Chapter 1: Containing the Spread of COVID-19 with IoT: A Visual Tracing Approach

Authors:
*Pallav Kumar Deb (Indian Institute of Technology Kharagpur, pallv.deb@iitkgp.ac.in)
Sudip Misra (Indian Institute of Technology Kharagpur, sudipm@iitkgp.ac.in)
Anandarup Mukherjee (University of Cambridge, anandarupmukherjee@ieee.org)
Aritra Bandopadhyay (Manipal Institute of Technology, aritraban10@gmail.com)

Abstract:

In this chapter, we propose COVI-SCANNER, an IoT-based solution to overcome the issue of secondary spread of the virus from fomite spaces in public infrastructures. With the assumption that the infected individuals are already identified, we design COVI-SCANNER as two augmented reality-based phases: contamination and sanitization. In the contamination phase, we first detect the person and the corresponding bounding boxes for tracking his/her movements and highlight the spaces as they contaminate it. In the sanitization phase, we first identify cleaning/sanitization materials/objects and remove the markers as they pass (while cleaning) through the highlighted fomite spaces to ensure sanitization. Additionally, we minimize the delay by dividing the resource-constrained fog nodes into two dedicated sets for performing each phase before transferring to centralized servers for storage, which also reduces bottlenecks at the server. Through the extensive implementation of COVI-SCANNER, we observe that it operates with 81% accuracy on real-time data with a delay of at most 0.1 and 1.2 seconds for the contamination and sanitization phases, respectively. Further, due to the incorporation of the IoT-based architecture, we observe maximum upload and download rates of 700 Kbps in each phase.

1.1 INTRODUCTION

The outbreak of the COVID-19 virus has spread at an alarming rate all around the world. The intangible nature of the virus and its transmission mode makes containment challenging. Moreover, the droplets from an infected individual remain remnant in their hands in addition to transmission in the form of aerosols, which leads to contamination of public objects and spaces on interaction (touching and passage). The COVID-19 virus survives for almost 3 days on surfaces made of steel and plastic, and over 24 hours on cardboards (Chamola 2020). This gives rise to fomite spaces and such intractable contamination rapidly facilitates secondary spreading, especially in closed public environments. As a healthy person interacts with these fomite spaces, the virus enters into the body and binds itself to cellular receptors and starts multiplying, leading to fatality. Such a high risk of secondary transmission mandates the need for rigorous contact tracing (Gan 2020). In such scenarios, Internet of Things (IoT)-based solutions have the potential to combat the COVID-19 virus by tracking and containing the fomite spaces. The use of smart solutions based on image and video processing for tracking fomite spaces is beneficial in restricting secondary spreading in closed environments. Such image processing-based solutions require high configuration devices for seamless execution. However, adopting legacy and affordable IoT infrastructures to deploy such monitoring technologies is beneficial.

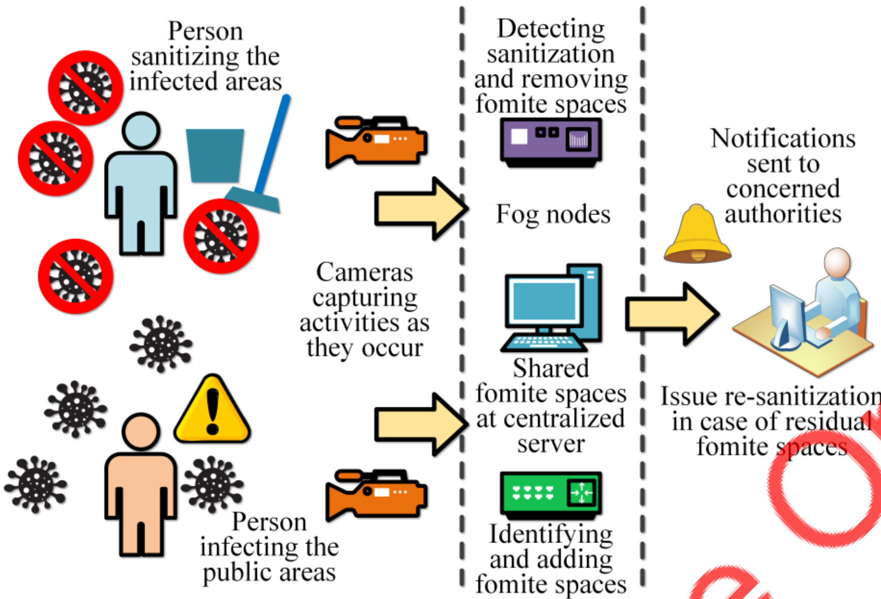

Figure 1.1: Overview of the proposed COVI-SCANNER scheme.

In this chapter, we propose COVI-SCANNER, an Augmented Reality (AR)-based fomite space monitor in closed environments on a fog-enabled IoT platform. As shown in Figure 1.1, the COVI-SCANNER scheme works in two phases: 1. *Contamination* and 2. *Sanitization*. In the contamination phase, COVI-SCANNER records the videos from the cameras for tracking the movement of the infected individuals and detects the spaces/objects that the person interfaces/touches. On detecting any such contact, the COVI-SCANNER highlights them (virtually) as potentially contaminated/fomite places. In the sanitization phase, the COVI-SCANNER tries to look for cleaning materials such as brooms, wipes, mops, and others. Upon detection of these objects, the COVI-SCANNER tries to detect cleaning activity by the person interacting with the sanitization objects. In case the person starts cleaning and the object passes through the fomite places identified during the contamination phase, it removes the demarcations. In case the cleaning activity is complete and all of the demarcations are not removed at the end of the sanitization phase, COVI-SCANNER raises an alarm and notifies the concerned authorities. This method allows us to ensure the existence of minimal fomite places in public areas and reduce the secondary spread. Towards this, we use the readily available machine learning models: 1. MobileNet-SSD (Younis 2020) for detecting the person and objects in the contamination phase, and 2. 3D-ResNet (Hara 2018) for identifying the cleaning activity in the sanitization phase. It may be noted that we assume that the concerned authorities have information on the infected individuals and recognize them using available face recognition techniques (Khan 2019). We further propose a fog-based IoT architecture to deploy the proposed COVI-SCANNER scheme on pre-installed infrastructures. We account for the computational complexity in the AR-based operations and propose assigning different fog nodes for each contamination and sanitization phase. Since the two phases share the information on the fomite spaces, we propose storing the points in a shared database with provisions for real-time updates on highlighting and removal from the two phases. In this chapter, we propose COVI-SCANNER, an AR-based fomite space monitor in closed environments to minimize the secondary spreading of the COVID-19 virus. For the ease of deployment on legacy infrastructures, we propose a fog-enabled IoT architecture comprising of resource-constrained devices. The major highlights of this chapter are as follows:

