·Article·

Virtual-reality-based digital twin of office spaces with social distance measurement feature

Abhishek MUKHOPADHYAY[1], G S Rajshekar REDDY[1], KamalPreet Singh SALUJA[1], Subhankar GHOSH[1], Anasol PEÑA-RIOS[2], Gokul GOPAL[2], Pradipta BISWAS[1*]

1. *Indian Institute of Science, Bangalore, India*

2. *BT Plc, UK*

*** Corresponding author,** pradipta@iisc.ac.in

Received: 16 May 2021 **Accepted:** 21 September 2021

Citation: Abhishek MUKHOPADHYAY, G S Rajshekar REDDY, KamalPreet Singh SALUJA, Subhankar GHOSH, Anasol PEÑA-RIOS, Gokul GOPAL, Pradipta BISWAS. Virtual-reality-based digital twin of office spaces with social distance measurement feature. Virtual Reality & Intelligent Hardware, 2022, 4(1): 55—75
DOI: 10.1016/j.vrih.2022.01.004

Abstract **Background** Social distancing is an effective way to reduce the spread of the SARS-CoV-2 virus. Many students and researchers have already attempted to use computer vision technology to automatically detect human beings in the field of view of a camera and help enforce social distancing. However, because of the present lockdown measures in several countries, the validation of computer vision systems using large-scale datasets is a challenge. **Methods** In this paper, a new method is proposed for generating customized datasets and validating deep-learning-based computer vision models using virtual reality (VR) technology. Using VR, we modeled a digital twin (DT) of an existing office space and used it to create a dataset of individuals in different postures, dresses, and locations. To test the proposed solution, we implemented a convolutional neural network (CNN) model for detecting people in a limited-sized dataset of real humans and a simulated dataset of humanoid figures. **Results** We detected the number of persons in both the real and synthetic datasets with more than 90% accuracy, and the actual and measured distances were significantly correlated (r=0.99). Finally, we used intermittent-layer- and heatmap-based data visualization techniques to explain the failure modes of a CNN. **Conclusions** A new application of DTs is proposed to enhance workplace safety by measuring the social distance between individuals. The use of our proposed pipeline along with a DT of the shared space for visualizing both environmental and human behavior aspects preserves the privacy of individuals and improves the latency of such monitoring systems because only the extracted information is streamed.

Keywords Virtual environment; Digital twin; 3D visualization; Convolutional neural network; Object detection; Social distancing

1 Introduction

The COVID-19 pandemic is already considered one of the worst human disasters since the Second World War. The pandemic is spreading at different paces in different geographic regions, and efforts are already

in place to adapt to a "new normal". Social distancing is undoubtedly an effective strategy for slowing the spread of the disease in the workplace and other crowded spaces such as shopping centers. Enforcing social distancing is often challenging, however, owing to various factors such as ignorance and the nature of human activity, among other aspects. An automatic method for measuring and alerting a deviation from social distancing can be an effective way to stop the spread of the disease as offices and shopping centers gradually reopen in affected areas.

Social distancing can be measured automatically using computer vision technology by detecting the presence of individuals within the field of view of a camera. Most modern computer vision systems operate based on machine learning technology, which in turn depends on appropriate training and testing datasets. Despite a plethora of datasets for autonomous vehicles and facial images, among others, generating an appropriate dataset for social distancing measurement systems is a challenge under the present circumstances because many offices, shopping centers, and public spaces are either closed or are operating with a reduced number of people. However, as more people start returning to the workplace, an automatic system for accurately detecting and calculating the optimal number of people in indoor places would be of significant value for accelerating the return to normal activities. For this approach, a system trained in an outdoor environment may not work as well as that trained in an indoor environment where variables such as the background color, lighting, and even the posture of personnel will be different. Following a similar logic, we must consider that even a system trained on one particular indoor environment may not work well in another indoor environment.

In this context, we propose a digital twin (DT) of a workspace through an interactive and immersive virtual reality (VR) experience. Users can move around the space virtually and remotely, as they would in the real world. The benefits of using DTs as a visualization medium are multifold. First, a DT provides an interactive and intuitive virtual experience that can be used in VR. Users can navigate around the virtual environment as they would in the real world. Second, a virtual environment protects the privacy of the occupants through abstract humanoid figures compared to a direct video feed. In the virtual world, a virtual camera was simulated at the same position from where the real-world feed was recorded. We then mapped the two-dimensional centroid coordinates onto the feed of the virtual camera. Moreover, through a ray-cast operation, the two-dimensional coordinates are mapped to the three-dimensional coordinates of the virtual world, and hence, the movements of people are simulated in real time. In addition, to help us debug the performance of the system, we used data visualization techniques to explain the working of a complex machine learning system, such as a convolutional neural network (CNN). As the main contribution of this study a synthetic data generation system is validated using a VR digital twin. Whereas earlier studies have taken a similar approach for traffic datasets or robot control, we validated the VR DT for detecting persons inside office workspaces.

The remainder of this paper is organized as follows. Section 2 presents a literature survey on VR-based workspace simulators, particularly in the context of COVID-19 and different human detection systems. The subsequent sections present a case study for developing a VR-based simulator, training a CNN with a synthetic dataset, and explaining the operation of the CNN with both real and synthetic datasets using appropriate data visualization techniques. The final sections highlight the utility and value addition of the system and provide some concluding remarks.

2 Related work

2.1 Digital twins

The first digital twin implementations date back to NASA's Apollo program[1], in which live missions were

used to replicate the problem scenarios faced by a crew 100000 miles away. NASA[1] formalized the definition of a DT in 2012 as an integrated multiphysics, multiscale, and probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, and other available data to mirror the circumstances of its corresponding flying twin. Tao et al. highlighted state-of-art approaches in industrial DTs[2], according to which DTs have been implemented in three key application areas: (1) product design, (2) production, and (3) prognostics and health management (PHM), with the majority of focus primarily on (3). Khajavi explored the use of DTs in a smart building scenario by replicating a part of the front facade[3]. The facade was visualized by assigning different yellow shades to the respective lux values received from the sensor. Several commercial solutions have emerged owing to their diverse possibilities and benefits. One example is the Azure Digital Twins (ADT)[4], a cloud-based service that aims to democratize DT deployment by providing software as a service solution. Steelcase, a company known for workspace designs, developed a space-sensing sensor network using ADT[5]. By implementing a suite of wireless infrared sensors, they generated analytics on how their spaces were being utilized, which in turn was used to enhance the reliability and efficiency. ICONICS[6] also utilized ADT to create a virtual representation of a physical space to improve the energy efficiency, optimize space usage, and lower costs.

