SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

IoT-Based and Teachable Machine Platform for Covid-19 Prevention and Control

Siti Murni Hanum Md Ariff[1], Norlela Ishak[1,*], Mazidah Tajjudin[1]

[1] School of Electrical Engineering, College of Engineering, Universiti Teknologi Mara, 40450 Shah Alam, Selangor, Malaysia

ABSTRACT

This paper aims to develop a solution for premises requiring compliance with Standard Operating Procedures (SOP) for COVID-19 safety. To address potential outbreaks and human errors in manual entrance supervision, a face mask detection model is developed using Teachable Machine in which a dataset of 4092 images is used to train the model. This system utilizes an ultrasonic sensor (HC-SR04) to gauge the distance and a non-contact infrared temperature sensor (MLX90614) for body temperature screening. Data is displayed on an I2C LCD and published on an IoT platform (Blynk's dashboard) for remote monitoring. A healthy body temperature (36.1℃ to 38℃) triggers the face mask detection model. If the person wears a mask, a green LED lights up, and the barrier opens for entry; otherwise, a red LED indicates no mask, and entry is denied. Notification is also prompted upon the detection of an unhealthy body temperature. The system achieves the following; 98.43% accuracy for body temperature scanning, 98% for face mask detection, and 97.5% success rate during the implementation of the face mask detection system.

Keywords:

Face mask detection; Teachable machine; Infrared sensor; MLX90614; Blynk; ESP32-CAM; Ultrasonic sensor; HC-SR04

1. Introduction

1.1 Research Background

As we enter the year 2023, the global fight against the COVID-19 pandemic continues to show signs of progress, with the virus's relentless spread finally beginning to slow down. The tireless efforts of healthcare professionals, widespread vaccination campaigns, and stringent public health measures have all contributed to this positive trend. However, despite the encouraging progress, the spectre of new variants looms on the horizon, reminding us that the battle is far from over as health authorities and scientists are vigilant about the emergence of new variants, as evident in [1] where new cases caused by the new Covid-19 XBB.1.16 or the Arcturus variant has been reported as of recently.

According to [2], while Malaysia is currently in transition towards Covid-19 Endemic phase, it is still advised to wear face masks in crowded and confined places [3] especially in health facilities and

* *Corresponding author.*
E-mail address: norlelaishak@uitm.edu.my

https://doi.org/10.37934/araset.56.1.163182

public transportation where transmission rates are higher. The virus's ability to mutate and evolve remains a significant concern, as new variants have the potential to be more transmissible, cause more severe disease, or even evade existing immunity from vaccines and prior infections. The threat of future outbreaks lingers, underscoring the importance of continued surveillance, research, and adaptation in our approach to public health.

Notably, during the pandemic's peak, public premises and establishments implemented strict entry protocols to safeguard public health such as the Standard Operating Procedure (SOP). Generally, manual labour played a pivotal role in ensuring compliance with safety measures. Dedicated staff supervised entry points, meticulously checking that everyone wore masks, maintained physical distancing, and screened each individual's body temperature. These labour-intensive procedures were crucial in minimizing the risk of transmission within enclosed spaces and contributed to the broader efforts to mitigate the spread of the virus. However, the inherent flaws with manual supervision have called for more integrated and automated solutions and systems for entry screening and authorization.

Our proposed automated system addresses a critical research gap in the effective enforcement and monitoring of health safety measures, specifically mask-wearing compliance and body temperature screening. By integrating real-time updates and data visualization on Blynk's web and mobile dashboards, our system enhances visibility and accessibility for both premise owners and authorities. The inclusion of a notification feature on mobile phones upon detecting unhealthy body temperatures ensures swift response to potential health risks. The continuous data log on Google Sheets facilitates long-term trend analysis, aiding in the identification of patterns and anomalies. Moreover, our system allows for remote monitoring, enabling authorities to oversee multiple locations simultaneously. One of the key contributions of our study lies in the use of low-cost and easily accessible components, making it feasible for widespread implementation. Additionally, the integration of a readily available machine learning platform enhances the system's adaptability and effectiveness in accurately detecting non-compliance and abnormal body temperatures. Overall, our research bridges the existing gap by providing a comprehensive, cost-effective, and technologically advanced solution for public health safety in various settings.

1.2 Literature Review

1.2.1 Face mask detection model

Recent research and development on COVID-19 prevention systems saw the implementation of face mask detection model such as in [4] where a deep learning algorithm via image classification method Mobilenet-v2 was applied in OpenCV while utilizing machine learning packages and libraries from TensorFlow and Keras to train the face mask detection model. A dataset of 3830 images were used where half of the sample contains images of faces with masks on while the remaining half contains bare faces. Before training the model, the loaded dataset would first undergo pre-processing through image resizing, array conversion and labelling. Then, 80% of that data is used for training while the remaining 20% is used for testing the model. Upon the completion of 20 epochs (the number of iterations the model will work through the dataset), the model is found to achieve 99% accuracy. Similarly, in [5] the same algorithm using Mobilenet-v2 was implemented to achieve 96.85% accuracy.

Das *et al.,* [6] proposed another approach for recognizing hybrid facial masks through the application of straightforward yet effective machine learning programs. The significance of their work lies in developing a surveillance task performer capable of detecting face masks even in dynamic scenarios. To achieve this, they employed a Sequential CNN (Convolutional model, strategically fine-

tuning the model's parameters to ensure accurate mask detection while mitigating overfitting issues. Tools such as TensorFlow, Keras, OpenCV and Sci-Kit Learn were used to facilitate pre-processing of image samples by resizing and colour conversion such as binarization from RGB (Red Green Blue) images. By leveraging these tools, they are able to process raw data into a suitable format for training and validating their facial mask recognition system which ultimately attained up to 95.77% and 94.58% accuracy in two different datasets.

Various other approaches and work for developing a face mask detection model are proposed in [7,8] such as YOLO v4 algorithm and Convolutional Neural Network architecture, which resulted in 99.5% accuracy.

However, a simpler approach for developing a face mask detection system is shown in [9,10] where the author trains a face mask detection model using a web-based GUI machine learning tool known as Teachable Machine, where accuracies comparable to that of other machine learning techniques can be attained, as evident in [9] where the face mask detection model achieves 100% and 92.5% accuracy for its mask wearing and non-mask wearing classes respectively. In [10], a dataset of 2 classes (faces with mask on and faces without mask) is used to train the model which is then exported as a Tensorflow.js and hosted on a web server that is accessed by the ESP32-CAM for running the model. A similar approach for face mask detection using Teachable Machine is also demonstrated in [11,12]. Exporting the trained model generates a model path that can be specified in the programming or the web server that hosts the video stream to access the model on Teachable Machine. Carney *et al.,* [13] highlights the following regarding Teachable Machine's contribution; a user-friendly interface for developing machine learning classification models without extensive machine learning experience or coding expertise, while providing an easier prototyping tool than other existing workflows.

