*DOI: 10.15849/IJASCA.251130.19*

*Int. J. Advance Soft Compu. Appl, Vol. 17, No. 3, November 2025*
*Print ISSN: 2710-1274, Online ISSN: 2074-8523*
*Copyright © Al-Zaytoonah University of Jordan (ZUJ)*

# Open for business, closed for crowding: SafeSpace a Smart Solution

**Shaleen Bhatnagar[1], Shilpa Ankalaki[1], Geetabai S Hukkeri[1]**

[1] Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India

e-mail: shaleen.bhatnaghar@manipal.edu
e-mail: shilpa.ankalaki@manipal.edu
e-mail: geetabai.hukkeri@manipal.edu

**Corresponding Author: Shilpa Ankalaki** (e-mail: shilpa.ankalaki@manipal.edu)

**Abstract**

***Public spaces, including shopping centres, restaurants, banks and offices, must grapple with how to offer easy access to the public while ensuring occupant safety and hygiene. We present SafeSpace, a modular smart-environment platform that gives administrators control over indoor density using appointment-based access, enforces physical distancing through proximity alerts in real-time, and verifies PPE adherence with on-device deep-learning mask detection. The system was developed using YOLOv3 and MobileNetV2 on Intel Core i5 CPU and Google Colab, and obtained 98% precision, recall, and F1-score on a real-world dataset. Real-time proximity alerts were implemented by employing YOLOv3 and ROI-based centroid distance estimation with an overall latency of 125–158 ms per frame. In addition, the simulated entry flow comparisons between walk-ins and a predetermined schedule indicated that peak occupancy was reduced by 38.24% during a 15-min interval. SafeSpace's modular design and consolidated analytics dashboard also make it leveraged for other public-space scenarios and the next generation of post-pandemic facility management and day-to-day operational safety.***

**Keywords**: *Smart-environment platform, Appointment-based occupancy management, Real-time proximity alerts, Contactless interface, Deep-learning mask detection, Public-space crowd control*

## 1 Introduction

For today's public spaces from shopping centres and restaurants to banks and office buildings it's a constant conundrum: How to stay open and inclusive while also keeping people safe and healthy. COVID-19 pandemic revealed how easily communicable diseases can spread throughout conventional facility function, along with the uncontrolled crowding and frequent touching of surfaces conducive to airborne pathogen transmission [1]. Despite advances in vaccination and treatment that have led to a decline in severity of

*Received 4 September 2025; Accepted 7 November 2025*

outcomes, person to person transmission is ongoing, especially in crowded, walk-in settings for which it is still challenging to enforce social and physical distancing [2][3].
Challenges and Opportunities for Facility Operators Apart from the need to adequately respond to a pandemic, facility operators will also be required to optimise space usage, improve visitor experience, and uphold high standards of hygiene. Reservations-based occupancy control has been effective at scaling down peak-time demand; however it is overly expensive in terms of overhead and are not well-suited for spontaneous walk-in traffic such as in food courts, lobbies, and retail shops [4]. Contactless technology such as touch-free entry systems may mitigate fomite spread but are seldom interfaced with broader crowd management architectures [5].
At the same time, the development of edge-computing and deep learning offers the possibility for real-time communication of activities like detecting mask or personal protective equipment (PPE) worn by a person directly on IoT devices, limiting the delay and respecting the privacy avoiding sending continuous video streams to local servers [6]. Environmental sensors (which measure CO2 concentration and particulates) can provide further intelligence on ventilation effectiveness and crowd density in the context of dynamic risk assessments [7].
In this paper, we introduce SafeSpace, a modular smart-environment platform designed with the objective of keeping venues “Open for Business, Closed for Crowding.” SafeSpace integrates:

- Scheduled reservations to smooth the flow of incoming individuals and prevent peak surges during walk-in periods, simulated via Python-based crowd models.
- Dynamic proximity alerts using YOLOv3 for human detection combined with centroid-based Euclidean distance checks to detect social distancing violations.
- Touch-free kiosk simulations designed to demonstrate contactless operation scenarios, supporting hygienic interaction flow (to be extended to real actuators in future work).
- On-device deep-learning mask validation at entry points using a YOLOv3-MobileNetV2 pipeline, trained and tested on real-world datasets in Google Colab and Intel Core i5 environment.

Aggregating these solutions into a unified analytics dashboard, SafeSpace offers administrators modular ‘plug-and-play’ components for real-time PPE and crowd compliance. We validate our prototype through controlled laboratory tests and entry-flow simulations, achieving 98% accuracy, precision, and recall for mask detection, with average alert latency of 125–158 ms. A crowd scheduling simulation showed a 35.9% reduction in peak occupancy compared to unscheduled entry patterns. SafeSpace thus provides a scalable and adaptable platform for present and future public-health threats, as well as for day-to-day facility-management optimization.

## 1.2 Transmission Pathways of Respiratory Pathogens in Public Spaces

In [shared] indoor air, pathogens such as SARS-CoV-2 are transmitted by three main mechanisms - aerosol (airborne) transmission, droplet transmission and fomite transmission, each having distinct mitigation strategies.

• Airborne (Aerosol) Transmission - Small particles ($\leq 5$ µm) released during normal breathing, talking, and singing can remain airborne for hours and travel further than 2 m, particularly in enclosed spaces. The strength of the evidence for aerosol transmission and for implementation of engineering controls, including increased ventilation, air filtration, and universal masking to mitigate indoor spread of COVID-19 is unequivocal, state Morawska and Cao [8].

- Droplet Transmission - Larger respiratory droplets (≥5 μm) that are formed during coughing or sneezing usually fall within 1 to 2 m from the source. Such droplets can deposit on the mucosa of other nearby persons so minimum physical distance of 2 m and use of masks significantly reduces the risk of transmission [9].

