Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5626764, 13 pages
https://doi.org/10.1155/2022/5626764

Research Article

Autonomous Robots for Deep Mask-Wearing Detection in Educational Settings during Pandemics

Huma Zia,[1] Marah Alhalabi,[1] Maha Yaghi,[1] Amer Barhoush,[1] Omar Farag,[1] Mohammad Alkhedher,[2] Adel Khelifi,[3] Ahmed M. A. Ibrahim,[4] and Mohammed Ghazal[1]

[1] *Electrical, Computer and Biomedical Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi 59911, UAE*
[2] *Mechanical Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi 59911, UAE*
[3] *Computer Science and Information Technology Department, College of Engineering, Abu Dhabi University, Abu Dhabi 59911, UAE*
[4] *Mansoura University, Mansoura, Egypt*

Correspondence should be addressed to Mohammed Ghazal; mohammed.ghazal@adu.ac.ae

Received 3 March 2022; Revised 3 July 2022; Accepted 8 July 2022; Published 27 July 2022

Academic Editor: Xiaohui Yuan

The COVID-19 pandemic has severely impacted various aspects of life, where countries closed their borders, and workplaces and educational institutions shut down their premises in response to lockdowns. This has adversely affected the lives of everyone, including millions of students worldwide, socially, mentally, and physically. Governments and educational authorities worldwide have taken preventive measures, such as social distancing and mask wearing, to control the spread of the virus. This paper proposes an AI-powered autonomous robot for deep mask-wearing detection to enforce proper mask wearing in educational settings. The system includes (1) Simultaneous Localization and Mapping framework to map and navigate the environment (i.e., laboratories and classrooms), (2) a multiclass face mask detection software, and (3) an auditory system to identify and alert improper or no mask wearing. We train our face mask detector using MobileNetV2 architecture and YOLOv2 object detector classification. The results demonstrate that our robot can navigate an educational environment while avoiding obstacles to detect violations. The proposed face mask detection and classification subsystem achieved a 91.4% average precision when tested on students in an engineering laboratory environment.

1. Introduction

The COVID-19 epidemic, which the World Health Organization (WHO) has labelled a global pandemic, has severely impacted people's lives. This highly infectious disease has caused a rapid increase in COVID-19 incidents around the world and triggered the need for immediate countermeasures [1]. Countries enforced strict laws and regulations to reduce the transmission of the virus and prevent its spread [2], especially following the lift of the nationwide lockdown. For instance, some counties have made wearing face masks in public mandatory [3, 4], including the United Arab Emirates (UAE), as it is an efficient way to limit the spread of the virus [5, 6].

The pandemic disrupted teaching and learning in schools and universities. It has been necessary for educators, students, institutions, and parents to adapt, implement measures, and make optimal use of the resources, technologies, and instructional methodologies that are currently accessible [7]. Many universities in the UAE have already implemented alternative educational methods such as online, hybrid, and blended learning, making teaching and learning more adaptable and accessible to students' needs. While blended learning incorporates elements of online course delivery, alternative approaches include the hybrid model, which combines online course delivery with in-person sessions. Additionally, the hybrid approach in some universities includes the need for students to attend online and other face-to-face lab sessions and assessments while adhering to the COVID-19 guidelines of face mask wearing and social distancing. However, the lack of awareness and failure to comply with such rules allows for the disease's unrestricted spread. Currently, the implementation techniques for mask enforcement are primarily human-monitored, making them challenging to enforce successfully in highly populated venues [8].

Researchers and manufacturers constantly aim to develop systems that can aid in pandemic preventive measures, specifically robotics. With a total market share of 27 billion USD, the prominence of service robots at this day and age is unwavering [9], especially with estimations pointing towards the imminent integration of service robots in our daily lives [10]. Moreover, the pandemic has encouraged the development of systems designed to identify recurring violations relating to social distancing measures such as mask detection or the distance between two individuals. Such systems or robots can also approach the violators and notify them to adhere to the set measures [11]. Furthermore, there is a strong emergence of tracking systems that can track social distancing metrics. Most of these systems are efficient, accurate, and easily applicable to any CCTV surveillance camera in any environment, regardless of visual challenges such as occlusion [12]. In addition to service robots and detection systems, the pandemic has also triggered the development of tracing applications used on smartphones; such applications allow the identification of any person in contact with an infected person prior to them knowing about the infection [13]. Whereas privacy concerns regarding such applications have risen, they remain effective and highly used in countries such as the UAE [14]. Such developments in the field will aid us in developing an automated mask detection monitoring system capable of detecting mask violations and notifying the violator. Various factors make face mask detection algorithms challenging. This includes various mask types, differing degrees of obstruction, varying angles, implementation of detection models on machines with limited computing capabilities, poor-quality images, facial expressions, and lack of real-world image database [4].

Deep learning artificial intelligence approaches are being widely used for face mask detection algorithms. Authors of [15] developed a hybrid model that consists of a feature extraction component using Resnet50 followed by a classification component using Support Vector Machine (SVM). The AI model was trained and tested on three different datasets and achieved a reasonable accuracy. Another implementation based on deep learning face mask detection is presented in [2], where the authors have used the YOLOv3 model along with a novel data augmentation technique to detect face masks. Their data augmentation methodology involves filtering images through greyscale and Gaussian blurring. In [5], the authors utilize YOLOv4 to detect whether pedestrians adhere to the rules of face mask wearing or not, especially at night time. Alok et al. [16] proposed CNN and VGG16 model to detect people not wearing a mask. Their work utilizes data augmentation, normalization, and transfer learning to build the model. They train their model on Google Colab using Tensorflow. For the dataset, the public domain Simulated Masked Face Dataset (SMFD) is used. The dataset included 1315 images, 657 for no mask and 658 with mask, which were included for the training set, 142 for validation, and 194 for testing. Other studies that deploy deep learning-based models for the detection of face masks are presented in [4, 8, 17, 18].

In [19], the authors presented a face mask-wearing condition identification method that addresses a classification problem for three categories based on unconstrained 2D facial pictures by merging image super-resolution and classification networks (SRCNet). They train and test the model using the Medical Masks Dataset, which includes images of people without face masks, people wearing face masks incorrectly, and others wearing them correctly. However, one of the study's limitations is that the dataset utilized is quite limited, limiting the study's ability to cover all postures and situations.

A two-stage real-time face mask detection and classification is proposed in [3]. In the face detection step, the detector filters out nonfaces and divides the facial areas into two groups based on their location on the face. The authors trained and tested both models using benchmark datasets. Thus, the proposed detector performs well and has a good level of accuracy compared to other detectors.

Authors of [20] developed a real-time face mask detection model. They use a Haar cascade classifier and YOLOv3 for face and mask detection, respectively. This system has been built as a safety solution for office entrance. The DL model has been trained on 7000 samples, 5000 training, 1000 validation, and 1000 testing. The algorithm achieved up to 83% precision. This proposed algorithm can work in real time with 30fps, and it uses image enhancement techniques to improve accuracy.