1.2.2 Body temperature detection

Though researches thus far have suggested a variety of methods as a means of detecting body temperature, several recent studies have implemented the use of non-contact Infrared sensors, such as in [14] where an IR sensor is devised to provide an alternative to the conventional thermometers which require skin contact especially at the areas of mouth and armpit. The authors addressed how the rise of COVID-19 have heightened the demand for contactless and non-invasive means of measurement, and have thus designed a system to be placed at point of entries that takes a person's temperature at a distance using IR sensor. This measured temperature is then displayed on a 16x2 LCD and entry is either allowed or denied based on the specified range. Other papers such as [15,16] proposed a similar usage of IR sensor (MLX906124) for detecting body temperature in indoor premises to minimize contact with any potential virus carrier. The same system adopts the use of ultrasonic sensors to ensure that the person is within the appropriate range for body temperature screening. To ensure accurate temperature readings, authors in [17] who too, utilized the MLX90614 IR sensor for their design of a contactless and intelligent epidemic prevention system based on PYNQ (Python Productivity for Zynq) performs a calibration on their IR temperature sensor. This calibration was done with against a water bath to provide for a constant temperature reference source, where the temperature of the water bath is compared with readings from the MLX90614 to calculate the offset that needs to be corrected.

1.2.3 Internet of things platform

In a recent work, Narayanan *et al.,* [18] designed an IoT-based occupancy monitoring and screening system as a response to COVID-19 whereby an ultrasonic sensor is used to detect the distance between the person and the system in order to initiate the temperature scanning process taken by an IR temperature sensor. This value is then displayed on a locally mounted LCD and the parameters are logged into an online Google spreadsheet and uploaded to the Thingspeak's cloud server to provide the graphical representation of the data via their online dashboard, in effort to facilitate data collection and remote monitoring. The integration of temperature screening systems with IoT platforms is also demonstrated in [19,20] where all sensory data is sent to Blynk's cloud server to be visually displayed on their dashboard, which is made available in both web browsers and as a mobile application. If any one of the displayed parameters exceeds the threshold level, a notification is prompted and an email is sent to notify and alert concerned parties.

2. Methodology

2.1 Block Diagram

Figure 1 illustrates the block diagram of the system whereby an Arduino UNO microcontroller board and an ESP32-CAM development board (equipped with an OV2640 camera and built-in Wi-Fi module) were used. The Arduino UNO board accepts 3 types of inputs; an MLX90614 temperature sensor, a HC-SR04 ultrasonic sensor and the ESP32-CAM board. The 4 outputs involve a 2x16 I2C LCD, red and green LEDs, an SG90 servo motor and the ESP32-CAM board. Note that the ESP32-CAM communicates to Arduino UNO via serial communication as to allow two-way communication. It receives inputs from Arduino UNO which is then relayed to Blynk and Google Sheets. At the same time, ESP32-CAM accesses Teachable Machine platform to run the face mask detection model which then returns the mask wearing percentage to Arduino UNO to control its outputs accordingly.

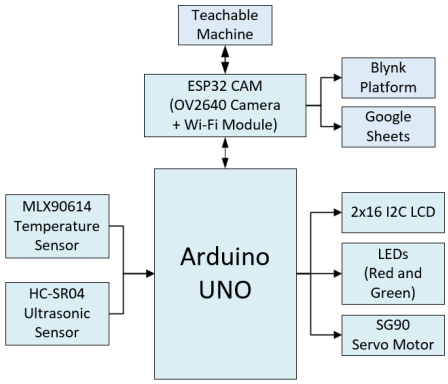


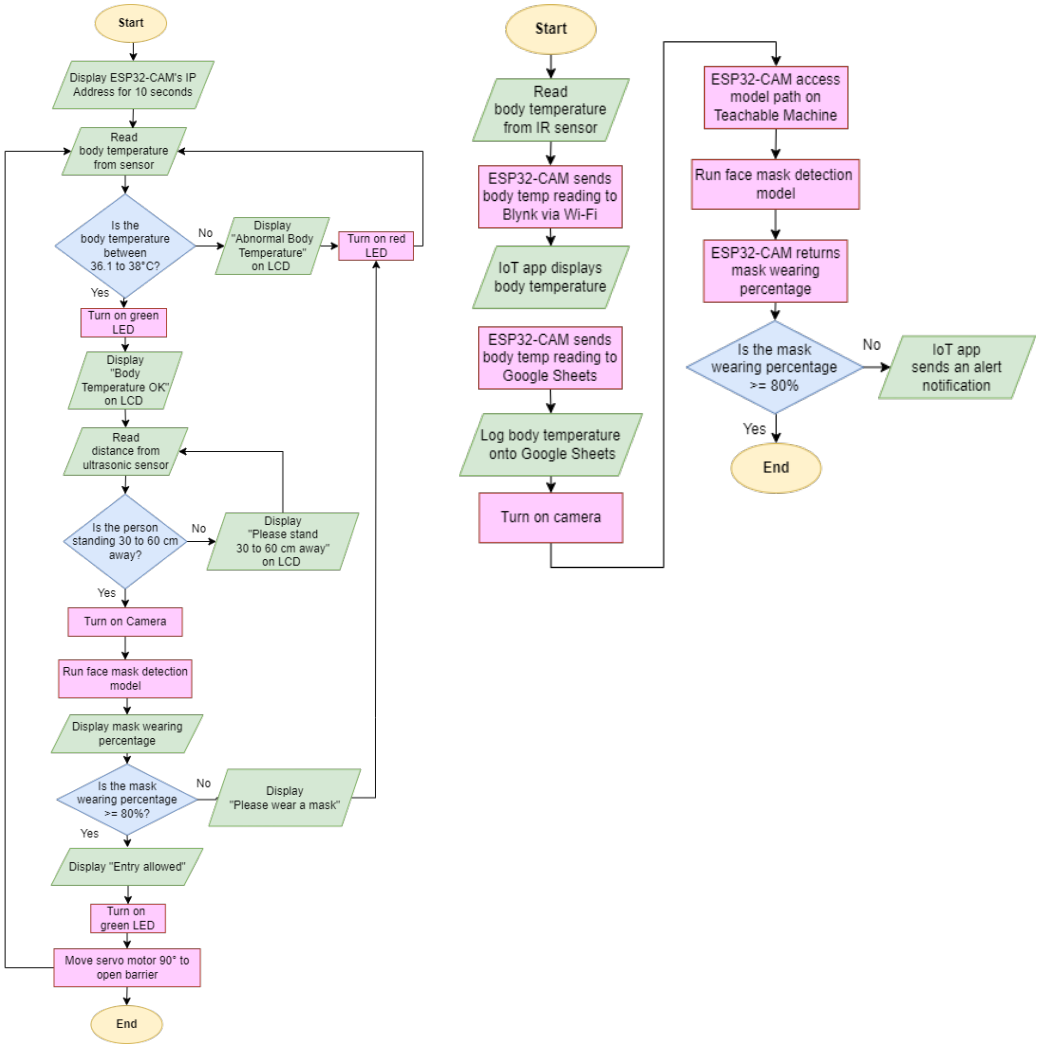
Fig. 1. Block Diagram of the System

2.2 Flowchart

Figure 2(a) depicts the flowchart for the system's hardware. When initiated, it reads the body temperature of the person. A condition is placed where only body temperatures of 36.1℃ to 38℃ is allowed to proceed with the subsequent steps. Else, the LCD will print "Abnormal temperature" with the red LED turned on as an indicator that entry is denied. Otherwise, if the body temperature is within the permitted range, the system will then read the value from the ultrasonic sensor to determine their position away from the camera. If they are between 30 to 60 cm away, which is the

optimal distance for face mask scanning, the system will proceed to turn on the camera to capture a real-time live stream of the person's face. The face mask detection model is then run to discern if the person has complied with mask wearing rules, which is expressed in percentage (%). If the person's mask wearing percentage is below the 80% threshold (meaning they are not wearing a mask or wearing it incorrectly), the LCD will print "Please wear a mask" and the red LED will turn on. If it exceeds the 80% threshold, the green LED will turn on as an indicator and the servo motor is rotated 90° degrees to open the barrier, thus allowing entry.