2.2 Digital twins in COVID-19

Through real-time sensor data and accurate simulations, DTs can play a vital role in reducing the spread of COVID-19. Milne et al. modeled a city in Australia to understand the effectiveness of social distancing and reported that such distancing has been a substantial factor in flattening the epidemic curve[7]. A consortium among Aalto University, the Finnish Meteorological Institute, the VTT Technical Research Centre of Finland, and the University of Helsinki[8] studied the transmission of the virus by modeling possible scenarios in indoor spaces. They examined various situations, such as when a person coughs in an aisle in a grocery store. In a blog post by Sharma[9], the authors concluded that the traditional workplace model is ineffective in managing social distancing. In addition, Unity Technologies[10] built an open-source simulator concept for visualizing the spread of COVID-19 in a fictitious three-dimensional grocery store environment. Large industry players, including Google and Amazon, have also attempted to make social distancing hassle-free in indoor and outdoor spaces. Google released a web application called SODAR[11], which uses WebXR technology to help workers maintain the necessary distance. It operates by drawing a 2m circle around a user when walking and alerts the user if another person enters the circle. Amazon[12] also developed a mirror-like tool that helps employees observe physical distancing in an office workspace. Augmented reality and machine learning techniques are applied to provide visual feedback to employees. It portrays a person as being inside a red circle when entering within 6 feet of any other person.

2.3 Person detection

Pedestrian or person detection is a key research area in the computer vision domain. It has applications in autonomous vehicles, video surveillance, and robotics. In the early stage of pedestrian/person detection research, people used Haar wavelet features[13-15] or component-based pedestrian detection[14,16,17]. With an increase in computational power, researchers have started using gradient-based representations[17-20] and the deformable part-based model (DPM) and its variants[19,21,22]. Hosang et al. first used CNNs for pedestrian detection[23]. Although fast and faster RCNN methods have performed well for general object detection, they are unable to detect smaller pedestrians owing to the low resolution of the feature map. Zhang et al.

addressed this issue through feature fusion using a boosted forest technique[24]. In addition, Cao et al. introduced unified multi-layer channel features (MCFs) that integrate handcrafted features (HOG+LUV) in each layer of a CNN[25]. Tian et al. also optimized a pedestrian detection task using semantic segmentation to improve the hard negative detection[26]. To overcome the problem of occlusions and variations in illumination and lighting, Xu et al. proposed a cross-modality learning framework with input images from RGB and thermal cameras[27]. Wang et al. also addressed the occlusion problem by proposing a bounding box loss function, called a repulsion loss function[28].

2.4 Visualization of CNN

Although CNN-based object recognition has achieved an impressive performance, the use of a CNN faces the challenge of working with a black box. The features learned in different layers of the CNN are difficult to understand, unless we can visualize how they work. Explainable AI (XAI) appears to overcome these concerns, providing transparent models (white box) that allow humans to understand how an AI decision has been made; therefore, they do not rely on data only, but can be improved through human observations[29]. A brief literature survey on the application of CNN visualization techniques can be found elsewhere[30].

2.5 Summary

Past studies have primarily focused on the use of DTs in industrial scenarios[2]. Despite the literature on the use of twins in workspaces, only Nikolakis et al. focused on mapping the position and posture of an individual using expensive depth cameras[31]. Synthetic data have also proven to be a successful alternative for generating annotated datasets and is particularly essential during a pandemic. Moreover, we infer that the existing state-of-art object detection models fail to detect humans with the same degree of accuracy as they do in general object detection. Numerous approaches have been proposed to overcome this limitation. In this study, we address these limitations using the approaches detailed in the subsequent sections.

3 Our proposed approach

A unified approach for modeling DTs has not yet been developed. According to Tao et al.[2], a generic framework is critically required. The authors also outlined five dimensions that should be addressed while modeling a twin: a physical part, a virtual part, data, their connections, and service modeling. In collaboration with BT, we created a DT of their office workspace in Bengaluru, India. We built a three-dimensional (3D) representation of a 12-person meeting room and the surrounding area using the modeling, physics simulation, and rendering capabilities of Unity 3D. The physical and virtual worlds are connected through sockets. Specifically, we mapped the weather properties of the space, such as the temperature and humidity, measured using a DHT-11 sensor. Furthermore, through a ray-cast operation, the two-dimensional coordinates are mapped to the three-dimensional coordinates of the virtual world, and hence the movements of people are simulated. To ensure that the twin is as photorealistic as feasible for data generation, we employed the ray tracing[32] tools of Unity instead of the traditional Rasterized renderer. Ray tracing is a rendering technique that involves the tracing of individual rays of light as it bounces off virtual objects in a scene. Specifically, we used the path tracing algorithm[33] of Unity with a sample count of 4096, that is, the algorithm traces 4096 rays of light and requires 4096 frames to generate a single image. Hence, if the simulation runs at 30fps, it will take approximately 2.3min to generate a single image. To automate the process and increase the diversity of the dataset, we utilized the perception package of

Unity. We were able to generate high-fidelity ray-traced synthetic datasets of humanoids in a sitting or standing pose using the perception package and its in-built randomizer. By exploiting this randomizer, the humanoid poses, that is, the position and orientation, were changed according to a random seed with each iteration. By randomizing the poses on a fixed *z*-axis, we were also able to ensure that the humanoids did not clash with one another.

3.1 Planned physical setup

In the planned deployment, each meeting room has a set of weather monitoring sensors and cameras (Figure 1). Data from the sensors and cameras were collected and processed on a local computer. Processing involves noise cancelation from sensor readings through low-pass filtering and calculating the number of people inside each meeting room using a CNN. The processed data are sent to a central computer equipped with a high-end graphics processing unit (GPU) using network sockets. The VR-based DT will be deployed on this machine and updated with a real-time sensor feed. A demonstration video of the implementation can be found at https://youtube/XGYvDnwbyhM, and a web version can be found athttp://cambum.net/BT/BTWebGL/.