Several approaches proposed in the literature include robots designed to navigate and automate the process of face mask detection autonomously. In [21], a TurtleBot3 robot was used alongside a LiDAR sensor for obstacle detection. The sensor scans and maps the environment using a 3D visualization software available in the ROS environment. Authors of [22] designed and built a robot that will assist authorities in preventing the transmission of COVID-19 and its outbreaks. With biosensors and temperature detectors, the robot can check for the virus. It also deploys a deep learning artificial intelligence-based face mask detection. In [23], the authors built a mobile robot called Thor that classifies people wearing masks from those who are not. This robot is trained using ResNet50 to primarily detect unmasked people and provide them with a mask to limit the virus spread. The model's accuracy was reasonable despite the challenging nature of the dataset.

To the best of our knowledge, none of the existing mask-wearing systems has catered primarily for students' health nor tested in educational settings. This research work proposes an AI-powered self-driving robot to enforce student mask wearing through automated face mask detection and classification techniques to address these limitations. We design our robot to autonomously navigate an educational environment using Simultaneous Localization and Mapping (SLAM), especially in laboratories and classrooms, while avoiding obstacles. The face mask detection and classification system uses MobileNetV2 and YOLOv2 to detect students with or without face masks and classify them into three categories. The system uses bilingual auditory alerts to notify students who are not wearing their masks or wearing their masks incorrectly. This proposed system aims to limit the spread of the COVID-19 virus in educational institutions, especially in laboratories and classrooms.

The remainder of the paper is structured as follows. The Materials and Methods section presents the methodology of our proposed research work, while the Results and Discussion section discusses the results of our proposed system and subsystems. We conclude our work in the Conclusions section.

2. Materials and Methods

2.1. Proposed System Overview. We propose an AI-powered self-driving robot to enforce student mask wearing that consists of face mask detection and autonomous navigation subsystems as shown in Figure 1. The autonomous navigation subsystem consists of a TurtleBot3 robot and a LiDAR sensor. This subsystem allows the robot to navigate the educational environment autonomously, determining the optimum path, mapping its surroundings, and avoiding obstacles. The face mask detection subsystem simultaneously gets activated while navigating a laboratory or a classroom. We train a machine learning model to detect and classify students into three main categories: wearing masks, not wearing masks, and wearing masks incorrectly. The proposed system deploys auditory alerts to warn students without masks or wearing them incorrectly. Through such preventive measures, our approach ensures that students are wearing their masks correctly at all times, controlling the transmission of the disease.

2.2. Robot Design. Our autonomous robot consists of a Turtlebot3 Burger base with a built-in 360 degree LiDAR for obstacle detection, SLAM and navigation, a gyroscope, and an accelerometer. The robot is equipped with a Raspberry Pi 3, an OpenCR control board to configure and control the sensors and motors, respectively, and a battery. The robot also consists of a camera for real-time video capturing, a Bluetooth speaker for alerts, and an NVIDIA Jetson Xavier for running the face mask detection algorithm. The robot design, including the camera and other components' positions, is illustrated in Figure 2.

2.3. Autonomous Navigation Subsystem. The proposed robot navigates the educational premises autonomously using the ROS framework as illustrated in Figure 3. The robot begins by creating a static map of the surroundings using the SLAM method, fed with 360 LiDAR sensor data. We first use a joystick to drive the robot around manually. The ROS Navigation stack's Gmapping [24] SLAM method is then used to create the map of the environment accordingly, where the map's accuracy depends on the accuracy of the localization. The LiDAR sensor's odometry data and the data from the motor encoders and gyroscope are used for localization. The navigation algorithm then takes the odometry data, the LiDAR sensor stream, and the static map of the surroundings and outputs the velocity commands accordingly to the motor driver, where they are then input to the navigation stack.

While the robot is navigating a laboratory or a classroom, it may encounter various static and dynamic obstacles that may obstruct its path. The robot performs a cautious reset in the first recovery behavior, clearing the barriers identified from the local cost map. If no path can be found because the obstacles have not been cleared, the robot rotates in its current location and checks whether the obstacles have been removed. If it is still obstructed, the cost map is reset entirely by clearing all obstacles. If the robot still cannot discover a way, it will perform one final rotation in its place after clearing the cost map. The robot will then abort the mission if none of the recovery attempts succeed.

2.4. Face Mask Detection and Classification Subsystem. We train the face mask detector to identify students without masks or wearing masks incorrectly and classify them into three categories: correct, incorrect, or no mask. A Logitech C920 camera is mounted on the robot to provide real-time feedback while navigating the laboratory or a classroom setting. We use three face mask detection datasets [25–27] to develop the proposed detection model. These datasets include people wearing face masks properly, improperly, and without masks at all. Conducting many experimental tests to increase accuracy, performance, and generalization has led to 8 different datasets throughout this project. The first dataset (refer to Table 1) collected had 4,400 images. The dataset contained images of people wearing face masks correctly and was not equally distributed. The model trained based on this dataset did not satisfy our requirements, so we have increased the dataset size. The second dataset contained a total of 9,200 images. The model's average precision has improved, but it was not reliable enough. The dataset size continued to increase until we reached a point where the device used for training runs out of memory while training more than 15,200 images. To further improve our model, we increased the number of categories into three categories (correctly, incorrectly worn face masks, and without face masks). We started with a dataset size of 6,600 images (2,200 images per category). Starting with small dataset sizes is critical to test whether the model's average precision will improve after increasing the dataset size and reducing the weight on the detection models since the more trained data used, the slower the detection model will be. The dataset that built the highest precision models contained significant differences than the previous ones. The finalized dataset included more generalized images based on face angles, distances, and mask colors. Each dataset will be used to create a unique face mask detection model. We will evaluate the precision and recall of the models to find the best model

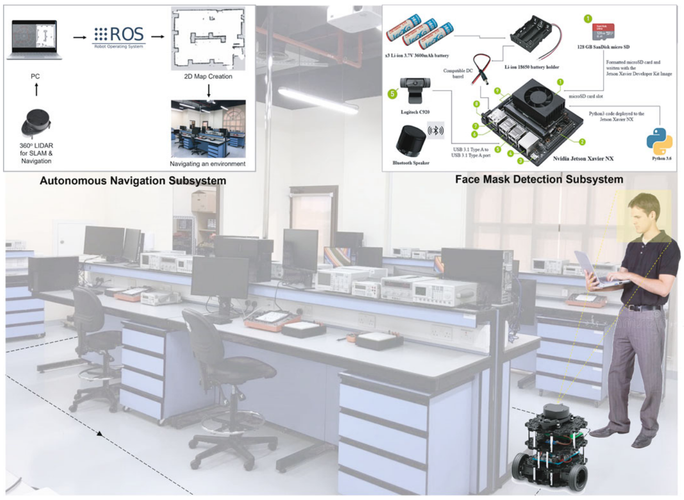

ROS
Robot Operating System
PC
2D Map Creation
360° LIDAR for SLAM & Navigation
Navigating an environment
Autonomous Navigation Subsystem
x3 Li-ion 3.7V 3600mAh battery
Li-ion 18650 battery holder
128 GB SanDisk micro SD
Formatted microSD card and written with the Jetson Xavier Developer Kit Image
Compatible DC barrel
Logitech C920
microSD card slot
Python3 code deployed to the Jetson Xavier NX
Bluetooth Speaker
USB 3.1 Type A to USB 3.1 Type A port
Nvidia Jetson Xavier NX
Python 3.6
1
2
3
4
5
6
7
8
9
Face Mask Detection Subsystem

Figure 1: System overview diagram.