- ***COVI-SCANNER:*** We propose an AR-based method for tracking and monitoring fomite spaces to reduce the secondary spread of the COVID-19 virus.
- ***Modular Operations:*** As image processing techniques involve complex computations, we divide COVI-SCANNER into two phases: *Contamination* and *Sanitization*. Such modules helps in reducing the load from the devices.
- ***IoT-Based Architecture:*** To facilitate easy deployment on resource-constrained devices, we propose an IoT-based architecture that assigns dedicated fog nodes for performing each phase and sharing their data from a common database server in real-time.
- ***Evaluation:*** To show the feasibility of the proposed COVI-SCANNER scheme, we implement and deploy in lab-scale and present the observed results.

Example Scenario: Consider a closed environment such as hospitals, industries, and any other environment with cameras installed for monitoring. In such environments, fog nodes assigned for the contamination phase in COVI-SCANNER identifies the infected individuals. It then creates a bounding box around the person and tracks the person, particularly the arms and its interaction with the nearby objects. In the case of detecting such interactions, COVI-SCANNER marks them as fomite spaces on the screen. On the other hand, fog nodes assigned for the sanitization phase in COVI-SCANNER first identifies the necessary cleaning objects such as wipes, mops, brooms, and others to then detect the cleaning activity. As the individual cleans the regions, COVI-SCANNER tracks the cleaning objects and removes the highlighted portions. The COVI-SCANNER notifies the concerned authorities in case all the fomite spaces are not sanitized. The fog nodes operating under each phase share their data through a central database server for simultaneously updating the data in real-time.

1.1.1 Motivation

The recent outbreak of the COVID-19 virus has spread rapidly all around the world. The intangible nature of the virus and its mode of spread of the droplets in the form of aerosols and contact has increased the challenges of containing it. Current solutions depend on contact tracing and self-assessment tests by individuals (Menni 2020), which is not reliable as they may enter false information to avoid isolation and containment. The remnants of the virus in an infected individual, particularly in the hands cause contamination of the objects and spaces with each touch, giving rise to fomite spaces. Healthy individuals acquire this virus by interacting with the same set of fomite spaces. The virus stays alive on cardboard and plastic surfaces for a duration of 24 hours to 3 days, respectively. Such a lifespan increases the risk of spreading to more number of individuals each day, which mandates the need for efficient sanitization. The intractable transmission mode of the virus necessitates IoT-based solutions for monitoring and tracking fomite spaces in closed public areas to reduce secondary spread. Moreover, we propose an AR-based method (COVI-SCANNER) to combat the secondary spread of the virus. However, image processing techniques involve complex operations for execution. Facilitating such operations on legacy infrastructures requires task distribution techniques for seamless deployment. Such issues act as our motivation for designing COVI-SCANNER and its IoT-based architecture. We envision such solutions to help in restricting the secondary spread of the COVID-19 virus in closed public environments.

The rest of the chapter is organized as follows. We present some of the existing works in literature and techniques for combating the COVID-19 virus in section 1.2. In section 1.3, we present the system model. We discuss our observations in section 1.4 and finally conclude in section 1.5.

1.2 OVERVIEW OF TECHNIQUES FOR COMBATING COVID-19

In this section, we present some of the current literature on the COVID-19 virus and IoT solutions towards combatting it. We then present some of the existing works on computer vision, human activity recognition, and fog/edge computing solutions.

1.2.1 COVID-19 and IoT

The COVID-19 virus needs a week to show initial symptoms of infection and some do not show any symptoms at all (asymptotic cases). The authors in (Benreguia 2020) identified the issue and the significant difference in the documented and the actual count of positively infected individuals due to it. They proposed an IoT-based solution to track the known individuals and locations they have been. As healthy individuals visit the same location, the proposed solution warns them of potential infection. Wang *et al.* (Wang 2020) exploited the Social Internet of Things (SIoT) for identifying social relationships and potentially infected individuals as they come in contact with positively infected patients. They used a graph theoretic approach coupled with reinforcement learning to realize the proposed patient identification method. Researchers are also exploring machine learning (ML) methods for combatting the pandemic. Waheed *et al.* (Waheed 2020) focused on the lack of data due to the recent outbreak and the major drawback of the ML methods. They developed synthetic x-ray images by considering the effects of the COVID-19 virus and trained an Auxiliary Classifier Generative Adversarial Network (ACGAN) for identifying infected individuals with high precision. The authors in (Hussain 2020) have provided a comprehensive description of the ML methods and its role in combatting the COVID-19 virus.