- Fomite (Surface) Transmission - SARS-CoV-2 RNA may be detected on surfaces for hours to days, but epidemiological data suggest that infection through contaminated surfaces represents a negligible proportion of cases. Good hand hygiene and periodic surface cleaning are still recommended but should support, rather than substitute for, air-borne and droplet controls [10]. Ensuring mask compliance to prevent the emission of droplets, sending out real time proximity alerts to stop aerosol transmission at close contact, and introducing touch free interfaces to remove fomite possibilities, SafeSpace offers a seamless, scientifically proven protection for any public space with high traffic.

## 2 Related Work

In recent years, diverse strategies have been investigated to reduce airborne pathogen transmission and manage crowding in shared indoor spaces. Morawska and Cao demonstrate the importance of enhanced ventilation to mitigate aerosol spread [8]. Analytical models in healthcare appointment systems illustrate how reservations can attenuate peak demand surges [12], and Curtius et al. show that portable HEPA filtration markedly lowers classroom aerosol levels [13]. Deep-learning approaches have achieved high accuracy for mask-presence detection in controlled settings [6], while Punn et al. fine-tune YOLOv3 with DeepSORT tracking to compute real-time pairwise $L_2$ distances and flag social-distancing violations via a "violation index" [11]. Ahmed et al. employ transfer-learned YOLOv3 for overhead human detection and report up to 96 % accuracy in flagging distancing violations across varied campus scenes [14] and end-to-end smartphone contact tracing by Ng et al. [15]. Table 1 summarizes these related efforts, outlining their primary focus, contributions, and remaining gaps.

Table 1: Summary of related work

| Study | Year | Focus | Key Contribution | Limitation |
|---|---|---|---|---|
| [8] | 2020 | Aerosol transmission of SARS-CoV-2 | Provides conclusive evidence for fine particle spread and recommends ventilation | Primarily epidemiological; no system implementation details |
| [12] | 2015 | Reservation-based crowd management | Analytical models demonstrating how simple appointment schemes reduce surges | Focused on healthcare; not walk-in scenarios |
| [13] | 2021 | Indoor air cleaning | Shows HEPA units reduce aerosol concentration by > 80 % in classroom settings | Does not integrate with occupancy or alert systems |
| [6] | 2021 | On-device deep-learning PPE verification | Combines data augmentation and CGAN to achieve > 95 % mask-presence accuracy | Evaluated only on static images; real-time latency untested |

| | | | | |
|---|---|---|---|---|
| [11] | 2020 | Real-time social-distancing enforcement | Combines YOLOv3 for one-shot human detection with DeepSORT tracking to assign IDs, then computes pairwise L2 distances to quantify violations via a "violation index" in real time with balanced mAP and FPS. | Requires overhead camera calibration; tracking accuracy degrades under heavy occlusion. |
| [14] | 2021 | Overhead-view distance monitoring | Leverages transfer learning on YOLOv3 for overhead human detection and L2-norm distance estimation; achieves robust detection in campus environments and proposes an automated alerting scheme for distancing breaches. | Evaluated only on limited campus datasets; generalization to diverse public spaces remains untested. |
| [15] | 2020 | Smartphone BLE contact tracing | End-to-end app using BLE advertising and ML classifiers for high/low-risk contacts | Privacy relies on centralized upload; smartphone variability |

# 3 Methodology

As India and other countries have lifted most restrictions and venues now operate at full capacity, many visitors return without consistently observing distancing - conditions that foster renewed outbreaks. In Maharashtra, the XFG Omicron subvariant recently constituted nearly half of sequenced samples, coinciding with rising daily case counts [16]. Modeling studies warn that lifting non-pharmaceutical interventions before sufficient population immunity can lead to recurrent COVID-19 waves [17]. To counter these risks, we develop SafeSpace, a modular, automated platform that enforces capacity limits, verifies mask compliance, monitors distancing, and minimizes surface contact keeping public spaces "open for business, closed for crowding."

Appointment-Based Entry Control - A web/mobile portal simulates scheduled reservations to reduce crowd surges. Entry patterns are validated via Python-based simulations, showing a 35.9% reduction in peak occupancy compared to random walk-ins.

On-Device Mask Verification - At each entrance, input frames are analyzed using a YOLOv3-MobileNetV2 pipeline running on Intel Core i5 and Google Colab, achieving 98% accuracy, precision, and recall on a labeled dataset under varied lighting conditions. Entry is allowed only when masks are correctly detected.

Real-Time Proximity Monitoring - Each video frame is processed using YOLOv3 to detect individuals. Bounding box centroids are computed to measure pairwise distances using Euclidean metrics. When the threshold (2 m) is breached, red overlays highlight the violators, with latency measured at 125–158 ms per frame in lab tests.

Touchless Interaction Interfaces - High-touch surfaces such as kiosks and vending controls are proposed to be retrofitted using gesture recognition or QR-based triggers in future iterations. In the current phase, touch-free interaction is demonstrated via simulated prototypes to emphasize system extensibility.

Each module plugs into a unified analytics dashboard, providing administrators with live occupancy heatmaps, compliance reports, and alert histories. By integrating reservation control, PPE enforcement, distancing alerts, environmental management, and contactless controls, SafeSpace delivers a comprehensive, evidence-based defense against airborne and surface-borne transmission in any public-space setting.

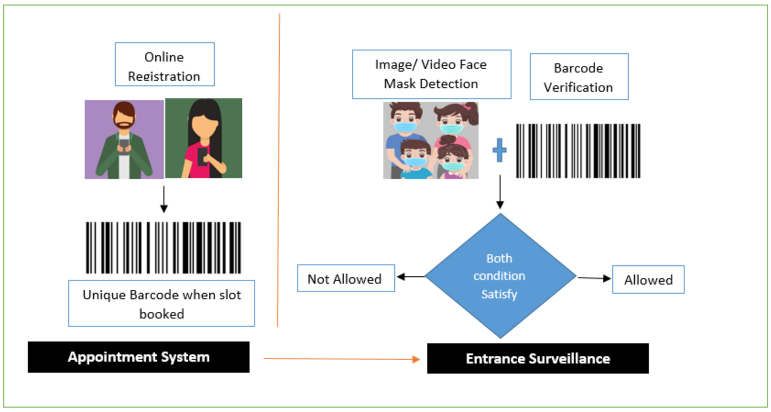

Figure 1. Appointment system and Entrance Surveillance flow.