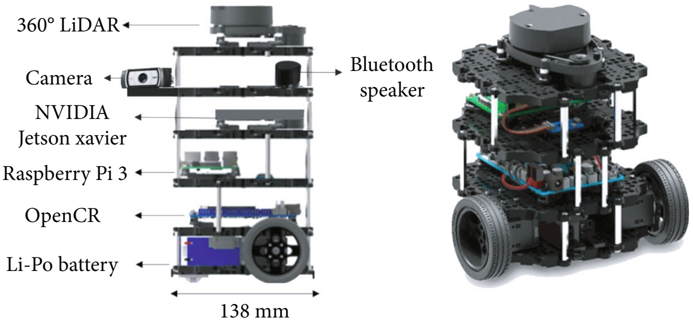


Figure 2: Robot hardware design.

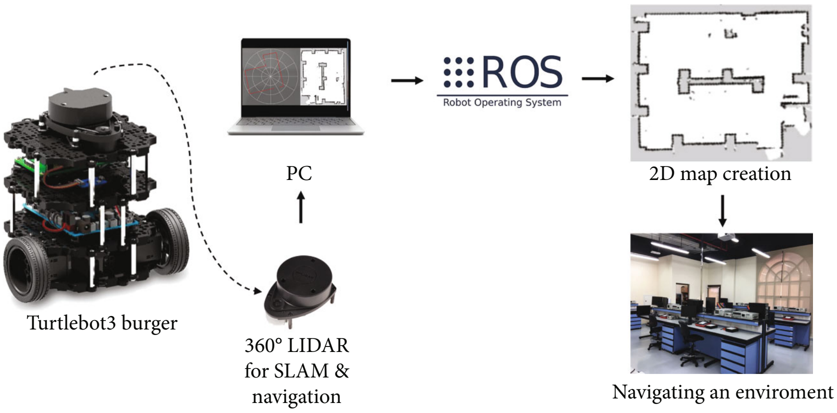


Figure 3: Autonomous navigation subsystem diagram.

performance to be used for this study. The summary of each dataset created can be presented in Table 1.

Due to the limited adequate amount of data available for training the face mask detector AI model, image augmentation is performed. Images are rotated, zoomed in and out, and shifted to generate various versions of each picture and improve accuracy [28].

We use the Caffemodel and prototxt for the implementation for the detection of facial masks. Each frame is input through a pretrained face detector model designed to identify and crop every detected face with a confidence of 70% or higher. The cropped image is then scaled to 224 x 224 pixels, RGB encoded, and inputted into the classifier along with the cropped face's X and Y coordinates. We perform face mask detection using the pretrained MobileNetV2 network, a light-weight deep CNN model, and initialize the networks with the weights of the pretrained models trained on the ImageNet dataset.

Table 1: Datasets created.

Name	Total images	Categories	Models generated
Dataset 1	15,200	Correct, incorrect	Model 1
Dataset 2	9,200	Correct, incorrect	Model 2
Dataset 3	4,400	Correct, incorrect	Model 3
Dataset 4	10,000	Correct, incorrect, without	Model 4
Dataset 5	6,600	Correct, incorrect, without	Model 5
Dataset 6	12,000	Correct, other	Model 6
Dataset 7	12,000	Incorrect, other	Model 7
Dataset 8	12,000	Without, other	Model 8

Table 2: Models created in experiment 1.

Name	Total images trained	No. of epochs	Batch size	Categories
Model 1	15,200	12	32	Correct, incorrect
Model 2	9,200	12	32	Correct, incorrect
Model 3	4,400	12	32	Correct, incorrect

We conduct three main experiments to develop a detector that can detect face masks of students' faces with high accuracy and precision. All detectors in the upcoming experiments were trained using Keras and TensorFlow2. In the first experiment, different dataset sizes with two categories only were used to generate various models for evaluation. The models created in this experiment can be shown in Table 2.

In the second experiment, different dataset sizes with three categories were used to generate models for evaluation. The models created in this experiment can be shown in Table 3.

In the third experiment, one dataset containing three categories was divided to generate three models in which one category is dominant to the other categories within each model. The dominant category contains half the dataset size while the other categories share the other half equally. The models created in this experiment can be shown in Table 4.

The output of the face mask detector includes the location of the bounding boxes for each detected face, a colored label, and the confidence score of those predictions. To further classify the images, we use the You Only Look Once (YOLO) object detector [29], shown in Figure 4 to detect cases where students are wearing a full mask, nose exposed, or chin mask and customize the bilingual alerts provided to the user.

Table 3: Models created in experiment 2.

Name	Images trained	No. of epochs	Batch size	Categories
Model 4	10,000	11	32	Correct, incorrect, without
Model 5	6,600	11	32	Correct, incorrect, without

Table 4: Models created in experiment 3.

Name	Images trained	No. of epochs	Batch size	Categories
Model 6	12,000	14	128	Correct, other
Model 7	12,000	14	128	Incorrect, other
Model 8	12,000	14	128	Without, other

The full subsystem pipeline is illustrated in Figure 5.

3. Results and Discussion

3.1. Autonomous Navigation Results. We tested the AI-powered robot in the engineering labs at Abu Dhabi University, UAE, and in various lab-like and classroom environments. Such settings include workstations, chairs, desks, and other objects. The robot successfully mapped the lab's overall shape and the static objects using the LiDAR sensor, as illustrated in Figure 6. It travelled through the goal points defined on the planner without colliding with any obstacles to arrive at the final destination. The local planner constructs a map once an obstacle appears that takes approximately 30 seconds to get around the obstacle and to the goal point.

3.2. Face Mask Detection and Classification Results

3.2.1. Face Mask Detection Results. The trained models from the face mask detection and classification subsystem were used to generate a total of 6 face mask detectors to be tested in real time. All detectors will then be compared to find the best performance detector for this research work. Table 5 shows the testing results of each trained detector in real time.

From Table 5, Detectors 4 and 6 showed the highest average precision compared to the other detectors. The real-time results of the captured testing images can further support the performance of all detectors. Figure 7 shows the detection accuracy of Detector 6 for a single student captured by the robot, which has the highest precision out of all other detectors.

In Figure 8, the robot captures 2 students in the laboratory environment in two different scenarios.