1.2.2 Computer Vision

In this section, we highlight some of the works in literature focusing on the development of computer vision techniques and their applications. Morales *et al.* (Morales 2019) used a combination of VGG-16 network and convolutional LSTM layers to detect violent robberies through CCTV camera footage. Wong *et al.* (Wong 2020) proposed a novel approach to re-identification of a person on a campus using multiple CCTV cameras by assigning higher weights to combinations of parts that helped in re-identification by evaluating the relative performances of each of these combinations. Khandelwal *et al.* (Khandelwal 2020) used face detection and person detection algorithms to raise alarms if people were detected not wearing masks or not following social distancing rules and also implemented the same in manufacturing plants having multiple CCTV cameras.

1.2.3 Human Activity Recognition

In this chapter, we use human activity recognition to distinguish between a normal person and a person who is cleaning infected areas. This field has seen a lot of research work in recent years mainly due to the increasing popularity of deep learning. Xu *et al.* (Xu 2019) proposed a deep learning model that draws inspiration from the inception network and Gated Recurrent Unit (GRU) to predict human activity by detecting inputs in the form of waveform data from multiple sensors attached to the body. Gnouma *et al.* (Gnouma 2019) proposed to use a dynamic frame skipping method and used Gaussian Mixture Model for foreground detection, both of which reduced the time taken for silhouette extraction which is required for human activity recognition. Noori *et al.* (Noori 2019) trained a Recurrent Neural Network consisting of Long Short Term Memory cells on the OpenPose dataset to predict activity performed by a human from different camera angles.

1.2.4 Fog/Edge Computing

Edge Computing helps in keeping computational and storage units closer to the devices for reducing bandwidth usage and response time. Abdellatif *et al.* (Abdellatif 2019) discussed the challenges of using edge computing concepts in healthcare systems and discussed in depth how wearable sensors and medical devices on the edge of the network could be used to monitor the health conditions of patients while ensuring user privacy is maintained as well. Cao *et al.* (Cao 2015) proposed a real-time fall detection system for stroke patients by dividing the computation of analytics between smartphones with accelerometers having lower computational speed and edge nodes with higher computational speed to reduce and rectify false detections. Barthelemy *et al.* (Barthelemy 2019) proposed an edge computing architecture using a live feed from CCTV cameras across a smart city by using popular lightweight algorithms like YOLO V3 (Redmon 2018) and Simple Online and Real-time Tracking (SORT) algorithm to perform object detection and object tracking respectively. Deb *et al.* (Deb 2020) proposed a digital stethoscope SkopEdge for counting the number of heartbeats by exploiting the features of edge computing. They proposed sending the recorded audio in the most suitable format based on the network state and the device conditions. IoT devices such as SkopEdge have the potential of addressing the problems in the identification of the COVID-19 virus.

1.2.5 Synthesis

From the discussion in this section, it is evident that IoT solutions have a major role to help combat the rapidly spreading COVID-19 virus. These solutions help in tracking the positively infected individuals and identifying potentially asymptotic carriers based on the locations that they visit. Additionally, ML methods also help in diagnosing the patients and categorizing them as infected or healthy individuals. We also notice that the popularly available pre-trained ML models based on image processing help in maintaining social distancing and identifying violators. Since ML techniques involve computationally complex operations, fog/edge computing methods help in deploying the trained models on resource-constrained devices. Although we notice a myriad of applications, we observe a lacuna in reducing the secondary spread of the virus from fomite spaces. Towards this, we propose the COVI-SCANNER scheme using an IoT architecture to first identify and highlight the fomite spaces (contamination). Then, we propose identifying cleaning objects and the corresponding activity to trace and remove the highlighted fomite spaces (sanitization).

1.3 SYSTEM MODEL

In this section, we present the IoT network architecture adopted for the proposed COVI-SCANNER scheme along with the information flow. Additionally, we also explain the solution technique for realizing COVI-SCANNER.

1.3.1 IoT-Based Network Architecture for COVI-SCANNER

We consider a set of pre-installed surveillance cameras $C = \{c_1, c_2, \ldots, c_p\}$ responsible for monitoring the closed environments, as shown in Figure 1.1. The images and videos from these cameras pass through fog nodes before reaching the centralized storage server. In this chapter, we consider the routers, switches, and other devices present at the edge of the network as fog nodes. Considering a set of fog nodes $F = \{f_1, f2_{,\ldots,} f_q\}$, we split them into two subsets for performing each of the two phases in COVI-SCANNER. In other words, $F_c \subset F$ fog nodes for contamination and $F_s \subset F$ fog nodes for sanitization. It may be noted that the fog nodes in F_c and F_s may not be mutually explicit. The fog nodes with high configurations may execute both the modules for contamination and sanitization, respectively. It may be noted that the selection of the optimal set of fog nodes is beyond the scope of this chapter and we rely on available works on resource allocation (Tran 2018). Fog nodes executing the contamination routine tracks the infected individuals and highlight the fomite spaces on the screen. It then sends the details of each of the fomite spaces for storage and access by the fog nodes assigned for the sanitization phase. As the cleaning activity proceeds and the fog nodes detect the sanitization of the fomite spaces, it removes the corresponding information from the server. The data in the server is simultaneously accessible by both sets of fog nodes so that the concerned personnel may view the results on a screen in real-time. In case the cleaning activity is over and fomite spaces continue to persist in the region, the server may issue an alarm for notifying the concerned authorities.

