Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 935-951
© Universiti Tun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Development of Face Mask Detection System Using Arduino Alert Feature

Siveshwarn Thangaveloo[1], Suhaila Mohd. Yasin[1]*,

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.053
Received 14 June 2022; Accepted 20 May 2023; Available online 30 June 2023

Abstract: Coronavirus Disease 2019 (COVID-19) has emerged as a serious public health issue, causing severe acute respiratory illness in humans. The limited availability of pharmaceutical interventions for the COVID 19 outbreak has compelled governments all over the world to impose precautions and hygienic measures to prevent virus transmission. One of these important measures is wearing a face mask in public places, which significantly reduces the risk of transmission. Hence, a system for monitoring face mask detection using a machine learning model is developed. The system is developed to design a face mask detection system that ensures public health and safety by detecting individuals who do not wear their masks properly or not wearing masks at all and to develop a face mask detection system using an object-oriented approach and machine learning by training it using various data models/sets to improve its detection rate and accuracy .Using the Iterative Design Model, the system incorporates Arduino alert feature to alert faces that are captured by the camera with the absence of a face mask. Python programming language is used in the system development. Subsequently, the system is then tested to a selected group of users to determine its applicability.

Keywords: Covid-19, Face mask, IoT.

1. Introduction

In December 2019, a new and worryingly contagious primary pneumonia broke out in Wuhan, China. The new disease, called coronavirus disease 2019 (COVID-19), was later found to be caused by a previously unknown zoonotic coronavirus, named severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [1]. One of the transmission routes of COVID-19 is through droplets of saliva or nasal secretions. This can be problematic in crowded places especially when an infected person coughs or sneezes since the SARS-CoV-2 virus is highly transmissible. As of now, there is no specific treatment publicly available for COVID-19 and infections had to be curbed through other prevention methods such as social distancing, frequent handwashing and wearing face masks in public. Studies have shown that wearing masks can reduce the risk of coronavirus transmissions [2] which makes it one of the more effective prevention methods. According to the World Health Organization (WHO), the right way to

Corresponding author: ysuhaila@uthm.edu.my
2023 UTHM Publisher. All rights reserved.

wear a mask is by adjusting the mask to cover the mouth, nose, and chin [3]. The protection from the disease will be greatly reduced if the mask is not worn properly.

At present, there are security guards arranged at the entry points in business premises and private residences to ensure the guests and customers wear their face masks and to check their temperature to curb the outspread of the Covid-19 virus. These workplaces must follow a strict health and safety protocols such as wearing masks, avoiding crowding and social distancing. Health authorities are working hard on ensuring that these businesses follow all the protocols for keeping their employees safe. They are conducting regular inspections and even shutting down establishments that continue to violate the Covid-19 safety protocols. Due to this, the security personnel must conduct inspections manually where it requires them to ask the customers to write down their name on a logbook or scan the MySejahtera application's QR code. Aside from this, they also need to check the customers' vaccination status and temperature within proximity before letting them enter the premise.

However, not only this safety protocol exposes the guards to asymptomatic Covid-19 carrier, but its inefficiency also leads to overcrowding at the entrances. Inevitably, the manual process increases the risk of spreading the disease and is time-consuming. If an asymptomatic Covid-19 customer is not wearing a face mask, the customer jeopardizes the security guard's health and safety including other innocent customers/ bystanders.

To solve these problems, the Face Mask Detection System is proposed where everyone who enters an establishment is scanned visually and artificially to detect a face mask visibility by using face recognition technology and object detection. The system will be implemented by utilizing Artificial Intelligence based on machine learning which prioritizes the health and safety of Malaysian citizens. This proposed system will also be connected to a repository which will be recording the time, date, and the faces of the non-compliant students.

This paper is organized in four sections. Section 1 explains the background of the project. Section 2 gives the related work. Section 3 briefs about the methodology. Section 4 describes about the results and discussions and section 5 concludes the project.

2. Related Work

To develop the proposed system, three existing systems with similar main features as the proposed system have been studied. Those systems are as follows.

2.1 Detection System of Facial Patterns with Masks in New Normal Based on The Viola Jones Method

In a study carried out by Jauhari [4], the aim was to detect facial patterns that indicated the presence of facial masks in images. For this, it was based on Single Board Computer (SBC) Raspberry Pi. A face detection system based on the Viola Jones algorithm [citation to the original, principal work] was used to obtain efficient, rapid, and accurate results [4]. Developed in 2001 by Paul Viola and Michael Jones [citation], the Viola-Jones method is an object-recognition/detection framework that allows the detection of image features in real-time. Despite being an outdated framework as it was created in 2001, Viola-Jones algorithm is quite powerful, and the applications which implement the algorithm have proven to be exceptionally notable in real-time face detection. This method allows the adjustment of the cascade classifier to determine the area of the face in the image where it only detects mask and no mask. This existing system uses Python as it's software library and AdaBoost as it's classification model. This system made by Jauhari uses an Internet of Things (IOT) device namely Raspberry Pi which resembles the Arduino Board in the proposed system. Using an Arduino board for the proposed system seems to be very likely as it is much cheaper, and it is easier to integrate devices with Arduino compared to Raspberry Pi as it is open-sourced.

2.2 Face Mask Detection System Using Deep Learning

Kurlekar [5] have developed a face mask system that can be implemented in offices, airports, and public places in general. This application can detect face masks in static images as well as in real-time videos. To do this, the application used Computer Vision methods or ideas and Deep Learning, incorporating OpenCV, TensorFlow, and Caffe. The datasets used in this existing system were mask and no mask only using the CNN classification model. A few of the software library used by this existing system is like the proposed system which are OpenCV and TensorFlow. This system also uses the same classification model as the proposed system which is the MobileNetV2.