3.3. Face Mask Detection Model Evaluation 261. In this section, we evaluate the performance of both the baseline

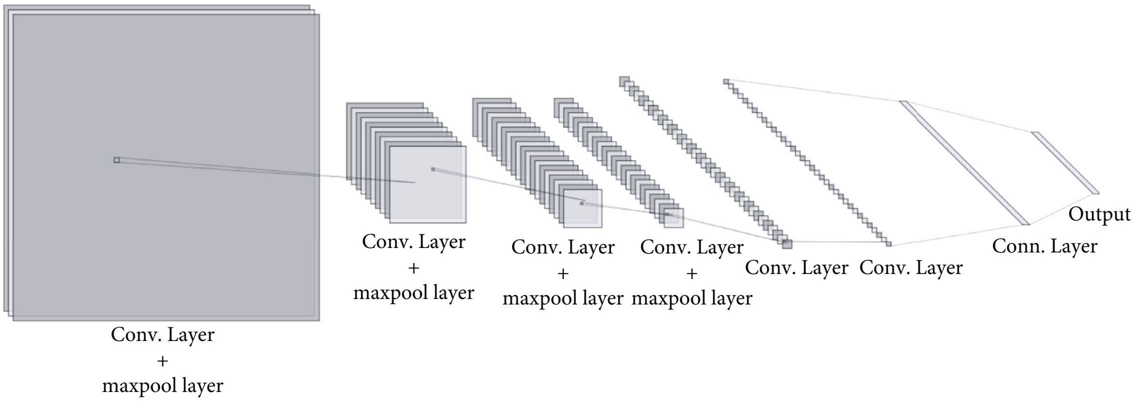

Conv. Layer + maxpool layer
Conv. Layer + maxpool layer
Conv. Layer + maxpool layer
Conv. Layer + maxpool layer
Conv. Layer
Conv. Layer
Conn. Layer
Output

Figure 4: YOLOv2 network architecture.

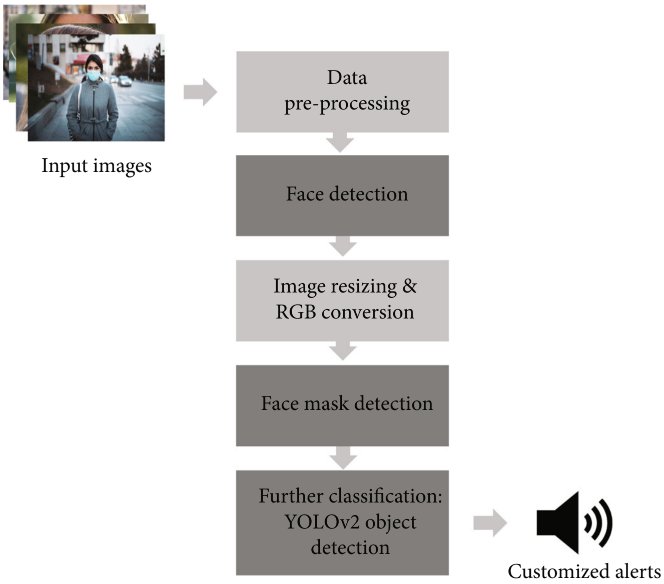


Figure 5: Face mask detection pipeline.

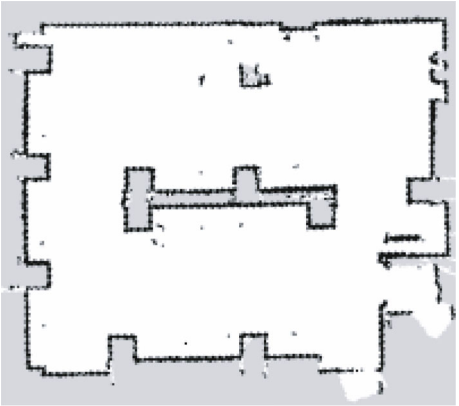


Figure 6: Map generated of the lab environment using the LiDAR sensor.

(*Detector 4*) and improved model detectors (*Detector 6*) and discuss the main differences that led to better precision and performance. To describe the model's performance, we construct a confusion matrix, as shown in Figure 9.

We primarily focus on the model's predictive ability, precision, and recall, instead of the classification time of the model. We elaborate on these two metrics as follows:

(i) *Precision* quantifies how correct the model's positive predictions are, where positives mean correctly worn masks and negatives mean incorrectly worn masks, in our case. This means that the more the model correctly classifies the label "correctly worn masks," the more precise it will be. It can be computed using Equation (1).

$$Precision = \frac{TP}{TP + FP}. \tag{1}$$

(ii) *Recall* is otherwise referred to as *Sensitivity* or *True Positive Rate (TPR)*. This measure refers to how many people wearing their mask correctly did the

Table 5: Performance of the MobileNetV2 detectors.

Name	Models used	Avg. precision	Avg. recall
Detector 1	Model 1	0.692	0.562
Detector 2	Model 2	0.624	0.511
Detector 3	Model 3	0.538	0.445
Detector 4	Model 4	0.775	0.654
Detector 5	Model 5	0.707	0.633
Detector 6	Model 6	0.914	0.654

AI model miss out of all people who wore their face masks correctly. It can be determined using Equation (2).

$$Recall = \frac{TP}{\mathrm{TP} + FN}. \quad (2)$$

Due to the nature of this research work, precision is a significant metric to report. We want to ensure that our AI model can correctly identify students who wear their masks correctly from those who do not. In other words, if the AI model predicts that the student is wearing their face mask correctly while they are not, the chances of spreading the infection increases due to those wearing the face incorrectly but not alerted.

We evaluate the proposed AI models and report on their performance in terms of their precision. We also construct Receiver-Operating Characteristic (ROC) curves using True Positive Rates, TPR, vs. False Positive Rate (FPR). To construct an ROC curve, we need to:

(1) Use the face mask detection model to produce a probability of correctly worn masks (P (correctly worn masks)) in each frame captured from the live stream camera. The total number of test instances (frames) that will be used is 100

(2) Sort the instances in descending order according to the P (correctly worn masks).

(3) Count the number of TP, FP, TN, and FN after applying a threshold to each unique P-value (correctly worn masks).

(4) Calculate the TPR using Equation (2) and FPR using Equation (3).

$$FPR = \frac{FP}{FP + TN}. \quad (3)$$

Following the previous steps, we build a table that will help us construct an ROC curve for the model. We implement the following to improve the performance of the face mask detection model:

(i) Increase the number of augmented images showing students wearing face masks correctly in varying angles

(ii) Use binary classification (one vs. all classification) and repeat the experiment three times. The first time to classify the correctly worn mask images vs. the rest, while the second time to classify the incorrectly worn face mask vs. the rest. The last experiment classifies students without a face mask vs. the rest

We train the improved face mask detection model on 100 frames and construct its ROC curve. We display the first 20 frame instances for the improved face mask detection model along with their respective positive class probabilities and confusion matrix metrics in Table 6.

The constructed ROC curve for the improved model taken from Table 6 along with the ROC curve for the base and default models can be seen in Figure 10.