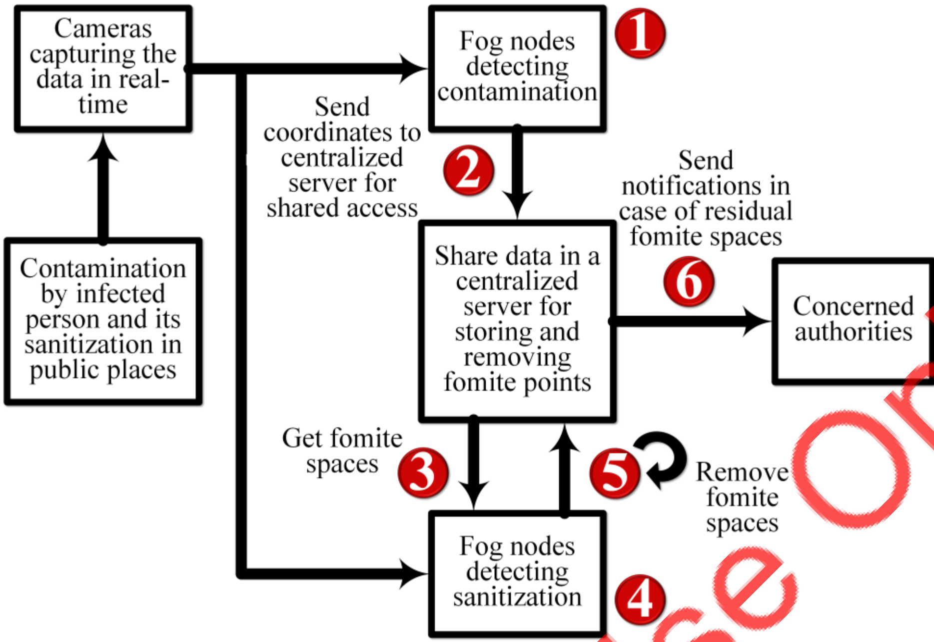

Figure 1.2: Information flow in the proposed COVI-SCANNER scheme.

1.3.2 Information Flow in COVI-SCANNER

Figure 1.2 depicts the information flow among the devices mentioned in Section 1.3.1. The F_c fog nodes determine the fomite spaces and their coordinates in the contamination phase (Step 1). They send the coordinates of the infected regions along with the details of the bounding boxes to the centralized servers (Step 2). These coordinates are accessible by the sanitization phase to keep track of the cleaning of the fomite spaces (Step 3) and for performing the execution routine (Step 4). The fog nodes remove the demarcated fomite spaces as the cleaning activity proceeds (Step 5). In the case of residual fomite spaces after sanitization, the server notifies the concerned authorities (Step 6). It may be noted that there is no need for manually removing the demarcated fomite spaces as the sanitization phase will automatically remove the same on detecting each sequence of cleaning activities (recurrent Step 5).

1.3.3 Proposed Solution

In this section, we describe the proposed solution technique and the adopted pre-trained ML models for realizing COVI-SCANNER. Conforming to the network architecture in Section 1.3.1 and the information flow in Section 1.3.2, we present each of the phases in COVI-SCANNER separately in the subsequent sections.

1.3.3.1 The Contamination Module

We execute this module on the F_c set of fog nodes. We use the MobileNet-SSD network available in the deep learning module of OpenCV for object detection. The MobileNet-SSD network detects and returns coordinates of the bounding boxes for 20 classes of objects. Among these 20 classes, we only use the class labeled as *person* for our application and ignore the other classes. As mentioned earlier, we assume that the infected individuals are known beforehand and facial recognition techniques may be used for detecting them on the screen. We use the person class as proof of concept for the proposed COVI-SCANNER scheme. The MobileNet class of convolutional neural networks uses depth-wise separable convolutions followed by 1 × 1 pointwise convolutions which allows these models to be much more computationally much faster (Howard 2017). This model allows us to detect and draw bounding boxes around all person objects detected in each frame. This module is entirely run on the contamination devices.

Each person detected from the developed model p_i has its own bounding box coordinates, represented by the upper left point (x_1^i, y_1^i) and the lower right point (x_2^i, y_2^i). For demarcating the places where each person pi has visited and the interacted objects, we calculate a fomite point $fp_i \in R^2$ for each person object. We calculate the fp_i in such a manner to make sure that the fomite point is nearer to the points in the frame which helps in tracking the cleaning objects during the activity. Mathematically, we calculate fp_i as:

$$\boldsymbol{fp_i} = \left(\frac{x_2^i + x_1^i}{2}, \frac{9y_2^i + y_1^i}{10}\right) \quad \textbf{(1.1)}$$

This fomite point fp_i is used for tracking person objects from one frame to the next. Let the set of all person objects detected in a particular frame j be P^j while the set of all person objects detected in the next frame be P^{j+1}. Let each person object in frame j be $p_i^j \in P^j$ and each person in the next frame j+1 be $p_i^{j+1} \in P^{j+1}$. Let fomite point of each person object p_i^j be fp_i^j, such that $fp_i^j \in F_i^j$. Initially, we calculate the Euclidean distances between all pairs of fomite points in P^j and P^{j+1}. We define each of the distances such as between fp_i^j and fp_i^{j+1} as *dist*((fp, fp_i^{j+1})). To track each person on the screen effectively, we rely on the following set of rules:

- ***Already Detected Person:*** For any fixed p_i^j, we consider the person to be already detected in p_i^{j+1} if *dist*((fp_i^j, fp_i^{j+1})) is minimum $\forall\ fp_i^{j+1} \in F_i^{j+1}$. We use this to rediscover already detected people from the previous frame j in the current frame j + 1.
- ***New Person Detection:*** We demarcate an object as a new person and identities on the screen to all objects in P^{j+1} that were not mapped as a person in P^j in the previous step. This is used when a new person who was not present in the frame j has just entered in the next frame j + 1.
- ***Terminate Tracking:*** If p^{j+1} remains unassigned to any person object in P^j for 10 consecutive frames, then we remove this person p_i from the screen and store the list of all its corresponding fomite points in the centralized server for future reference. This is used to stop trying to find the closest fp_i^{j+1} to a particular fp_i^j belonging to a person p_i^j who has already exited the area of the frame.