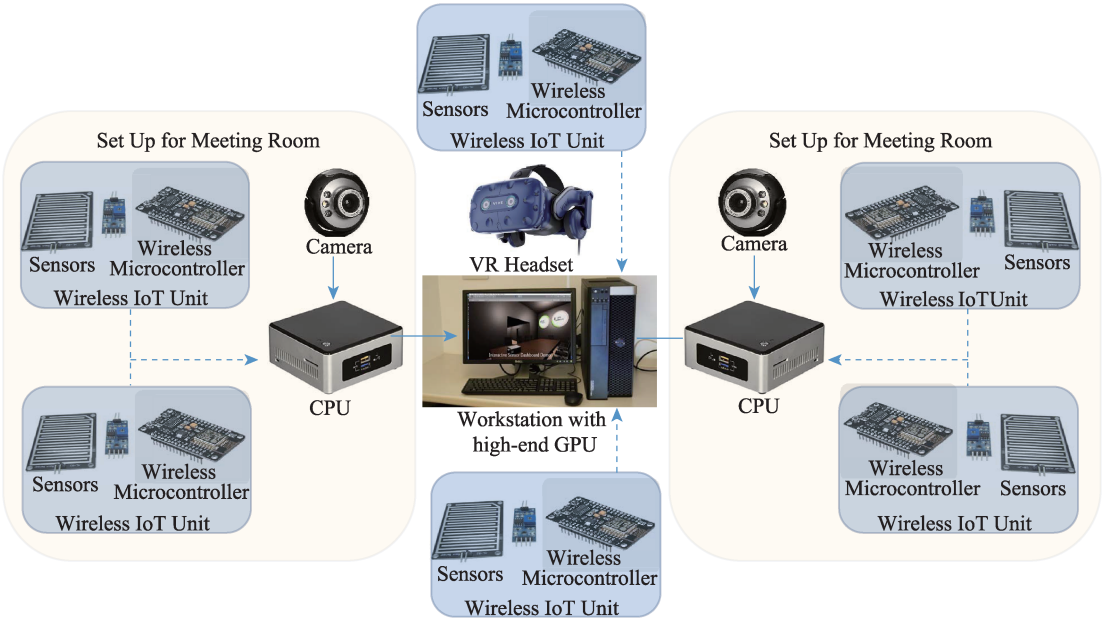


Figure 1 Planned setup of the VR-based DT.

Figure 1 shows a schematic diagram of the planned deployment of the DT implementation, gathering real-time data from a camera and IoT sensors (temperature and humidity). A similar setup was deployed earlier for the smart manufacturing capabilities[34].

3.2 Social distancing measurement through person detection

The system was designed in such a way that the system input devices (i.e., sensors and camera) need to be implemented within a physical office space. However, given the current situation in which most office spaces are still closed in many parts of India owing to the pandemic, the system cannot be deployed in the designated office space. Even after the office spaces are reopened, it will take a long time to generate appropriate data to validate the CNNs for person detection. Hence, we planned to generate synthetic data using a VR-based digital twin to validate a CNN-based person detection model. We chose YOLOv3 as our

person detection model based on the following studies:

(1) The performance of YOLOv3 was compared with a Faster RCNN, Mask RCNN, SSD, and RetinaNet in terms of accuracy and latency, and it was found that YOLOv3 is better than the other models[35,36].

(2) Redmon et al. also reported that YOLO performs better than other models when they fine-tune the model with artwork images and tested it on a synthetic dataset[37].

Finally, we measured the distance between each pair of humanoids and calculated the correlation of this measurement with that from the actual distance within the virtual environment. We describe the training, validation of YOLOv3, and distance measurement in detail in Section 5.

3.3 Explanation through visualization

To understand how well the CNN performs, we investigated two different types of CNN visualization techniques.

(1) The first type is visualizing the intermediate layers of a CNN model following Zeiler and Fergus[38]. This visualization technique is useful for understanding how successive convnet layers transform their inputs. It also gives us an idea of what types of features are extracted by different filters of different layers of the CNN model from the input images.

(2) The second, Grad-CAM based visualization[39], aims to understand which part of the image has a maximum association in predicting person classes. To obtain the class discriminative localization map corresponding to a specific class, we calculated the gradient with the feature maps of the last convolutional layer. These gradients were globally average pooled to obtain weights corresponding to the class, followed by a weighted combination of activation maps; finally, we applied a ReLU function. Thus, we obtained a coarse heatmap of the same size as the feature map in the last convolutional layer of the CNN model. In the final step, we resized the coarse heatmap to the input image size and overlapped the input image. Thus, the Grad-CAM-based heatmap helps us visualize which part of the image has a maximum association with the class of interest.

We applied these two techniques on both synthetically generated and real images to determine whether there are any differences in extracting features for predicting persons in the images. In the following sections, we describe our approach for developing a VR-based digital twin and use it to train and explain the function of the CNN in detail.

4 VR simulator development

4.1 Modeling

The construction of an accurate virtual twin requires precise information about the geometrical dimensions and physical properties of the object. Moreover, there is more than one way to implement such a twin. Building information modeling (BIM)[40] is a growing technology used in the AEC industry that advances the planning and design of infrastructure by portraying the building properties in 3D. BIM has been used in several previous studies[41,42], as well as in commercial services such as Tridify[43], PiXYZ[44], and Unity Reflect[45] to expedite the process of importing BIM files into game engines such as Unity. Another technique, highlighted by a Siemens patent[46], is the use of depth scanners to generate a point cloud illustration of a room and then match the point cloud data with the corresponding objects. However, owing

to the immediate lockdown and social distancing measures enforced in the wake of the COVID-19 pandemic, the above techniques are infeasible and could not be duly arranged. Hence, we manually modeled a part of the office workspace with the aid of an architectural drawing for our approach. We started with a meeting room that could accommodate a total of 12 people and then continued to the encompassing areas. We used Probuilder[47] and ProGrids[48] for modeling and rapid prototyping. The 3D models for workspace furniture were procured from the online markets TurboSquid[49] and Sketchfab[50] and accordingly placed within the environment.

4.2 Realistic rendering

Through multiple photographs taken using standard digital cameras, we were able to ascertain the different materials that made up the meeting room, and we aimed to replicate these materials in the twin through physically based rendering (PBR). PBR materials[51] enable the physical simulation of real-life material properties such that they accurately reflect the flow of light and thereby achieve photorealism. A PBR material entails several parameters such as albedo, metallic, and smoothness properties, as well as normal, height, diffuse, and occlusion maps. The respective texture maps used in our twin for the walls and floor mat were obtained from Freepbr.com.

Global illumination (GI) is one of the most significant factors determining how realistically a twin can resemble a real object. GI facilitates realistic light rendering by bouncing light off of surfaces; that is, it accounts for indirect light within the scene. We employed Baked GI for our environment, which entails computing the lighting and generating lightmap textures beforehand and is therefore computationally inexpensive during runtime. Its counterpart, Realtime GI, involves calculating the light during runtime and places a substantial load on the GPU. Furthermore, reflection probes are placed within the environment to simulate reflections and strengthen photorealism. Finally, the Post-Processing tool of Unity is used to implement anti-aliasing, ambient occlusion, color grading, and auto-exposure. The final results are shown in Figure 2.