2.3 Smart Screening and Disinfection Walkthrough Gate (SSDWG)

Smart Screening and Disinfection Walkthrough Gate (SSDWG) was created by Hussain in 2021 [6]. It is a rapid and efficient virus spread detection and control system based on IoT (Arduino and Ultrasonic sensors), for all places of entry. In addition to the registering body temperature through temperature sensors that do not require physical contact, the system is also capable of differentiating people who wear face masks from those who do not. For the classification, it was also added not only if they were wearing a mask but also their correct use such as nose out of the face mask. For this classification, VGG-16, MobileNetV2, InceptionV3, and ResNet-50 have been utilized using a transfer learning approach. The use or non-use of the mask was implemented through deep learning in real-time. The obtained precision was 99.81 using VGG-16 and MobileNetV2, respectively. In addition, the classification of the type of mask, either N-95 or surgical masks, has also been implemented. This existing system detects mask, no mask and nose out. The face detector for this system is YOLO v3 while its software library is Keras. One of the IOT device used in this existing system is an Arduino board, which is like the proposed system. This existing system and the proposed system will both be using MobileNetV2 as a medium to conduct machine learning.

Table 1 presents a comparison of the works that have been discussed. The works are compared based on several features namely type of detection, face detector, classification model, software library and IOT device. Next, these features are compared to the system to identify the important features that should be present in the face mask detection system.

Table 1: Comparison table

Features/System	Detection System of Facial Patterns with Masks in New Normal Based on The Viola Jones Method (Jauhari, 2021),	Face Mask Detection System Using Deep Learning (Kurlekar, 2021)	Smart Screening and Disinfection Walkthrough Gate (SSDWG) (Hussain, 2021)	Real Time Face Mask Detection System Utilizing Machine Learning Technology for Tution Cikgu Mani
Type of Detection	Mask/No Mask	Mask/No Mask	Mask/No Mask/Nose Out	Mask/No Mask/Improper Mask Wearing
Face Detector	Cascade Viola Jones	-	YOLO v3	Single shot multibox
Classification Model	AdaBoost	CNN	VGG-16, MobileNetV2, InceptionV3, ResNet50	MobileNetV2 FPNLite 320x320
Software Library	Python	TensorFlow, OpenCV, Caffe	Keras	Python, TensorFlow and OpenCV
IOT Device	Raspberry Pi	-	Arduino and Ultrasonic sensors	Arduino

3. Methodology/Framework

In this project, the iterative model was selected as the development guide. This iterative model contains of five main phases namely the planning phase, analysis and design phase, implementation phase, testing phase and evaluation phase. The analysis and design phase, and the implementation phase are done together. These phases performed repeatedly until the system fulfills the proposed requirements and objectives.

3.1 System Development Workflow

There are total of six phases from the iterative design model. Table 2 summarizes each of the phases, activities, and deliverables of this project.

Table 2: Deliverable Table

Phase	Activity	Output
Planning	- Identify the Objectives, Problems and Scope. - Identify the System Requirements.	- Proposal. - Gantt Chart
Requirement	- Information analysis. - Data gathering. - Identifying required programming languages and technologies. - Create ERD, Context Diagram, Use Case diagram.	- The functional and non-functional requirements of the system. - Software Requirement.
Design and Analysis	- Design system programming.	- UML diagrams of the proposed system - The storyboard of the proposed system.
Build and Implementation	- Creating a System Husk. - Creating a Data Set/ Model. -Training the AI. -Preparing Coding for Prototype. - Refining the prototype according to user preference. - Create a working repository (database).	- A functioning prototype.
Testing	- Prototype testing.	- The test report
Evaluation	- Prototype is presented to the client. - Suggestions and feedbacks are gathered and analyzed.	- The user feedback report
Deployment	- Improve Prototype. - Code and Test Final Prototype. -Document Final Prototype. -Completion of Finalized System - Finalized system is presented to the stakeholders for actual use.	- Completed system.

3.2 Hardware and Software Specifications

In this section, the hardware and software specification required to develop the Face Mask Detection system is described. Table 3 shows the hardware specifications while Table 4 shows the software specifications of the system.

Table 3: Hardware specifications

Hardware Requirements	Description
Computer	ASUS ROG Zephyrus G15 2021
Processor	AMD Ryzen 9 5900HS
Random Access Memory (RAM)	32GB
Solid State Drive (SSD)	2TB
Peripheral Devices	Arduino Uno Board, External Camera

Table 3: Hardware specifications (cont)

Table 4: Software Specifications

Software Requirements	Description
Operating System (OS)	Window 10
Programming Platform	IDLE 3.7 (64-bit)
Database	Local Repository
Documentation	Microsoft Word

3.3 Functional and Non-Functional Requirements

During the requirement phase, the functional and non-functional requirements of the system are elicited from the stakeholder, who is the owner of the tuition center. Table 5 describes the functional requirements, whereas Table 6 present the non-functional requirements of the system.

Table 5: Functional requirements

No	Modules	Functionalities
1	Learning Module	-The system should be able to produce more accurate results by learning from provided data sets.
2	Detection Module	- The system must be able to detect face masks worn by the students. -The system must be able to recognize whether the face mask is present or improperly worn by the students/bystanders.
3	Security Module	- The system will sound the buzzer when it detects someone wearing their face mask improperly or not wearing them at all.
4	Database Module	-This system must be able to record and store data such as time and date of entry together with the faces of the non-compliant students in the repository.

Table 6: Non-Functional requirement

No	Requirements	Description
1	Performance	The system’s uptime must be more than 99% as it should be operating continuously unless a maintenance activity is in progress.
2	Security	The system will be utilising a buzzer, which is connected through an Arduino board that acts as an alert system to notify when there is a breach.