1.3.3.2 *The Sanitization Module*

We execute this module entirely on the set of sanitization fog node (F_s) devices. We pass the bounding box of each detected person as input to a separate deep learning model to detect whether the person is performing the cleaning activity or not. If the person is detected to have been performing the activity of cleaning then its focus point is calculated. Let this focus point of cleaner person object be f_C. Let $M \in R^2$ be the set of the history of all fomite points of all persons detected so far. Let m_i be each fomite point in M. To remove the fomite points of people in areas where cleaning activity has occurred we remove all m_i from M, such that $dist(f_C, m_i) < D_{max}$

In our implementation of COVI-SCANNER, we consider D_{max} = 50 pixels. We use the pre-trained 3D-ResNet deep learning model which is trained on the Kinetics Human Action Video Dataset (Kay 2017) for identifying up to 400 different types of human activities to detect the cleaning activity. Out of all the 400 activity classes, we use only 2 classes of *cleaning floor* and *mopping floor* and categorize them as cleaning. We do so by taking the previous N frames as input and predicting the activity in the $(N+1)^{th}$ frame. In this chapter, we use N=16 and N=10 while performing our experiments.

1.4 PERFORMANCE EVALUATION

In this section, we present our lab-scale experimental setup and present our observations on deploying COVI-SCANNER on the proposed IoT architecture.

1.4.1 Experimental Setup

We use two arbitrary systems with i3 and i5 processors and assign them for executing the contamination and sanitization modules. We install the MobileNet-SSD and 3D-ResNet ML models on the concerned devices and use the videos from the cameras as inputs in each case. We use Python 3.7 platform for realizing the proposed COVI-SCANNER scheme and to present its performance.

1.4.2 Results

In this section, we present and discuss the performance of the proposed COVI-SCANNER scheme on deployment. Towards this, we first present its output in identifying and removing the fomite spaces in each phase, followed by its corresponding accuracy. We then demonstrate the delay in executing each phase together with the upload rate for the contamination and sanitization phases along with the download rate at the centralized server.

Figure 1.3: Identification of fomite spaces contaminated by infected individuals in the contamination phase.

1.4.2.1 *Output from Contamination Phase*

We capture one of the instances from the contamination phase. As mentioned earlier, we assume that the identity of the infected individuals is known beforehand and they may be identified using available face recognition techniques. In this chapter, we consider any random person to be infected with the COVID-19 virus for proof of concept. We observe in Figure 1.3 that COVI-SCANNER detects a person on the screen (bounding box). As the person moves (steps 1 through 4), the contamination routine highlights the floor as fomite space with several fomite points (red dots). We comment that the proposed COVI-SCANNER identifies the fomite spaces efficiently.

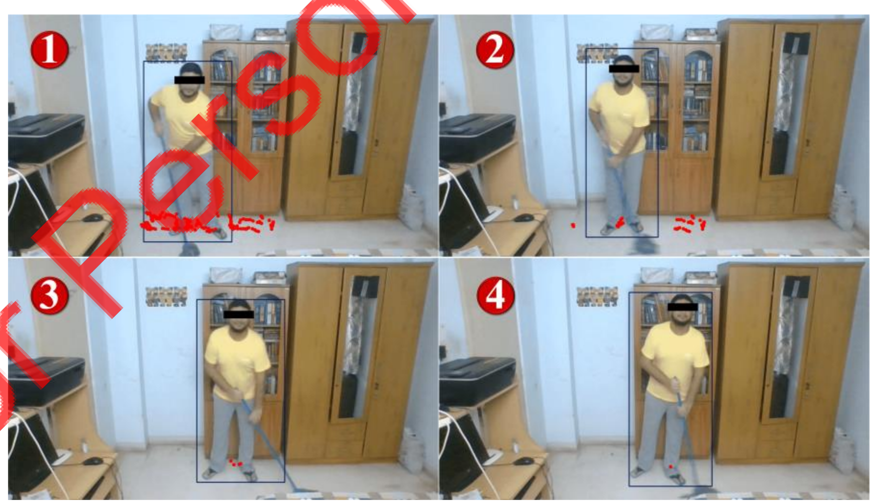

Figure 1.4: Removal of the fomite spaces identified in the contamination phase on detection of cleaning activity in sanitization phase.

1.4.2.2 *Output from Sanitization Phase*

We capture the same instance corresponding to that in Section 1.4.2.1. We observe in Figure 1.4 that the sanitization phase in COVI-SCANNER first identifies an individual (bounding box) and then the broom. Upon detection of the cleaning activity, the COVI-SCANNER starts removing the fomite spaces. At the end of the cleaning activity (steps 1 through 4), we observe that almost all of the fomite spaces are removed as the cleaning object passes through them. We may safely comment that the proposed COVI-SCANNER works efficiently in tracking the sanitization of the detected fomite spaces.

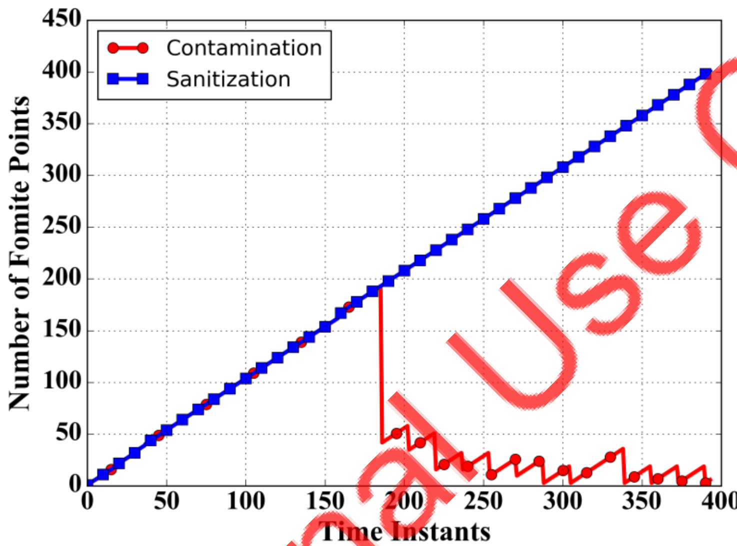

Figure 1.5: Number of fomite spaces detected and removed on execution of each phase.