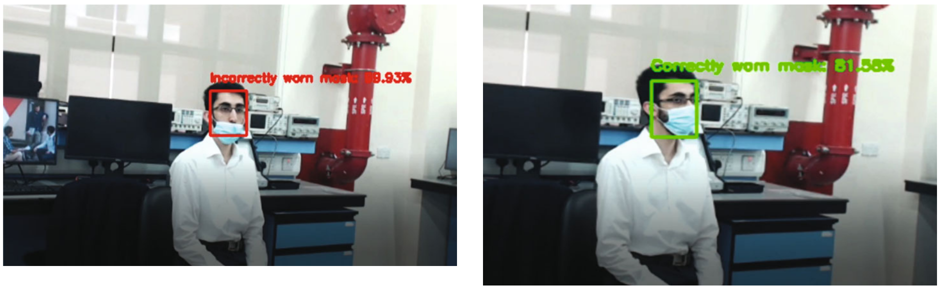
As can be seen in Figure 10, the AUC for the improved model is greater than the default classifier, which means it is more precise than the base model presented in green. We summarize the major differences between the base model initially used and the improved face mask detection model in Table 7.

We observe that the improved model's detection speed is affected by the dataset size, computing compatibility (GPU), and the camera's quality. A high-quality camera with better shutter speed and exposure will increase the performance significantly. Similarly, deploying the model on a more powerful GPU with high compatibility can improve performance.

3.4. Face Mask Classification Results. We present the outcome of the YOLOv2 face mask classification system. The test images are labelled *Full Mask*, *Nose Exposed*, or *Chin Mask*. Figure 11 demonstrates the effectiveness of our model in detecting and classifying students wearing masks from varying angles regardless of the color of the mask worn.

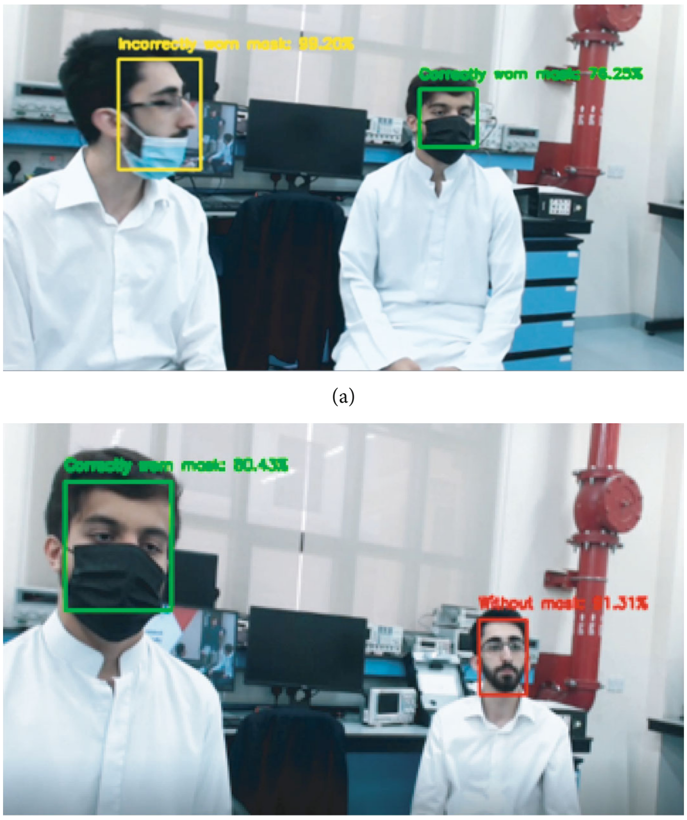
The system can also accurately classify students wearing their masks incorrectly as can be seen in Figure 12, where the student had his nose exposed in (a) and both his nose and mouth exposed in (b).

3.5. Computational Complexity and Inference Speed. The frames captured using the camera mounted on the robot go through different functions. Each function/process is unique when it comes to complexity. In addition, the more complex the process is, the more time the frame needs to be processed. Frames captured by the camera start their processing journey. At first, the frame gets resized since the models trained are based on 224x224 pixel images. After that, the spatial dimensions of the frame get extracted, and a blob gets constructed. This constructed blob will pass through the pretrained face detector to extract the confidence and face coordinates of the captured frame. Next, the frame with high confidence (face detection probability above 70%) will pass through the face mask detectors (3 in total). Each face mask detector will give a prediction based on the input frame, and the face mask detector with the



(a) (b)

Figure 7: Face Mask Detector testing accuracy for a student wearing a face mask (a) incorrectly and (b) correctly.


Incorrectly worn mask: 99.20%
Correctly worn mask: 76.25%
Correctly worn mask: 80.43%
Without mask: 81.31%

(a)

(b)

Figure 8: Face Mask Detector testing accuracy for two students in the lab, where (a) one student is wearing a face mask incorrectly while the other student is wearing it correctly and (b) one student is wearing a face mask correctly while the other one is not wearing a face mask.

		Predicted class	
		Class 1: correct	Class 2: correct
Actual class	Class 1: correct	a (TP)	b (FN)
	Class 2: incorrect	c (FP)	d (TN)

TP: True positive
FN: False negative
FP: False positive
TN: True Negative

Figure 9: Confusion matrix.

highest accuracy will select the most accurate label to label the input frame. Finally, the processed frames will be displayed using OpenCV with the corresponding face boundaries and labels. To compute the complexity of this process, we calculated the frame rate (FPS) before and after the face mask detection process. The frame rate without mask detection using real-time video capturing was 25 frames per second. Once faces and masks are detected, the frame rate drops to 10 frames per second. This means that it takes 2.5 seconds additionally per frame to get processed, detected, and labelled. The device used to obtain such results was a laptop with a 2.4 GHz CPU. Testing on GPU was

Table 6: Testing instances up to 30 frames, probability of correctly worn masks, true class, prediction, TP, FP, TPR, FPR, recall, and precision for the improved face mask detection model.

#	P (+)	True class	Prediction	TP (++)	FP (-+)	TPR (recall)	FPR	Precision
1	99.99	+	+	1	0	0.015	0	1
2	99.97	+	+	2	0	0.290	0	1
3	99.93	+	+	3	0	0.044	0	1
4	99.92	+	+	4	0	0.059	0	1
5	99.92	+	+	5	0	0.074	0	1
6	99.91	+	+	6	0	0.088	0	1
7	99.91	+	+	7	0	0.103	0	0.857
8	99.88	+	+	8	0	0.118	0	0.875
9	99.85	+	+	9	0	0.132	0	0.889
10	99.85	+	+	10	0	0.147	0	0.900
11	99.81	+	+	11	0	0.162	0	0.909
12	99.75	+	+	12	0	0.176	0	0.917
13	99.71	+	+	13	0	0.191	0	0.923
14	99.71	+	+	13	1	0.191	0.031	0.929
15	99.70	+	+	14	1	0.206	0.031	0.933
16	99.70	+	+	15	1	0.221	0.031	0.938
17	99.69	+	+	16	1	0.235	0.031	0.941
18	99.65	+	+	17	1	0.250	0.031	0.944
19	99.63	+	+	18	1	0.265	0.031	0. 947
20	99.62	+	+	19	1	0.279	0.031	0.950
21	99.57	+	+	20	1	0.294	0.031	0.952
22	99.51	+	+	21	1	0.309	0.031	0.955
23	99.36	+	+	22	1	0.324	0.031	0.957
24	99.12	+	+	23	1	0.338	0.031	0.958
25	99.12	+	+	24	1	0.353	0.031	0.960
26	99.08	+	+	25	1	0.368	0.031	0.962
27	99.00	+	+	26	1	0.382	0.031	0.963
28	98.93	+	+	27	1	0.397	0.031	0.964
29	98.91	+	+	28	1	0.412	0.031	0.966
30	98.86	+	+	29	1	0.426	0.031	0.967

inapplicable since the GPU capability was below 3.0. As for the Jetson Xavier NX, the compute capability is 7.2, which can run the detection process in real time with a frame rate of 8 frames per second. The computing device plays a critical role in speeding up the detection process. Additionally, compatible versions of TensorFlow 2, CUDA, and cuDNN can accelerate deep learning significantly. cuDNN provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers. [30]