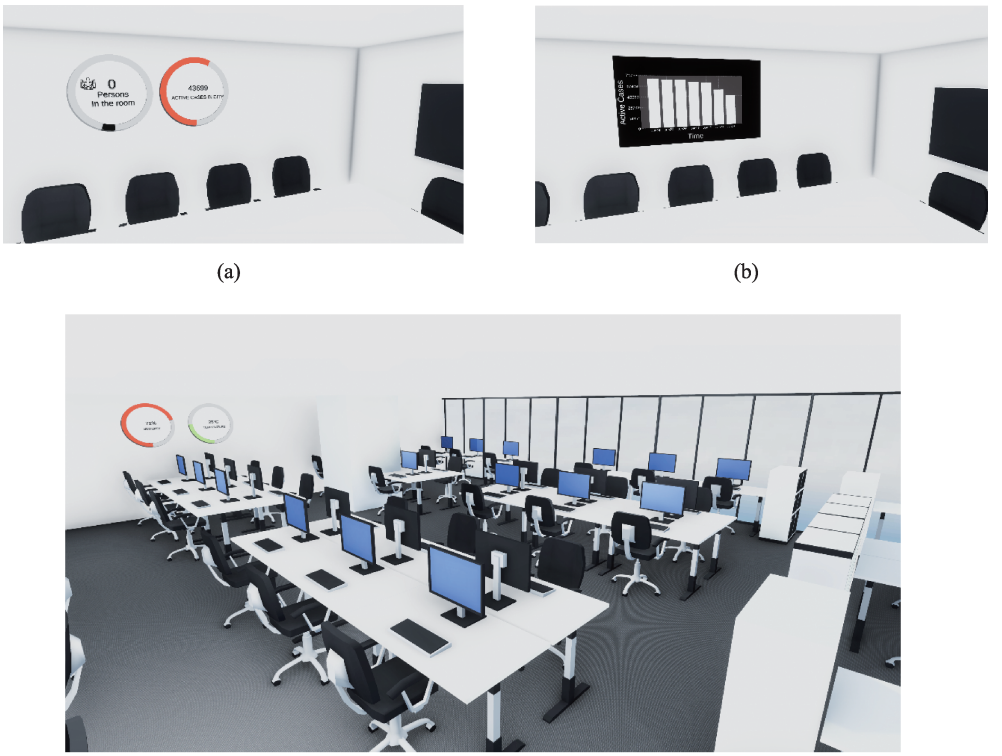
(a) (b)

Figure 2 Digital twin rendered with baked global illumination and post-processing using Unity.

For smoother processing, we optimized the twin by deleting several unnecessary polygons, such as the height adjusters in the chairs and trays underneath the desks. Low-poly humanoid models were placed within the environment for recognition by the person detection model. Their behavior was driven using NavMeshAgents of Unity[52]. Here, the agents avoid each other and other obstacles in a scene through spatial reasoning obtained from a baked NavMesh. We also enabled ray-traced rendering in a virtual environment by employing the path-tracing algorithm of Unity. In this context, physically based rendering is a category of virtual materials that mimics the physical properties of real-world materials. Finally, we compared the performances of rasterized rendering and raytracing.

4.3 Interactive dashboards

We configured interactive dashboards inside a VR-based workspace simulator, which displayed real-time sensor data such as the temperature and humidity, and the latest statistics regarding the coronavirus pandemic at the place of deployment. The sensors were interfaced with the VR machine through their respective wireless modules. These wireless modules used a peer-to-peer connection to communicate with a VR machine using the UDP protocol at a frequency of 1 Hz. Data obtained from the temperature and humidity sensors are shown as separate circular bars (Figure 3c). Instantaneous values are converted into time-series values when users dwell using an eye gaze while wearing an HTC Vive Pro Eye headset or when selecting a dial using a hand-controller, thereby providing a detailed view (Figure 3b). The color of the circular bars changes if the value exceeds a predefined threshold (Figure 3c). Any abrupt changes in the sensor readings are also reflected instantly through both visual and haptic feedback. Haptic feedback is generated through the hand controller. The live sensor data values can be used further for making decisions regarding air conditioning or maintaining the room temperature of the office workspace.



(a) (b)

(c)

Figure 3 VR Model of the office space.

In addition, the dashboard displays real-time statistics from the coronavirus pandemic obtained from the COVID-19-India API (https://api. covid19india. org/). The dashboard shows the number of active cases within the region where the actual workspace is located. The data are shown as a circular bar (Figure 3a) depicting the number of active cases to date. When a user dwells using an eye gaze, detailed statistics are shown for the latest phase[53] as a bar graph (Figure 3b).

4.4 Connecting CNN to VR environment

The physical implementation involves processing live video in a separate computer and sending the number of people detected in the live video feed to the VR setup. However, at the present stage of

development, and given the previously mentioned constraints from COVID-19, we connected the CNN model for detecting humanoids within the VR environment through a real-time streaming protocol (RTSP) connection, streaming the game view of the Unity camera to the CNN where the person detection process (as described in Section 5.1) occurs. Once the person detection results are obtained, we filter our predictions using the corresponding confidence scores. We select persons with a confidence score of greater than 0.6, and if such a person is found, we stream the results back to Unity through a UDP connection. Currently, there is no in-built option in Unity to stream its camera view; therefore, we built a custom solution using the FFmpeg module and an RTSP server. These functions were implemented to stream the Unity view through the RTSP connection. Because the CNN processing speed differs from the streaming speed of Unity, we considered the latest sample of the RTSP buffer to pass on to the CNN. We tested the person detection model on videos recorded in both real and virtual worlds. The model processes each frame and localizes persons/humanoids when detected in the frame (Figure 4). Localization is achieved by annotating the bounding box around the person. Figure 4 shows each person annotated with a bounding box labeled with a number.

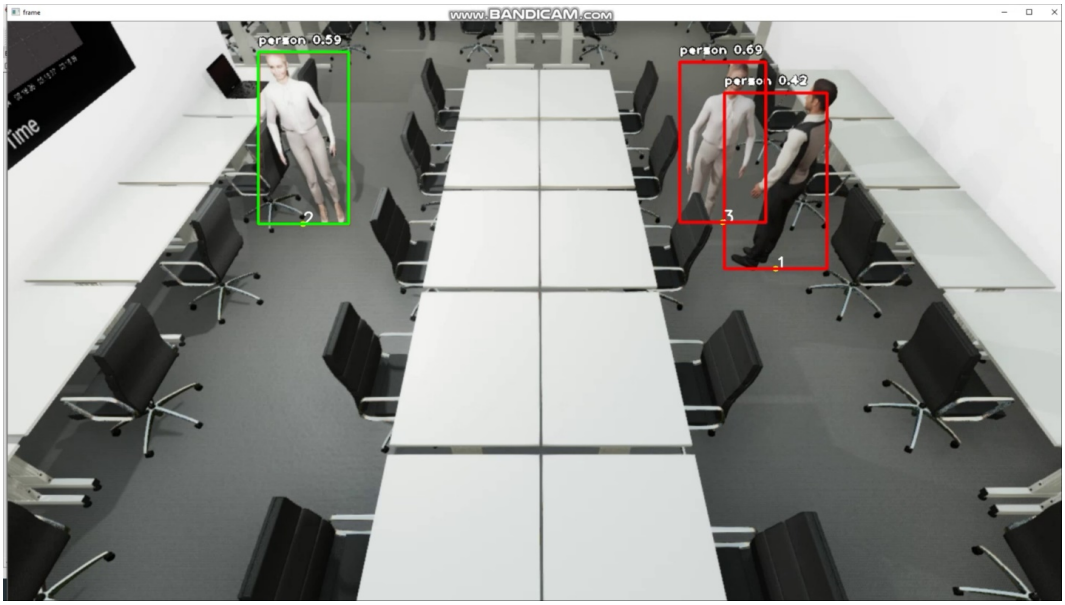


Figure 4 Humanoids detected by person detection model and annotated with bounding boxes. Here, the bounding boxes turn to red if social distancing is violated and are otherwise green.