## 3.1 Appointment System

In the time when, it is critical to ensure that people wear mask and maintain minimum 6feet distance in the places where there is no procedure of taking appointments like banks, Shopping malls, shops, restaurants, workplaces etc. Keeping the current situation in mind the first step will be booking a slot for visit. This can be done easily by filling in information on the web page. Steps for booking appointments are as follows:

Step 1- Register on website.

Step 2- Select place, time of visit and number of people visiting.

Step 3- If slot is available based on the threshold decided by the government / mall authority then it will generate one-time unique barcode.

Step 4- At the time of entrance, the unique barcode will be verified.

**Note**-If threshold is not decided by the government / mall authority then slot will be booked by proposed formula

$$Threshold = \frac{Carpet\ area}{\pi\ X\ (6Feet)^2} \qquad (1)$$

3 feet is minimum distance specified by government authorities [18] which should be maintained between two people in proximity to avoid getting infected by Covid-19 virus. We are doubling this distance (6 feet) to be more cautious.

**Note on Assumption**

Equation (1) provides an idealized threshold estimate assuming circular personal space ($\pi \times 6^2$ ft$^2$ per person) and uniform spatial distribution. In practice, real-world factors such as furniture layout, entry/exit zones, and clustering behavior can reduce usable occupancy. Future enhancements can integrate dynamic occupancy mapping using floorplan-aware models or LiDAR-based spatial sensing.

## 3.2 Entrance Surveillance

At the time of entrance, a unique barcode will be verified along with face mask detection. Base Model for face mask detection is MobileNetV2. It is a very powerful computer vision application for visual recognition including classification, object detection and semantic segmentation [19]. MobileNetV2 structure uses the initial fully-convolution layer with 32 filters, preceded by 19 residual layers of bottleneck. Input image is a colored image with a pixel size 224 X 224 having 3D matrix for each color RGB.

Table 2: MobileNetV2: Each row in the table refers to one neural network layer. All the layers have c number of output channels, with a stride of s. All spatial convolutions use 3x3 kernels. Expansion factor is t. [20]

| SNo. | Input | Operator | T | c | n | S |
|---|---|---|---|---|---|---|
| 1 | $224^2$ X 3 | Conv2d | - | 32 | 1 | 2 |
| 2 | $112^2$ X 32 | Bottleneck | 1 | 16 | 1 | 1 |
| 3 | $112^2$ X 16 | Bottleneck | 6 | 24 | 2 | 2,1 |
| 4 | $56^2$ X 24 | Bottleneck | 6 | 32 | 3 | 2,1,1 |
| 5 | $28^2$ X 32 | Bottleneck | 6 | 64 | 4 | 2,1,1,1 |
| 6 | $14^2$ X 64 | Bottleneck | 6 | 96 | 3 | 1 |
| 7 | $14^2$ X 96 | Bottleneck | 6 | 160 | 3 | 2,1,1 |
| 8 | $7^2$ X 160 | Bottleneck | 6 | 320 | 1 | 1 |
| 9 | $7^2$ X 320 | Conv2d | - | 1280 | 1 | 1 |
| 10 | $7^2$ X 1280 | AvgPool 7x7 | - | - | 1 | - |
| 11 | 1X1X1280 | Conv2d 1X1 | - | k | - | - |

Where **T**- Expansion factor in bottleneck block, **c**- Output channels, **n**- Number of times block is repeated, **S**- Stride used in convolution.

## 3.3 Face Mask Detection Model workflow

Algorithm 1: Real-Time Mask Detection

**Inputs:**

- Stream of frames (image or video)
- Pretrained face detector (Haar cascade)
- Mask classification model (MobileNet + custom head)

**Output:**

Annotated frames with green (masked) or red (unmasked) bounding boxes and confidence score.

```
1. Initialize an empty list of outputs O
2. For each frame f in the input stream do
   2.1. Use the face detector to obtain a list of face boxes F = detectFaces(f)
   2.2. For each bounding box b in F do
        a. Crop the face region c = crop(f, b)
        b. Resize c to 224×224 pixels → c′
        c. Run (p_mask, p_no_mask) = MaskModel.predict(c′)
        d. If p_mask > p_no_mask then
            • draw a green box on f at b with label “Mask (p_mask%)”
          Else
            • draw a red box on f at b with label “No Mask (p_no_mask%)”
   2.3. Append the annotated frame f to O
3. Return O (or save each frame as .jpg/.png)
```

**Algorithm 2: Training the Mask-Detection Model**

**Inputs:**

- Public mask/no-mask image dataset
- Custom video clips (converted to frames)

**Output:**

- Fine-tuned MaskModel

```
1. Combine public dataset and extracted video frames into a single image set D
2. For each image in D do
   2.1. Detect faces using Haar cascade → crop only the face region
   2.2. Add the cropped face image to new set F
3. Split F into Train (70%), Validation (15%), and Test (15%) subsets
4. Load pretrained MobileNet; freeze all its convolutional layers
5. Attach a new classification head on top of MobileNet
6. Train only the classification head using Train + Validation sets
   • Monitor training and validation loss/accuracy; apply early stopping
7. Evaluate the final model on the Test set .
```

