maass, "Singular Values for ReLU Layers," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 9, pp. 3594–3605, 2020, doi: 10.1109/TNNLS.2019.2945113.
- [24] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2017-May, pp. 2684–2691, 2017, doi: 10.1109/IJCNN.2017.7966185.

- [25] Y. Zhang, E. Zhang, and W. Chen, "Deep neural network for halftone image classification based on sparse auto-encoder," *Eng. Appl. Artif. Intell.*, vol. 50, pp. 245–255, 2016, doi: 10.1016/j.engappai.2016.01.032.
- [26] M. Wang, S. Lu, D. Zhu, J. Lin, and Z. Wang, "A High-Speed and Low-Complexity Architecture for Softmax Function in Deep Learning," *2018 IEEE Asia Pacific Conf. Circuits Syst. APCCAS 2018*, pp. 223–226, 2019, doi: 10.1109/APCCAS.2018.8605654.
- [27] D. Zhu, S. Lu, M. Wang, J. Lin, and Z. Wang, "Efficient Precision-Adjustable Architecture for Softmax Function in Deep Learning," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 67, no. 12, pp. 3382–3386, 2020, doi: 10.1109/TCSII.2020.3002564.
- [28] E. Kristiani, C. T. Yang, and C. Y. Huang, "ISEC: An Optimized Deep Learning Model for Image Classification on Edge Computing," *IEEE Access*, vol. 8, pp. 27267–27276, 2020, doi: 10.1109/ACCESS.2020.2971566.
- [29] J. Sepúlveda and S. A. Velastín, "F1 score assesment of Gaussian mixture background subtraction algorithms using the MuHAVi dataset," *IET Semin. Dig.*, vol. 2015, no. 5, pp. 1–6, 2015, doi: 10.1049/ic.2015.0106.
- [30] L. Dagne, "Flutter for Cross-Platform App and SDK Development," *Metrop. Univ. Appl. Sci.*, no. May, 2019, [Online]. Available: [https://www.theseus.fi/bitstream/handle/10024/172866/Lukas\\_Dagne\\_Thesis.pdf?sequence=2&isAllowed=y](https://www.theseus.fi/bitstream/handle/10024/172866/Lukas_Dagne_Thesis.pdf?sequence=2&isAllowed=y).