Once Unity receives the object results, we add or delete humanoids inside the virtual environment. Digital humanoid models are composed of more than 90 different links/joints and 140 degrees, similar to those used in many biomechanical models of the human body[54]. We used motion capture data of Mixamo[55] to automatically rig the armature of the humanoid (base skeleton rig) to reflect realistic human poses.

4.5 Comparison with similar approaches

The ParallelEye dataset[56] applies an approach similar to our method using a VR-based synthetic dataset for autonomous vehicles. UnrealROX[57] is another tool built over the Unreal engine to generate photorealistic synthetic datasets but is targeted more toward research into robotic vision. The tool focuses on simulating a broad range of indoor robotic activities in terms of both object interactions and pose. We extended the idea for a different use case and compared the system in terms of accuracy with a real dataset.

Different methods for generating synthetic datasets use variable autoencoders (VAE)[58−60] and generative adversarial networks (GANs)[61]. We also compared our approach using the same setup. We ran real images through a GAN implementation. The GAN consists of a generator, which tries to fool another network, known as a discriminator, that learns to distinguish between real and fake images. We used one version of a GAN, called SinGAN[62], which is an unconditional generative model that can be trained on a single image. The model learns the internal distribution of the image patches[63−66] using multi-scale adversarial training and can generate similar images of different scales. This model is similar to a GAN model, except that the training samples are patches of the input image rather than a set of images, and the network consists of a pyramid[67−69] of GANs of different scales. As the authors of the study on SinGAN claimed, it might produce unrealistic and distorted results on coarser scales. Still, we were able to generate realistic fake images on finer scales that were indistinguishable from the real image. At finer scales, the generator learns a smaller patch distribution than at a coarser scale, giving better results in smaller scales and preserving the global structure of the image (Figure 5).

Figure 5 (a) original input images (b) random samples from a single image at n=6, n=11, and n=25.

There are no existing VAE-based algorithms that take a single image and can synthesize as many fake images as desired. If we have a sufficient dataset, the VAE can capture the distribution and generate more data from the same distribution. Conventional GANs have problems of non-convergence [70] and mode collapse [71], which researchers have improved over time. Although a SinGAN model can synthesize more indistinguishable fake images similar to the original image, as shown in Figure 5, it offers less customization compared to a VR-based DT. As shown in earlier sections, in a DT, we can easily change the number, clothing colors, and postures of the persons in an image dataset while keeping the background and ambient light constant. Although it was found that a VAE can detect the camera rotation and emotion of Frey faces [72], neither VAE nor GAN can add multiple objects or persons in an image while keeping a few features constant and varying the others.

5 Accuracy comparison of person detection

5.1 Model preparation

We used a transfer learning technique to fine-tune the model with a person dataset downloaded from the Open Images Dataset[73]. This dataset contains both real and artwork images. We used 2022 images in total with the "person" label showing single or multiple people. We separated the complete dataset into an 80:20

ratio for training and validation. We prepared the dataset by converting the annotation files into an xml format. The existing annotation files were in Darknet format, which is the actual backend used for YOLO training. We trained the model using the model using the "Keras" backend. The model was trained for 200 epochs, with a batch size of 4. We used an NVIDIA GeForce RTX 2070 GPU to train the model and conduct a performance testing on the images. Finally, we tested our trained model on both real and synthetic sequences of images.

5.2 Data preparation

To test the model on real and synthetic images, we recorded short videos in the physical world (Figure 6d) and the virtual environment (Figure 6d). In the physical world video, we recorded multiple situations such as occluding persons and varying lighting conditions in the same office space used for VR modeling (Figure 6c). In the virtual environment video, we recorded the ambient light and room setup as constant parameters and the following as independent parameters: (1) changing the number of humanoids in the frames (one to four humanoids), (2) posture of humanoids (seating and standing), and (3) occluding humanoids (yes or no). We tested a total of 9000 images divided into three classes: real images, synthetic images generated without ray tracing, and synthetic images generated through ray tracing. Because ray tracing is computationally intensive, we considered synthetic images without ray tracing, and in a practical implementation, we may need to deploy the DT without ray tracing based on the availability of the GPUs. We calculated the accuracy of our model using the following formula: $Accuracy = (TP + TN)/(TP + FP + FN + TN)$, where TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively.

(a) (b) (c) (d)

Figure 6 Training and test data samples: (a) Training data with real person image, (b) training data with artwork image, (c) testing on synthetic world image in which a false negative result was obtained, and (d) test result on real-world image in which a false positive result was obtained with four persons detected despite only three persons being present.

5.3 Results

We found an overall accuracy of 96.044% (standard error of 0.186) for real images, 96.981% (standard error of 0.126) for synthetic images without using ray tracing, and 94.25% (std error of 0.974) for synthetic images using ray tracing. We analyzed the accuracies to determine if the performance of the CNN was significantly different between real and synthetic images. We listed the accuracies for all conditions (different numbers of persons, postures, and occlusions) separately and found that, except when one person was occluded, the interquartile range for all conditions was zero, and the median and first and third quartiles were 100% for both real and synthetic images (Table 1, where the numbers in the brackets indicate the interquartile range). Table 2 lists the proportion of images for all conditions, with an accuracy of less than 100%. Owing to this skewness of the samples, we did not conduct standard ANOVA and median tests.