Figure 2(b) shows the flowchart of the system's software, where it will initially read the body temperature. Then, the ESP32-CAM board (with built-in Wi-Fi module), will transmit this data (body temperature value) to Blynk to be displayed on the mobile and web dashboard, to allow for remote monitoring. The same data is logged onto Google Sheets. Then the camera is turned on and ESP32-CAM will access Teachable Machine to run the face mask detection identify weather the person is wearing a mask or not. This mask wearing percentage is then returned to Arduino UNO. If the mask wearing percentage is less than the 80% threshold, the Blynk will send a notification to alert its user.



(a) (b)

Fig. 2. Flowchart (a) Flowchart of system's hardware (b) Flowchart of system's software

2.3 Circuit Diagram

Figure 3 shows the system's hardware connection. The hardware is powered by a 9V battery or power supply.

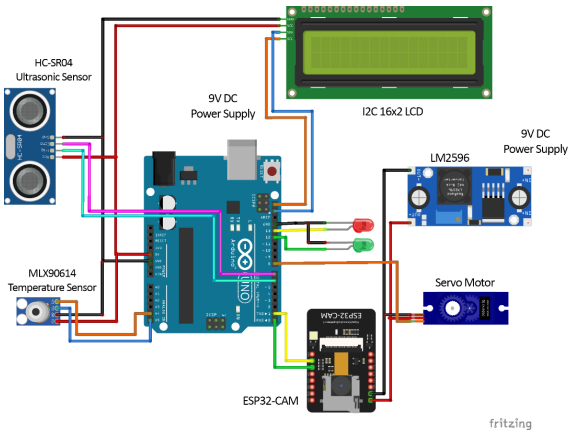
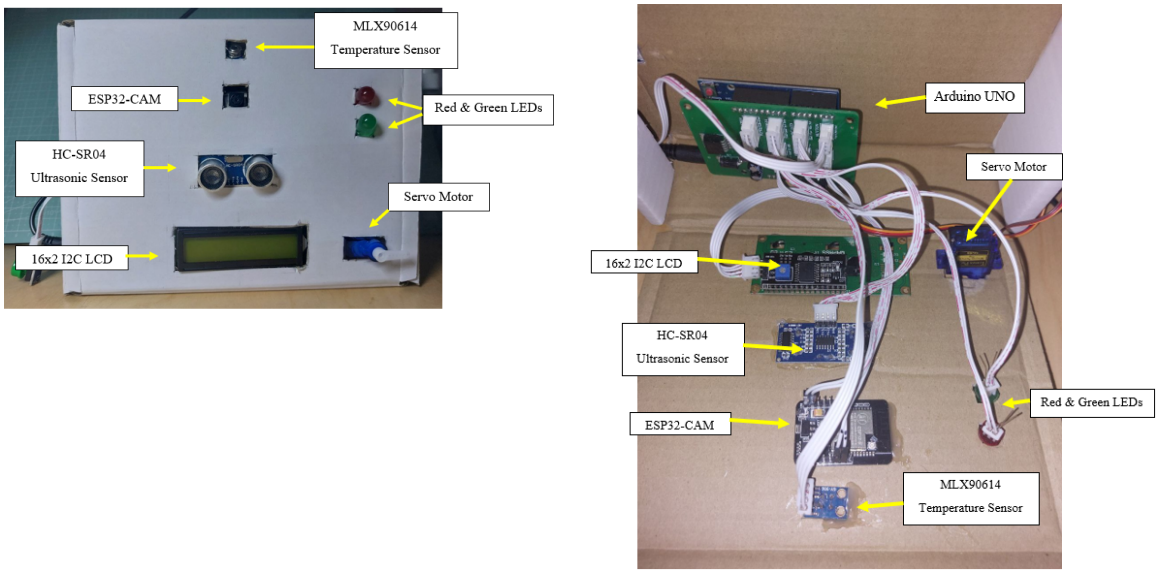


Fig. 3. Circuit diagram of the system

2.4 Hardware Setup and Implementation

Figure 4(a) shows the front view of the system's hardware while Figure 4(b) shows the inside view. The hardware setup and implementation of the system can be seen in Figure 5. The hardware is encased inside a 20 x 6 x 16 cm (length x width x height) box and is placed by the entrance where it stands at approximately 1.57 m tall from the ground. This height is decided after taking into account the average Malaysian female and male height which were 157.1 cm and 169.2 cm respectively which produced an average height of 163. 1 cm according to [21]. Considering that is easier by nature for a taller person to bend down than it is for a shorter person to reach tall heights, it is best that the hardware is placed at heights that accommodate to shorter people, hence the 1.57m height was chosen.



(a) (b)

Fig. 4. Hardware view (a) Front view (b) Inside view

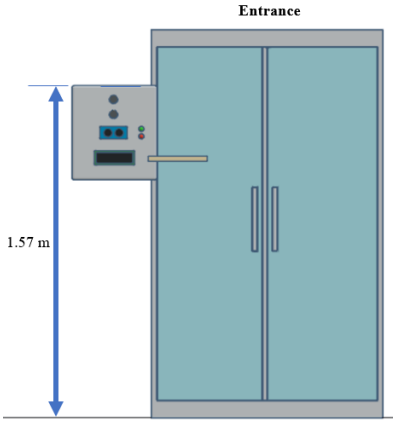


Fig. 5. Hardware setup

2.5 Proposed System and Working Principle

2.5.1 Body temperature detection

An MLX90614 IR temperature sensor was used to obtain the body temperature reading. As it is a non-contact temperature sensor, it can sense a wider range of temperatures than most digital sensors, with an object temperature measurement ranging from -70 to 382.2℃ and ambient temperature, while having a resolution of 0.02℃ and a standard accuracy of ±0.5℃ in room temperature as per its specifications.

According to [22], the average body temperature for an adult is 37℃, but the normal body temperature may range between 36.1 to 37.2℃ or more, depending on your level of activity and age. However, a reading of 38℃ for forehead (temporal artery) is generally a sign of fever. Thus, 36.1 to 38℃ can be assumed as the normal temperature range to be adopted in this particular system.

2.5.1.1 MLX90614 calibration

Before use, temperature sensors need to be calibrated because they can drift over time and exhibit inaccuracies in their readings. Calibration is the process of comparing the sensor's output to a known reference or standard temperature and adjusting the sensor's response to minimize any deviations or errors. As previously mentioned, authors in [17] used a water bath for the calibration of MLX90614 for a consistent source of temperature reference. In our system, the ice bath method was used where the difference in temperature (as tabulated in Table 1) is taken to be the calibration offset that is then applied as a constant correction factor to the sensor's readings, which in our case is found to be 0.028 ℃.

Table 1
Temperature calibration against ice bath

No.	Ice Bath Temperature (℃)	Measured Temperature (℃)	Error (℃)
1	0	0.02	0.02
2	0	0.02	0.02
3	0	0.02	0.02
4	0	0.04	0.04
5	0	0.04	0.04
Average			0.028

2.5.1.2 MLX90614's optimal distance

Another critical parameter of the MLX90614 that is worth noting is its 90° cone-shaped field of view (FOV) depicted in Figure 6, which can be defined as the angle in which the sensor is sensitive to or detects thermal radiation. Theoretically, the greater the distance between you and the sensor, d the greater the sensing area.