1.4.2.3 Detection and Removal of Fomite Spaces

We perform a cumulative count of the fomite points with each iteration for each of the phases. We observe in Figure 1.5 that the number of fomite spaces increases linearly in the contamination phase. This is because, with each progress in the execution, the contamination routine keeps tracking the infected individual and marks the fomite spaces. On the other hand, we notice that on the detection of the cleaning activity, the sanitization phase removes the corresponding fomite points efficiently. We notice almost 80% removal of the fomite points. We observe a steep decrease in the number of fomite points as the cleaner enters the scene for the first time. This is because the fomite points near the cleaner initially get removed in bulk, compared to the later time instants. Although we observe significant removal of the fomite points, we hardly notice 100% removal of the same. Intuitively, this may be because the cleaning object does not pass through all the fomite points or it may be due to the accuracy limitations of the pre-trained models. In the next section, we present the accuracies in each phase.

Table 1.1 Accuracy, precision, recall, and F1-score of the COVI-SCANNER scheme.

Videos	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Video 1	82.75%	85.71%	80.00%	82.75%
Video 2	81.08%	85.25%	76.47%	78.78%
Video 3	82.05%	85.35%	77.77%	79.99%
Mean values	81.96%	85.43%	78.08%	80.51%

1.4.2.4 *Accuracy of the COVI-SCANNER Phases*

We arbitrarily use 3 videos as inputs to the COVI-SCANNER routine and tabulate the accuracy scores in Table 1.1. We briefly define each of the columns for understanding the results. Accuracy is the quality of correctness of the proposed scheme. Precision represents the correctly identified results and recall represents the number of correctly identified points in comparison to the actual one. The F1-score is the harmonic mean of the precision and recall results, represented as F1 = 2/(recall^{-1}+precision^{-1}). On average, we observe an accuracy of 81%. We acquire the precision and recall values to identify the reason behind such low accuracies. We notice that on average, COVI-SCANNER has a precision of 85% which suggests that it has a fairly high rate of fomite space detection. However, we observe low recall values of 78%, suggesting that the COVI-SCANNER identifies fomite spaces that may not be contaminated. Intuitively, we attribute this behavior to the occlusions that may be present in the video and that of the accuracies of the pre-trained MobileNet-SSD and 3D-ResNet models with 80% and 75%, respectively. In the future, we plan to address this issue and increase the accuracy of the proposed COVI-SCANNER scheme.

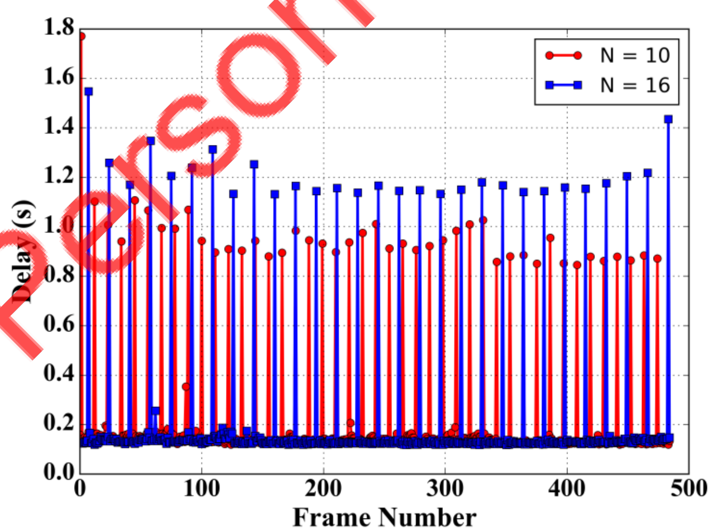

Figure 1.6: Delay in processing each frame with varying N.

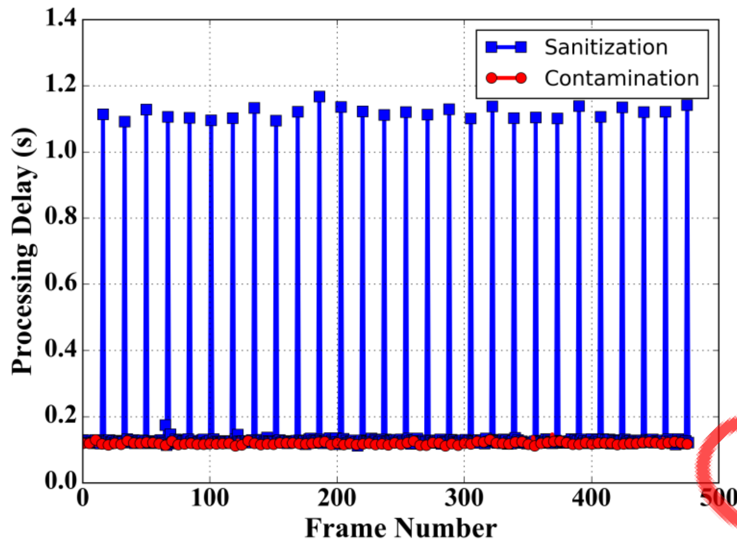

Figure 1.7: Delay in processing the frames in the contamination and sanitization phases.

