






Article

Edge Computing for Smart-City Human Habitat: A Pandemic-Resilient, AI-Powered Framework

Atlanta Choudhury ¹, Kandarpa Kumar Sarma ^{1,*}, Debashis Dev Misra ², Koushik Guha ³
and Jacopo Iannacci ^{4,*}

- ¹ Department of Electronics and Communication Engineering, Gauhati University, Guwahati 781014, Assam, India; atlantachoudhury07@gauhati.ac.in
 - ² Department of Computer Science and Engineering, Faculty of Engineering and Technology, Assam Down Town University, Guwahati 781026, Assam, India; debashis.misra@adtu.in
 - ³ Department of Electronics and Communication Engineering, NIT, Silchar 788010, Assam, India; koushik@ece.nits.ac.in
 - ⁴ MicroSystems Technology Research Unit, Center for Sensors and Devices (SD), Fondazione Bruno Kessler, Povo, 38123 Trento, Italy
- * Correspondence: kandarpaks@gauhati.ac.in (K.K.S.); iannacci@fbk.eu (J.I.)

Abstract: The COVID-19 pandemic has highlighted the need for a robust medical infrastructure and crisis management strategy as part of smart-city applications, with technology playing a crucial role. The Internet of Things (IoT) has emerged as a promising solution, leveraging sensor arrays, wireless communication networks, and artificial intelligence (AI)-driven decision-making. Advancements in edge computing (EC), deep learning (DL), and deep transfer learning (DTL) have made IoT more effective in healthcare and pandemic-resilient infrastructures. DL architectures are particularly suitable for integration into a pandemic-compliant medical infrastructures when combined with medically oriented IoT setups. The development of an intelligent pandemic-compliant infrastructure requires combining IoT, edge and cloud computing, image processing, and AI tools to monitor adherence to social distancing norms, mask-wearing protocols, and contact tracing. The proliferation of 4G and beyond systems including 5G wireless communication has enabled ultra-wide broadband data-transfer and efficient information processing, with high reliability and low latency, thereby enabling seamless medical support as part of smart-city applications. Such setups are designed to be ever-ready to deal with virus-triggered pandemic-like medical emergencies. This study presents a pandemic-compliant mechanism leveraging IoT optimized for healthcare applications, edge and cloud computing frameworks, and a suite of DL tools. The framework uses a composite attention-driven framework incorporating various DL pre-trained models (DPTMs) for protocol adherence and contact tracing, and can detect certain cyber-attacks when interfaced with public networks. The results confirm the effectiveness of the proposed methodologies.

Keywords: deep learning; deep transfer learning; contact tracing; face mask; social distancing; edge computing; cyber-attack; federated learning



Citation: Choudhury, A.; Sarma, K.K.; Misra, D.D.; Guha, K.; Iannacci, J. Edge Computing for Smart-City Human Habitat: A Pandemic-Resilient, AI-Powered Framework. *J. Sens. Actuator Netw.* **2024**, *13*, 76. <https://doi.org/10.3390/jsan13060076>

Academic Editor: Lei Shu

Received: 11 September 2024

Revised: 28 October 2024

Accepted: 31 October 2024

Published: 6 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The global situation surrounding the COVID-19 pandemic has significantly improved, with decreasing infection rates, albeit not having been entirely eradicated. Despite widespread immunization efforts, the emergence of new variants and continued infections indicate that COVID-19 may persist alongside other influenza-like illnesses as an ongoing challenge for humanity [1]. Experts emphasize the importance of curtailing virus transmission through human-to-human contact, including aerosol transmission, which is typical of many influenza-like diseases. Guidelines from the World Health Organization (WHO) recommend measures such as maintaining a minimum distance of six feet or two meters between individuals (known as social distance) in crowded areas and wearing masks to prevent virus spread [2]. Given

these recommendations, there is a need for the development of monitoring tools to reinforce mask-wearing standards in public places and uphold prescribed bio-safety measures [3]. Contact tracing remains a crucial component in preventing virus spread, with governments and agencies advocating for the use of privacy-preserving contact-tracing applications to collect relevant data while safeguarding user privacy [4]. Research indicates that contact tracing and associated policies are effective in preventing the spread of COVID-19 and similar contagious diseases [5]. By enabling infected individuals to voluntarily quarantine themselves, these measures help mitigate the impact of community containment efforts, reducing overall suffering within nations and communities [6,7].

The combination of maintaining social distance, wearing masks, and implementing contact tracing has been recognized as crucial in preventing the widespread infection of influenza-like viruses, including COVID-19 [8]. It is equally important to implement measures for monitoring adherence to social distancing and mask-wearing guidelines [9]. Similarly, effective solutions are needed for contact tracing and maintaining databases that allow for frequent updates and accessibility to aid in preventing the spread of viruses associated with influenza outbreaks. These elements are essential in designing pandemic-resilient infrastructure for human habitation.

Among the array of technological solutions, the Internet of Things (IoT) stands out, leveraging sensor packs, wireless communication networks, and artificial intelligence (AI)-driven automated decision-making [10,11]. Furthermore, advancements in edge computing [12], deep learning (DL) [13], and deep transfer learning (DTL) [14] have empowered IoT to play a decisive role in healthcare, particularly in combating outbreaks of influenza-like viruses such as COVID-19. The application of IoT in healthcare has given rise to Medical IoT (MIoT) or Internet of Healthcare Things (IoHT), offering significant benefits to stakeholders. These technologies facilitate the effective tracking of agents (patients, medical professionals, and resources), automatic data sensing, authentication, and diagnostic decision support within health management systems [15]. This convergence of technology is commonly referred to as the Internet of Health Things (IoHT).

In many studies focusing on designing and implementing technologies for monitoring social distancing, mask-wearing, and contact tracing, these processes are often carried out independently. Despite the extensive use of AI tools in these endeavors, there exists an opportunity to integrate IoT, edge, and cloud computing along with AI to create comprehensive frameworks for effectively monitoring adherence to protocols related to influenza viruses, including COVID-19 outbreaks [16,17]. In this work, we present the design of a pandemic-compliant mechanism for monitoring adherence to influenza virus protocols, including social distancing norms and mask-wearing, as well as contact tracing. We utilize the Grove AI HAT and Raspberry Pi 4 combination, connecting multiple sensors to form edge nodes that collaborate with a cloud server hosting various machine learning (ML)/DL tools within a residential premise. Specifically, we employ DL pre-trained models (DPTM) such as RESNET-50, MobileNetV2, and SocialdistancingNet-19 within an attention-driven composite framework named Hybrid Multi-Head Attention-Aided Multi-Tasking Deep Network with Diffusion Stability (HMAAHDNDS). This framework utilizes samples from open-source databases to monitor protocol adherence and conduct contact tracing. Additionally, it generates images and text descriptions of violators and is equipped to detect cyber-attacks when connected to public networks. We benchmark our approach using various AI models, including feed-forward (FF) Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), VGG-16, Random Forest (RF), Decision Tree (DT), ResNet-50, and MobileNetV2. The results obtained from this integrated system, also referred to as the internet of intelligent infrastructure things (IoIIT), demonstrate the effectiveness of our proposed approach.

The specific contributions of the work are the following:

1. The design of an integrated framework with sensors, intelligent decision support, and notification mechanism for effective monitoring of the adherence to bio-safety norms notified for influenza-type virus infections as part of human habitation;

2. Incorporating the capability to simultaneously perform the monitoring of face-mask-wearing, social-distance norm adherence, executing contact tracing, text-to-image translation, and cyber-attack detection;
3. The implementation of multiple perceptive edge nodes using Grove AI Hat and Raspberry Pi4 along with several sensors and cameras for effective monitoring of adherence to bio-safety norms and deployed and tested as part of a residential complex;
4. Most importantly, formulating a cloud-resident DPTM-based hybrid architecture (HMAAHDNDS) which is trained (using open-access databases) to carry out face mask discrimination, social distance norms monitoring, contact tracing, text-to-image translation, and cyber-threat detection when deployed as part of a residential premise with multiple inputs coming through several edge nodes.

2. Proposed Pandemic-Compliant Perceptive Infrastructure Design

Extensive analysis of the existing literature highlights a significant opportunity for devising solutions that seamlessly integrate various aspects such as face mask detection, mask classification, the selection of appropriate mask types, adherence to social distancing guidelines, and contact tracing in the context of influenza viruses, including COVID-19. It is evident that an AI-driven approach holds the potential to automate these processes, manage accumulated data effectively, generate insightful analytics, ensure higher reliability, and establish a connection with confirmed cases of infection. Figure 1 illustrates an integrated methodology that combines adherence to social distancing norms, the recognition of face masks, and contact tracing pertaining to influenza viruses, including COVID-19. Notably, our focus extends to developing a comprehensive strategy for monitoring protocol adherence during pandemics, encompassing social distancing norms and correct mask usage. This involves leveraging the Grove AI HAT [18] and Raspberry Pi 4 combination, which connects multiple sensors to form edge nodes operating in tandem with a cloud server. These components are integral to residential settings and employ a suite of AI tools. To elucidate the operational dynamics of the proposed approach, a concise overview of the relevant background is provided below.

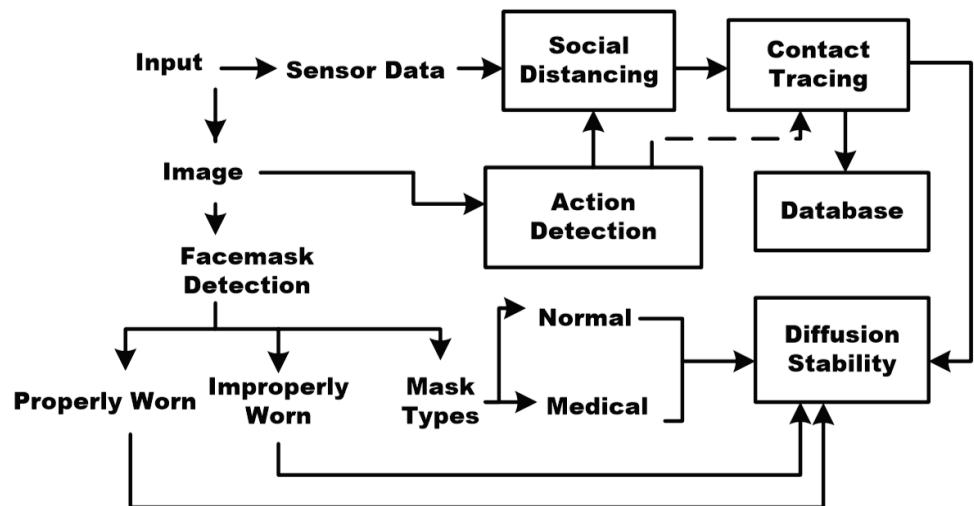


Figure 1. Block diagram of the proposed approach.