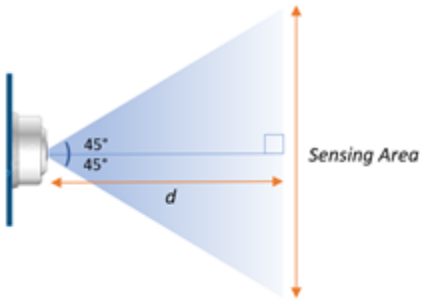


Fig. 6. MLX90614's FOV diagram

Using trigonometry, we are able to formulate the sensing area with the following equation:

$$Sensing\ Area = 2\ \times d\ \times Tan\ (45°) \tag{1}$$

Using Eq. (1) we know that for every 1 cm further you are from the sensor, the sensing area grows by 2 cm. This means that the further a subject is from the sensor, a lot more of ambient space and objects may occupy its FOV and subsequently the sensor will read the average temperature of all said objects. Therefore, it is vital for the subject or person to completely fill the sensor's field-of-view so that an accurate reading can be taken. Since the MLX90614 has varying applications and its specs does not specify the optimal distance in which body temperature readings can be taken, the following experiment is done:

Table 2
Measured temperature and actual forehead temperature at different distances

Distance (cm)	Measured Temperature (℃)	Actual Forehead Temperature (℃)	Error (℃)
2	36.80	36.8	0.00
4	36.80	36.8	0.00
6	36.80	36.8	0.00
8	36.78	36.8	0.02
10	36.78	36.8	0.02
12	36.74	36.8	0.06
14	36.68	36.8	0.12
18	36.68	36.8	0.12

To ensure contactless temperature reading, 2 cm is chosen as the minimum distance and the experiment is repeated at increments of 2 cm further away from the MLX90614 sensor. For every distance, a temperature reading is taken with the MLX90614 on the forehead area and another forehead temperature reading is taken with a temporal artery thermometer. Temporal artery thermometers also utilize infrared to measure the heat on surface of the skin, which is a result of blood moving through the temporal artery in the forehead. It does not require insertion to any body parts and is generally regarded as a non-invasive way to estimate core temperature. Results in Table

2 show that measured temperature readings from distance of 2 to 6 cm away from the sensor produces no error when compared with the actual forehead temperature and can thus be regarded as the optimal distance for non-contact body temperature reading using the MLX90614. Using Eq. (1), we find that distances of 2 to 6 cm produce a sensing area of 4 to 12 cm as depicted in Figure 7. Therefore, this is the range specified and used in our system to trigger the screening for body temperature.

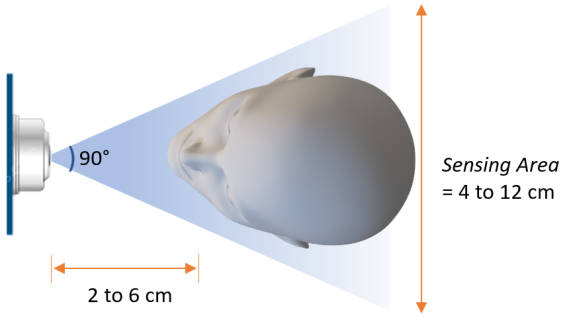


Fig. 7. Optimal Distance for Non-Contact Temperature Reading Within 90° FOV

2.5.2 Serial communication

Body temperature reading is then sent from the MLX90614 to Arduino UNO and subsequently to ESP32-CAM via a device-to-device communication protocol known as Universal Asynchronous Receive Transmit (UART)/ serial communication which can be done by connect the TX (transmit) pin of one board to the RX (receive) pin of the other board, and vice versa as seen in Figure 8. Following this, the Software Serial library which is built-in within Arduino IDE is used and initialization is done with functions such as mySerial(RX_PIN, TX_PIN) along with mySerial.begin(baudRate) to start the software serial communication. After setting up the serial communication on both boards, function Serial.write() or Serial.print() is called on the sender board to send data, and Serial.read() on the receiver board to receive data. This communication can be bidirectional by implementing similar code on both ends.

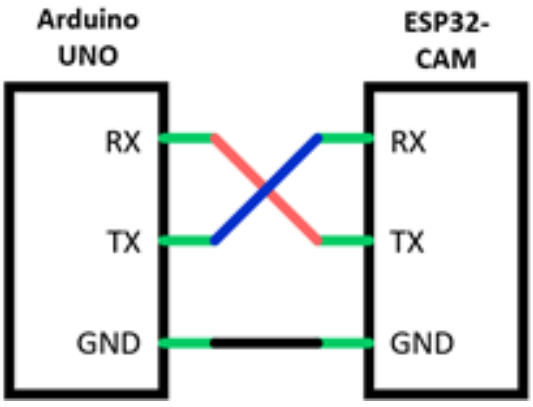


Fig. 8. Universal Asynchronous Receive Transmit (UART) communication between Arduino UNO and ESP32-CAM

2.5.3 Data logging

Using Google Apps Script, sensor values can also be logged onto an online spreadsheet (Google Sheets) which continuously updates the latest body temperature value for as long as the system runs. By deploying the Google Apps Script as a web app and, a URL (Uniform Resource Locators) is generated and is inserted into the Arduino IDE sketch where HTTP POST request is used to send sensor data to the Apps Script URL obtained when deployed. Using the HTTPClient library, we can establish a HTTP request that publishes these data into Google Sheets as a way of data logging. Figure 9 shows the general workflow.

Fig. 9. Using Google Apps Script to publish readings on Google Sheets

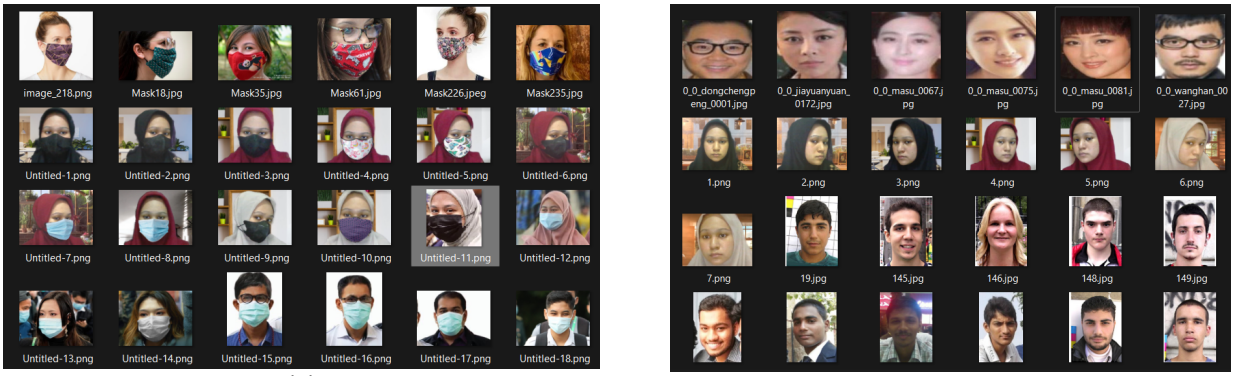
2.5.4 Face mask detection model

Teachable Machine is a web-based tool developed by Google's Creative Lab that allows users to create custom machine learning models for various tasks, such as image classification, sound classification, and pose estimation. The process involves training the model by providing examples of different classes, and the tool handles the complex aspects of machine learning behind the scenes. The general workflow that is implemented within this system to train and develop the face mask detection model can be seen in Figure 10 below.