Table 1 Median accuracy across different conditions

Conditions	Real image	Synthetic image without ray tracing	Synthetic image with ray tracing
One person standing without occlusion	100(0)	100(0)	100(0)
One person standing with occlusion	100(0)	100(0)	100(0)
One person sitting without occlusion	100(0)	100(0)	100(0)
One person sitting with occlusion	100(0)	100(0)	100(0)
Two persons standing without occlusion	100(0)	100(0)	100(0)
Two persons standing with occlusion	100(0)	100(0)	100(0)
Two persons sitting without occlusion	100(0)	100(0)	100(0)
Two persons sitting with occlusion	100(25)	100(0)	100(43.75)
Three persons standing without occlusion	100(0)	100(0)	100(15)
Three persons standing with occlusion	100(0)	100(20)	100(25)
Three persons sitting without occlusion	100(0)	100(0)	100(0)
Three persons sitting with occlusion	100(20)	100(0)	75(33.33)

Table 2 Proportion of images with accuracy of less than 100%

Conditions	Real image	Synthetic image without ray tracing	Synthetic image with ray tracing
One person standing without occlusion*	3.28	0.00	0.00
One person standing with occlusion	0.00	0.00	0.00
One person sitting without occlusion*	0.00	9.48	0.00
One person sitting with occlusion*	0.36	0.00	10
Two persons standing without occlusion	0.00	0.00	0.00
Two persons standing with occlusion	18.25	18.97	10
Two persons sitting without occlusion	0.00	1.83	0.00
Two persons sitting with occlusion*	37.59	0.33	40
Three persons standing without occlusion*	15.69	1.00	30
Three persons standing with occlusion	22.63	29.62	30
Three persons sitting without occlusion*	11.31	0.00	0.00
Three persons sitting with occlusion*	44.89	20.63	60

Notes: *Statistically significant differences among the seven conditions.

5.4 Distance measurement

Bertoni et al. worked on the silhouettes of people in an outdoor environment using 3D distances[74]. In this study, we measured the distance between persons in an indoor environment. We first fixed the camera at a particular height of the room in a virtual environment. We then recorded the distance between each pair of humanoids detected in each frame in pixels using the Unity tool. We used a trained model to generate a set of bounding boxes and a unique ID for each humanoid. To measure the distances between detected humanoids, we calculated the distance between humanoids from bounding box references generated by YOLO following Punn's study[36] using video recorded by a road surveillance camera. We calculated the bounding boxes and the corresponding centroids for each box in a frame recorded through the VE. The bounding boxes are shown in red or green, and the corresponding centroids are shown in yellow, as indicated in Figure 4. We computed the pairwise Euclidean distance between centroids using (1) and by applying a $p \times p$ matrix, where p denotes the number of persons detected at any instance.

$$d(m,n) = \sqrt{\sum_{i=1}^{p}(m_i - n_i)^2} \tag{1}$$

Here, p is a two-dimensional space, and m and n are two centroids in a 2D space. Finally, we measured the correlation of this measurement with the distance measured through the virtual environment, the correlation coefficient of which was r=0.99, p<0.01 (Figure 7).

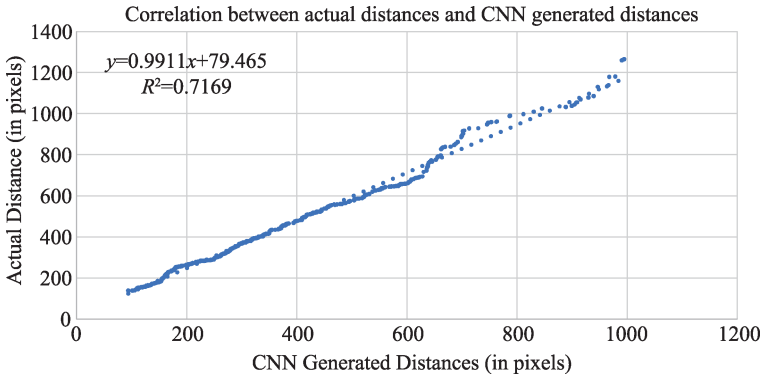


Figure 7 Scatter plot of distances measured from virtual environment and through a CNN.

5.5 Discussion

We compared the performance of a representative CNN model among real and synthetically generated images using a VR DT and noted that the interquartile range of accuracies across different combinations of numbers of persons, occlusions, and postures was zero for the first, second, and third quartiles at 100%. The highest number of images with less than 100% accuracy occurred when one or more persons were occluded, and was similar for both real and synthetic images. We observed a few cases (for example, two sitting persons with occlusions) in which the accuracy on the real images was comparatively lower than that on the synthetic images. This might be due to uncontrolled lighting illumination, the similar color contrast between the clothes of the persons and the background, and other factors. The difference in accuracies among the three conditions was less than 2%, which does not have a significant effect on the practical use. The calculated distances from the synthetic image correlated with the 0.99 coefficient under real distances.

We found three conditions (two persons sitting with an occlusion, three persons standing without an occlusion, and three persons sitting with and without an occlusion) in which the accuracy on real images was comparatively less than on synthetic images without ray tracing. We observed that, although the model was able to detect persons under such conditions in the real world, the false positive rate was higher. This might be due to uncontrolled ambient lighting conditions in the real world and an indistinguishable similarity between clothes colors and the background in the images.

6 Explanation through visualization

In the previous section, we mention that YOLO had the lowest accuracy when one or more persons were partially occluded. To understand this result, we used an intermediate layer visualization technique and the Grad-CAM technique to explain the performance of the person detection model. Grad-CAM calculates each pixel value of the feature maps in the last convolution layer on the predicted class[39]. It does not require any information related to bounding box regression, which is typically used to localize an object in an image. Because the YOLO model does not allow reading data from intermittent layers, we used a VGG16 classification model pretrained with the ILSVRC ImageNet dataset. We prepared our dataset by combining five different classes (i.e., *airplane*, *bicycle*, *car*, *motorbike*, and *face*) of images downloaded

from Kaggle, Google Image, the Caltech face image dataset, and the Georgia Tech Face database. We trained the model using a total of 3513 images separated into training and validation datasets (80∶20) for 100 epochs. To understand how the CNN model can classify the input image, we need to understand how our model sees the input image by looking at the output of its intermediate layers. We visualized the activations in the $\left(\frac{n}{4}\right)th$ convolutional layer, $\left(\frac{n}{2}\right)th$ convolutional layer, $\left(\frac{3n}{4}\right)th$ convolutional layer, and $n-th$ convolutional layer of the trained model. To visualize the heatmap generated by the Grad-CAM method, we used a pretrained VGG16 model. Although this pretrained model does not include any person class, it has different classes related to clothes (e.g., *t-shirt* and *jeans*), which are relevant for the localization of the individual in the image. We generated a heatmap corresponding to the *t-shirt* and *jeans* classes to identify people in the images. We visualized the performance of the CNN for person prediction in both real and synthetically generated images. We generated the output of the CNN from the layers mentioned above for both types of images to understand whether the CNN handles synthetically generated and real images differently or in the same manner. We found that the first few convolution layers of the model extracted the basic features (edges and contours) of the object and retained the maximum information from the input image (Figures 8b and 9b). As we found deeper in the model, the activations became less visually interpretable (Figures 8c−8e and 9c−9e). The model started to extract abstract features (e.g., patch-based features such as the texture of the body parts of a humanoid in Figure 8 or a person in Figure 10). At a deeper level of network resolution, the feature map starts decreasing, whereas the spatial information increases. If we observe all four feature-map outputs (Figures 8c−8e and 9c−9e), it is evident that in each transformation model, the background or any irrelevant information is eliminated, and useful information related to the class of objects is refined.