3	Portability	The establishment owner should prepare a laptop or a desktop computer with an external webcam, or a combination of Arduino board and monitor also with an external webcam.

4. Results and Discussion

The results and discussion section presents data and analysis of the study. This section shows diagrams that models the system. The results are divided into several segments that shows the designs of the diagrams involved to complete this project and then followed by the actual system interfaces and code segments. The discussion section consists of the functional and User Acceptance Testing.

4.1 Use Case Diagram

The Use Case diagram illustrates how users interact with the system that is going to be developed [7]. Since most of the process are automated, there is little user interaction in this system. 3 actors are defined in the diagram which are students, administrators, and security personnel. The use case diagram is small due to most of the activities being automated and their little interaction with end-users. The students will automatically provide visual input which is their faces when they are within the range of the camera. Meanwhile, the security personnel can monitor this video feed from a distance. The system will automatically detect whether there is a face mask available or not on the face of students. If a face mask is found or worn, the buzzer will activate which will alert both the student and security personnel. The administrator will have the ability to create, read, update, and delete the dataset as they find fit for the system and to improve its detection accuracy. The Use Case diagram is shown in the appendices as Figure 1.

4.2 Sequence Diagram

The student will first provide their visual input which is their face. This data will be directed to the core system and then directed to trained model to verify whether there's a mask present on the face. The results of the detection will be sent back to the system, where the system will direct it back to security personnel to give them a clear visual output. If the trained model does detect a face mask on the face, a command will be directed to the buzzer that is connected through the Arduino board. This buzzer will start vibrating which will directly alert both the user/ student and security personnel. The picture of the person not wearing face mask and the date and time which they entered the premise is then saved in the repository. The administrator can then access the repository to retrieve that information and train the model further using the pictures gathered to increase the accuracy of the system. The Sequence diagram is shown in the appendices as Figure 2.

4.3 Class Diagram

There are five important classes that will be included when developing the project which are Student, System, Arduino, Buzzer, and Repository. Each class contains its own special attributes and methods to ensure that the Face Mask Detection System works properly. The Class diagram is shown in the appendices as Figure 3.

4.4 Requirement Traceability Matrix

Table 7 presents the requirement traceability matrix for the system. It allows all requirements of the system to be identified systematically. The requirement traceability matrix for the system consists of requirements identified for each module, namely Learning, Detection, Security and Database Modules.

Table 7: Requirement's Traceability Matrix

Allocated	Descriptions
REQ_100	**Learning Module**
REQ_101	• Data model is trained using a face model and the data sets provided
REQ_200	**Detection Module**
REQ_201	• User (students) are required to pass in front of the external camera
REQ_202	• Visual input from the user (student) is detected by the system

REQ_203	• System passes visual input data to TensorFlow within the system.
REQ_204	• TensorFlow sends back information of object detected (face mask)
REQ_205	• System provides visual output (display) to user (security personnel) the presence of face mask.
REQ_206	• System shows green box around user's (student) face if face mask is present.
REQ_207	• System shows red box around user's (student) face if face mask is not present.
REQ_300	**Security Module**
REQ_301	• System sends message to the Arduino board if no mask is present on user (student) face.
REQ_302	• Arduino sounds buzzer to alert the users (student and security personnel)
REQ_400	**Database Module**
REQ_401	• System captures image of user (student) not wearing mask.
REQ_402	• System stores captured image of student in repository.
REQ_403	• System stores time and date of image captured

Table 7: Requirement's Traceability Matrix (cont)

4.4 Flowchart Diagram

The system is divided into two phases, namely the software and hardware phase. The software phase consists of the process of preparing the dataset, preparing the training model, and training the face mask detector. Next, the hardware phase will proceed with the hardware-related process in the system. The Flowchart diagram is shown in the appendices as Figure 4.

4.6 Schema Table

Table 8 is the table from the repository (database) that have been designed and extracted from the class diagram. Database table as shown in Table 8 is used to store the faces of the non-compliant students together with the date and time that the system detects the anomaly.

Table 8: Database table

Attributes	Data Type	Size	Key	Descriptions
ID	int	10	None	Identifier
Image	blob	1e+7 (10mb)	None	Student image during anomaly
Date & Time	DATETIME	8	None	Date and time of anomaly detection

4.7 Arduino Board Design

Figure 5 depicts the Arduino board design that is used for the alert feature of the system. The alert is used during the hardware phase in the flowchart of the system, (see Figure 4). The alert is activated every time a non-compliant face is detected, by emitting a buzzing sound. Figure 5 is shown in the appendices.

4.8 Interface Design

User interface is the medium that connects the user to the system through the graphical features is a system. Some regular features found in most of the computer software or systems are buttons, readable text, and harmony colors. User interface designs are compulsory for users to interact with the system. Therefore, these features along with many other graphical interface designs support users to understand and handle the system better. However, since this project does not require any interaction with users, there are only two designs for the interface. These two designs consist of what the system shows when a face is detected and then a face mask is detected. Figures 6 and 7 will show the two-interface design in the face mask detection system. Figure 8 shows the interface in the repository, followed by their corresponding code segments in Figures 9, 10 and 11 which can be seen in the appendices.

4.9 Test Cases

The Real Time Face Mask Detection System has been tested using test cases for each module. The modules involved in the testing process are the Learning, Detection, Security, and the Database modules. Table 9 shows the functional testing for Real Time Face Mask Detection System for each module.