2.1. Setting Up IoT, Edge, and Cloud Computing Framework for Protocol Monitoring and Contact Tracing

The primary focus of our proposed solution centers on developing an IoT and edge computing setup (depicted in Figure 2) tailored for monitoring compliance with mask-wearing and social distancing guidelines, alongside enabling efficient contact tracing. The edge computing infrastructure comprises a Grove AI HAT [19] and a Raspberry Pi

4 (model B). Each entry point in a residential premise with four entrances is equipped with an IoT pack containing a camera, an infrared (IR) thermometer, an ultrasonic sensor, and a proximity sensor. The Grove AI HAT interfaces with the camera, which is crucial for monitoring adherence to physical distancing norms. In case of violations, a loudspeaker alerts individuals, prompting necessary action.

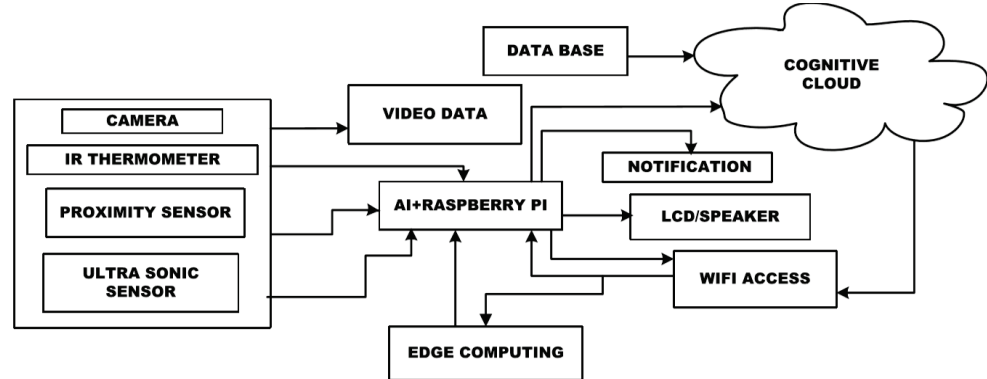


Figure 2. Schematic showing the connection of the IoT–edge computing node to the cloud server for monitoring of the compliance with face-mask-wearing and social distance norms and execute contact tracing.

Similarly, an IR thermometer measures an individual’s temperatures non-invasively during entry into the premises. The ultrasonic sensor notifies of any movement through the gates. The Grove AI Hat and Raspberry Pi4 along with the assortment of sensors form an edge node. One such node is deployed in each of the four entrances. The edge node has sufficient AI-aided decision-making capability. The Raspberry Pi handles data recording and processing, and maintains a tally of individuals passing through the entrances. Additionally, there is the provision for integrating a display to showcase temperature readings, social distancing metrics, compliance status, etc. (shown in Figure 3). The camera and display connect to the Grove AI HAT via a 24-pin serial communication connector.

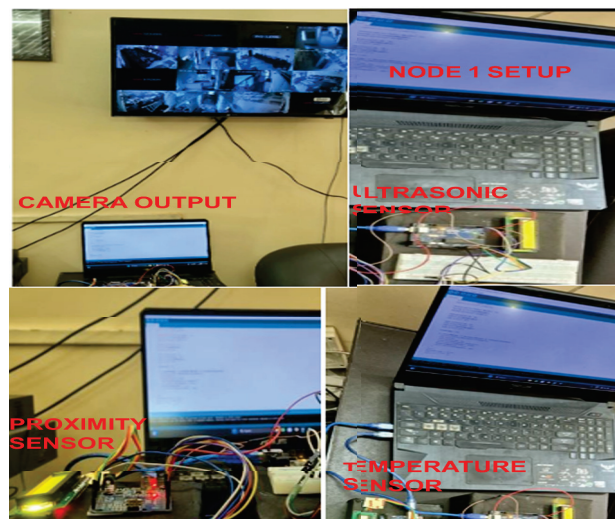


Figure 3. One of the edge nodes of the system.

The ultrasonic and IR temperature sensors, along with the Raspberry Pi, are interconnected, ultimately linking to the Grove AI HAT and a cloud server via Wi-Fi. Remote control of the setup is facilitated through VNC Connect software [20]. Utilizing edge computing, calculations are performed during the processing stage for each sensor’s data about an individual scan. Following that, in a multi-tasking scenario, the edge comput-

ing platform chooses a sensor node based on its location and updates the cloud server's records accordingly.

The combination of IoT, AI, and Wi-Fi access completes the process of data collection, processing, local decision-making, and communication. Typically, both wired and wireless networks are utilized by various applications to directly interface with the edge nodes and the cloud server computing platform. Additionally, both the cloud servers and edge nodes record all parameters, enabling them to calculate results based on global information. In our experiments, we have deployed four such edge computing nodes at the entrance of a residential complex as illustrated in Figure 4. As individuals enter, each sensor takes readings and transmits them to the edge computing node. Initial decisions regarding the actions to be taken, including monitoring compliance with mask-wearing and social distancing norms, and executing contact tracing, are made at the edge node level. This information is shared with the cloud server and stored in a database. For each of these actions, the edge node provides decision support, aided by dedicated Deep Neural Networks (DNNs) deployed on the cloud server. These DNNs undergo end-to-end learning and are trained to handle tasks such as ensuring compliance with mask-wearing, adhering to social distancing norms, and executing contact tracing. Details of each DNN type configured for these purposes are provided in subsequent sections. The Wi-Fi access boasts data rates of 50 Mbps with an optical fiber backhaul of 1 Gbps. Latency is a crucial aspect of the setup. We test the setup with three Wi-Fi configurations (50 Mbps, 40 Mbps, and 30 Mbps), 4G (10 Mbps), and 3G (1 Mbps), finding latencies ranging from 3 ms (at 50 Mbps) to 100 ms. The different sensors/components used in the system are detailed in Table 1, while the configuration of the edge node and cloud servers is outlined in Table 2. A photo of an edge node is shown in Figure 3. It depicts the display showing real-time capture from different camera feeds, one laptop working as the central controller of the nodes. Further, it shows the presence of the sensors with LCD displays and the Raspberry Pi-Grove AI pack.

You Only Look Once (YOLO) version 4 (YOLOV4) is used for local decision-making based on vision captured by the cameras especially to perform face mask detection, classification type, proper wearing and social distance monitoring. The YOLOV4 uses a pre-trained architecture, carries out one-shot training and detection, and is reliable for fast real-time applications. The entire processing takes place smoothly due to synchronous operation between the edge nodes and the cloud server, which is restricted only by the latency of the network connectivity. The range of the sensors and other related connectors are shown in Table 1. The cameras (Philips, HSP3500) for face mask handling are found to be effective up to 25 m in daylight and 10 m in low light.

The IoT framework designed for this purpose comprises four layers: the perception layer, transport layer, processing layer, and application layer. The perception layer consists of a camera, IR thermometer, ultrasonic sensor, and proximity sensor, each deployed at the entrance of the gates. The transport layer facilitates communication between the nodes and the cloud server, primarily utilizing Wi-Fi access (XLT240170), which has a maximum range of 150 m with various backhaul support options. A detailed summary of the sensor types used is provided in Table 1. The processing layer is constituted by the proposed architecture outlined in Section 2.2, which undertakes tasks such as detecting face masks, classifying masks, ensuring proper mask-wearing, monitoring social distancing norms, executing contact tracing, generating images of violators, and predicting cyber-attacks.

Reading through the sensors, face mask detection (proper wearing, appropriate type, and wearing or not) and social distance norms observance are carried out at the edge node level also. The details of the configuration of the edge node are shown in Table 2.

Table 1. Different sensors used for the work and key features.

Sl. No.	Sensor Type	Specifications
1	Ultrasonic sensor	HC-SR04-SEN-15569 Analog and digital connection. Baud Rate = 9600 I/P voltage = 5 v Working temp = −150–700 c Sensing angle = 300 cone Ultrasonic frequency = 40 KHz; Range = 2 cm–400 cm
2	Infrared Temperature Sensor	MLX90614 Vcc pin of the sensor with the Vin of the node Operating voltage = 3.6 v to 5 v Supply current = 1.5 mA Object temperature range= −700–382.20 c Ambient temperature range= −400 c to 1250 c FOV = 800 Distance between object and sensor = 2 cm to 5 cm
3	Proximity Sensor	PSAM8/ HC-SR04 Vcc-a +5 v GND of sensor is connected to GND of Raspberry Pi; Data Pin (IR sensor) is connected to PIN 16 Supply voltage = 10 v to 30 v Frequency = 800 Hz Working temp = −250–700 c Long range detection Range = 30 mm–50 mm Range up to 600 cm
4	IR camera	MLX90640 Vcc-a + 5 v GND of sensor to GND of Raspberry Pi Out a GPIO pin Small size; low cost; Supply voltage = 3 v; FOV (2 options): 550 × 350 and 1100 × 750 target temperature = −400 c–3000 c 32 × 24 pixels in IR array; I2C protocol, Range = 1 cm–100 cm
5	Wi-Fi Router	XLT24017 Wireless LAN Transmission range = 150 m; Frequency = 2.4 GHz; High stability built in PLL; Low clutter leakage; Size (L × W): 18.3 × 17.6 mm, Range = 30 m–100 m
6	Conventional CCTV Camera	Philips HSP 3500 Wireless Low light range −10 m; Day time −20–25 m

Table 2. Configuration of the edge node and the cloud servers.

Item	Description	Node
DNN model	YOLOv4	Raspberry Pi 4 (Edge)
Programming based on	Python	
RAM in GB	4 GB	
Processor	ARM Cortex-72	

Table 2. Cont.

Item	Description	Node
Clock in GHz	1.5	
GPU	Pi 3 A+: Broadcom VideoCore IV; 400 MHz	
DNN model	Kmodel	Grove AI HAT (Edge)
Programming based on	MicroPython	
RAM in GB	0.008	
Processor	M1 K210 RISC-V	
Clock in GHz	0.4–0.6	
GPU	KPU	
Maximum training time	121 min	
Response time after training	2 s	
DNN model	RESNET-50, YOLOv4, MobileNet, VGG-16, SocialDistanceNet-19 etc.	Google Colab
Programming based on	Python	
RAM in GB	32	
Processor	Intel® Xeon® Gold 5318N	
Clock in GHz	2.25	
GPU	Tesla T4	

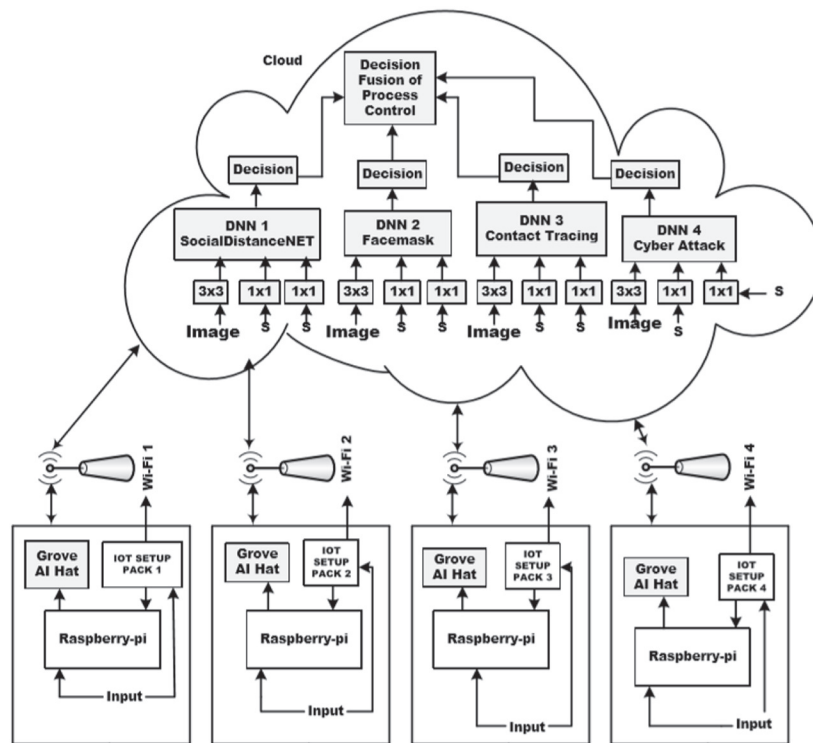


Figure 4. Schematic outlining the deployment of the edge computing nodes and connection with a cloud server as part of a residential block.