Fig. 10. General workflow for training and developing model

2.5.4.1 Dataset and training process

To develop a face mask detection model, Teachable Machine is used to train the model using a dataset containing 4092 images that is divided into two class; one class being the positive samples (Wearing Mask Class) containing images of people in face masks (2162 images), and the other class being the negative samples (Without Mask Class) containing images of people without masks (1930 images). A portion of this dataset is obtained online from a GitHub repository in [23] with an added 60 new image samples taken consisting of people wearing and not wearing masks in various head poses and mask colours and patterns as seen in Figure 11(a) and Figure 11(b) for “Mask Wearing” Class and “Without Mask” class respectively. By diversifying the representation of head poses, background variations and illumination or lighting, the accuracy of the trained face mask detection model may be improved. 85% of said dataset is used as training sample whilst the remaining 15% is used as test samples for measuring the model’s performance metrics.



(a) (b)

Fig. 11. Image samples for dataset classes (a) Wearing Mask class (b) Without Mask class

The parameters set during model training are as follows; Epoch: 50, Batch Size: 16. Epoch refers to the number of times the dataset is being fed to the model during training. Theoretically, the higher the epoch, the more accurate the resulting model is upon the completion of the training, but consequently results in longer training time. Batch size is a set of samples used in one iteration of training. In this case, 4092/16 gives us a total of 256 batches to be completed in one epoch. Figure 12 shows an example of one of the trained model outcomes.

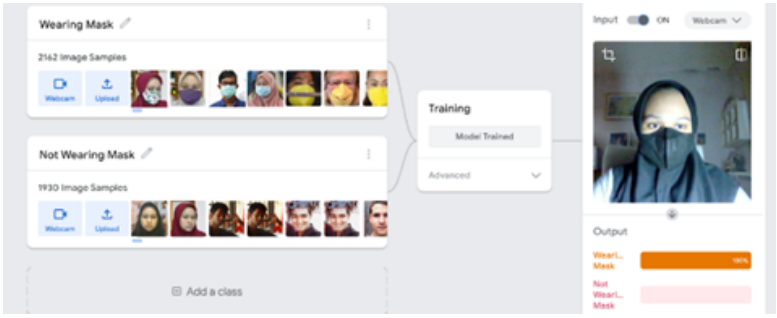


Fig. 12. Trained model outcome

2.5.4.2 Programming the ESP32-CAM

This model is then exported as a shareable link or model path that is specified in the ESP32-CAM programming. To upload the code to the ESP32-CAM development board using Arduino IDE, an FTDI USB to UART converter is used to interface the board with the computer without a level shifter. The connectivity can be seen in Figure 13 below.

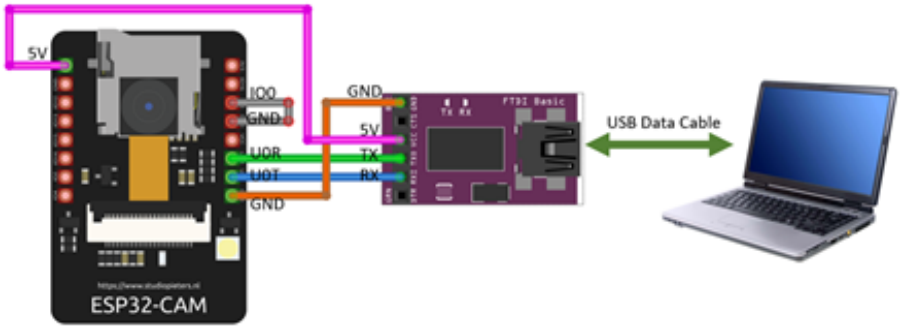
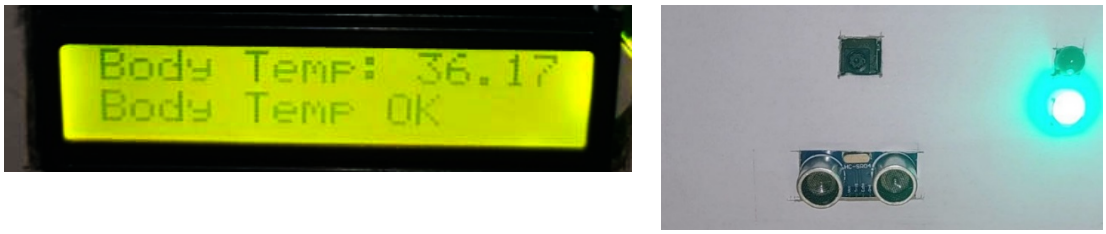


Fig. 13. ESP32-CAM and FTDI connectivity to upload code from a computer

3. Results and Discussion

3.1 Body Temperature Detection, Monitoring and Alert System

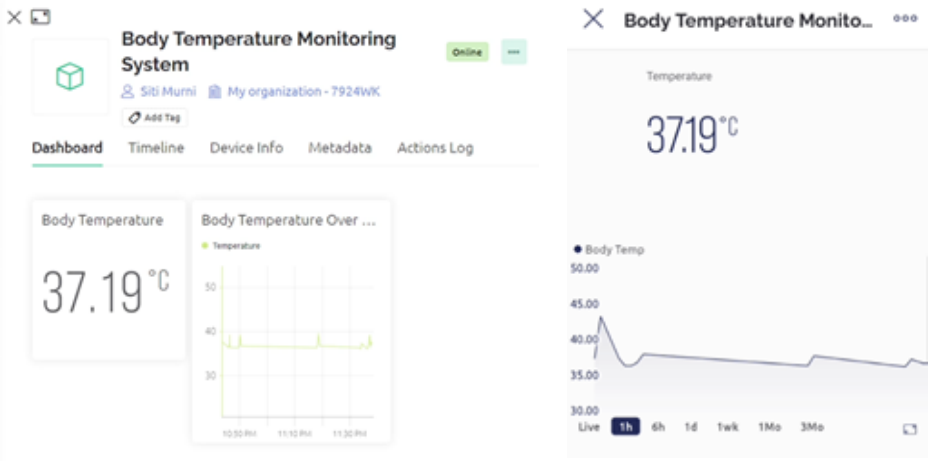
The body temperature that was detected by the MLX90614 sensor is displayed on the LCD. Figure 14(a) below shows the LCD when a normal body temperature is taken. Figure 14(b) shows how the green LED will turn on to indicate that the person has passed the body temperature screening.



(a) (b)

Fig. 14. Hardware results (a) LCD display (b) Green LED status

The body temperature reading is then sent to Blynk's cloud server to be displayed on their dashboard, to allow for remote monitoring. This can be seen in Figure 15(a) and (b) where the latest body temperature is displayed on the left, as well as a visual representation of body temperature readings over time in the form of a chart.



(a) (b)

Fig. 15. Real-time updates and data visualization on Blynk's web dashboard
(a) Web dashboard (b) Mobile dashboard

Figure 16 shows the notification sent to Blynk's mobile application users to alert relevant authorities of when abnormal body temperature is detected.