Table 9: Test Cases

		Module: Learning		
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
J1-1	To get a data model using the data sets provided.	The module produces a data model to be imported in the system.	The module produces a data model to be imported in the system.	Pass
		Module: Detection		
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
J2-1	Visual Input is provided by the system.	The system starts a video stream where it detects in real-time.	The system starts a video stream where it detects in real-time.	Pass
J2-2	System detects a person when they are wearing a mask.	The system shows a green box around the person's face if mask is detected.	The system shows a green box around the person's face if mask is detected.	Pass
J2-3	System detects a person when they are not wearing a mask.	The system shows a red box around the person's face if mask is detected.	The system shows a red box around the person's face if mask is detected.	Pass
		Module: Security		
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
J3-1	The buzzer connected to the Arduino buzzes when it detects a person wearing a mask signalling that they can enter.	The buzzer buzzes when it detects the presence of a face mask on a person.	The buzzer buzzes when it detects the presence of a face mask on a person.	Pass
		Module: Database		
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
J4-1	The system captures the image in the video stream when it detects an anomaly.	The system captures the images of the ones not wearing a face mask.	The system captures the images of the ones not wearing a face mask.	Pass

J4-2	The system stores the captured images in the repository.	When the database (repository) is opened captured images can be found stored in there.	When the database (repository) is opened captured images can be found stored in there.	Pass
J4-3	The system records the time and date of the captured images.	While accessing the database, the time and date of the captured images must be shown	While accessing the database, the time and date of the captured images is shown	Pass

Table 9: Test Cases (cont)

4.10 User Acceptance Test

A user acceptance test was conducted using a Google form. The form was then broadcasted to the stakeholders along with the system and its instructions. A scale of 1 to 5 was used where 1 is the lowest rating and 5 is the highest rating. The users were in the age range of 17 to 48. Figures 12, 13, 14, 15, and 16 shows the results of the user acceptance test.

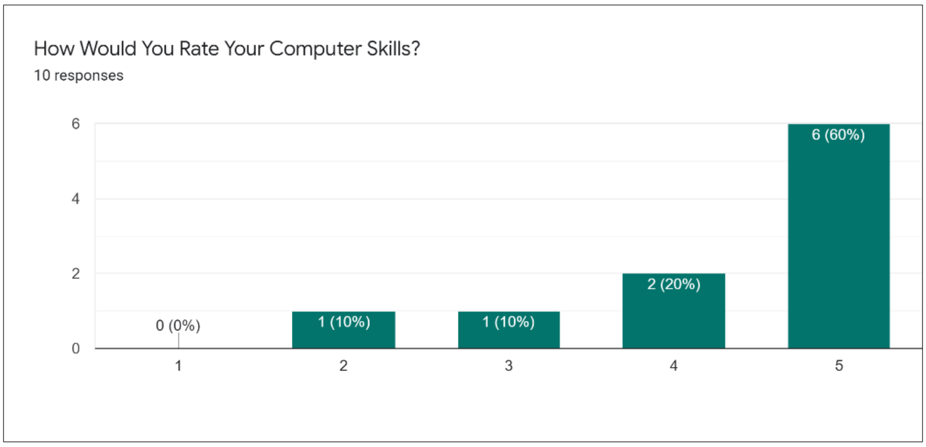

Figure 12: Test Results for The User’s Computer Skills.

Based on Figure 12, it can be deduced that most of the people at the stakeholder’s organization have very good computer literacy.

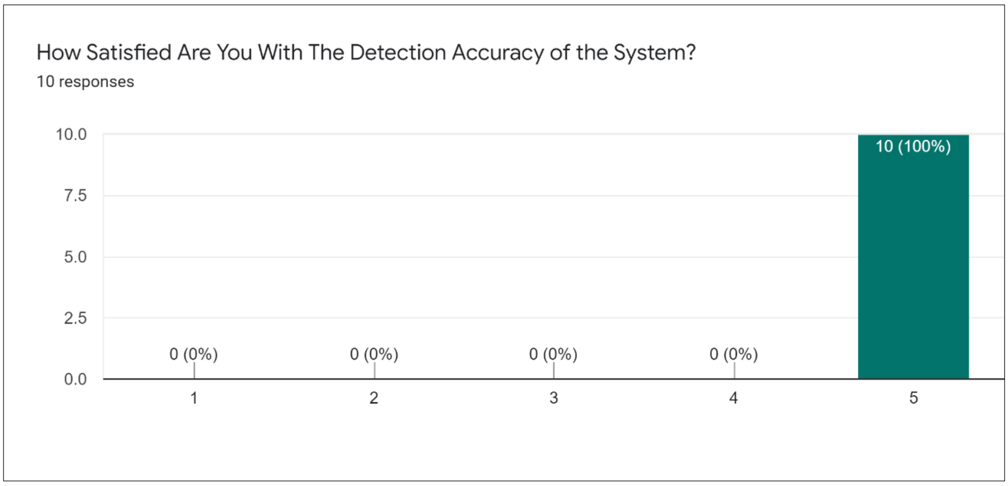

Figure 13: Test Results for Detection Accuracy of The System.

Based on Figure 13, it can it clear that the system's detection accuracy is very satisfactory for the users.