1.4.2.5 *Delays in Executing COVI-SCANNER and its Phases*

Execution of image processing routines involves complex computations. We account for the high resource demand and propose executing the phases in the COVI-SCANNER scheme periodically after every N frames. Figure 1.6 depicts the time taken in seconds for each frame to be processed in the fog nodes. We observe spikes of almost 1.75 seconds after the N frames due to the delays in acquiring results from the pre-trained activity detection models. As expected, we observe in Figure 1.6 that the frequency of the spikes decreases as we increase N from 10 to 16. This is because of the frequency of the execution of the COVI-SCANNER routines. However, the performance of the COVI-SCANNER starts deteriorating if N is increased as it will miss demarcating most of the fomite spaces, implying the existence of a tradeoff between accuracy and the processing delay of COVI-SCANNER. In this chapter, we limit our experiments up to N=16.

We present a granular insight into the delays necessary for executing each phase. In Figure 1.7, we observe that the sanitization phase endures more delay compared to that of the contamination phase. This is because the cleaning activity detection model in the sanitization phase requires more time to process as it needs to identify the person, the cleaning objects, and the activity than the person identification and tracking module (contamination). This is because the sanitization phase identifies the person and the object and then the cleaning activity. The contamination phase on the other hand only needs to identify the person and then start tracking.

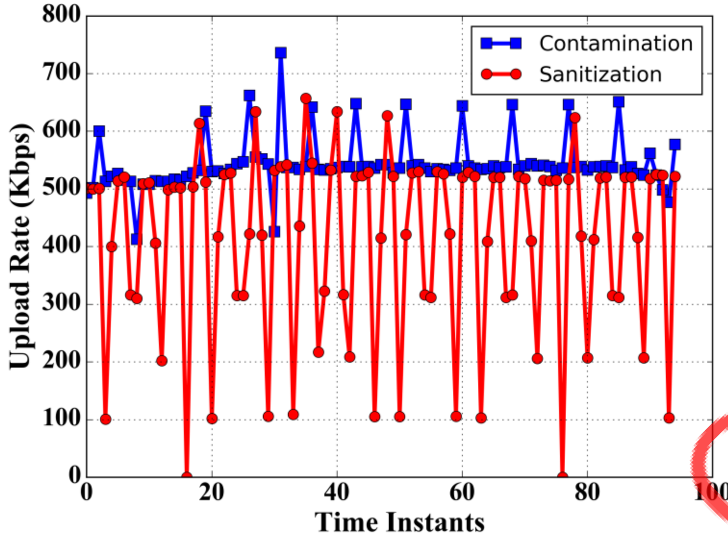

Figure 1.8: Upload rate in both contamination and sanitization fog nodes.

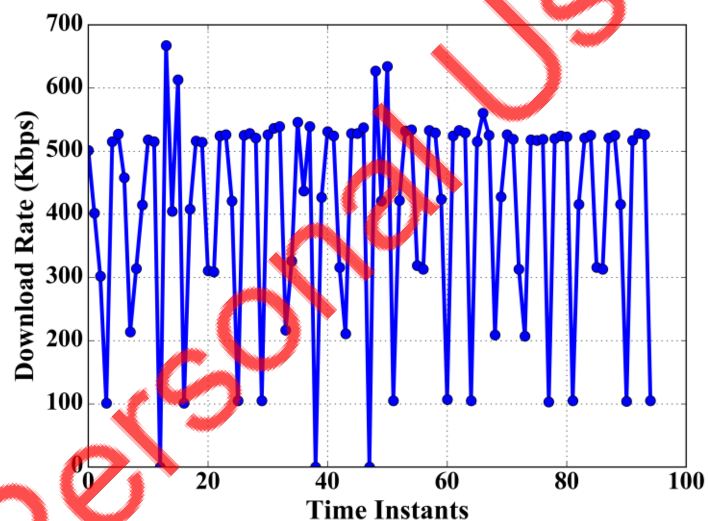

Figure 1.9: Download rate at the centralized storage server.

1.4.2.6 Upload and Download Rates

We capture the upload and download rates using the command line application *iftop* available in Linux distributions. We observe in Figure 1.8 that the contamination phase needs a higher upload rate of almost 800 Kbps, compared to that of the sanitization phase. We observe this variation in the upload rates as the sanitization phase sends only the information about the time instants and coordinates of the fomite points for removal. The contamination phase needs to send the video along with the fomite points while uploading the data for storage at the centralized server. Corresponding to the upload rates from the fog nodes, we observe a maximum download rate of 700 Kbps at the centralized server in Figure 1.9. Such high data rates at the fog nodes and the centralized servers are common because of the large size of videos and images from the cameras, implying the dependency of the proposed COVI-SCANNER on the quality of the network. In the future, we plan to use lightweight streaming protocols to reduce the data rates for each device.

1.5 CONCLUSION

In this chapter, we proposed an AR-based solution named COVI-SCANNER for limiting the secondary spread of the COVID-19 virus in closed public environments. COVI-SCANNER works in two phases: *contamination* and *sanitization*. In the contamination phase, COVI-SCANNER identifies the infected individuals and tracks them to detect fomite spaces and highlight them accordingly. In the sanitization phase, COVI-SCANNER identifies the cleaning objects and the corresponding cleaning activity to remove the fomite points from the contamination phase. In the case of residual fomite points on the screen after the cleaning activity is complete, the centralized server notifies the concerned authorities for re-sanitization. We achieve the identification of the individuals and the concerned activities using pre-trained MobileNet-SSD and 3D-ResNet models. Through deployment and extensive experimentation, we presented the efficiency of the proposed COVI-SCANNER scheme along with the necessary delays and data rates. We hope that with solutions like COVI-SCANNER, we may restrict the secondary spread of the COVID-19 virus in public places and ensure its sanitization.