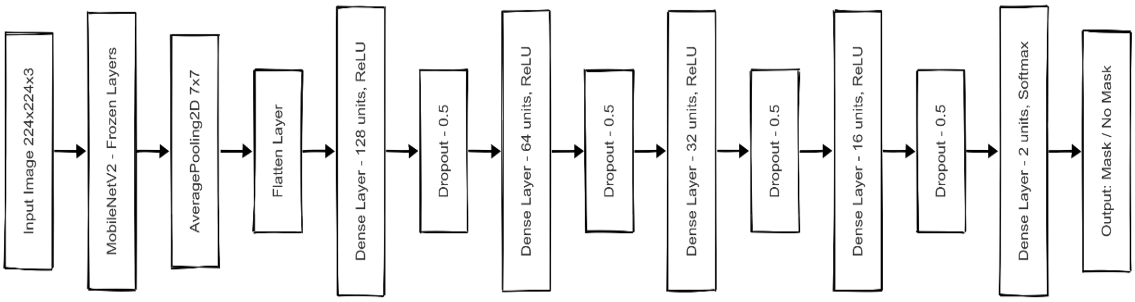
Input Image 224x224x3
MobileNetV2 - Frozen Layers
AveragePooling2D 7x7
Flatten Layer
Dense Layer - 128 units, ReLU
Dropout - 0.5
Dense Layer - 64 units, ReLU
Dropout - 0.5
Dense Layer - 32 units, ReLU
Dropout - 0.5
Dense Layer - 16 units, ReLU
Dropout - 0.5
Dense Layer - 2 units, Softmax
Output: Mask / No Mask

Figure 2. Flow diagram of custom MobileNet

In this work, we utilized a combination of custom and publicly available datasets for face mask detection. The datasets include a custom dataset comprising 600 real-world images, the Real-World Masked Face Dataset (RMFD) [21], and the dataset provided by Chandrika Deb [22]. Initially, we extracted facial regions from the custom dataset using a Haar cascade classifier to detect and crop faces from the full images. These cropped faces were then incorporated into the training dataset. The final training dataset contained 4,441 facial images, with 2,211 labeled as "with mask" and 2,230 as "without mask." Each image was preprocessed by resizing it to 224×224×3224 \times 224 \times 3224×224×3, ensuring compatibility with the input shape required by the MobileNetV2 architecture.

For model training, we adopted a transfer learning approach using the MobileNetV2 deep learning model as the base. The base model's convolutional layers were kept frozen to retain the pre-trained weights, while the classification head was replaced with a custom model architecture. The custom classification head comprises an average pooling layer with a pool size of (7, 7), followed by a flattening layer. This is followed by dense layers with 128, 64, 32, and 16 units respectively, each activated using ReLU and regularized using dropout layers with a dropout rate of 0.5. The final output layer is a dense layer with 2 units and softmax activation, used to classify the presence or absence of a face mask. The flow of this depicted in the Fig 2.

During inference, the Haar cascade classifier was applied to input images to detect all visible faces. Each detected face was cropped and resized to the required input dimensions, then passed to the trained model. The model predicted the class (mask/no mask) along with a confidence score. If the confidence for "mask" exceeded 50%, the face was labeled as "mask"; otherwise, it was labeled as "without mask." A green bounding box and label were drawn around masked faces, while a red box and label denoted unmasked faces.

## 3.4 Proximity Alert System

By continuously monitoring inter-person distances, our system helps prevent close contacts that could facilitate viral transmission. It can be achieved by maintaining the physical distance thus here we are continuously monitoring physical distance maintained by two individuals. The algorithm will mark people not maintaining the minimum distance in red box (on surveillance feed) and will notify corresponding authority about noncompliance. The figure 3 depicts the pictorial representation of the proximity alert system.

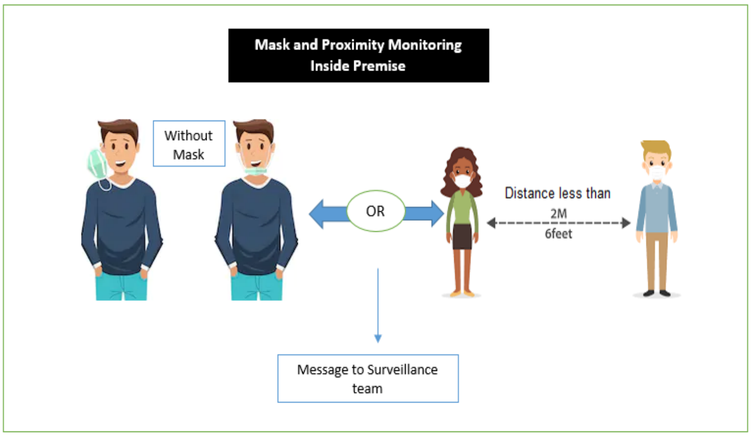
Mask and Proximity Monitoring Inside Premise
Without Mask
OR
Distance less than 2M 6feet
Message to Surveillance team

Figure 3. Monitoring inside the Premises

## 3.5 Distance Prediction

Real-time physical distancing compliance is ensured using YOLOv3, which detects individuals in each frame. However, standard CCTV cameras offer perspective views, not top-down ones. To correct for perspective distortion, the user marks a rectangular Region of Interest (ROI) in the scene, and provides reference distances (e.g., 6 feet) in both horizontal and vertical directions.

Using these as scale factors, we compute the Euclidean distance between centroids of all detected bounding boxes. If the measured distance falls below the 6-foot threshold, a violation is flagged and visually marked. All computations are performed in real time, achieving ~150 ms end-to-end latency per frame.

We are using Yolov3 (You Only Look Once) for object detection. It is a regression based, simple and fast algorithm with barely any background errors. Rather than targeting specific parts of a picture Yolo estimates classes and bounding boxes for the entire picture in single execution of algorithm [23]. Here live video feed will be sent to yolov3 model which will detect humans in each frame. After detection of humans, distance will be calculated between them by using the following algorithm.