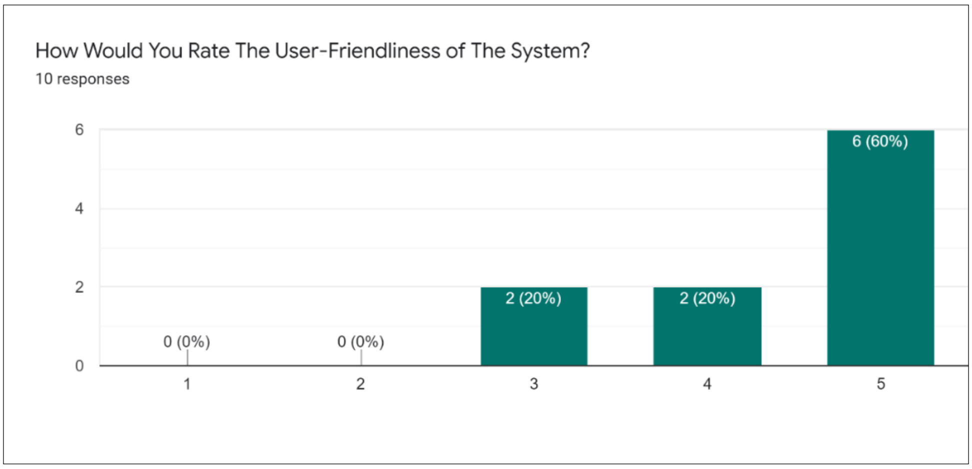

Figure 14: Test Results for the User-Friendliness of The System.

Based on the outcome of the test, it is vivid that most of the users (60%) found that the system was user friendly enough for them to use it without any issues while less than half of them (40%) rated it 3 and 4 on the scale provided.

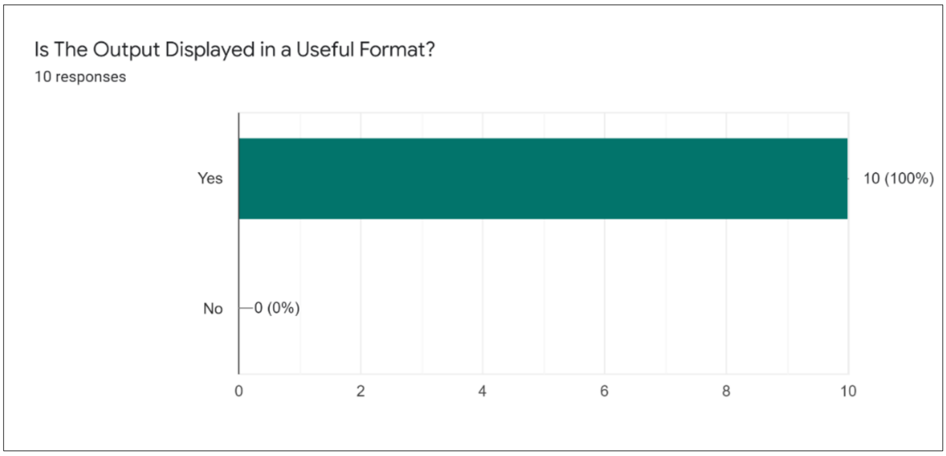

Figure 15: Test Results for The Usefulness of The Output Displayed.

Figure 15 shows that the users found the output displayed at the end was very useful which when the captured images are shown.

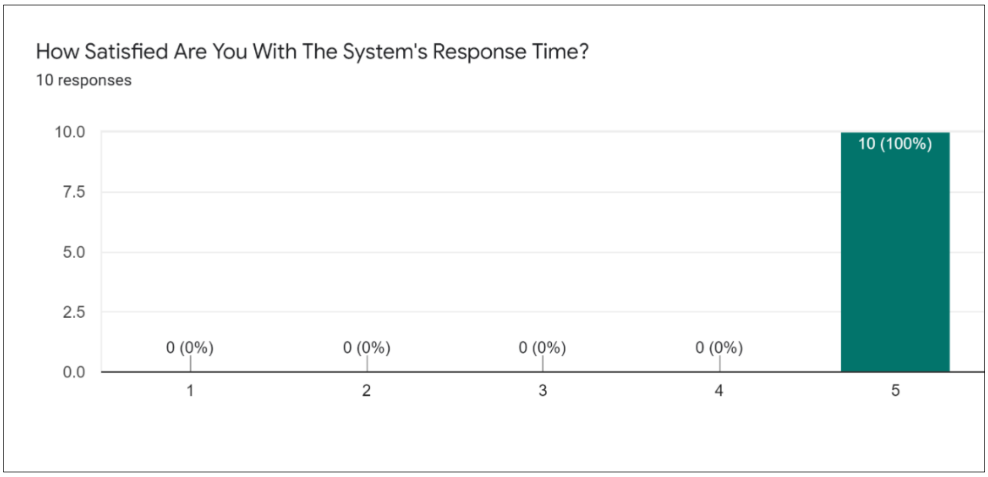

Figure 16: Test Results for The System's Response Time

The bar graph in Figure 16 above shows that all the users who tested out the system rated the system's response time as very satisfactory.

5. Conclusion

In conclusion, the Real Time Face Mask Detection System is proposed to make the flow of students and guests smoother and to ensure everyone's health and safety by detecting the presence of face mask on their face. The proposed system can hopefully improve the preventive measures taken during the pandemic and reduces the risk of overcrowding at an establishment's entrance, thereby minimizing the exposure to the COVID-19 disease in the workplace.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix

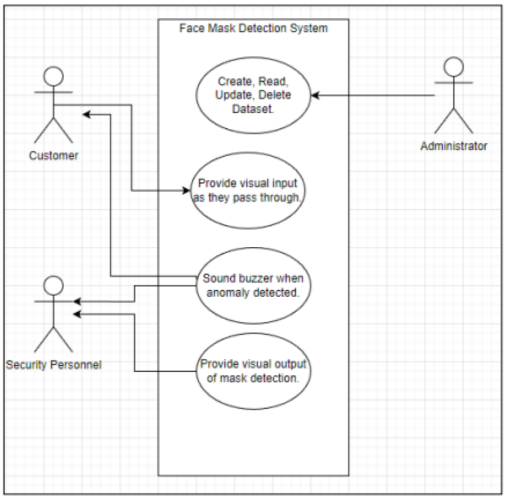

Figure 1: Use case diagram

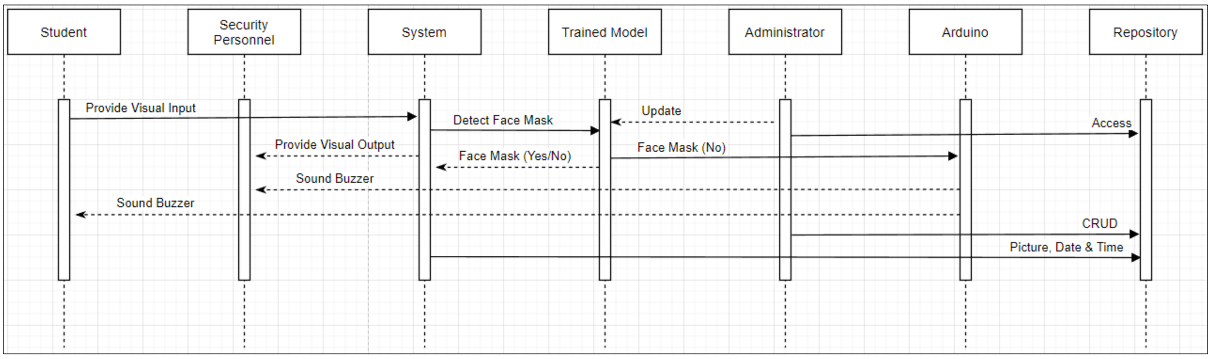

Figure 2: Sequence diagram

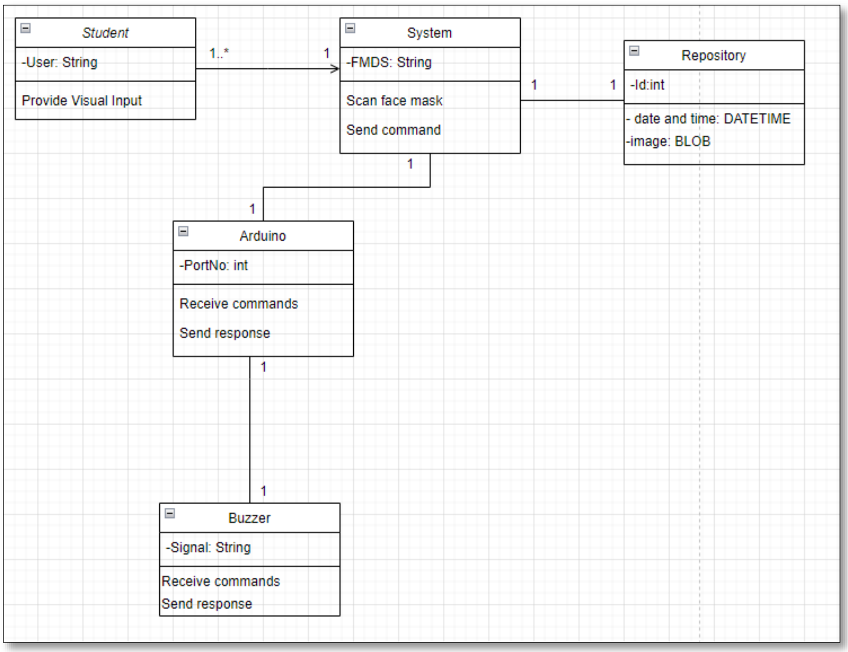

Figure 3: Class Diagram

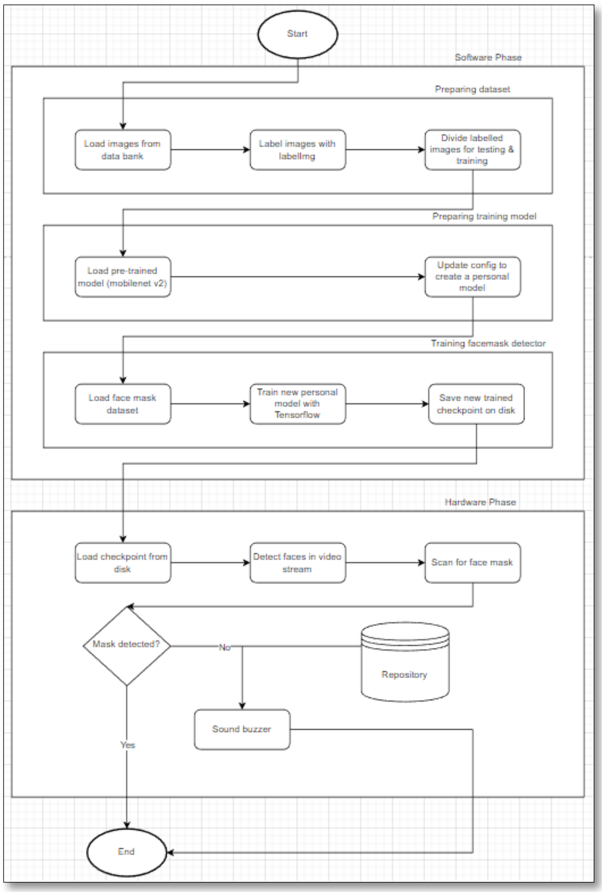

Figure 4: Flowchart diagram

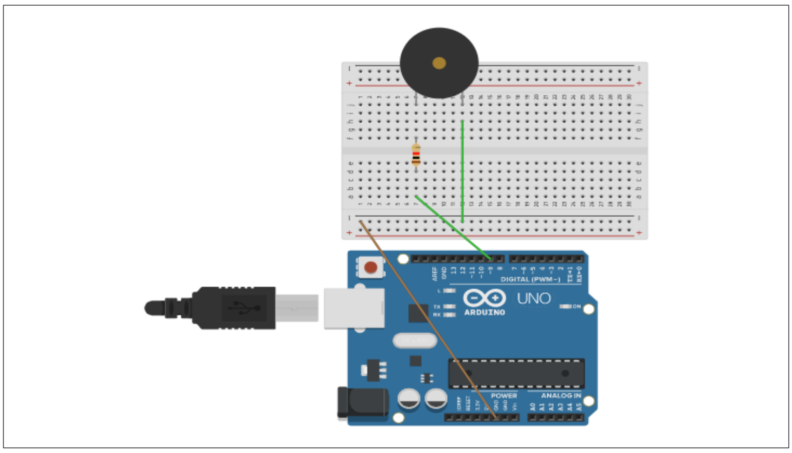

Figure 5: Design of Arduino Uno connected with buzzer on solderless breadboard

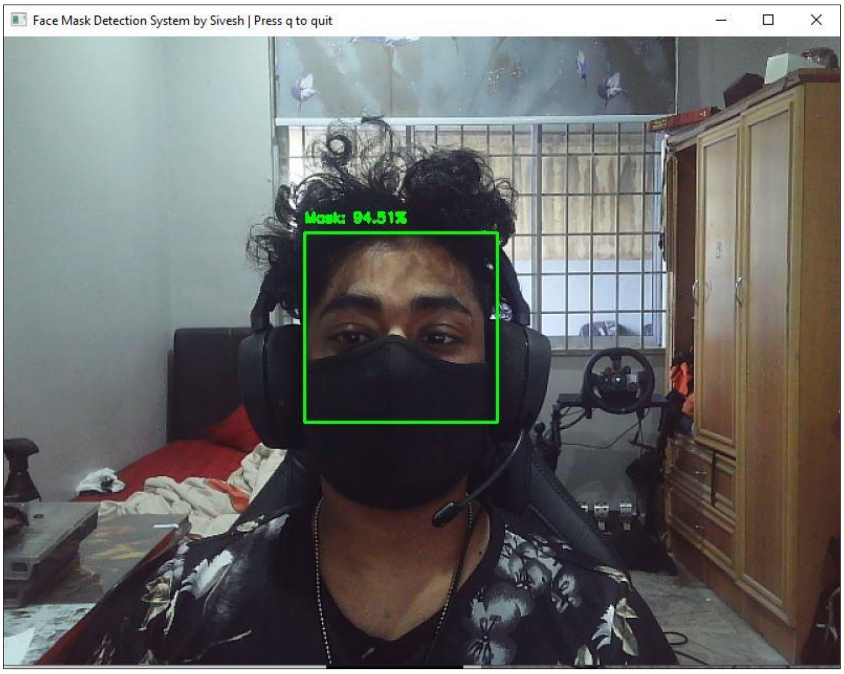

Figure 6: System interface when it detects a person with mask

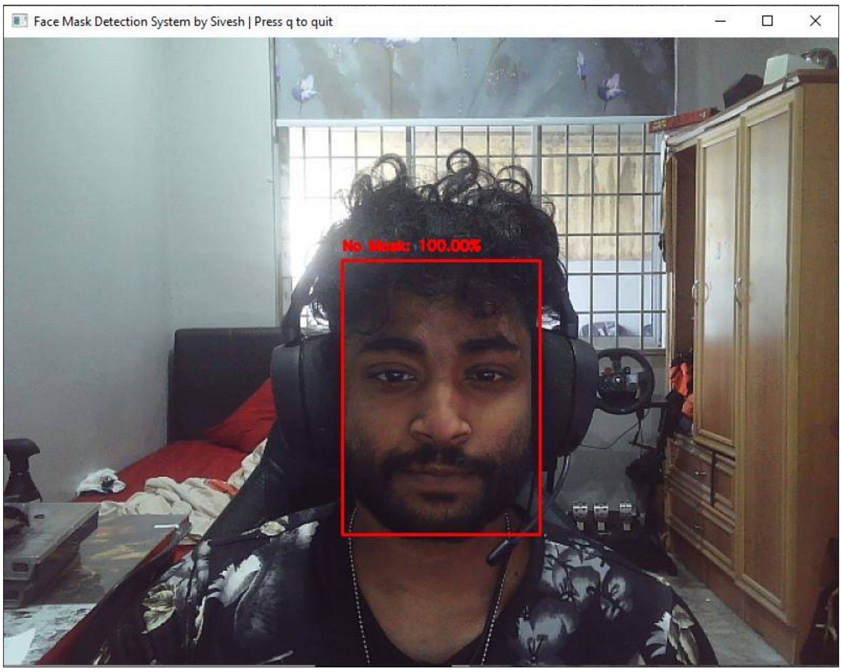

Figure 7: System interface when it detects person without mask

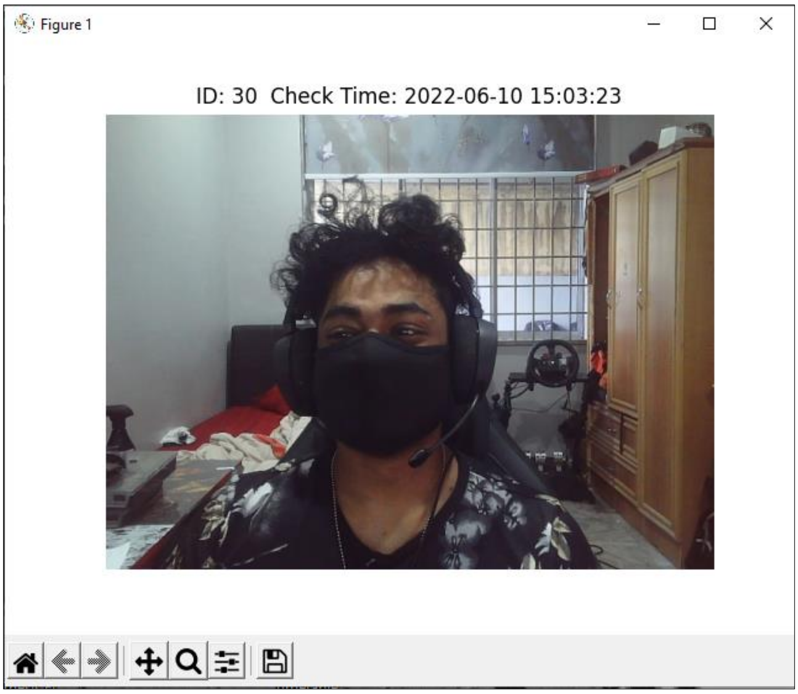

Figure 8: Database interface where it shows the time and date when the system detects an anomaly

```
        # only make predictions if at least one face was detected
        if len(faces) > 0:
                # for faster inference we'll make batch predictions on all
                # faces at the same time rather than one-by-one predictions
                # in the above `for` loop
                faces = np.array(faces, dtype="float32")
                preds = maskNet.predict(faces, batch_size=32)

        # return a 2-tuple of the face locations and their corresponding
        # locations
        return (locs, preds)

# load our serialized face detector model from disk
prototxtPath = os.path.join(os.getcwd(), 'face_detector', 'deploy.prototxt')
weightsPath = os.path.join(os.getcwd(), 'face_detector', 'res10_300x300_ssd_iter_140000.caffemodel')
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk
maskNet = load_model("mask_detector.model")

# initialize the video stream
vs = VideoStream(src=0).start()
```

Figure 9: Code segment for importing data model and setting up video stream

```
# loop over the frames from the video stream
while True:
        # grab the frame from the threaded video stream and resize it