In this future, we plan to extend this work by increasing the accuracy of detecting and removing the fomite points. Additionally, we also plan to incorporate lightweight streaming protocols to reduce the data rates at the fog nodes and the centralized server.

References

Abdellatif, Alaa Awad and Mohamed, Amr and Chiasserini, Carla Fabiana and Tlili, Mounira and Erbad, Aiman. 2019. "Edge Computing for Smart Health: Context-Aware Approaches, Opportunities, and Challenges." *IEEE Network* 196-203.

Barthelemy, Johan and Verstaevel, Nicolas and Forehead, Hugh and Perez, Pascal. 2019. "Edge-Computing Video Analytics for Real-Time Traffic Monitoring in A Smart City." *Sensors* 2048.

Benreguia, Badreddine and Moumen, Hamouma and Merzoug, Mohammed Amine. 2020. "Tracking COVID-19 by Tracking Infectious Trajectories." *arXiv preprint arXiv:2005.05523.*

Cao, Yu and Chen, Songqing and Hou, Peng and Brown, Donald. 2015. "FAST: A Fog Computing Assisted Distributed Analytics System to Monitor Fall for Stroke Mitigation." *IEEE international conference on networking, architecture and storage (NAS).* 2-11.

Chamola, Vinay and Hassija, Vikas and Gupta, Vatsal and Guizani, Mohsen. 2020. "A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact." *IEEE Access* 90225--90265.

Deb, Pallav Kumar and Misra, Sudip and Mukherjee, Anandarup and Jamalipour, Abbas. 2020. "SkopEdge: A Traffic-Aware Edge-Based Remote Auscultation Monitor." *IEEE International Conference on Communications (ICC).* 1-6.

Gan, Wee Hoe and Lim, John Wah and David, KOH. 2020. "Preventing Intra-Hospital Infection and Transmission of COVID-19 in Healthcare Workers." *Safety and Health at Work.*

Gnouma, Mariem and Ladjailia, Ammar and Ejbali, Ridha and Zaied, Mourad. 2019. "Stacked Sparse Autoencoder and History of Binary Motion Image for Human Activity Recognition." *Multimedia Tools and Applications* 2157-2179.

Hara, Kensho and Kataoka, Hirokatsu and Satoh, Yutaka. 2018. "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and Imagenet?" *IEEE Conference on Computer Vision and Pattern Recognition.* 6546-6555.

Howard, Andrew G and Zhu, Menglong and Chen, Bo and et al. 2017. "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications." *arXiv preprint arXiv:1704.04861.*

Hussain, Adedoyin Ahmed and Bouachir, Ouns and Al-Turjman, Fadi and Aloqaily, Moayad. 2020. "AI Techniques for COVID-19." *IEEE Access* 128776-128795.

Kay, Will and Carreira, Joao and Simonyan, Karen and et al. 2017. "The Kinetics Human Action Video Dataset." *arXiv preprint arXiv:1705.06950.*

Khan, Muhammad Zeeshan and Harous, Saad and Hassan, Saleet Ul and et al. 2019. "Deep Unified Model for Face Recognition Based on Convolution Neural Network and Edge Computing." *IEEE Access* 72622-72633.

Khandelwal, Prateek and Khandelwal, Anuj and Agarwal, Snigdha. 2020. "Using Computer Vision to enhance Safety of Workforce in Manufacturing in a Post COVID World." *arXiv preprint arXiv:2005.05287.*

Menni, Cristina and Valdes, Ana M and Freidin, Maxim B and et al. 2020. "Real-Time Tracking of Self-Reported Symptoms To Predict Potential COVID-19." *Nature medicine* 1-4.

Morales, Giorgio and Salazar-Reque, Itamar and Telles, Joel and Diaz, Daniel. 2019. "Detecting Violent Robberies in CCTV Videos Using Deep Learning." *IFIP International Conference on Artificial Intelligence Applications and Innovations.* Springer. 282-291.

Noori, Farzan Majeed and Wallace, Benedikte and Uddin, Md Zia and Torresen, Jim. 2019. "A Robust Human Activity Recognition Approach Using Openpose, Motion Features, and Deep Recurrent Neural Network." *Scandinavian Conference on Image Analysis.* Springer. 299-310.

Redmon, Joseph and Farhadi, Ali. 2018. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767.*

Tran, Tuyen X and Pompili, Dario. 2018. "Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks." *IEEE Transactions on Vehicular Technology* 856-868.

Waheed, Abdul and Goyal, Muskan and Gupta, Deepak and et al. 2020. "Covidgan: Data Augmentation Using Auxiliary Classifier Gan for Improved COVID-19 Detection." *IEEE Access* 91916-91923.

Wang, Bowen and Sun, Yanjing and Duong, Trung Q and Nguyen, Long D and Hanzo, Lajos. 2020. "Risk-Aware Identification of Highly Suspected COVID-19 Cases in Social IoT: A Joint Graph Theory and Reinforcement Learning Approach." *IEEE Access* 115655-115661.

Wong, Peter Kok-Yiu and Cheng, Jack CP. 2020. "Monitoring Pedestrian Flow on Campus with Multiple Cameras using Computer Vision and Deep Learning Techniques." *CIGOS 2019, Innovation for Sustainable Infrastructure* 1149-1154.

Xu, Cheng and Chai, Duo and He, Jie and Zhang, Xiaotong and Duan, Shihong. 2019. "InnoHAR: A Deep Neural Network for Complex Human Activity Recognition." *IEEE Access* 9893-9902.

Younis, Ayesha and Shixin, Li and Jn, Shelembi and Hai, Zhang. 2020. "Real-Time Object Detection Using Pre-Trained Deep Learning Models MobileNet-SSD." *The 6th International Conference on Computing and Data Engineering.* 44-48.