2.2. Proposed Hybrid Multi-Head Attention-Aided Multi-Tasking Deep Network with Diffusion Stability (HMAAHDNDS)

The central focus of our proposed work lies in the design of a Hybrid Multi-Head Attention-Aided Multi-Tasking Deep Network with Diffusion Stability (HMAAHDNDS) (Figure 5). At its core, the design incorporates several DPTMs, including RESNET-50, MobileNetV2, and SocialdistancingNet-19, each trained independently to address tasks such as face mask detection, proper face mask recognition, face mask classification, monitoring social distancing norms, and contact tracing (Table 3) [21,22]. Additionally, the HMAAHDNDS, as a composite setup, is equipped to perform tasks such as image generation from text inputs and the detection of cyber-attack types (Table 4). Figure 5 provides a detailed layout of this architecture. The incorporation of attention layers within the network facilitates optimized processing, outlier cancellation, and focused analysis of specific feature map details. Following that, several fully connected (FC) layers produce composite feature maps and text outputs, which are then fed into a generator network to trigger the production of images depicting violators, improper mask-wearing by specific individuals (samples), and contextual scenarios within the camera range connected to the IoT package. Furthermore, the setup is adept at detecting cyber-attacks, with each network core (DPTM) trained for this purpose. Details regarding the outcomes of cyber-attack detection will be discussed later. Operating on a cloud platform, each primary task, such as face mask recognition, social distancing monitoring, and contact tracing, is rigorously tested and elaborated on in subsequent sections. The inputs are images and sensor input. Images are colored, and sensor feeds are taken at intervals of 5 s. A summary of the key activities, inputs, output types and decision states of the proposed system is shown in Table 3.

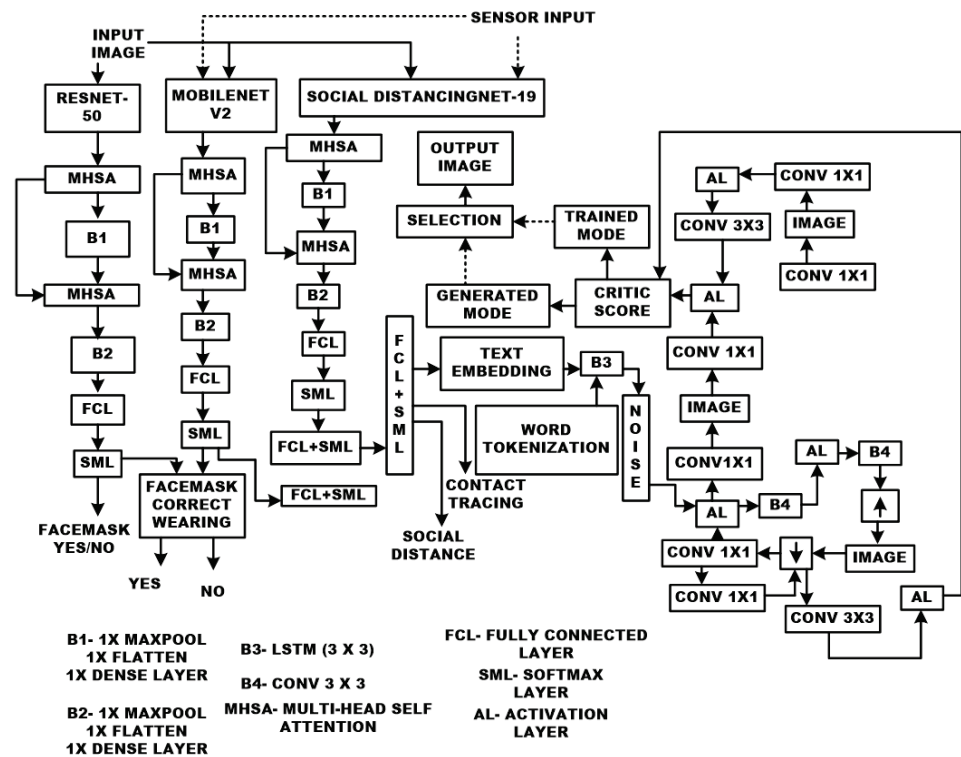


Figure 5. Hybrid Multi- Head Attention Aided Hybrid Deep Network with Diffusion Stability (HMAAHDNDS).

Table 3. Various activities and their decision states associated with the proposed HMAAHDNDS.

Sl. No.	Task	Input Types	Output Types
1	Face mask detection	Image	Yes/no binary decision
2	Proper face mask wearing	Image	Yes/no Male/female Indoor/outdoor
3	Classification of face mask	Image	Class 1, Class 2
4	Social distancing	Image and sensor data	Violation Yes/no
5	Contact tracing	Image and sensor data	Data
6	Cyber-attack	Text data, image	Class labels
7	Text-to-image generator	Text description from (1), (2), (3), (4) obtained in table and applied as CSV file	Image
8	Action detection	Image	Cough, sneeze

Table 4. Cyber-attacks associated with data-driven approaches of healthcare.

Sl. No.	Cyber-Attacks on Data Gathering Phase	Cyber-Attacks at Network Phase	Cyber-Attacks at Storage Phase
1	Phishing	Eavesdropping of health record	Cross site scripting
2	Log Access	Man in the middle attack	Weak authentication attack
3	Social Engineering Network	Data tempering	SQL injection attack
4	Brute force attack on passwords	Denial of the service attack	
		Data Interception	
		Spoofing and sniffing attack	

2.3. Face Mask Recognition Utilizing Multiple ML/DL Methods Deployed on Cloud Platforms

As discussed above, proper face-mask-wearing and recognition are crucial in the design of pandemic-compliant facilities and related infrastructures. For the design of a system that recognizes a proper face mask, it is necessary to configure an appropriate computer vision-based system with relevant decision support mechanisms. The fabrication of such a mechanism first in standalone form is key to the design of a pandemic-compliant infrastructure. It prepares a DPTM for such a task which is a part of the proposed hybrid block discussed in Section 2.2. The proposed method of proper face mask recognition is summarized using the block diagram shown in Figure 1.

The data used for training the learning-based methods are first separated for training, testing, and validation with a set of images having region of interests (RoIs) marked. Along with these, appropriately labeled samples are necessary for supervised learning. As already mentioned, we use several DPTMs, namely, ResNet-50 and MobileNetV2, for performing face mask detection to make the system suitable for real-world environments. A few other discrimination methods are used as benchmark approaches (Table 5). The details of the

different parameters of the benchmark methods and DPTMs used for face mask detection are shown in Table 5. Here, the term “batch size” typically refers to the number of samples processed in a single iteration of a ML/ DL network. Instead of feeding the entire dataset into the model at once, which can be computationally expensive and memory-intensive, the dataset is divided into smaller subsets called batches. Here, the layers used in the DPTMs for processing the inputs deal with the spatial resolution.

With face mask detection, the proper wearing of a face mask is also crucial. We perform a number of experiments to ascertain the correctness of wearing of the face mask. The methods and parameters used for this purpose are provided in Table 6.

The working of the system is based on certain steps. To start with, the pre-processing block prepares the datasets for the subsequent stages used. Kaggle datasets (<https://www.kaggle.com/mrviswamitrakaushik/facedatahybrid>, accessed on 12 December 2022) are adopted for the work. The dataset provides both faces with and without face masks. Around 7000 images (overall), of which 3725 depict people wearing face masks and 3828 depict people without them, are gathered into one set. Additionally, the images come in a variety of sizes, colors, backdrops, brightness levels, and contrast levels to suit any circumstance. There are two groupings made up of these datasets. The training set, which has roughly 6042 photos (80%), is one of them. The validation and testing datasets contain 755 photos (20%) each. The outputs of the pre-processing portions are applied to train the range of learning-based systems mentioned in Table 5. We use dropout rates as part of multi-modal factorized bilinear pooling (MFBP) with ResNet-50 and MobileNet after the training to avoid over-fitting. Then, to assess testing and compare the performance results before and after training, we employ the mean square error (MSE) as the cost factor and lower the learning rate using the adaptive moment estimation optimizer technique. Figure 1 shows the logic of the use of the DPTMs for face mask recognition. The related parameters and specifications are shown in Tables 5 and 6.

Table 5. Parameters of the different benchmark methods and DPTMs used for face mask detection (with MSE fixed at 10^{-6}).

Sl. No.	Method	Layer Type	Batch Size	Trainable Parameters	Epoch
1	ANN [23]	Pooling layer, dense layer	5	430,562	20
2	CNN	Pooling layer, dense layer	32	449,059	25
3	RF	Pooling layer, dense layer	32	425,256	15
4	DT	Pooling layer, dense layer	32	497,852	20
5	SVM [24]	Pooling layer, dense layer	32	422,285	30
6	RESNET-50	ResNet-50, pooling layer, dense layer	32	23,591,810	10
7	MOBILENET	MobileNetV2, pooling layer, dense layer	32	2,260,546	20
8	VGG-16 [19]	VGG-16-layer, pooling layer, dense layer	32	14,715,714	20

Table 6. Specifications of a few DPTM methods used for ascertaining the correctness of wearing facemask.

Sl. No.	Models	Trainable Parameters	Layer Type	Batch Size
1	MOBILENET	2,261,827	Conv2d, max pooling layer, dropout, dense layer, flatten layer	$7 \times 7 \times 1280$
2	RESNET-50	449,059		$244 \times 244 \times 32$
3	CNN	1,759,842		$244 \times 244 \times 32$

Similarly, we perform a series of experiments to ascertain the classification of face masks used by people during pandemic situations in public places (Figure 1), and the specifications of the classifiers used are summarized in Table 6. After a series of trials, it is found that the ResNet-50 performs best in terms of accuracy and processing time, and it becomes the main cloud-resident tool for carrying our decision support related to face masks.