3.6. Convolutional Neural Network (CNN) Architectures. Inference speed and mean average precision (mAP) are critical in CNNs. Choosing the correct CNN to train object detection models will enhance the model's performance. The depth and types of layers (such as convolution, batch normalization, and rectified linear unit (ReLU) activation) are the main characteristics of CNNs. The deeper the CNN is, the more precise the trained model will be. However, deep CNNs are heavier than CNNs with few layers, which means they will be much slower. Precision comes at a cost, so choosing a CNN that can be both precise and fast is very important. For this project, the model is pretrained on a CNN with enough speed to achieve exact results. Moreover, MobileNetV2 SSD CNN architecture was used to balance precision and speed. To further support our choice, we present in Table 8 a comparison between some of the different pretrained CNNs based on speed and mAP. [31]

3.7. Enhancing the Generalization of the Proposed Model. The generalization of deep learning models using data augmentation helps ensure model optimization. Data augmentation is a technique to increase the number of training samples by modifying the already existing data. In [28], a full-stage data augmentation framework is proposed to improve the accuracy of deep CNN for image classification. Two benchmarks CIFAR-10 and CIFAR-100, based on coarse-grained and fine-grained tiny images dataset, were used in the study. The experimental results for the study on the coarse-grained dataset CIFAR-10 and dataset

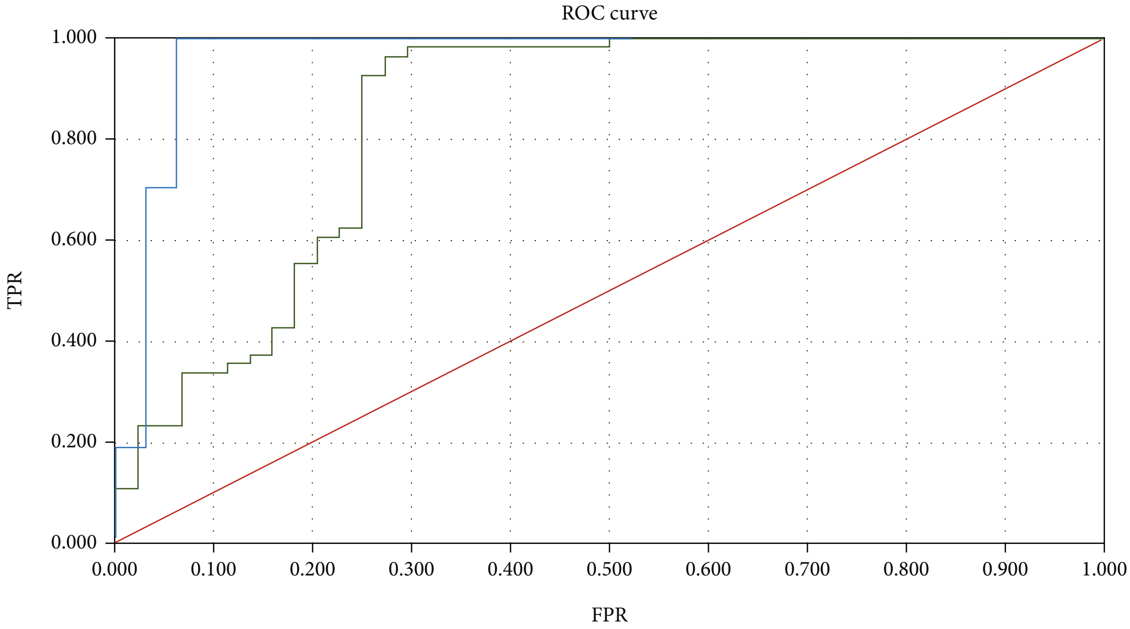


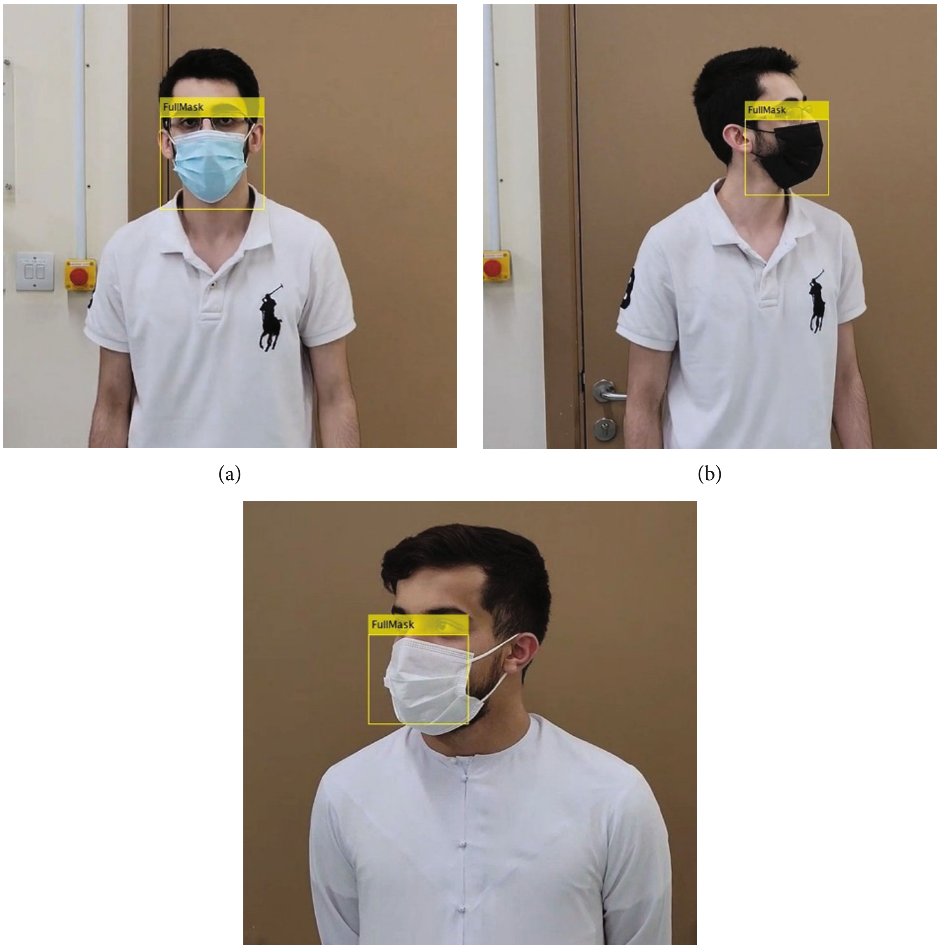
Figure 10: Comparison of the ROC curves of the improved face mask detection performance model shown in blue vs. the baseline model shown in green vs. the default classifier in red.