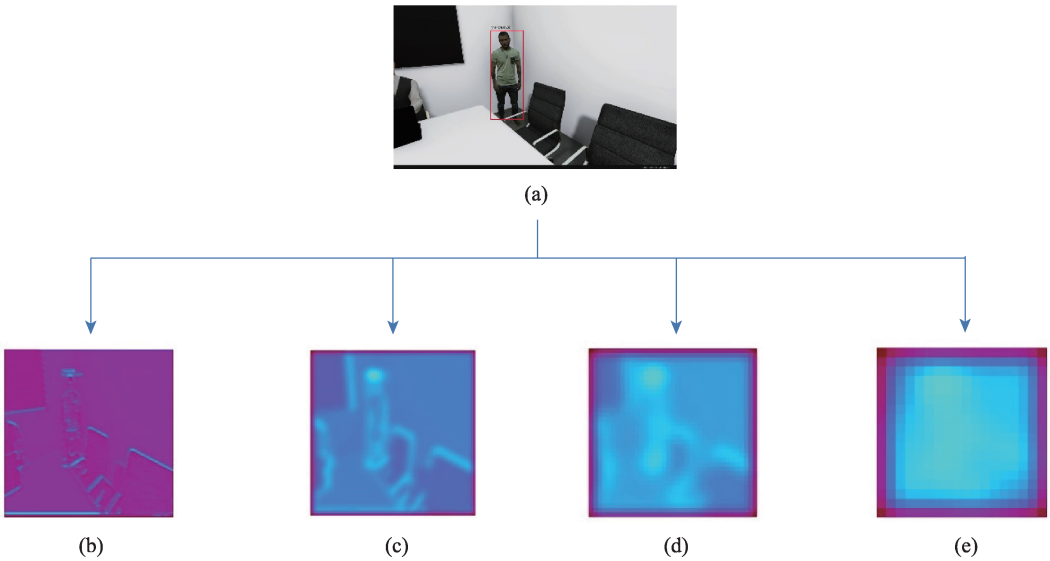


Figure 8 (a) Input image (humanoids in a synthetic generated image). A red box indicates that YOLO detected a person. (b) 28th channel of the activation of the 3rd convolution layer, (c) 28th channel of the activation of the 7th convolution layer, (d) 28th channel of the activation of the 10th convolution layer, and (e) 510th channel of the activation of the 13th convolution layer (please note that this figure is best viewed in its digital form).

We also visualized heatmaps of the class activation to understand which part of the object allowed the model to correctly classify the object. In this context, the class activation map indicates which part of an image corresponds to a class of objects. In Figure 10, we show a heatmap of synthetically generated images in real-world images (Figures 10a−10c). We found that different body areas were strongly

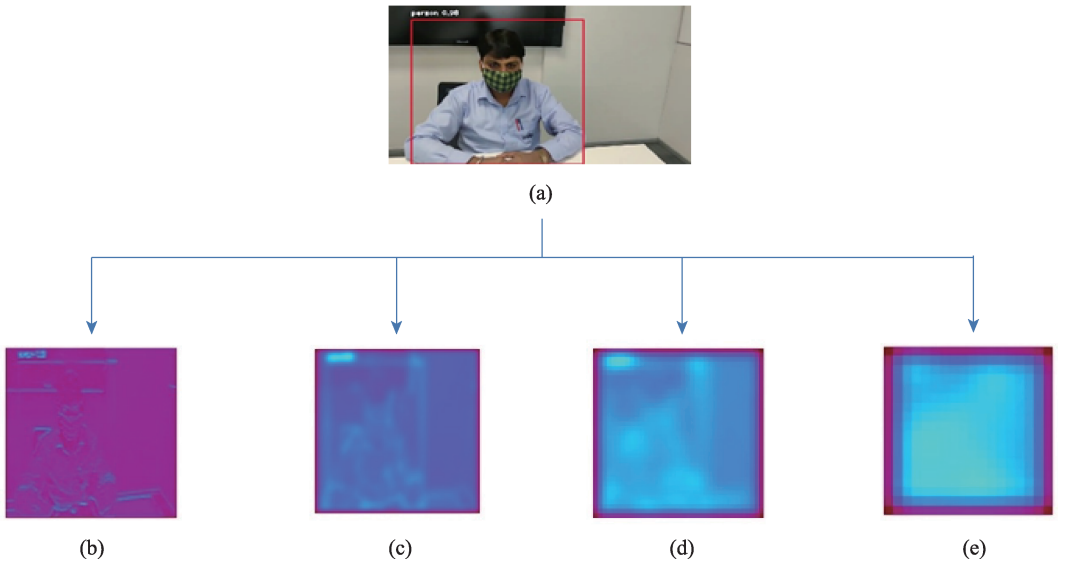


Figure 9 (a) Input test image (person in real-world image), (b) 28th channel of the activation of 3rd convolution layer, (c) 28th channel of the activation of 7th convolution layer, (d) 28th channel of the activation of the 10th convolution layer, and (e) 510th channel of the activation of the 13th convolution layer (please note that this figure is best viewed in its electronic form).

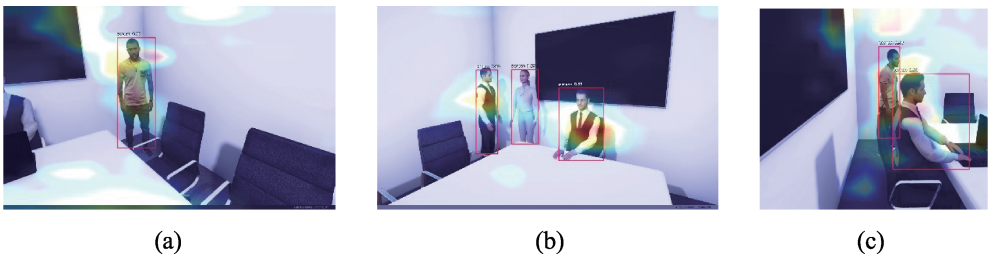


Figure 10 Grad-CAM based heatmap for three different situations where the performance of YOLOv3 differed in terms of accuracy (shown in the red bounding boxes). (a) YOLOv3 failing to detect a partially occluded person, (b) YOLO detecting all individuals, and (c) YOLOv3 detecting all individuals with different postures and different colored clothing.

activated, with brown corresponding to the highest gradient score and cyan corresponding to the lowest gradient score. Heatmap-based visualization helps us identify the part of the image that contributes to the false positive or false negative results.