2.4. Monitoring Compliance with Social Distancing Norms (SDNs)

To prevent the spread of infectious diseases, such as those transmitted through droplets and micro droplets, adherence to social distancing measures is crucial as recommended by the WHO [25]. Governments and concerned agencies require mechanisms to continuously monitor compliance with social distancing norms. Automated monitoring frameworks based on pattern recognition, object detection, and AI tools have been considered for this purpose. In object detection scenarios, popular methods include region-based and unified-based algorithms [23]. The Region-Based Convolutional Neural Network (RCNN), based on recurrent clustering, is effective but relatively slower due to the requirement to run procedures for each generated region [26]. An alternative, the Fast-RCNN, enhances speed by processing all regions simultaneously and utilizing a regional proposal network (RPN) instead of targeted search [27]. Improved performance is expected from several popular DPTMs, including MobileNetV2, ResNet-50, and Social DistancingNet-19 [24]. The logic for detecting compliance with social distancing norms is depicted in Figure 1, where inputs include images and video streams. Relevant parameters and configuration details are provided in Table 7.

The YOLO network, renowned for its capability of enabling real-time detection and enhancing the accuracy of identifying small objects, is a specific technology employed in a unified approach [28]. Another efficient method, the Single Shot Multi-box Detector (SSD), has also been suggested in [29]. These techniques utilize Convolutional Neural Networks (CNNs) to construct feature maps from images for object detection. By employing surveillance cameras installed in public areas, YOLO or SSD techniques can be utilized to detect individuals. A system linked to these cameras continuously monitors the live image stream, enabling the identification of individuals who violate social distancing regulations [30]. The YOLO network is particularly relevant to the edge nodes constituted by the Raspberry Pi 4 (Table 2).

The composite system also includes a social distance algorithm. As a result, once the server recognizes an infected person through a cough signal and a sneeze action, it notifies the decision support system, edge computing nodes, and the local populace, and triggers certain procedures linked with the social distancing guidelines. The cough and sneeze actions of a specific person are learned by the system, and if the person enters the gates of residential premises, it records the necessary vitals through the installed sensors. This process helps in placing the person under observation.

In Figure 1, the logical approach to the design and development of the smart social distancing system is summarized. The social distance observance is described by locating people wearing masks and then finding the physical separation between them.

Table 7. Parameters of various DL models for the continuous monitoring of social distance norms observance (MSE 10^{-6}).

Sl. No.	Model	Parameters					
		No. of Layer	Epoch	Optimizer	Training Dataset	Learning Objective	Composition of Layers
1	ResNet-50	Input layer-1 Zero padding layer-2 Relu layer-1 Dense layer-6	20	Adaptive moment estimation (Adam)	4098	Scale conjugate gradient	Conv layer 224 × 224 × 32 Max-pooling layer 112 × 112 × 32
2	MobileNetV2	Input layer-1 Zero padding layer-2 Relu layer-2 Dense layer-2	20	Adaptive moment estimation (Adam)	3843	Scale conjugate gradient	7 × 7 × 1280
3	SocialdistancingNet-19	Input layer-1 Zero padding layer-2 Relu layer-1 Dense layer-2	20	Adaptive moment estimation (Adam)	2598	Scale conjugate gradient	Conv layer 244 × 244 × 32 Maxpooling layer 112 × 112 × 32

The system initially focuses on detecting masks worn by individuals. Identifying a mask on a face serves as a confirmation of a person’s presence [31]. Subsequently, the distance between two individuals wearing masks is calculated through pixel-to-spatial separation conversion. Facial recognition is then employed to ascertain the identity of potential violators. A record of these violations is generated, and notifications are sent to relevant stakeholders, including the individuals involved. This mechanism is crucial for facilitating contact tracing. After conducting a series of standalone trials, it is ascertained that the SocialDistancingNet-19 is the most suitable model for monitoring compliance with social distancing norms. It is chosen as the cloud-resident tool to oversee adherence to these norms. In this capacity, it functions as a DPTM and undergoes further end-to-end learning using online samples received from edge nodes. At the edge node level, the system extracts, stores, and shares the initial features captured by video cameras. The higher-level inputs are then transmitted to the cloud-resident SocialDistancingNet-19, which continues training using inputs from all remaining feed points (as depicted in Figure 5).

2.5. Implementing Contact Tracing in Public Spaces and Residential Communities

During the peak period (2020–2022) of the severe acute respiratory syndrome caused by SARS-CoV-2, traditional contact-tracing methods became overwhelmed [32]. Public health organizations resorted to manual contact tracing by interviewing infected individuals to identify potential contacts. Exposed contacts were advised to seek medical attention, undergo treatment, and self-quarantine if necessary. While this strategy helped reduce transmission, it faced challenges and created opportunities for technological interventions to enhance reliability. Digital contact tracing emerged as a solution to scalability, notification delays, recall errors, and identifying contacts in public spaces. It utilizes electronic data to trace exposures to illnesses. Most COVID-19 contact-tracing apps estimate the proximity between smartphones using Bluetooth signal strength and determine exposure status based on proximity duration and distance from an infected individual. However, privacy and security concerns prompted debates over the architecture used for data gathering in tracking apps. Three main system architectures—centralized, decentralized, and hybrid—were employed for COVID-19 tracing applications, with the final architecture often blending elements of both centralized and decentralized approaches [33]. The proposed contact-tracing

technique’s block diagram is depicted in Figure 6. Seven alternative architectures (VGG-16, ResNet-50, MobileNetV2, SVM, DT, KNN, and LR) have been tested to determine the most reliable network for executing contact tracing within a cloud-resident framework integrated with intelligent edge nodes. In this approach, potential contacts are identified based on received inputs, initiating forward tracing and ascertaining infections. Infected individuals are isolated and treated, while symptom recognition during quarantine ensures continued care. Contacts of isolated cases are identified, and infected individuals are monitored for symptom severity to determine appropriate treatment levels. Quarantine, testing, and medication for contacts of infected individuals occur concurrently, with updates continuously logged in the database. Table 8 outlines the settings and configurations of the various ML and DL architectures utilized for executing the contact-tracing mechanism.

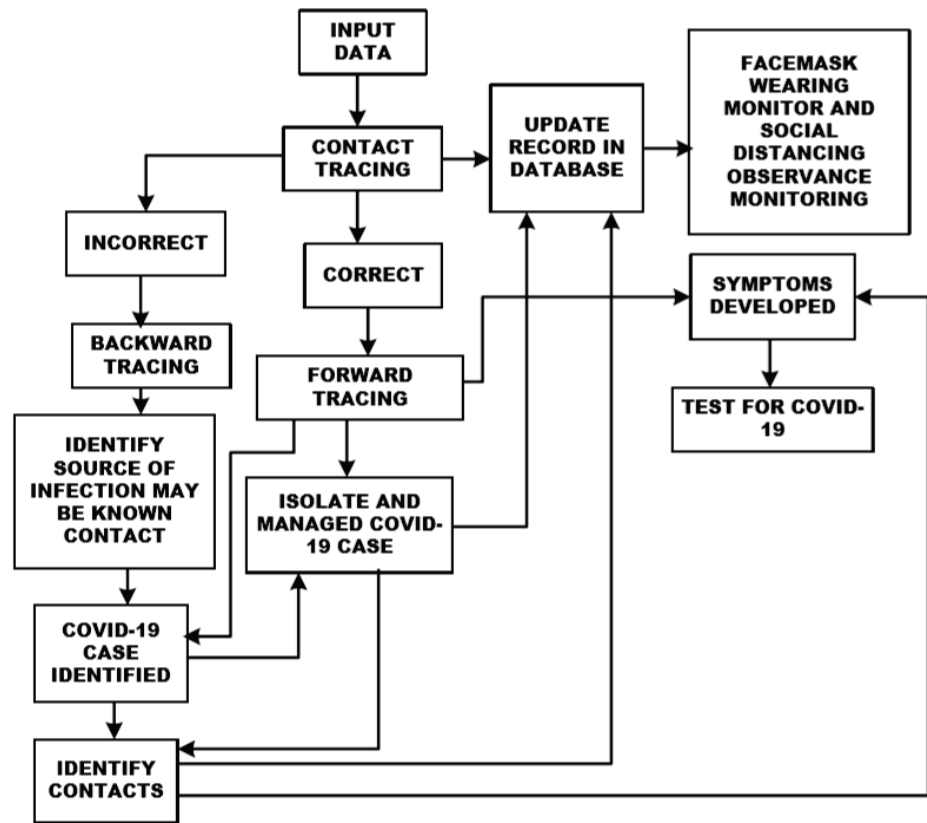


Figure 6. Block diagram of proposed contact-tracing technique.

Table 8. Parameters of different DL models for contact tracing (MSE 10^{-6}).

Sl. No.	Model	Parameters					
		No. of Layers	Training	Epoch	Optimizer	Learning Objective	Composition of Layers
1	ResNet-50	Input layer-1 Zero padding layer-2 Relu layer-1 Dense layer-6	4098	20	Adaptive moment estimation (Adam)	Scale conjugate gradient	Conv layer $224 \times 224 \times 32$; max-pooling layer $112 \times 112 \times 32$
2	MobileNetV2	Input layer-1 Zero padding layer-2 Relu layer-2 Dense layer-2	3843	25	Adaptive moment estimation (Adam)	Scale conjugate gradient	$7 \times 7 \times 1280$

Table 8. Cont.

Sl. No.	Model	Parameters					
		No. of Layers	Training	Epoch	Optimizer	Learning Objective	Composition of Layers
3	VGG-16	Input layer-1 Zero padding layer-2 Relu layer-1 Dense layer-2	2598	25	Adaptive moment estimation (Adam)	Scale conjugate gradient	Conv layer 244 × 244 × 32 max pooling layer 112 × 112 × 32
4	SVM	Conv2d, max	8450	20	Adaptive moment estimation (Adam)	Scale conjugate gradient	Conv layer 244 × 244 × 32 max pooling layer 122 × 122 × 32
5	DT	pooling, dropout,	8450	20			
6	KNN	dense layer,	8450	20			
7	LR	flatten layer	8450	20			

2.6. Text-to-Image Generation

Generating visuals from text is critical for graphically representing continuous monitoring data and conditions around residential premises. It helps to raise public awareness by illustrating difficult situations and gives training examples for improving network efficiency. To perform text-to-image creation, the input text is transformed into a meaningful representation, such as a feature vector, which is then used to generate an image that corresponds to the description.

The system initially receives text input and conducts tokenization to extract key textual features. Subsequently, images and text data from the dataset are combined before proceeding to the training phase. Here, the generator and discriminator play pivotal roles. The generator synthesizes images, while the discriminator classifies them. This iterative process continues until the generator can proficiently create images based on the input data. Ultimately, a trained model capable of producing visuals from the textual input is obtained [34]. The flow logic and constituent blocks are illustrated in Figure 5.

The Long Short-Term Memory (LSTM) layer is a crucial component within the block, executing sequence modeling and context retention. Within the system, the LSTM layer takes in text embeddings, which are then processed through a dense layer and concatenated with a latent vector for generating a response. To enrich the dataset, a normal distribution is employed for sampling additional data from the latent vector. The output of the dense layers undergoes reshaping and concatenation with the activation volumes of convolution blocks in the generator. This is achieved by passing the text embedding through various dense layers with outputs of varying sizes. The generator includes an up-sampling layer subsequent to a convolution block housing two 3 × 3 kernel-sized convolution layers. Additionally, a 1 × 1 convolutional layer with three kernels is applied to the activation volumes of the generator. Pixel-wise normalization is utilized within the generator. In the discriminator, after an average pooling layer, the convolution block comprises two convolution layers with 3 × 3 kernel sizes. Finally, the resulting output passes through a linear activation function, which determines whether the image is authentic or false.