Table 7: Differences between the base and improved models.

Differences	Base model	Improved model
Number of models	1 model	3 models
Categories	Correct, incorrect, and without	Correct, incorrect, and without
Dataset size	10,000 images	12,000 images
Mask colors	Bright mask colors only	Bright and dark mask colors
Face angles	Facing the camera directly	Face angles between 45 and 90 degrees
Frame rate	12 FPS	10 FPS
Inference time	83 ms	100 ms
Avg. precision	0.775	0.914
Avg. recall	0.654	0.654

CIFAR-100 demonstrated 93.41% and 70.22% accuracy, respectively. In another study [32], deep transfer learning method is used for facial diagnosis from uncontrolled 2D face images of various diseases like beta-thalassemia, hyperthyroidism, Down syndrome, and leprosy with a relatively small dataset of 350 face images. The experiments showed 90% accuracy and demonstrate the effectiveness of CNN for feature extraction of small datasets but emphasize the need for data augmentation to increase the ability of the model to detect more diseases with higher accuracy to perform facial diagnosis. This research [12] proposes real-time AI platform for people detection and social distancing measure, and social distancing classification of individuals using thermal camera. YOLO-v4-Tiny is used for model development, which is a lighter version of YOLO-v4. Two datasets were used of 1000 and 950 images, respectively. The dataset was collected from different sources on the Internet of people sneaking, walking, and running in different body positions. The final algorithm achieved up to 95% accuracy and was deployed in Nvidia Jetson devices.

Based on the literature to better optimize our model, we used utilized data generalization for our developed algorithm. Our model's ability to adapt to new unseen data relies on different factors, such as face angles, distance, and mask color, which are critical in improving the model's generalization ability. Moreover, numerous experimental tests were conducted to enhance the models' generalization ability. In the first few experiments, the detection models struggled to detect face masks at sharp angles between 45 and 90 degrees, so we have improved our dataset by increasing the images of people facing the camera at an angle to reach about 60% of the total dataset. Furthermore, we noticed in our tests that the model's accuracy against masks of dark colors is low. To avoid increasing the number of images containing people wearing a dark face mask (such as black or brown), we have used data augmentation to grayscale a considerable portion

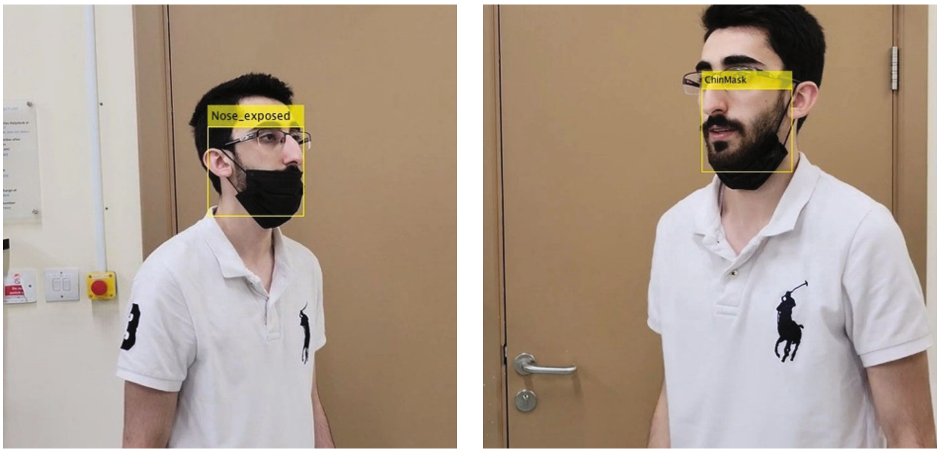


(a)

(b)

(c)

Figure 11: Full face mask classification using YOLOv2 object detection for a student wearing (a) blue mask and facing forward, (b) black mask and facing to the left, and (c) white mask and facing to the right.



(a)

(b)

Figure 12: (a) Nose exposed and (b) mask worn under the chin incorrectly detected with the proposed model.

of our training dataset. Hence, the model now detects dark and light-colored masks more accurately. In addition, it is well-known that faces become unrecognizable at long distances. Likewise, face masks can be difficult to detect from a distance. The pixel ratio between people's faces and face masks will change significantly. Two steps were taken to

Table 8: Differences between some COCO pretrained models.

Model name	Speed	mAP	Outputs
SSD MobileNet v2 320x320	19	20.2	Boxes
SSD MobileNet V1 FPN 640x640	48	29.1	Boxes
CenterNet HourGlass104 512x512	70	41.9	Boxes
EfficientDet D1 640x640	54	38.4	Boxes
SSD ResNet50 V1 FPN 640x640 (RetinaNet50)	46	34.3	Boxes
Faster R-CNN Inception ResNet V2 640x640	206	37.7	Boxes
SSD ResNet101 V1 FPN 640x640 (RetinaNet101)	57	35.6	Boxes

solve this limitation. In the first step, we included images that contain multiple people shown at different distances while wearing face masks correctly and not. In the second step, we have used data augmentation to apply "zoom-out" on the training data. With these two solutions, we have managed to sort out the distance issue. The model can now detect face masks from distances that can reach 6 meters.

4. Conclusions

In conclusion, we propose an AI-powered self-driving robot to enforce student mask wearing in educational settings during pandemics. We design and build the robot to navigate and map lab and classroom environments autonomously. Simultaneously, the face mask detection and classification system can identify students wearing masks from those wearing masks incorrectly or without a mask at all. Our bilingual and customized auditory system alerts students with no masks or incorrectly wearing their masks. We propose a mask-wearing robot as a solution to prevent the spread of the disease in educational premises, primarily laboratories and classrooms. Our face mask detection system is trained on a dataset based on 3 Kaggle datasets, including various images of people wearing face masks properly with different colors, not wearing face masks properly, and people with no mask. We train the model using MobileNetV2 architecture and classify the face masks correctly, incorrectly, or not wearing a mask. The training process resulted in different face mask detectors for real-time performance and precision testing. We use the YOLO object detector to classify the images into students wearing full masks, nose exposed, or chin masks. The face mask detector with three integrated models showed the highest performance and precision (77.5%) of all other face mask detectors. The improved performance detector had more images of students facing 90-degree angles. In addition, the better performance detector had three models in which each model had a dominant category (Correct, Incorrect, and Without) and achieved a precision of 91.4%. The overall system can operate for 2 hours and be extended using higher-capacity batteries. We tested the proposed approach in the lab and concluded that it efficiently alerts students not wearing a mask or wearing it incorrectly while navigating the environment. The system's limitations include small obstacles that are not in the LiDAR's range of vision, which can be overcome by using ultrasonic sensors. Moreover, to reduce camera blur and jitter and increase the prediction accuracy, we have programmed the robot to stop when detecting a face, capture an image, and continue moving.