        # to have a maximum width of 800 pixels
        frame = vs.read()
        frame = imutils.resize(frame, width=800)

        # detect faces in the frame and determine if they are wearing a
        # face mask or not
        (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

        # loop over the detected face locations and their corresponding
        # locations
        for (box, pred) in zip(locs, preds):
                # unpack the bounding box and predictions
                (startX, startY, endX, endY) = box
                (mask, withoutMask) = pred

                # determine the class label and color we'll use to draw
                # the bounding box and text
                label = "Mask" if mask > withoutMask else "No Mask"
```

Figure 10: Code segment for face mask detection and labelling

```
#view all the data
if db.view_all():
     records = db.view_all()
     for record in records:
        # Decode the image
        img = cv2.imdecode(np.frombuffer(record[1], np.uint8), -1)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
         # showing image
        plt.imshow(img)
        plt.axis('off')
        plt.title("ID: " + str(record[0]) + "  Check Time: " + str(record[2]))
        plt.show()
else:
     print("No data found")
```

Figure 11: Code segment for viewing all data in the repository

References

[1] Hussain. (2021). IoT and Deep Learning Based Approach for Rapid Screening and Face Mask Detection for Infection Spread Control of COVID-19. Applied Artificial Neural Networks, 11(8), 3495. Retrieved from https://doi.org/10.3390/app11083495

[2] Lina Draudviliene, A. M. (2017). Assessment of quantitative and qualitative characteristics of ultrasonic guided wave phase velocity measurement technique. Retrieved from https://link.springer.com/article/10.1007/s10921-017-0404-x