2.7. Enhancing Cyber-Attack Detection

The risk of cyber-attacks looms large whenever a computer-driven system is connected to the broader internet network. In this context, the composite system’s cyber-attack detection capability is developed to continuously monitor data traffic, discerning unauthorized access or activities within a networked environment. Cyber-attacks can manifest in two primary phases—during training and testing. Several common cyber-attacks targeting IoT-based setups include phishing, log access breaches, social engineering attacks, and brute force attacks aimed at passwords [35].

The provision for the detection of cyber-attack has been incorporated into the system because of the fact that it is an information and communication technology (ICT)-driven

framework. Such a framework is bound to be affected at some point in time by a cyber-intruder, which may be external or internal. Moreover, there is always a possibility that mobile device access to information generated by the system should be added as an extension of the work. In such a situation, the cyber-threat will be a reality. Further, the internal cyber-threat will always be present.

Database creation is necessary, as the records of sensor readouts and decisions generated for individuals will continuously expand. As a result, cyber-threats will always be present. The system can be operated as a standalone system when it is not connected with mobile devices/apps. In the present form, the experiments are only intended to validate the system's capability against a generalized cyber-threat.

There are different types of cyber-threat detection methods. Here, ResNet-50 and MobilenetV2 are used as classifiers to detect cyber-attacks. However, Linear Regression (LR) and VGG-16 classifiers are taken as benchmark methods for a similar purpose to ascertain the capabilities of DPTMs. For this, a total of 30,190 (CSC2010HTTP) samples (<https://www.kaggle.com/datasets/omkargurav/face-mask-dataset/data>, accessed on 7 November 2021) are taken, of which 70% is used for training and 30% is used for testing. These samples are balanced. Social distancing and cyber-attack can be performed through a mobile app. Residential blocks require internet connectivity, while mobile devices are connected through the internet, and records are stored on the cloud. Therefore, the threat of cyber-attack will always be there. Cyber-attacks that are reported to be associated with data-driven approaches of healthcare monitoring systems are shown in Table 4.

3. Results and Discussion

The results obtained from the proposed approach have several components. We are highlighting the key outcomes obtained using different approaches adopted for the detection of face-mask-wearing, the observance of social distance norms, and contact tracing. Further, we report the impact analysis of the proposed work with regard to its deployment in a residential premise. The system's capacity to discriminate efficiently between cyber-attacks is been included. We report the results in a sequence, starting with the work related to face mask detection (Section 3.1), the correct or incorrect wearing of a face mask, as well as the classification of the face mask; the observance of social distance (Section 3.2); contact tracing (Section 3.3); impact assessment of the proposed architecture (Section 3.4); validation of the capabilities in detecting cyber-attacks within the proposed system (Section 3.5); validation of text-to-image transformation (Section 3.6); and performance of the edge nodes (Section 3.7).

3.1. Experimental Outcomes of Face Mask Detection

The cornerstone of an effective design for face mask detection hinges on various sub-systems within the system, with the reliability of the ML/DL-aided decision support system being paramount. In our approach, we leverage a spectrum of ML and DL techniques to identify the optimal method for dependable face mask detection. Our methodology also encompasses the capability to distinguish between correct and incorrect face mask usage. To validate our approach, a series of experiments are conducted to differentiate between appropriate face-mask-wearing. Training and testing are conducted using a dataset comprising over 6000 samples across multiple iterations. Approximately 80% of the data is allocated for training purposes, with the remaining data reserved for validation and testing. During real-time testing, samples gathered from field visits to potential deployment locations are utilized. Images of individuals both wearing and not wearing masks, as depicted in Figures 7a and 7b, respectively, are instrumental in training and testing DL models for face mask detection. Additionally, Figure 7c showcases datasets illustrating improper face mask usage, which serve as valuable resources for refining and enhancing face mask detection algorithms and models, constituting a crucial aspect of our ongoing efforts. These datasets contribute to the robustness of the DL models.

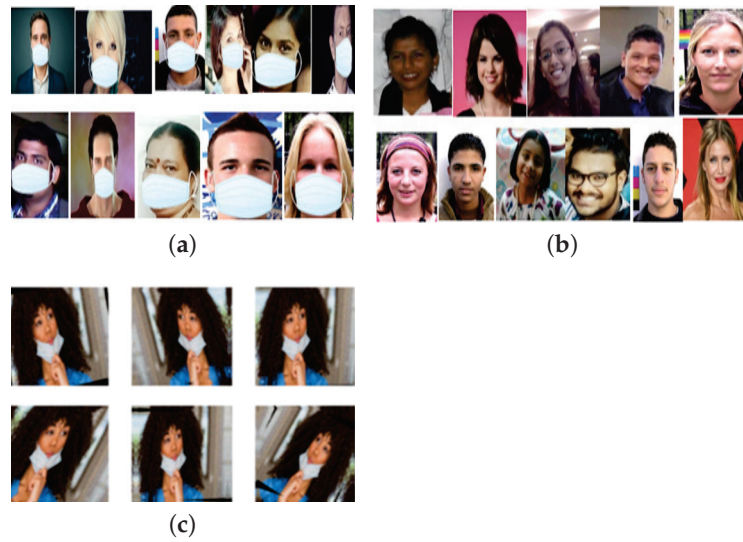


Figure 7. Examples of face masks datasets (a) with masks (b) without mask and (c) incorrect face mask.

Medical face masks are commonly crafted from materials like cloth, non-woven fabric, or disposable paper. They can be either disposable or reusable, serving as essential personal protective equipment (PPE) to safeguard both patients and healthcare workers against the transmission of infectious diseases. Consequently, medical face masks find extensive utilization in healthcare settings such as hospitals and clinics. Examples of datasets featuring normal masks and medical face masks are depicted in Figures 8a and 8b, respectively.



Figure 8. Datasets of (a) NG face masks and (b) Medical face masks.

Table 9 shows the summary results obtained from a class of methods trained to detect correct face mask wear. We clearly observe that VGG-16, conventional CNN, ResNet-50, and MobileNetV2 provide better reliability.

In our experiments concerning the detection of proper face mask wear through computer vision and learning-based techniques, we note that MobileNetV2 outperforms ResNet-50 in terms of accuracy, as evidenced by the summarized results presented in Table 10. Notably, Table 9 showcases a reported 100% accuracy by [15]. Upon implementing the approach detailed in [16], we confirm achieving 100% accuracy with small datasets. However, when working with larger datasets, this accuracy significantly decreases.

Table 9. Comparison results of eight different classifiers for mask-wearing detection (MSE 10^{-6}).

SL. No.	Model	Class	Recall	Support	Confusion Matrix	F1 Score	Precision	Accuracy (%)	Accuracy (%) (Previous Work)
1	ANN	0	0.33	761	[634 127]	0.81	0.79	80	75.5 [14]
		1	0.78	750	[168 5321]	0.80	0.82	80	
2	DT	0	0.71	729	[519 210]	0.70	0.68	70	100 [15]
		1	0.69	782	[240 542]	0.71	0.72	70	
3	RT	0	0.83	729	[605 124]	0.78	0.73	77	100 [15]
		1	0.72	782	[222 560]	0.75	0.82	77	
4	SVM	0	0.31	729	[594 135]	0.75	0.69	74	100 [15]
		1	0.67	782	[261 521]	0.72	0.79	74	
5	VGG-16	0	0.99	761	[588 5]	0.98	0.96	98	96 [16]
		1	0.96	750	[23 592]	0.98	0.99	98	
6	CNN	0	0.96	761	[568 25]	0.94	0.93	94	93.4 [16]
		1	0.96	750	[43 572]	0.94	0.96	94	
7	RESNET50	0	0.99	745	[651 94]	0.93	0.99	94	90.1 [16]
		1	0.89	765	[4 751]	0.94	0.89	94	
8	MOBILENET V2	0	0.92	756	[704 52]	0.93	0.92	92	92.64 [16]
		1	0.93	754	[62 692]	0.92	0.93	92	

Table 10. Comparative analysis of ResNet-50 and MobileNetV2 for incorrect-face-mask-wearing detection.

Sl. No.	Model	Class	F1 Score	Precision	Support	Recall	Confusion Matrix	Accuracy (%)
1	ResNet-50	0	0.88	0.99	147	0.80	[117 23 7]	92
		1	0.90	0.85	137	0.97	[1 133 3]	92
		2	0.96	0.93	131	0.99	[0 1 130]	92
2	MobileNetV2	0	0.91	0.92	147	0.87	[131 9 7]	97.2
		1	0.94	0.90	137	0.97	[3 133 1]	97.2
		2	0.92	0.94	131	0.90	[8 5 118]	97.2
3	ResNet-50 [31]	-	-	-	-	-	-	47.91
4	MobileNetV2 [30]	-	-	-	-	-	-	92.64

Additional experiments on face mask type detection: Comparing ResNet-50 and MobileNetV2. Table 11 illustrates the outcomes, showcasing ResNet-50’s superior accuracy over MobileNetV2. Our implementation of the work detailed in [36], with parameter adjustments and rigorous training, achieves the accuracy reported in Table 11.

Table 11. Assessment of performance ResNet-50 and MobileNetV2 in face mask classification.

Sl. No.	Method	Class	F1-Score	Confusion Matrix	Recall	Precision	Accuracy (%)
1	ResNet-50	0	0.93	[252 17]	0.94	0.92	94
		1	0.95	[22 357]	0.94	0.95	94
2	MobileNetV2	0	0.63	[146 123]	0.54	0.74	75
		1	0.79	[52 327]	0.86	0.73	75
3	CNN	0	0.74	[198 71]	0.71	0.74	79
		1	0.82	[68 311]	0.82	0.81	79
4	MobileNetV2 [30]	-	-	-	-	-	97.08
5	ResNet-50 [31]	-	-	-	-	-	92.49
6	CNN [20]	-	-	-	-	-	70

Comparison of ResNet-50 results with previous studies: Our study aligns with prior research, utilizing identical datasets, methodologies, and testing procedures as depicted in Table 12. Our ResNet-50-based model exhibits a 4% decrease in performance due to extensive data diversity, the inclusion of on-field samples, and variations in illumination. Anticipated enhancements in results are foreseeable with increased exposure to similar samples, especially those with complex backgrounds. Table 12 exclusively features studies employing ResNet-50, highlighting its widespread utilization in such applications. Our findings encompass outcomes under diverse conditions, including variations in illumination, background, and sensor capture quality. We conduct comprehensive model implementations across various datasets and diversity settings, yielding comparable results. Figures 9–11 illustrate model accuracies, training time losses for ANN, CNN, VGG-16, MobileNetV2, and ResNet-50, and classification performance metrics for ResNet-50 and MobileNetV2.

Table 12. Performance evaluation of ResNet-50 with the proposed methods (face-mask-wearing).