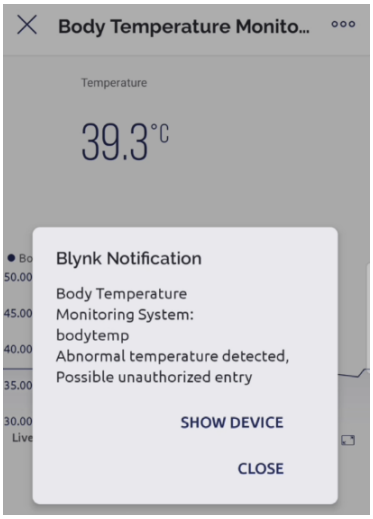
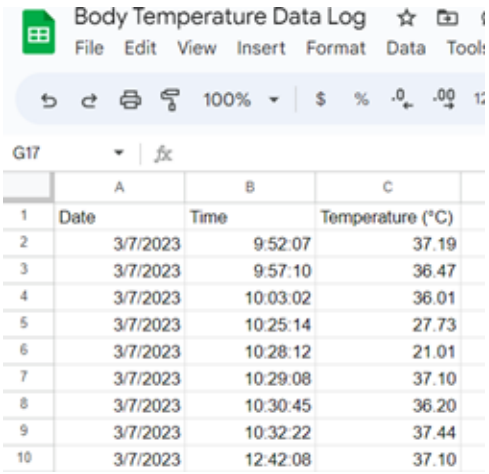


Fig. 16. Alert Notification

Figure 17 shows how the data is logged onto Google Sheets, with date and timestamp listed so users are able to sort or access data by their desired date or time.



	A	B	C
1	Date	Time	Temperature (°C)
2	3/7/2023	9:52:07	37.19
3	3/7/2023	9:57:10	36.47
4	3/7/2023	10:03:02	36.01
5	3/7/2023	10:25:14	27.73
6	3/7/2023	10:28:12	21.01
7	3/7/2023	10:29:08	37.10
8	3/7/2023	10:30:45	36.20
9	3/7/2023	10:32:22	37.44
10	3/7/2023	12:42:08	37.10

Fig. 17. Data Log on Google Sheets

3.1.1 Performance analysis and accuracy

To gauge the performance of our body temperature detection system, comparison is done against other non-contact infrared temperature sensors that are commercially available and typically used for entrance screening. The Beurer FT65 is a multi-functional infrared thermometer with several modes for measuring object, ear and forehead temperature. Forehead temperature mode acts the most similar to how the MLX90614 temperature is implemented in the system, and is therefore used for benchmarking. The results are as tabulated in Table 3 where it was found that the two measuring devices produces an average difference of 0.06℃, and is therefore rather comparable and similar in terms of its readings. Note that all subjects are adults and that the body temperature readings are taken indoors at room temperature conditions during daytime. The MLX90614 readings are taken from 4 cm distance away which is within the optimal range as found via experiment previously in 2.5.1.1. These variables are kept constant throughout the experiment to ensure fair comparison. Note that the MLX90614 has an accuracy of ±0.5℃ while the Beurer FT65's forehead temperature mode has an accuracy of ±0.2 ℃.

Table 3
Performance difference between the MLX90614 with the Beurer F65

Subject (Person)	Temperature (℃)		Absolute Difference (℃)
	MLX90614	Beurer FT65	
1	37.19	37.20	0.01
2	36.47	36.50	0.03
3	36.04	36.10	0.06
4	37.10	37.10	0.00
5	36.20	36.40	0.20
6	37.44	37.50	0.06
7	37.17	37.10	0.07
8	37.08	37.10	0.02
Average			0.06

To evaluate the accuracy of the MLX90614, the measured body temperature readings need to be compared against the actual body temperature. In this case, an oral thermometer (Rossmax TG380) is used to get the actual body temperature readings. Results can be seen in Table 4 and is visualized in Figure 18. Note that every measured body temperature reading is taken at a distance of 4cm which is within its optimal range. It can be seen that the actual body temperature (measured with an oral thermometer) records higher reading than that of a body temperature taken on the surface (contactless) due to internal body temperature being higher than skin temperature.

Table 4
Measured temperature and actual temperature difference

Person	Measured Temperature (℃)	Actual Temperature (℃)	Deviation (℃)	Percentage Error (%)	Accuracy (%)
1	36.45	36.90	0.45	1.22	98.78
2	36.63	37.10	0.47	1.27	98.73
3	37.03	37.60	0.57	1.52	98.48
4	36.16	36.70	0.54	1.47	98.53
5	36.33	37.20	0.87	2.34	97.66
6	36.28	36.80	0.52	1.41	98.59
7	36.19	36.70	0.51	1.39	98.61
8	36.88	37.60	0.72	1.91	98.09
Average			0.58	1.57	98.43

Comparison between the two produces an average difference of 0.58℃, which is consistent with findings in [24] which states that forehead temperature is 0.3 to 0.6 ℃ lower than that of orals. A percentage error of 1.57% is also produced, while the accuracy of our body temperature scanning system is found to be 98.43%.

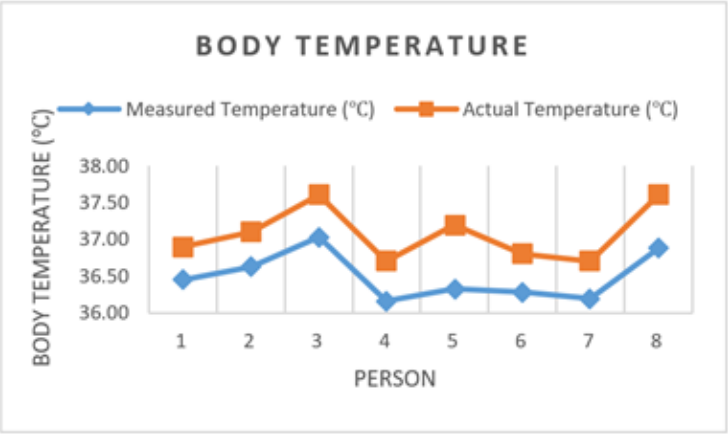


Fig. 18. Measured and actual body temperature of 8 people

3.2 Face Mask Detection System

For every mask detection percentage over 80%, the LCD will display "Entry Allowed" with the green LED turned ON (as seen in Figure 19), while for any value that is less than the threshold, the LCD will display "Please wear a Mask" with the red LED turned ON (as seen in Figure 20). The video streaming web server for monitoring can be accessed by inputting the IP address on any web browser.

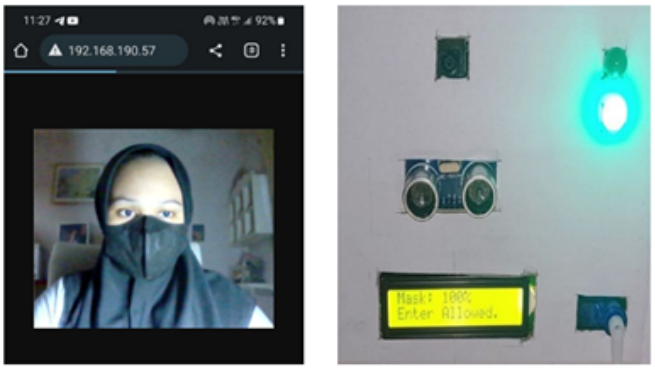


Fig. 19. Video streaming web server and the corresponding LCD and green LED for when mask is detected

Fig. 20. LCD and red LED being lit when mask is not properly worn

3.2.1 Model performance metrics

Upon completion of training, using 15% of the images in each class as testing samples gives us the following performance metrics.