Data Availability

The face mask detection datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Authors' Contributions

Marah Alhalabi, Maha Yaghi, and Amer Barhoush contributed equally to this work.

Acknowledgments

The authors would like to thank Sohrab Setoodeh and Zaid Mohammad for their roles in the implementation and testing. This research was funded by the Office of Research and Sponsored Programs in Abu Dhabi University with a grant number [19300578].

References

[1] R. Salgotra, I. Rahimi, and A. H. Gandomi, "Artificial intelligence for fighting the COVID-19 pandemic," in *Humanity Driven AI*, pp. 165–177, Springer, Cham, 2022.

[2] M. Prusty, V. Tripathi, and A. Dubey, "A novel data augmentation approach for mask detection using deep transfer learning," *Intelligence-Based Medicine*, vol. 5, article 100037, 2021.

[3] M. Putro, D. Nguyen, and K. Jo, "Real-time multi-view face mask detector on edge device for supporting service robots in the COVID-19 pandemic," *Intelligent Information and Database Systems*, vol. 12672, 2021.

[4] E. Mbunge, S. Simelane, S. Fashoto, B. Akinnuwesi, and A. Metfula, "Application of deep learning and machine learning models to detect COVID-19 face masks - a review," *Sustainable Operations and Computers*, vol. 2, pp. 235–245, 2021.

[5] Z. Cao, M. Shao, L. Xu, S. Mu, and H. Qu, "MaskHunter: real-time object detection of face masks during the COVID-19

pandemic," *IET Image Processing*, vol. 14, no. 16, pp. 4359–4367, 2020.

[6] V. Cheng, S. Wong, V. Chuang et al., "The role of community-wide wearing of face mask for control of coronavirus disease 2019 (COVID-19) epidemic due to SARS-CoV-2," *Journal of Infection*, vol. 81, no. 1, pp. 107–114, 2020.

[7] R. Vijayan, "Teaching and learning during the COVID-19 pandemic: a topic modeling study," *Education in Science*, vol. 11, no. 7, p. 347, 2021.

[8] R. Katari, "A comparative analysis of variant deep learning models for COVID-19 protective face mask detection," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 6, pp. 2841–2848, 2021.

[9] J. A. Gonzalez-Aguirre, R. Osorio-Oliveros, K. L. Rodríguez-Hernández et al., "Service robots: trends and technology," *Applied Sciences*, vol. 11, no. 22, p. 10702, 2021.

[10] S. H. Shah, O. M. Steinnes, E. G. Gustafsson, and I. A. Hameed, "Multi-agent robot system to monitor and enforce physical distancing constraints in large areas to combat COVID-19 and future pandemics," *Applied Sciences*, vol. 11, no. 16, p. 7200, 2021.

[11] M. Mende, M. Scott, J. van Doorn, D. Grewal, and I. Shanks, "Service robots rising: how humanoid robots influence service experiences and elicit compensatory consumer responses," *Journal of Marketing Research*, vol. 56, no. 4, pp. 535–556, 2019.

[12] S. Saponara, A. Elhanashi, and Q. Zheng, "Developing a real-time social distancing detection system based on YOLOv4-tiny and bird-eye view for COVID-19," *Journal of Real-Time Image Processing*, vol. 19, no. 3, pp. 551–563, 2022.

[13] E. Hernández-Orallo, C. T. Calafate, J. C. Cano, and P. Manzoni, "Evaluating the effectiveness of COVID-19 Bluetooth-based smartphone contact tracing applications," *Applied Sciences*, vol. 10, no. 20, p. 7113, 2020.

[14] D. Storm van Leeuwen, A. Ahmed, C. Watterson, and N. Baghaei, "Contact tracing: ensuring privacy and security," *Applied Sciences*, vol. 11, no. 21, p. 9977, 2021.

[15] M. Loey, G. Manogaran, M. Taha, and N. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement*, vol. 167, article 108288, 2021.

[16] A. Negi, K. Kumar, P. Chauhan, and R. S. Rajput, "Deep neural architecture for face mask detection on simulated masked face dataset against covid-19 pandemic," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 595–600, Greater Noida, India, 2021.

[17] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustainable Cities and Society*, vol. 66, article 102692, 2021.

[18] M. Almghraby and A. Elnady, "Face mask detection in real-time using MobileNetv2," *International Journal of Engineering and Advanced Technology*, vol. 10, no. 6, pp. 104–108, 2021.

[19] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," *Sensors*, vol. 20, no. 18, p. 5236, 2020.

[20] T. Q. Vinh and N. T. Anh, "Real-time face mask detector using yolov3 algorithm and haar cascade classifier," in *2020 International Conference on Advanced Computing and Applications (ACOMP)*, pp. 146–149, Quy Nhon, Vietnam, 2020.

[21] L. Galtarossa, "Obstacle avoidance algorithms for autonomous navigation system in unstructured indoor areas," October 2018, https://bit.ly/2Sla6AV.

[22] S. Senhaji, S. Faquir, and M. Ouazzani Jamil, "Towards robotics and artificial intelligence for the prevention of Covid 19 pandemic," *E3S Web of Conferences*, vol. 229, p. 01035, 2021.

[23] S. Snyder and G. Husari, "Thor: a deep learning approach for face mask detection to prevent the COVID-19 pandemic," in *SoutheastCon 2021*, Atlanta, GA, USA, 2021.

[24] Base_LOCAL_PLANNER-ROS Wiki, *Wiki.ros.org*, 2021, http://wiki.ros.org/base_local_planner.

[25] Face Mask Detectionhttps://www.kaggle.com/andrewmvd/face-mask-detection.

[26] Face Mask Detection Datasethttps://www.kaggle.com/omkargurav/face-mask-dataset.

[27] Face Mask Detection 12K Images Datasethttps://www.kaggle.com/ashishjangra27/face-mask-12k-imagesdataset.

[28] Q. Zheng, M. Yang, X. Tian, N. Jiang, and D. Wang, "A full stage data augmentation method in deep convolutional neural network for natural image classification," *Discrete Dynamics in Nature and Society*, vol. 2020, 11 pages, 2020.

[29] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *2017, IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017.

[30] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," *Journal of Big Data*, vol. 8, no. 1, p. 66, 2021.

[31] A. Teoh, *TensorFlow 2 Detection Model Zoo - CoCO mAP vs Speed*, 2020, https://public.tableau.com/app/profile/aaronteoh/viz/TensorFlow2DetectionModelZoo-CoCOmAPvsSpeed/COCOmAPvsSpeedms.

[32] B. Jin, L. Cruz, and N. Gonçalves, "Deep facial diagnosis: deep transfer learning from face recognition to facial diagnosis," *IEEE Access*, vol. 8, pp. 123649–123661, 2020.