Reference	Model	Classification	Accuracy (%)
[1]	ResNet-50	YES	89
[2]	ResNet-50	YES	81
[3]	ResNet-50	YES	95.8
[4]	ResNet-50	YES	88.2
[6]	ResNet-50	YES	98.6
[7]	ResNet-50	YES	97.9
[8]	ResNet-50	YES	92.5
This work	Proposed Model	YES	93.5

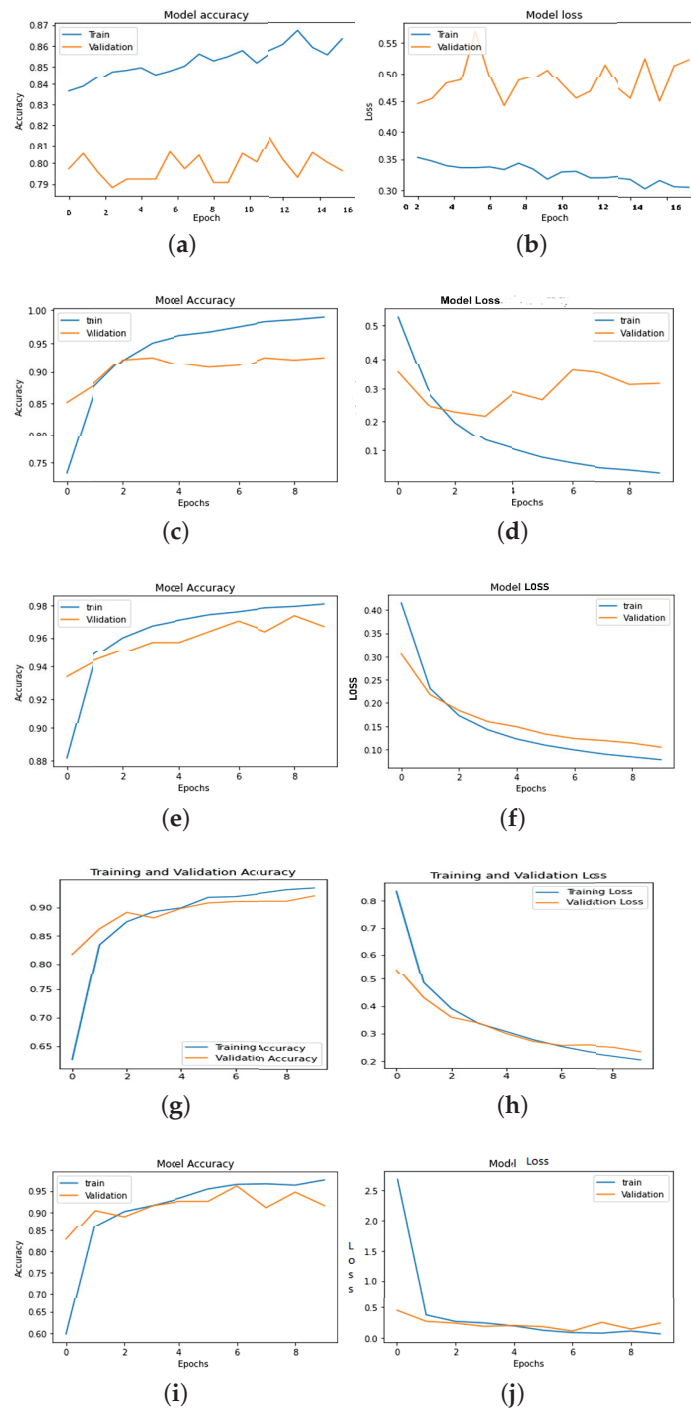


Figure 9. Model accuracy and loss of face-mask-wearing (a,b) ANN, (c,d) CNN, (e,f) VGG-16, (g,h) MobileNetV2 and (i,j) ResNet50.

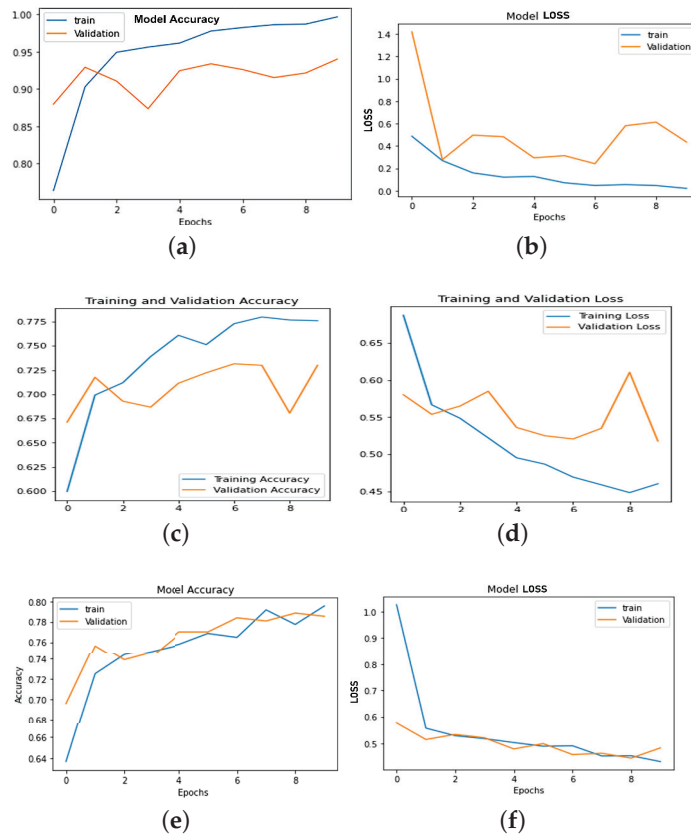


Figure 10. Face mask classification accuracy and loss: (a,b) ResNet-50, (c,d) mobileNetV2, (e,f) CNN.

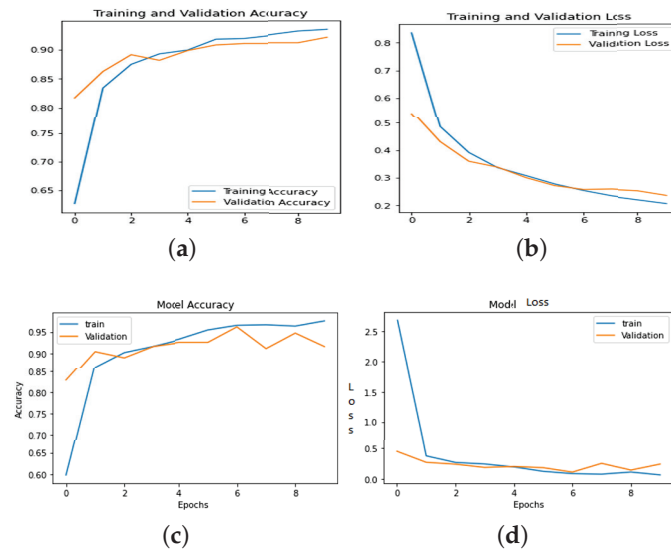


Figure 11. Incorrect face-mask-wearing accuracy and loss (a,b) MobileNetV2 (c,d) ResNet-50.

3.2. Evaluation of Observance of Social Distancing Measures

Social distancing refers to measures aimed at preventing the spread of infectious diseases, such as COVID-19, by reducing close contact between individuals. This can include staying at least 6 feet away from others, avoiding crowded places, working from home, and avoiding non-essential travel (as per WHO guidelines). The goal of social

distancing is to slow down the transmission of disease and reduce the burden on the healthcare system. For the monitoring of social distance observance, the camera is the most obvious sensor. In our work, with the camera, effective performance is obtained up to 20 m. The accuracy and F1-score are shown in Table 13.

Table 13. Accuracy and performance of the monitoring of the observance of social distance norms.

Sl. No.	Action	Model	Performance	
			F1-Score	Accuracy (%)
1	Person detection	ResNet-50	0.90	87.6
		MobileNetV2	0.95	89.9
2	Social distancing	SocialdistancingNet-19	0.89	94.5
3	[23]	SocialdistancingNet-19	–	92.8

We conduct experiments to detect the observance of social distancing norms. Table 13 summarizes the average performance for person detection and social distancing using specific methods. The model training accuracies for ResNet-50, MobileNetV2, and SocialDistancingNet-19 are compared, with SocialDistancingNet-19 achieving the highest accuracy at 94.5%. Table 7 displays the parameters of different DL methods employed for social distancing, reflecting the combinations and tuned states of the models utilized in this study. Ultimately, SocialDistancingNet-19 emerges as the most suitable option and is deployed as a cloud-resident network for social distance decision support.

3.3. Contact Tracing: Tracking and Managing Exposure Risks

The objective of contact tracing is to mitigate virus transmission, avert outbreaks, and decelerate disease spread. To facilitate contact tracing, the VGG-16, MobileNetV2, and ResNet-50 networks undergo training using pre-existing data. Table 5 outlines the configurations of these networks employed for this purpose. Current data are continually updated with previously known details, enriching the training process. Among these models, ResNet-50 achieves an accuracy of 96.1%. Table 14 illustrates a comparison of the results attained from proposed methods, encompassing VGG-16, MobileNetV2, ResNet-50, and several other previously reported methods.

Table 14. Comparison of results obtained from proposed method with previously reported works (contact tracing).

Sl. No.	Model	Accuracy of Previously Reported Work (%)	Accuracy (%)
1	VGG-16 [29]	89.6	92.3
2	MobileNetV2 [30]	78.5	93.4
3	ResNet-50 [25]	83.4	96.1
4	SVM [31]	91	94.6
5	DT [32]	87	91.3
6	KNN [32]	88	92.5
7	LR [32]	89	93.6

3.4. Impact Assessment of the Proposed System Architecture

Table 15 provides a comprehensive overview of the optimal performance of the proposed multi-node edge and cloud computing setup tailored for residential premises. It encompasses the analysis of deviations in sensor readings, decision support for face mask recognition, adherence to social distancing norms, contact-tracing capabilities, and latency response across various load conditions at both the node and system levels [37]. These deviations stem from the diverse array of tasks managed by each node, the continual variation in the sample types, the operation of different network types, and synchronization challenges. Despite these complexities, the system exhibits reliability. However, the primary bottleneck emerges from Wi-Fi access limitations and the performance of pickup devices. Notably, varying error levels occur at both the node and server levels. Learning and decision-making processes are shared between the nodes and the server. Errors may occur when an individual, who typically utilizes a specific gate on one day, appears at a different gate. In such instances, the server may make a true negative decision, which gradually stabilizes over time toward the desired outcome.

Table 15. Summary of optimal performance in a multi-node edge and cloud computing configuration for residential premises.