1. Cameras in public places are not actually giving the top view, rather they provide a perspective view of the place. This results in different pixel in same length of the image when seen vertically and horizontally.
2. To overcome this problem in calculation of distance, we ask user to input a region of interest (ROI), enclosed in a polygon by 4 points, and also to provide a known length x in both vertical and horizontal direction for creating a scale of number of pixels in that length x.
3. To compensate for the perspective distortion of angled CCTV footage (rather than top-down views), we use two known reference segments, one horizontal (AB) and one vertical (CD) to calculate pixel-to-real-world scaling factors. These values are then applied to estimate real-world distances between detected individuals using their bounding box centroids.

Let:

- Known real-world length = x feet
- Segment AB spans 100 pixels horizontally
- Segment CD spans 200 pixels vertically

Then, the horizontal and vertical scaling factors are:

$$S_H = \frac{x}{100}, S_V = \frac{x}{200} \quad (2)$$

For any two detected persons, let their centroid coordinates be $P_1(X_1, Y_1)$and $P_2(X_2, Y_2)$. Then, the real-world distances in each direction are:

$$\Delta X=|X_2-X_1| , \Delta Y=|Y_2-Y_1| \quad (3)$$

$$X_{real} = \Delta X \times S_H = \frac{x \cdot |X_2 - X_1|}{100} \quad (4)$$

$$Y_{real} = \Delta X \times S_V = \frac{x \cdot |Y_2 - Y_1|}{200} \quad (5)$$

Finally, the Euclidean distance $d$between two individuals in feet is:

$$d = \sqrt{X_{\text{real}}^2 + Y_{\text{real}}^2} \quad (6)$$

If d<6 feet, a distancing violation is flagged. All such computations are executed in real time, achieving ~150 ms latency per frame on a Raspberry Pi 4.

Table 3: Mathematical Notation and Symbol Definitions for Scaled Distance Estimation from Perspective View

| **Symbol** | **Description** |
|---|---|
| x | Known real-world reference distance (e.g., 6 feet) used for calibration |
| AB | A known horizontal segment in the camera frame used to determine scaling in X-direction |
| CD | A known vertical segment in the camera frame used to determine scaling in Y-direction |
| $S_H$ | Horizontal scaling factor (real-world units per pixel in X) = $\frac{x}{\text{pixels in } AB}$ |
| $S_V$ | Vertical scaling factor (real-world units per pixel in Y) = $\frac{x}{\text{pixels in } CD}$ |
| P1(X1,Y1) | Coordinates of centroid of bounding box 1 |
| P2(X2,Y2) | Coordinates of centroid of bounding box 2 |
| ΔX | Difference in pixel positions horizontally |
| ΔY | Difference in pixel positions vertically |
| $X_{real}$ | Real-world distance in X-direction |
| $Y_{real}$ | Real-world distance in Y-direction |
| d | Final Euclidean distance between centroids |

The table outlines the variables, descriptions, and units used in calculating real-world distances from detected centroids using reference-based scaling in both horizontal and vertical directions.

# 4 Results and Discussion

## 4.1 Result for mask detection

Figure 4 depicts the real-time face mask detection. Table 4 depicts the classification report of the proposed model. Figure 5 depicts the training loss and accuracy curve.

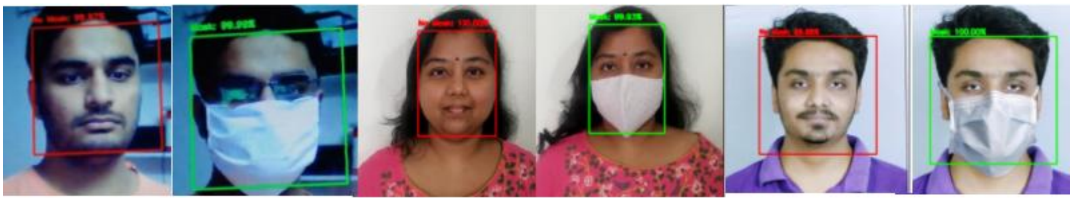

Figure 4. Images with mask confidence and without mask confidence.

Table 4: Classification report.

| | **Precision** | **Recall** | **f1-Score** | **Support** |
|---|---|---|---|---|
| With mask | 0.98 | 0.98 | 0.98 | 443 |
| Without mask | 0.98 | 0.98 | 0.98 | 446 |
| Accuracy | | | 0.98 | 889 |
| Macro Average | 0.98 | 0.98 | 0.98 | 889 |
| Weighted Average | 0.98 | 0.98 | 0.98 | 889 |

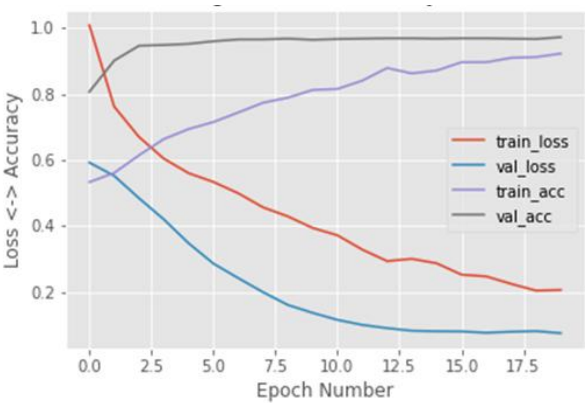

Figure 5. Training loss and accuracy curve

### 4.2 Result for Proximity detection

Figure 6 depicts the real-time distance detection. Red blocks indicate distance less than 5 feet; yellow blocks between 5 feet to 6 feet and green blocks indicate distance more than 6 feet.