As seen in Figure 21, the "Wearing Mask" class achieves 98% accuracy while the "Not Wearing Mask" class achieves 100%. Confusion matrix in Figure 22 describes the correct classifications and misclassifications in terms of true positives, false positives, false negatives and true negatives. The confusion matrix is typically used to evaluate the performance of a binary classification model like ours which consists of two classes, referred to as positive for "Wearing Mask" and negative for "Not Wearing Mask". In this case, the model incorrectly detects 5 samples of people wearing mask as not wearing any, which gives us 5 false positives. False positives (FP) are the number of instances that were incorrectly predicted as positive by the machine learning model when they actually belong to the negative class. In other words, false positives represent the cases where the model made a mistake by falsely identifying something as positive when it should have been negative. Meanwhile, other classifications made by the model are predicted correctly.

Accuracy per class

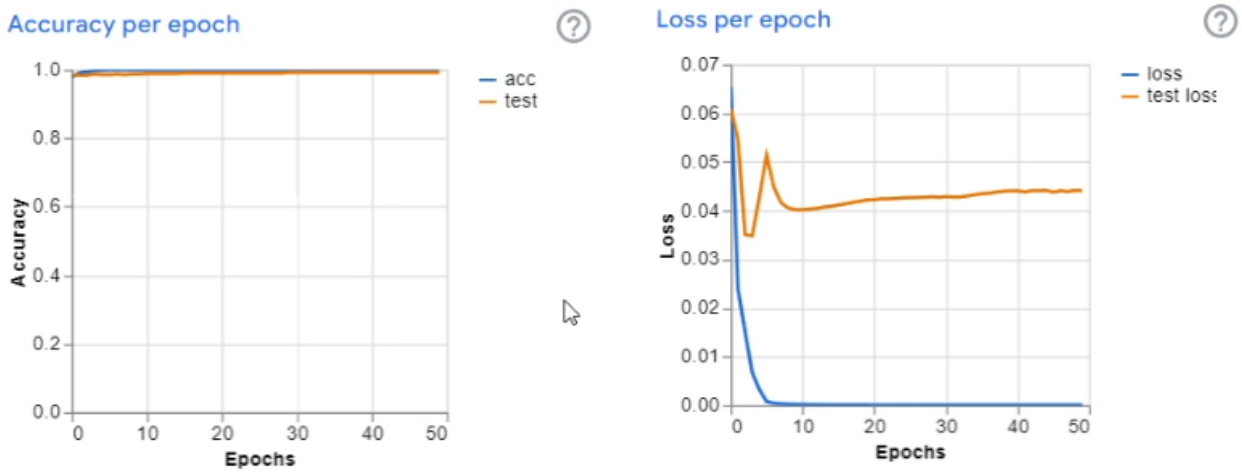
CLASS	ACCURACY	# SAMPLES
Wearing Mask	0.98	325
Not Wearing Mask	1.00	290

Fig. 21. Accuracy per class

Fig. 22. Confusion Matrix

Figure 23(a) and Figure 23(b) depicts the accuracy and loss as training completes; that is as epoch approaches 50. Accuracy tells us the number of classifications the model gets correct during training. In this case, accuracy of the model gets close to 100% upon finishing 50 epochs. Meanwhile, loss indicates the difference from the desired outcome. During each epoch, the model makes predictions on the training data, calculates the difference between these predictions and the actual target values (also known as the ground truth), and then updates its parameters to minimize this difference, which is represented by the loss. The loss function measures how well the model is performing on the training data. It quantifies the discrepancy between the predicted output and the actual target output. The goal of training a machine learning model is to minimize this loss, as it indicates how well the model is fitting the training data and learning the underlying patterns. The better the predictions

are, the lower the loss, which is associated with greater confidence in making its predictions. Two models may have the same accuracy but differ in terms of losses. In our case, the loss is rather low at roughly 0.04.



(a) (b)

Fig. 23. (a) Accuracy per epoch (b) Loss per epoch

3.2.2 Implementation success rate

While the model itself achieves 98% accuracy for "Wearing Mask" class and 100% for "Not Wearing Mask" as per 4.3.1, it is worth noting that such results are achieved against 15% of the dataset whose images are sourced from online sites. During the actual implementation of the system, the face mask detection system may have varying success rates depending on the camera's resolution, frame rate and transfer rate or Wi-Fi strength that affect the video stream in which the model takes inputs from. Additionally, in real life settings or practical applications, a person scanning for mask detection may have varying facial features, head poses and different coloured masks. To simulate these situations, 8 different subjects were chosen to undergo 5 trials of face mask detection. Note that every trial is done at 40cm away from the camera, and the lighting condition of the room is well illuminated, away from any windows that may influence the brightness of the captured images, and these conditions kept constant throughout to ensure fair comparison.

Table 5 shows the results for each trial of 8 different people which showed 97.5% success rate for face mask detection. Recall that in 4.3.1 the accuracy for face mask detection class was 98% when tested against 15% of the dataset. The small discrepancy between the model's 98% accuracy and the system's 97.5% success rate may be explained by the difference in the dataset's sample image resolution and OV2640's camera resolution. Since the model was trained with a dataset that composed of JPEG sample images from various websites, each sample possesses differing resolutions ranging from 100 x 100 (10 000 pixels) to 6000 x 4000 (24 megapixels) with the likelihood that they have went through some form of image compression during cropping and other image transformation. JPEG uses lossy compression which may cause some information to be lost. During implementation however, the model takes input from the OV2640 camera which consistently produces JPEG image captures at 1600 x 1200 (2 megapixel) resolution. Since models are trained to look at each pixel, which stores the necessary digital information, the lack thereof, inconsistency and variability of pixel count may influence how the model makes classifications and predictions. Such argument is supported in [25] where it is found that the accuracy of image classification is dramatically reduced when trained with datasets of compressed JPEG samples.

Table 5
Detection of face mask for 8 different people

Subject	Mask Detection Trial					Success Rate (%)
	1	2	3	4	5	
1	/	/	/	/	/	100
2	/	/	/	/	/	100
3	/	/	/	/	/	100
4	/	/	/	/	/	100
5	/	/	/	/	/	100
6	/	/	/	/	/	100
7	/	/	/	X	/	80
8	/	/	/	/	/	100
Average						97.5

4. Conclusion and Recommendations

To conclude, the system has managed to achieve the following results; an accuracy of 98.43% for the body temperature scanner, accuracy of 98% for face mask detection class, and success rate of 97.5% for the implementation face mask detection model and system. It is also found that the temperature scanning system performs rather similarly with other commercially available infrared sensors, with an average difference of 0.06 ℃. The system's integration of body temperature screening and face mask detection offers added functionalities to authorize entry exclusively for individuals who are healthy and non-infectious. This solution can be applied effectively in various enclosed environments, including the retail industry and workplaces, among others. Some limitations and challenges encountered during the development and testing of the system includes temperature variability whereby factors of ambient temperature, humidity, and physical activity of subjects may affect to the reading's inaccuracy, hence a controlled environment is used for testing to minimize error. Another challenge faced is the number of false positives the model commits during testing which results in false authorization for entry.