Sl. No.	Block	Specific Item	Average Error at System Level	Average Error at Node Level
1	Sensor type	Camera	3.80%	4%
		IR thermometer	3.50%	5%
		Proximity sensor	2.80%	3%
		Ultra-sonic	2.40%	3%
2	Decision support for face mask	Low (30 people/hour)	8.5%	9.2%
		Medium (50 people/hour)	8.8%	11%
		High (70 people/hour)	9.3%	12%
3	Decision support for social distance	Low (30 people/hour)	3.40%	9%
		Medium (50 people/hour)	4.50%	12%
		High (70 people/hour)	4.50%	13%
4	Decision support for contact tracing	Low (30 people/hour)	3.80%	8.5%
		Medium (50 people/hour)	3.40%	10%
		High (70 people /hour)	4.50%	12%
5	Latency variation (average)	Low (30 people/hour)	7.5%	9%
		Medium (50 people/hour)	11%	14%
		High (70 people/hour)	14%	15%

Table 16 presents the analysis of the response variability in social distance enforcement by SOCIALDISTANCINGNET-19. It highlights significant differences in compliance across demographics and geographic regions. This variability underscores the influence of social norms and local policies on individuals’ adherence to distancing measures. Table 17 presents a comprehensive overview of the effectiveness of various strategies implemented in the proposed system, evaluated in residential settings over a six-month period. The post-validation and deployment phases demonstrate response speeds (in milliseconds) suitable for real-world applications, despite extensive training and testing durations (in seconds). Similarly, a detailed analysis is conducted regarding social distance measurements using

the DPTM SOCIALDISTANCINGNET-19. The system’s results exhibit slight discrepancies due to inaccuracies in the distance calibration and spatial calculations derived from the pixel values. The DPTM exhibits a maximum error margin of 12% between the projected and actual distances. The devastation caused by the COVID-19 pandemic between 2020 and 2022 underscores the importance of mitigating the spread of contagious diseases by minimizing interpersonal contact. Hence, the development of an intelligent infrastructure capable of the continuous monitoring of social distance compliance [38], proper face mask usage, and contact tracing is imperative, particularly in high-risk areas.

The proposed method demonstrates high accuracy rates for face mask detection, ranging from 90% to 96% across different DL models. This facilitates reliable surveillance in crowded environments and provides guidance on correct mask usage. Additionally, the system employs human detection techniques, achieving accuracies of 87% with ResNet-50 and 90% with MobileNetV2, enhancing social distance monitoring reliability. Moreover, the contact-tracing accuracy rates using VGG-16, MobileNetV2, and ResNet-50 are reported as 92%, 93%, and 96% respectively, ensuring robustness and dependability. The collective effectiveness of each component underscores the system’s capability to facilitate continuous COVID-related behavioral monitoring as part of an intelligent infrastructure compliant with pandemic regulations. Given the challenges and limitations associated with manual intervention, the proposed system emerges as pivotal in the creation of intelligent human environments, offering a solution that is efficient, safe, and accurate.

Table 16. Analysis of computational complexity across various DL methods.

Sl. No.	Specific Purpose	Training Dataset (80%)	Testing Dataset (20%)	Technique	Response Time (ms)	Testing Time (s)	Training Time (s)
1	Face Mask Wearing	6042	1510	ANN	22.4	446	1790
				CNN	24.05	432	1875
				RF	20.9	446	1700
				SVM	22	430	1750
				DT	22.4	446	1790
				VGG-16	141.75	775	9300
				RESNET-50	133	770	8750
				MOBILENET	137.5	775	9025
2	Improper wearing of Face Mask	1664	415	RESNET-50	36.67	800	3000
				MOBILENET	41.33	820	3300
3	Classification of Face Mask	2598	648	CNN	24.58	475	1950
				RESNET-50	39.25	645	3000
				MOBILENET	42.16	670	3200
4	Social Distance Detection	4098	970	RESNET-50	60.5	570	4200
		3843	705	MOBILENET	65.16	590	4500
		2598	648	SOCIAL DISTANCING NET-19	58.83	570	4100
5	Contact Tracing	8450	2050	DT	112.83	430	7200
		8450	2050	RF	112.83	430	7200
		8450	2050	LR	112.83	430	7200
		8450	2050	SVM	112.83	430	7200
		2598	648	VGG-16	25.67	460	2000
		3843	705	MOBILENET	36.33	470	2650
		4098	970	RESNET-50	38.5	490	2800

Table 17. Analysis of response variability in social distance enforcement by SOCIALDISTANCINGNET-19.

Sl. No.	Samples	k-Value	Actual Distance (cm) (A)	Distance from k-Value (cm) (B)	% Error Between (A) & (C)	Value Predicted by AI Technique (cm) (C)	Dimension	Pixel Size (mm)
1	1	85.7	210	200	10.0	189	224 × 224 × 32	2.565
	2	43.85	220	210	9.3	199.5	112 × 112 × 32	5.131
	3	44.54	195	185	9.9	175.7	256 × 256 × 64	4.49
2	1	0.051	161	150	11.8	142	7 × 7 × 1280	3284
	2	63.48	182	170	11.3	161.5	96 × 96 × 32	2.993
	3	91.31	200	185	12.0	176	256 × 256 × 32	2.245
3	1	82.8	180	170	10.6	161	244 × 244 × 32	2.355
	2	81.87	205	195	9.8	185	224 × 224 × 32	2.565
	3	97.99	210	200	10.0	189	512 × 512 × 64	2.245

3.5. Validation of Capabilities in Detecting Cyber-Attacks Within the Proposed System

The confusion matrix of various classifiers tested for malicious and normal traffic is shown in Table 18. A total of 30,190 samples are taken for detecting cyber-attacks in the proposed system. The accuracy rates for cyber-attack using MobileNetV2, and ResNet-50 are 94.89% and 90.07%, respectively. However, VGG-16 is used as a benchmark to train this model, and the summary performance is shown in Table 19.

Table 18. Comparing confusion matrices from different classifiers (cyber-threat detection).

Sl. No.	Classifier	Predicted		
			Normal	Attack
1	Linear Regression (benchmark)	normal	32,421	56
		attack	145	22,415
2	VGG-16 (benchmark)	normal	32,545	23,864
		attack	125	56
3	MobileNetV2	normal	32,572	48
		attack	125	23,864
4	ResNet-50	normal	30,071	2549
		attack	119	23,870

Table 19. Assessment of the proposed model’s performance in predicting malicious activity.

Sl. No.	Pre-Trained Model	Parameters				
		Sensitivity (%)	Precision (%)	F-Measure (%)	Specificity (%)	Accuracy (%)
1	ResNet-50	90.66	74.26	75.73	97.07	90.07
2	MobileNetV2	93.93	90.56	82.12	97.82	94.89
3	VGG-16(benchmark)	92.32	74.05	80.35	96.28	88.36

3.6. Validation of Image Output from Text Input Within the Proposed System

The ability of the system to provide image output from text input is shown in Figure 12. For four different text descriptions, how the composite system is able to generate the corresponding representative images is shown in Figure 12. The text embedding that

contains the decision states obtained from the DPTMs is shown in Table 20. These text descriptions assist the text-to-image transformer in using the principles of diffusion stability to provide synthetic visual representations related to bio-safety norms as part of a pandemic-compliant, resilient, and perceptive framework.

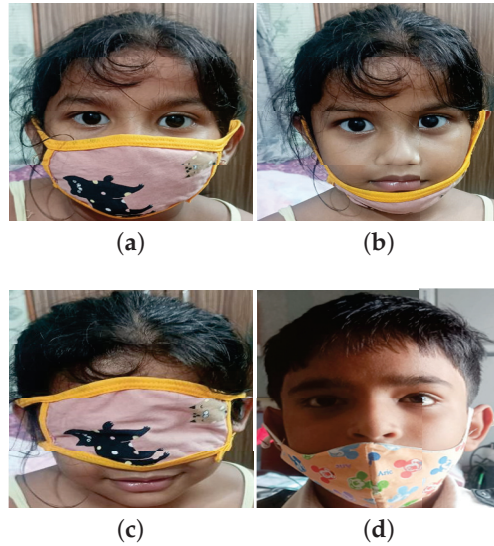


Figure 12. Text-to-image transform for four different cases of wearing masks: (a) correctly wearing a mask, (b) mask under mouth, (c) mask above mouth, and (d) mask under nose.

Table 20. Generation of text description from the decision states of the DPTMs.

Sl. No.	Face Mask Detection		Proper Face-Mask-Wearing		Classification of Face Mask		Social Distancing		Text Description
	Y1	N1	Y2	N2	Y3	N3	Y4	N4	
1	0	0	0	0	0	0	0	0	Null
2	0	0	0	0	0	0	0	1	Social Distance Not Violated
3	0	0	0	0	0	0	1	0	Social Distance Violated
4	0	0	0	0	0	0	1	1	Null
5	0	0	0	0	0	1	0	0	Not Proper Face Mask
6	0	0	0	0	0	1	0	1	Not Proper Face Mask Social Distance Not Violated
7	0	0	0	0	0	1	1	0	Not Proper Face Mask Social Distance Violated
8	0	0	0	0	0	1	1	1	Null
9	0	0	0	0	1	0	0	0	Proper Face Mask
10	0	0	0	0	1	0	0	1	Proper Face Mask Social Distance Violated

The key attributes of the proposed method compared to previously reported works are summarized in Table 21. The advantage of the proposed method is obvious.

Table 21. Key attributes of the proposed method vis-a-vis previously reported works.

Attributes	[6]	[7]	[30]	[31]	[27]	[26]	[18]	[23]	[34]	[35]	[36]	[37]	Proposed Method
Face Mask Detection	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y
Proper Face-Mask-Wearing	N	Y	N	N	Y	N	N	N	N	N	N	N	Y
Classification of Face Mask	N	N	Y	Y	N	N	N	Y	N	N	N	N	Y
Social Distancing	N	N	N	N	N	N	Y	N	N	N	N	N	Y
Contact Tracing	Y	Y	N	N	N	N	N	N	N	N	N	N	Y
Cloud Computing	N	N	Y	N	Y	Y	N	Y	Y	Y	Y	Y	Y
Edge Computing	N	Y	N	Y	N	N	Y	N	N	N	N	N	Y
Text-to-Image converter	N	N	N	N	N	N	N	N	Y	Y	N	N	Y
Cyber-Attack Detection	N	N	N	N	N	N	N	N	N	N	Y	Y	Y

3.7. Performance of the Edge Nodes

The edge nodes are tasked with taking temperature readings, face mask detection, and the observance of social distance norms. Contact tracing and cyber-attack detection are performed at the server level. The summary results of face mask detection (wearing or not, proper face-mask-wearing, and appropriate face-mask-wearing) and social distance observance are shown in Table 22. Here, YOLOV4 is used to perform the tasks.

Table 22. Average performance of four edge nodes in executing facemask detection, and observance of social distance norms (using YOLOV4).