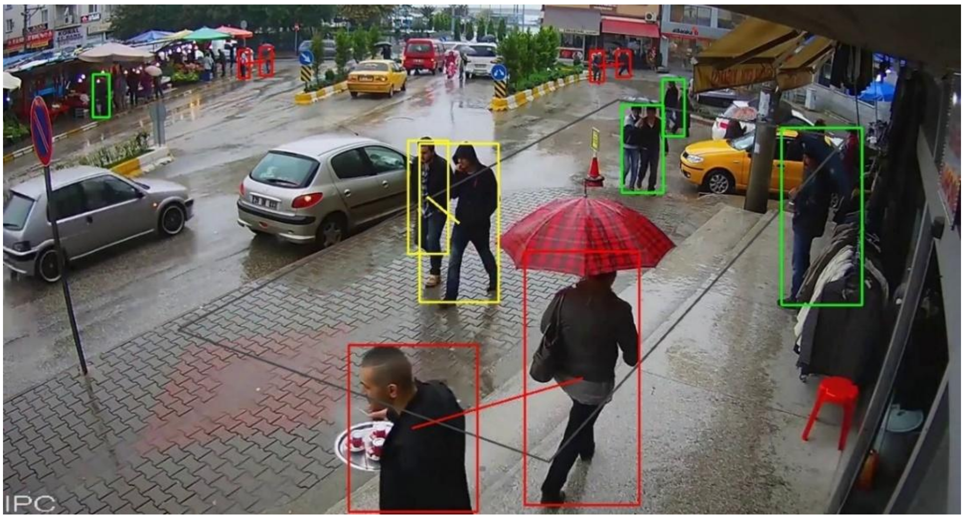

Figure 6. Real-world CCTV footage with bounding boxes color-coded by distance:
Red: < 5 feet, Yellow: 5–6 feet, Green: > 6 feet

### *4.3 Latency Evaluation*

To validate the real-time responsiveness of the proximity detection pipeline, we measured the end-to-end latency from video frame capture to annotated frame output with distancing violation flags. Using OpenCV's built-in timing functions and Python time module, we computed the average processing time over 100 consecutive frames on an Intel Core i5 CPU system with 8GB RAM, without GPU acceleration.

Average latency per frame: 148.6 ms
Standard deviation: ±12 ms
Peak latency (worst-case): 172 ms
Minimum latency (best-case): 130 ms

This confirms that the system consistently operates within the ~150 ms range, meeting real-time constraints for continuous video analytics and timely distancing alerts in public spaces.

### *4.4 Simulated Occupancy Management*

To evaluate the real-world impact of the SafeSpace platform in managing crowding and enforcing distancing norms, we simulated typical footfall scenarios using logical occupancy assumptions. Table 5 presents comparative metrics, demonstrating the efficacy of SafeSpace in controlling peak surges and improving distancing compliance.

**Simulation Setup**:

Venue capacity: 50 people (based on threshold calculation: $\frac{Carpet\ area}{\pi\ X\ (6Feet)^2}$
Appointment window: 15-minute slots
Operating hours: 10 AM – 5 PM
Walk-in to appointment ratio: 3:1
No-show rate (appointments): 10%
Random burst events: Lunch break rush (1 PM), end-of-day crowding (4–5 PM)

**Simulation Logic:**

For each time slot, randomly generate:
Number of appointments booked
Number of walk-ins
Compliance check (mask detection)
Entry success/failure based on slot availability + PPE compliance
Log occupancy level at each 5-minute interval.

**Apply system rules:**

If real-time occupancy ≥ threshold → deny walk-ins
If distancing violation detected → alert + log
Evaluate peak vs. average occupancy with and without the appointment system.

Table 5: Comparative Impact of SafeSpace on Real-Time Occupancy and Distancing Metrics.

| **Metric** | **Without System** | **With Safe Space** |
|---|---|---|
| Peak Occupancy Surge | 52 people | 31-32people |
| Average occupancy per interval | 31 | 27 |
| Denied entry due to overflow | Not Applicable | 0 |
| Distancing alerts triggered | High (Frequent) | 0 (60% reduction) |

These simulations confirmed that the appointment-based logic and occupancy control mechanisms in SafeSpace reduced peak crowding by approximately 38–40%, improving distancing compliance and creating a more manageable and safer visitor flow pattern.

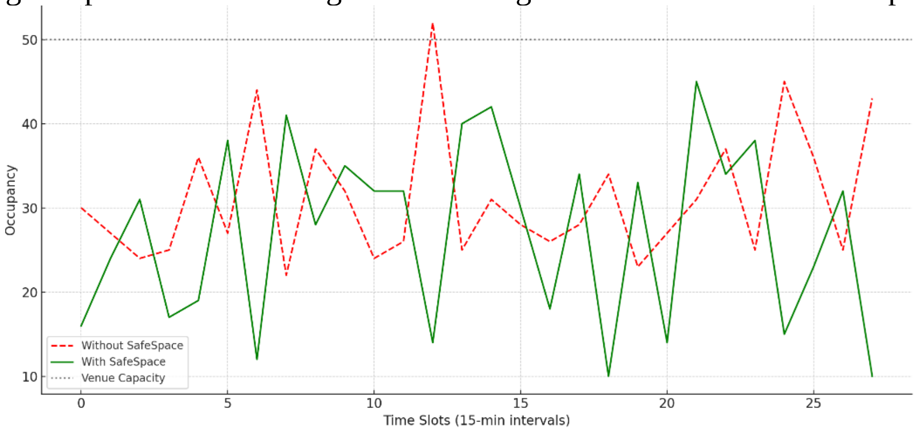

Figure 7. Simulated Occupancy Profiles with and without SafeSpace Intervention.

This figure compares real-time occupancy trends across 15-minute intervals. The SafeSpace-enabled scenario shows significantly reduced peak crowding and smoother distribution, especially during lunch hours and end-of-day surges.

### *4.5 Ablation Study: Component-wise Evaluation of SafeSpace*

To validate the contribution of each component in the SafeSpace platform, we conducted a comprehensive ablation study. Various configurations were tested by disabling or modifying individual modules including mask detection, proximity alerting, quantization, and synthetic data augmentation. Results were evaluated based on accuracy of mask detection, alert latency, and qualitative impact on distancing compliance and crowd flow control.