[3] World Health Organization. (2020, December 1). Coronavirus disease (COVID-19) advice for the public: When and how to use masks. Retrieved from World Health Organization: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/when-and-how-to-use-masks

[4] Jauhari, A. (2021). Detection system of facial patterns with masks in new normal based on the Viola Jones method. Journal of Physics: Conference Series, 2097. doi:10.1088/1742-6596/1836/1/012035

[5] Kurlekar. (2021). Face mask detection system using deep learning. Turkish Journal of Computer and Mathematics Education, 12(7), pp. 1327 - 1332. Retrieved from https://doi.org/10.17762/turcomat.v12i7.2845

[6] Hussain. (2021). IoT and Deep Learning Based Approach for Rapid Screening and Face Mask Detection for Infection Spread Control of COVID-19. Applied Artificial Neural Networks, 11(8), 3495. Retrieved from https://doi.org/10.3390/app11083495

[7] Nguyen, G. D. (2019). Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review, 52*(1), 77-124. doi:10.1007/s10462-018-09679-z

[8] Someshwar, D., Bhanushali, D., Chaudhari, V., & Nadkarni, S. (2020). Implementation of Virtual Assistant with Sign Language using Deep Learning and TensorFlow. *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (p. 600). Coimbatore: IEEE. doi:10.1109/ICIRCA48905.2020.9183179

[9] Mota Romero, E., Puente Fernández, D., Rodríguez Pertíñez, C., Árbol Fernández, G., Moreno Guerrero, S., & Montoya Juárez, R. (2022). Inadequate human resources, equipment, and training: A qualitative assessment of the objectives of the NUHELP end-of-life care programme in the context of the COVID-19 pandemic. Palliative Medicine, 026921632211030. https://doi.org/10.1177/02692163221103099

[10] Lina Draudviliene, A. M. (2017). Assessment of quantitative and qualitative characteristics of ultrasonic guided wave phase velocity measurement technique. Retrieved from https://link.springer.com/article/10.1007/s10921-017-0404-x