Task	Accuracy (%)	F1-Score	Precision
Face mask Detection	97.1	0.92	0.85
Social Distance norms	95.2	0.88	0.79

4. Conclusions

We have presented the description of a system architecture comprising edge computing nodes, an IoHT system equipped with sensor packs, and a composite attention-driven framework integrating various DPTM such as RESNET-50, MobileNetV2, and SocialdistancingNet-19. This framework serves the monitoring of protocol adherence, including social distancing norms and mask-wearing, as well as contact tracing utilizing samples from open-source databases. Additionally, the composite framework generates scene images and context, and identifies violators with text descriptions. Furthermore, it detects cyber-attacks when connected to public networks. Configured and trained to effectively address real-world challenges within residential premises, the constituent systems undergo rigorous testing. Notably, the accuracy rates for detecting mask presence and proper usage are 92% (ResNet-50) and 97.2% (MobileNetV2). However, facial recognition encounters difficulties when faces are inclined at certain angles, regardless of mask usage, necessitating the resolution of this issue. To discern between fabric masks and N-95 masks, ResNet-50 and MobileNetV2 are employed. Parallelism enables the simultaneous execution of mask categorization and social distancing calculations. Despite the system’s robustness, insufficient and non-diverse datasets limit its adaptability. For instance, an inadequate representation of beards may lead to occasional confusion with masks. Access to more comprehensive datasets could facilitate the training of more competent models. Contact tracing for COVID-19 and influenza involves identifying individuals with significant contact with confirmed virus carriers and monitoring them for symptoms. These data aid in

virus transmission prevention and outbreak management. Public health experts leverage contact-tracing results to understand virus spread within communities and inform pandemic response strategies. System performance is constrained by network latency, varying with traffic load. Enhancing bandwidth and physical connectivity effectively mitigates this limitation.

Author Contributions: Conceptualization, A.C. and K.K.S.; methodology, A.C., K.K.S. and D.D.M.; software, A.C. and D.D.M.; validation, A.C., K.K.S., K.G. and J.I.; formal analysis, K.K.S. and J.I.; data curation, A.C., K.K.S., K.G. and J.I.; writing—original draft preparation, A.C.; writing—review and editing, A.C., K.K.S. and J.I.; visualization, K.K.S.; supervision, K.K.S., K.G. and J.I.; funding acquisition, A.C. and J.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by SERB, DST, Government of India, under the project reference no. SUR/2022/001704.

Data Availability Statement: <https://www.kaggle.com/datasets/omkargurav/face-mask-dataset/data>, 7 November 2021.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Turni, C.; Lefringhausen, A. COVID-19 vaccines—An Australian Review. *J. Clin. Exp. Immunol.* **2020**, *7*, 491–508.
- Abid, H.; Mohd, J.; Raju, V. Effects of COVID 19 pandemic in daily life. *Curr. Med. Res. Pract.* **2020**, *10*, 78–79.
- Wang, Y.; Deng, Z.; Shi, D. How effective is a mask in preventing COVID-19 infection? *Med. Devices Sens.* **2021**, *4*, e10163. [[CrossRef](#)] [[PubMed](#)]
- Kwon, O. Evidence of the importance of contact tracing in fighting COVID-19. *Epidemiol. Health* **2022**, *44*, e2022006. [[CrossRef](#)] [[PubMed](#)]
- Jones, K.; Thompson, R. To use or not to use a COVID-19 contact tracing app: Mixed methods survey in Wales. *JMIR mHealth uHealth* **2021**, *9*, e29181. [[CrossRef](#)] [[PubMed](#)]
- Saidi, O.; Malouche, D.; Saksena, P.; Arfaoui, L. Impact of contact tracing, respect of isolation, and lockdown in reducing the number of cases infected with COVID-19. *Int. J. Infect. Dis.* **2020**, *113*, 26–33. [[CrossRef](#)] [[PubMed](#)]
- Stuart, R.M.; Abeysuriya, R.G.; Kerr, C.C.; Mistry, D.; Klein, D.J.; Gray, R.T.; Hellard, M.; Scott, N. Role of masks, testing and contact tracing in preventing COVID-19 resurgences: A case study from New South Wales, Australia. *BMJ Open* **2021**, *11*, e045941. [[CrossRef](#)]
- Prasad, A.; Kotz, D. ENACT: Encounter-based architecture for contact tracing. In Proceedings of the 4th International on Workshop on Physical Analytics, Niagara Falls, NY, USA, 19–23 June 2017; pp. 37–42.
- Vinuesa, R.; Azizpour, H.; Leite, I.; Balaam, M.; Dignum, V.; Domisch, S.; Felländer, A.; Langhans, S.D.; Tegmark, M.; Fuso Nerini, F. The role of artificial intelligence in achieving the Sustainable Development Goals. *Nat. Commun.* **2020**, *11*, 233. [[CrossRef](#)] [[PubMed](#)]
- World Health Organization. *Water, Sanitation, Hygiene and Waste Management for COVID-19: Technical Brief*; (No. WHO/2019-NCoV/IPC_WASH/2020.1); World Health Organization: Geneva, Switzerland, 2020.
- Ellis, R. WHO changes stance, says public should wear masks. Available online: <https://www.webmd.com/lung/news/20200608/who-changes-stance-says-public-should-wear-masks> (accessed on 10 December 2020).
- Jahromi, R.; Mogharab, V.; Jahromi, H.; Avazpour, A. Synergistic effects of anionic surfactants on coronavirus (SARS-CoV-2) virucidal efficiency of sanitizing fluids to fight COVID-19. *Food Chem. Toxicol.* **2020**, *145*, 111702. [[CrossRef](#)]
- Feng, S.; Shen, C.; Xia, N.; Song, W.; Fan, M.; Cowling, B.J. Rational use of face masks in the COVID-19 pandemic. *Lancet Respir. Med.* **2020**, *8*, 434–436. [[CrossRef](#)] [[PubMed](#)]
- World Health Organization. *Advice on the Use of Masks in the Context of COVID-19: Interim Guidance*; No. WHO/2019-nCoV/IPC_Masks/2020.4; World Health Organization: Geneva, Switzerland, 2020.
- Prajapati, A.C.; Patel, M.M.; Khanpara, H.J.; Shukla, R.P.; Christian, D.S.; Geriya, D.R. A Hospital based cross sectional study to find out factors associated with disease severity and length of hospital stay in COVID-19 patients in Tertiary Care Hospital of Ahmedabad city. *Indian J. Community Health* **2021**, *33*, 256–259. [[CrossRef](#)]
- Qin, B.; Li, D. Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. *Sensors* **2020**, *20*, 5236. [[CrossRef](#)] [[PubMed](#)]
- Ejaz, M.S.; Islam, M.R.; Sifatullah, M.; Sarker, A. Implementation of principal component analysis on masked and non-masked face recognition. In Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 3–5 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
- Pfitzer, B.; Steckhan, N.; Anrich, B. Federated Learning in a Medical Context: A Systematic Literature Review. *ACM Trans. Internet Technol.* **2021**, *21*, 50. [[CrossRef](#)]

19. Murphy, K.P. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning Series)*; The MIT Press: London, UK, 2018.
20. Din, N.U.; Javed, K.; Bae, S.; Yi, J. A novel GAN-based network for unmasking of masked face. *IEEE Access* **2020**, *8*, 44276–44287. [[CrossRef](#)]
21. Hariri, W. Efficient masked face recognition method during the covid-19 pandemic. *Signal Image Video Process.* **2022**, *16*, 605–612. [[CrossRef](#)] [[PubMed](#)]
22. Leung, N.H.; Chu, D.K.; Shiu, E.Y.; Chan, K.H.; McDevitt, J.J.; Hau, B.J.; Yen, H.L.; Li, Y.; Ip, D.K.; Peiris, J.S.; et al. Respiratory virus shedding in exhaled breath and efficacy of face masks. *Nat. Med.* **2020**, *26*, 676–680. [[CrossRef](#)] [[PubMed](#)]
23. Keniya, R.; Mehendale, N. Real-Time Social Distancing Detector Using SocialdistancingNet-19 Deep Learning Network. Available at SSRN 3669311 2020. Available online: <https://ssrn.com/abstract=3669311> (accessed on 11 August 2020).
24. Park, J.J.; Jung, J.; Eun, S.; Yun, S.Y. UI elements identification for mobile applications based on deep learning using symbol marker. *J. Inst. Internet Broadcast. Commun.* **2020**, *20*, 89–95. [[CrossRef](#)]
25. Ahmed, I.; Ahmad, I.M.; Jeon, G. Social distance monitoring framework using deep learning architecture to control infection transmission of COVID-19 pandemic. *Sustain. Cities Soc.* **2021**, *69*, 102777. [[CrossRef](#)] [[PubMed](#)]
26. Hussain, S.; Yu, Y.; Ayoub, M.; Khan, A.; Rehman, R.; Wahid, J.A.; Hou, W. IoT and deep learning based approach for rapid screening and face mask detection for infection spread control of COVID-19. *Appl. Sci.* **2021**, *11*, 3495. [[CrossRef](#)]
27. Snyder, S.E.; Husari, G. Thor: A deep learning approach for face mask detection to prevent the COVID-19 pandemic. In Proceedings of the Southeast Con 2021, Atlanta, GA, USA, 10–13 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
28. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. Available online: <http://pjreddie.com/yolo/> (accessed on 17 October 2024).
29. Vimal, C.; Shirivastava, N. Face and Face-mask Detection System using VGG-16 Architecture based on Convolutional Neural Network. *Int. J. Comput. Appl.* **2022**, *975*, 8887. [[CrossRef](#)]
30. Faisal, A.; Dharma, T.N.A.; Widayat, W.; Pasfica, G.R.; Satura, F.R. Comparative study of VGG16 and MobileNetv2 for masked face recognition. *J. Ilm. Tek. Elektro Komput. Dan Inform (JITEKI)* **2021**, *7*, 230–237.
31. Utomo, M.; Violita, F. Face Mask Wearing Detection Using Support Vector Machine (SVM). *Int. J. Informatics Dev.* **2021**, *10*, 72–81.
32. Naezullaev, A.; Muminov, Z.; Narzullaev, M. Contact Tracing of Infectious Diseases using Wi-Fi signals and Machine Learning classifications. In 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAJET), Kota Kinabalu, Sabah, 26–27 September, 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
33. Buccafurri, F.; De Angelis, V.; Labrini, C. A centralized contact-tracing protocol for the COVID-19 pandemic. *Inf. Sci.* **2022**, *617*, 103–132. [[CrossRef](#)]
34. Alzubi, A.A.; Matiah, M.A.; Alarifi, A. Cyber-attack detection in healthcare using cyber-physical system and machine learning system. *Soft Computing* **2021**, *25*, 12319–12332. [[CrossRef](#)]
35. Khriji, L.; Bouaafia, S. Secure Convolutional Neural Network-Based Internet-of-Healthcare Applications. *IEEE Access* **2023**, *11*, 36787–36804. [[CrossRef](#)]
36. Ku, H.; Lee, M. Text Control GAN: Text to image synthesis with controllable Generative Adversarial Networks. *Appl. Sci.* **2023**, *13*, 98. [[CrossRef](#)]
37. Sawant, R.; Shaikh, A.; Sabat, S.; Bhole, V. Text to Image Generation using GAN. In Proceedings of the 2nd International Conference on IoT Based Control Networks and Intelligent Systems, Kerala, India, 28–29 June 2021.
38. Loke, C.H.; Adam, M.S.; Nordin, R.; Abdullah, N.F.; Abu-Samah, A. Physical distancing device with edge computing for COVID-19 (PADDIE-C19). *Sensors* **2021**, *22*, 279. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.