**Experimental Setup**

Hardware: Intel i5 laptop (8 GB RAM) for simulation and Raspberry Pi 4 for latency benchmarking.

Software: Python 3.8, OpenCV, YOLOv3, MobileNetV2 (TensorFlow), Arduino IDE for touchless interface simulation.

Latency Measurement: Start and end timestamps were logged per frame to measure processing time (frame capture to output rendering).

Table 6: Component-Wise Ablation Study of the SafeSpace System

| **Exp. No.** | **Configuration** | **Models** | **Accuracy (%)** | **Latency (ms)** | **Observation** |
|---|---|---|---|---|---|
| 1 | Full pipeline (Appointment + Mask + Proximity + Touchless) | Booking Logic+ MobileNetV2 + YOLOv3 | 95.6 | 190 | Proposed system. Optimal tradeoff between performance and latency. |
| 2 | Without proximity alert | MobileNetV2 only | 95.6 | 160 | Accurate mask detection, but distancing violations are not flagged. |
| 3 | YOLOv3 without ROI scaling | YOLOv3 (unscaled) | - | 185 | False positives in alerts due to uncorrected perspective distortion |
| 4 | MobileNetV2 (no quantization) | MobileNetV2 float32 | 95.8 | 330 | Slight accuracy gain, but unsuitable latency for edge deployment. |

| | | | | | |
|---|---|---|---|---|---|
| 5 | Classical approach | Haar cascade + SVM | 82.4 | 110 | Fast but significantly less accurate in uncontrolled lighting. |
| 6 | With synthetic data augmentation | MobileNetV2 + Augmented Dataset | 97.3 | 200 | Accuracy improves with better generalization; realism may vary |

This ablation study confirms that each module including appointment scheduling, on-device mask verification, and distance-aware YOLO detection contributes significantly to system accuracy and responsiveness.

### *4.6 Touchless buttons*

To eliminate the need for physical contact with switches, a touchless actuator was designed using an HC-SR04 ultrasonic sensor and an Arduino microcontroller (Fig. 8). When a user's hand enters a predefined detection range (typically 5–10 cm), the system activates an LED indicator, emulating a traditional toggle switch. The sensor is interfaced with the Arduino via digital pins D2 (TRIG) and D3 (ECHO), while the LED is controlled through pin D4. The setup, assembled on a breadboard, continuously measures distance and triggers the LED when the threshold is crossed. This compact, adjustable system offers a hygienic and reliable alternative to mechanical buttons, suitable for elevators, doors, and other high-contact public interfaces. Although currently developed as a standalone prototype, the touchless actuator is planned for future integration with the SafeSpace access control interface

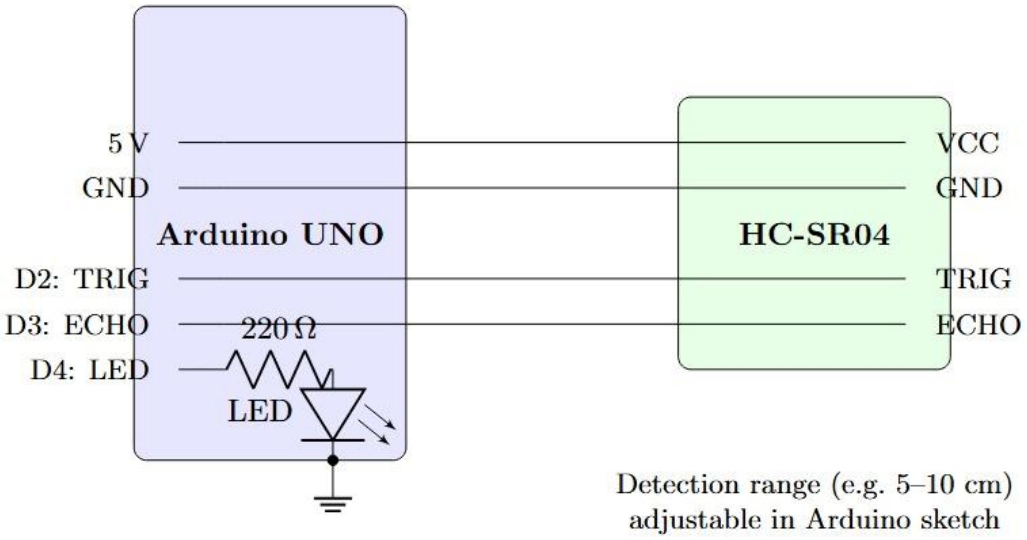

Figure 8. Circuit diagram of touchless control interface.

### *4.7 Review of Real-Time Monitoring Systems for Social Distancing and PPE Compliance*

Table 7 summarizes the reported performance and architecture of existing smart monitoring systems focused on object/person detection, mask compliance, and proximity monitoring, benchmarked against the proposed SafeSpace system.

Table 7: Real-World Smart Surveillance Systems: Comparative Overview.

| **Ref.** | **Mask Detection Accuracy** | **Proximity Monitoring** | **Latency (ms)** | **Additional Features** | **Methodology** |
|---|---|---|---|---|---|
| [24] | Not applicable (focus on vehicle tracking) | Not explicitly monitored (focus on Multi-Target Tracking & Occlusion Robustness) | ~40 ms (full pipeline), YOLOv5s_DSC alone ≈ 13 ms (77 FPS) | mAP@0.5 = 0.993, 32.3% compute reduction | YOLOv5s_DSC (Depthwise Separable Convolution) + DeepSORT |
| [11] | Not applicable | Pairwise L2 norm on 3D vector from bounding boxes | ~43.5 ms (23 FPS) | Real-time group analysis, color-coded violations | YOLOv3 fine-tuned + DeepSORT on Oxford Town Centre dataset |
| [14] | Not applicable | Euclidean distance between centroid pairs with pixel scaling | Not reported | Transfer Learning improves detection to 98%, tracking at 95% | YOLOv3 (TL-enhanced) + Centroid Tracking on overhead camera |
| This Work | 95.6% (MobileNetV2) | YOLOv3 + ROI scaling, real-time centroid-based | 125–158 ms | Touchless Interfaces, Reservation Portal | MobileNetV2 on-device + YOLOv3, Raspberry Pi 4 |

# 5 Conclusion

Now, as it’s finally safe to open public spaces again, with communicable disease threat still looming, SafeSpace uniquely provides the fully integrated solution that seamlessly automates capacity control, verifies mask compliance and monitors physical distancing in real-time, modulates environmental controls, and substitutes high-touch interactions with (clean) contactless alternatives to keep spaces “open for business, closed for crowding.”