Few recommendations for future work include the integration of facial recognition with the body temperature scanning system. This would enable the system to associate temperature readings with specific individuals, providing a more personalized monitoring experience. Geolocation tracking can also be incorporated to monitor body temperature in specific locations or zones. This can be particularly useful in crowded spaces or high-risk areas, allowing for targeted monitoring and response. To improve the face mask detection model, a dataset that is diverse in terms of mask types, colour, patterns, face angle and illumination will greatly increase its capability to detect masks. By training the face mask detection model on a diverse dataset that incorporates these factors, the model becomes more robust, generalizes better to different situations, and minimizes biases, thereby improving its overall performance.

Acknowledgement
The authors are grateful for the financial assistance provided by Universiti Teknologi MARA through LESTARI (Reference Code: 600-RMC/MYRA 5/3/LESTARI (034/2020). The authors would also like to thank the College of Electrical Engineering at UiTM Shah Alam in Selangor, Malaysia, for providing research facilities.

References

[1] N. A. Sulaiman, "Covid-19: Health Ministry reports Arcturus variant cases in KL, Selangor," *New Straits Times*, Kuala Lumpur, (2023). https://www.nst.com.my

[2] Kenyataan Akhbar, K. P. K. "Mac 2020–Situasi Semasa Jangkitan Penyakit Coronavirus 2019 (COVID-19) di Malaysia." (4).

[3] F. Nizam, "Health ministry eases Covid-19 guidelines; masks not mandatory on public transport and hospitals," *New Straits Times*, Kuala Lumpur, (2023). https://www.nst.com.my

[4] Jamini, Archana, Sai Sri Lalitha Perubhotla, Grishma Reddy Patlola, Sruthi Velpula, and G. L. Sumalata. "Face Mask and Temperature Detection for Covid Safety using IoT and Deep Learning." In *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, pp. 373-379. IEEE, 2022. https://doi.org/10.1109/ICCES54183.2022.9835768

[5] Sanjaya, Samuel Ady, and Suryo Adi Rakhmawan. "Face mask detection using MobileNetV2 in the era of COVID-19 pandemic." In *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, pp. 1-5. IEEE, 2020. https://doi.org/10.1109/ICDABI51230.2020.9325631

[6] Das, Arjya, Mohammad Wasif Ansari, and Rohini Basak. "Covid-19 face mask detection using TensorFlow, Keras and OpenCV." In *2020 IEEE 17th India council international conference (INDICON)*, pp. 1-5. IEEE, 2020. https://doi.org/10.1109/INDICON49873.2020.9342585

[7] Yu, Jimin, and Wei Zhang. "Face mask wearing detection algorithm based on improved YOLO-v4." *Sensors* 21, no. 9 (2021): 3263. https://doi.org/10.3390/s21093263

[8] Eswar, Medisetty, Nali Venkata Rao, Palla Dhanush, Meka Mohith Chowdary, and V. Hima Deepthi. "Real Time AI Based Face Mask Detector." In *2022 International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS)*, pp. 206-210. IEEE, 2022. https://doi.org/10.1109/ICETEMS56252.2022.10093259

[9] Sanjay, S., SS Nithish Soorya, R. Vengatesh, and KC Sri Hari Priya. "Security Access Control System Enhanced with Face Mask Detection and Temperature Monitoring for Pandemic Trauma." In *2022 2nd International Conference on Intelligent Technologies (CONIT)*, pp. 1-6. IEEE, 2022. https://doi.org/10.1109/CONIT55038.2022.9848266

[10] S. Srivastava. "Mask-Detection-Face-Recognition-Using-ESP32-Camera," *GitHub*, (2022). https://github.com/MasterSarvagya/Mask-Detection-Face-Recognition-Using-ESP32-Camera

[11] N. Suthar. "Face-mask-detection-Using-ESP32-CAM-Teachable-machine," *Github*, (2021). https://github.com/niteshsuthar/Face-mask-detection-Using-ESP32-CAM-Teachable-machine

[12] Agustian, Diki, Pande Putu Gede Putra Pertama, Padma Nyoman Crisnapati, and Putu Devi Novayanti. "Implementation of Machine Learning Using Google's Teachable Machine Based on Android." In *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, pp. 1-7. IEEE, 2021. https://doi.org/10.1109/ICORIS52787.2021.9649528

[13] Carney, Michelle, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. "Teachable machine: Approachable Web-based tool for exploring machine learning classification." In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pp. 1-8. 2020. https://doi.org/10.1145/3334480.3382839

[14] Mnati, Mohannad Jabbar, Raad Farhood Chisab, Azhar M. Al-Rawi, Adnan Hussein Ali, and Alex Van den Bossche. "An open-source non-contact thermometer using low-cost electronic components." *HardwareX* 9 (2021): e00183. https://doi.org/10.1016/j.ohx.2021.e00183

[15] Catalbas, Mehmet Cem. "A framework of multi-point infrared temperature screening system for COVID-19 pandemic." In *2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, pp. 1-4. IEEE, 2021. https://doi.org/10.1109/ISAECT53699.2021.9668508

[16] Vulpe, Andrei, Ciprian Lupu, and Cosmin Mihai. "Research on infrared body temperature measurement–virus spreading prevention." In *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1-4. IEEE, 2020. https://doi.org/10.1109/ECAI50035.2020.9223195

[17] Lingzhi, Wang, Wang Jianyuan, and Xu Yingjing. "Design of Contactless Intelligent Epidemic Prevention System based on PYNQ." *Engineering Letters* 30, no. 2 (2022).

[18] Narayanan, K. Lakshmi, R. Santhana Krishnan, and Y. Harold Robinson. "IoT Based Smart Assist System to Monitor Entertainment Spots Occupancy and COVID 19 Screening During the Pandemic." *Wireless Personal Communications* 126, no. 1 (2022): 839-858. https://doi.org/10.1007/s11277-022-09772-1

[19] Kulkarni, Pavankumar, Hamjad Ali Umachagi, and Mariya sultana Bakshi. "IOT Based Temperature and Oxygen Pulse Scan Entry System." In *2022 3rd International Conference on Communication, Computing and Industry 4.0 (C2I4)*, pp. 1-6. IEEE, 2022. https://doi.org/10.1109/C2I456876.2022.10051183

[20] Ranjana, S., Ramakrishna Hegde, and C. D. Divya. "Real Time Patient Monitoring System Using BLYNK." In *2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pp. 327-331. IEEE, 2021. https://doi.org/10.1109/DISCOVER52564.2021.9663681

[21] Wikipedia Contributors. "Average human height by country," *Wikimedia Foundation*, (2019). https://en.wikipedia.org/wiki/Average_human_height_by_country

[22] Mayo Clinic. "Fever: First aid Print," *Mayo Clinic*, (2022). https://www.mayoclinic.org/first-aid/first-aid-fever/basics/art-20056685

[23] C. Deb. "Face-Mask-Detection/dataset," *Github*, (2021). https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset

[24] S. Wright and C. Vandergriendt. "What Is the Normal Body Temperature Range?," *Healthline*, (2022). https://www.healthline.com/health/what-is-normal-body-temperature

[25] Lau, W-L., Z-L. Li, and KW-K. Lam. "Effects of JPEG compression on image classification." *International journal of remote sensing* 24, no. 7 (2003): 1535-1544. https://doi.org/10.1080/01431160210142842