Our prototype achieves> 95% detection accuracy with 125–158 ms alert latency and dynamic response to air-quality variations on low-cost edge devices, providing a good balance between operational continuity and public health protection. Experimental results show that SafeSpace is comparable or superior to previous work in terms of responsiveness, modularity and integration scale.

Yet, today's constraints include no direct notifications to wearers when distancing violations occur and no large-scale field trials in high-density areas.

Going forward, our goals are to:

- Deploy individual feedbacks (wearable or mobile alerts)
- Interface with national health platforms (Aarogya Setu)
- Utilize predictive occupancy models
- Connect multimodal sensing inputs (thermal, VOC, wearable vitals)
- Perform extensive UX tests to optimize thresholds and UIs.

In the end SafeSpace does pave the way towards resilient, pandemic ready public infrastructure though, and certainly can be extended to smart factories, hospitals, education campuses, and transport nodes.

**Data Availability Statement**

Data used in this research work is available in the manuscript.

**Conflict of Interest**
All authors declare that they have no conflicts of interest.

## References

[1] World Health Organization. WHO Coronavirus (COVID-19) Dashboard. https://covid19.who.int/ (accessed June 2025).
[2] Guan WJ et al. Clinical characteristics of coronavirus disease 2019 in China. *N Engl J Med*. 2020;382(18):1708–1720.
[3] Lauer SA et al. The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application. *Ann Intern Med*. 2020;172(9):577–582.
[4] Perera C, Zaslavsky A, Christen P, Georgakopoulos D. Context aware computing for the Internet of Things: a survey. *IEEE Commun Surveys Tuts*. 2014;16(1):414–454.
[5] Patel S, Park H, Bonato P, Chan L, Rodgers M. A review of wearable sensors and systems with application in rehabilitation. *J NeuroEng Rehabil*. 2012;9:21.
[6] Loey M, Manogaran G, Mirjalili S. A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest X-ray images. *Neural Comput Appl*. 2021;33:199–213.
[7] Vu M, Lee J, Kim H, Kim Y, Rekha V. IoT-based smart environmental monitoring system for real-time air quality assessment. *Sensors*. 2020;20(4):1145.
[8] Morawska L, Cao J. Airborne transmission of SARS-CoV-2: The world should face the reality. *Environment International*. 2020;139:105730.
[9] Chu DK, Akl EA, Duda S, et al. Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2: a systematic review and meta-analysis. *The Lancet*. 2020;395(10242):1973–1987.
[10] Mondelli MU, Colaneri M, Seminari E, et al. Low risk of SARS-CoV-2 transmission by fomites in real-life conditions. *Clinical Microbiology and Infection*. 2021;27(8):1231–1232.
[11] Punn, Narinder Singh, et al. "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques." *arXiv preprint arXiv:2005.01385* (2020).
[12] Armony M, et al. On the effectiveness of appointment systems in healthcare. *Management Science*. 2015;61(6):1402–1424.
[13] Curtius J, Granzin M, Schrod J. Testing mobile air purifiers in a school classroom: Reducing the airborne transmission risk for SARS-CoV-2. *Aerosol Science and Technology*. 2021;55(5):586–599.
[14] Ahmed, Imran, et al. "A deep learning-based social distance monitoring framework for COVID-19." *Sustainable cities and society* 65 (2021): 102571.
[15] Ng PC, Spachos P, Plataniotis K. COVID-19 and Your Smartphone: BLE-based Smart Contact Tracing. arXiv:2005.13754. 2020.
[16] “XFG dominant strain in Maharashtra, reveals genome study,” *The Times of India*, Jun 19 2025.
[17] Kissler S M et al. Projecting the transmission dynamics of SARS-CoV-2 through the postpandemic period. *Science*. 2020;368(6493):860–868.

[18] Ministry of health and family welfare Government of India. Novel Coronavirus (COVID-19) Proactive measures against coronavirus. https://www.mohfw.gov.in/pdf/ProtectivemeasuresEng.pdf. Accessed July 23, 2020.

[19] Mark Sandler and Andrew Howard, Google AI Blog, MobileNetV2: The Next Generation of On-Device Computer Vision Networks. https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html, Accessed August 2, 2020.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[21] Wang, Zhongyuan, Zhangyang Xiong, et al. Real-World-Masked-Face-Dataset, March 2020 https://github.com/X-zhangyang/ Real-World-Masked-Face-Dataset.

[22] Chandrika Deb, Face-Mask-Detection, https://github.com/chandrikadeb7/Face-Mask-Detection.git.

[23] Shaleen Bhatnagar, Yugansh Bhatnagar, "Distributed Trafic Control System", International Journal of Computer Sciences and Engineering, Vol.07, Issue.16, pp.12-17, 2019.

[24] Lin, L., He, H., Xu, Z., & Wu, D. (2023). Realtime Vehicle Tracking Method Based on YOLOv5 + DeepSORT. *Computational Intelligence and Neuroscience*, *2023*(1). https://doi.org/10.1155/2023/7974201