To understand how different independent variables contributed to the CNN performance, we tested synthetic images with different parameters. We started with an image of an occluded person, where YOLOv3 failed to detect individuals, and the accuracy dropped to 50% (Figure 10a). We found that the occlusion made it difficult for the model to obtain sufficient information from a partially occluded person and achieve localization, although it could generate a heatmap for whole body areas of a standing person who was fully visible (Figure 10a). We conducted a second check with a different image in which YOLOv3 was able to detect all people (Figure 10b). When we looked closely at the heatmap areas, we found a strong association between the bounding box region and the heatmap areas. Although the female humanoid in this image was weakly classified, the heatmap covered the maximum upper body parts visible in the image for all three individuals. As previously mentioned, brown corresponds to the highest gradient score, and cyan corresponds to the lowest gradient score. We tested the heatmap with a third image where

the humanoid figures were wearing different colors (green and white) and were positioned in different postures (i.e., one person was sitting, and a second person was standing). The Grad-CAM heatmap provided a strong visual cue regarding the location of these two different individuals (Figure 10c). These results confirm our idea of using this visualization technique to analyze the failure modes of a CNN model and take corrective steps, such as increasing the camera field of view and location to record a full-frontal view of the humanoids, or in this case, to increase the accuracy of the model.

7 General discussion

7.1 Summary

In this paper, a new method is proposed for validating the accuracy of a CNN through a customized synthetic video generated in an immersive environment. A case study demonstrated the possible application of this implementation in the development of an automated social distance measurement system in a physical office space. We presented the training and testing accuracy in detecting individuals using a CNN-based person detection model and used data visualization techniques to describe the operation of the model for both real-world and synthetically generated videos.

7.2 Accuracy of person detection

We achieved 100% median accuracy for person detection and a 0.99 correlation for physical distance measurements. The applications a CNN are rapidly evolving with new models frequently appearing in the literature, and the accuracy of person detection even under the presence of occlusions can be further increased using customized CNN models. However, it should be noted that this study is not focused on the development of a CNN for person detection, and rather proposes a new way of validating a CNN model using a VR-based synthetic dataset. If a different CNN model other than YOLOv3 is used, we can also train it with a synthetic dataset and can achieve a similar accuracy in a real-life deployment. Although the present social distance measurement algorithm works within the visual field of a webcam inside a room, future versions will implement 3D distance measurements such as Bertoni's[74] monocular 3D localization algorithm.

7.3 Utility

The proposed VR prototype will be deployed as a VR-based DT of an office space implementing real-time person detection and environmental variable monitoring capabilities through interactive dashboards. In addition, the VR interface will show real-time COVID-19 statistics at the place of deployment and measure the number of people in the space, as well as their relative position and posture. This can be extremely valuable for monitoring social distancing measures in office spaces. As a second benefit, an observer can undertake a detailed remote virtual walk through an office space, which would not be possible with a standard multi-screen video from a security camera.

The concept of validating a CNN through synthetic video can have utility beyond this particular use case of measuring the social distancing in an office space. For example, for both unmanned ground and aerial vehicles, synthetic videos can be used to validate machine vision systems where real-time video generation is difficult, for example, inside a hazardous location such as a nuclear power station or a high-security zone such as inside a military facility.

7.4 Value addition

During the past few months, a plethora of computer vision projects for calculating social distancing have been conducted. However, most of these systems have not been rigorously validated as traditional machine vision systems for autonomous vehicles or face recognition owing to a lack of appropriate data. Bertoni et al. algorithm was validated for outdoor environments but not for indoor office workspaces[74]. Our paper proposes a new method for validating a machine-learning-based person detection system using synthetically generated video in an immersive environment.

DTs are traditionally used to optimize or simulate a process lifecycle or the maintenance of assets. Our study proposes a new use of DTs to enhance workplace safety by measuring social distance.

We also demonstrated an application of the visualization technique using synthetic images to understand why CNN-based object detection models work or fail to detect individuals from an image. We compared the performance of YOLO using different independent parameters to understand how it operates under different situations. The heatmap-based visualization helped us obtain a visual explanation of the operation of the CNN model. This approach is novel in that a similar approach can be used for other CNN models for different applications, and is a step toward the collective goal of XAI.

8 Conclusion

This paper presented a VR-based DT implementation of a physical office space with the goal of using it as an automatic social distancing measurement system. The VR environment was enhanced with an interactive dashboard showing information collected from physical sensors and the latest statistics on COVID-19. Moreover, we implemented a CNN to identify people within a room. Given the present pandemic, where lockdown and social distancing measures have impeded the collection of real large-scale image and video data, we proposed a new technique for training and validating a CNN through synthetic images generated through the VR DT. We also used two different data visualization techniques to explain how a complex CNN operates, aiming toward the advancement of XAI, and used it to improve the CNN performance. We hope that the proposed solution will help measure the occupancy accurately and contribute to enhancing the safety of workspaces by enforcing social distancing measures.

Declaration of competing interest

We declare that we have no conflict of interest.

References

1 Glaessgen E, Stargel D. The digital twin paradigm for future NASA and US air force vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference20th AIAA/ASME/AHS Adaptive Structures Conference14th AIAA. Honolulu, Hawaii, Reston, Virigina, 2012,1818
DOI:10.2514/6.2012-1818

2 Tao F, Zhang H, Liu A, Nee A Y C. Digital twin in industry: state-of-the-art. IEEE Transactions on Industrial Informatics, 2019, 15(4): 2405−2415
DOI:10.1109/tii.2018.2873186

3 Khajavi S H, Motlagh N H, Jaribion A, Werner L C, Holmström J. Digital twin: vision, benefits, boundaries, and creation for buildings. IEEE Access, 2019, 7: 147406−147419
DOI:10.1109/access.2019.2946515

4 Khelifi I. Azure Digital Twins: Powering the next generation of IoT connected solutions, Microsoft Azure Blog. [2020-8-5] Available from: https://azure. microsoft. com/en-in/blog/azure-digital-twins-powering-the-next-generation-of-iot-

connected-solutions/

5 Steelcase 2019, Microsoft Corporation. [2020-8-5] Available from: https://customers. microsoft. com/en-US/story/steelcase-manufacturing-azureiot-en

6 ICONICS 2018, Microsoft Corporation. [2020-8-5] Available from: https://customers.microsoft.com/en-us/story/iconics-partner-professional-services-azure-iot

7 Milne G J, Xie S. The effectiveness of social distancing in mitigating COVID-19 spread: a modelling analysis. 2020

8 Vuorinen V, Aarnio M, Alava M, Alopaeus V, Atanasova N, Auvinen M, Balasubramanian N, Bordbar H, Erästö P, Grande R, Hayward N, Hellsten A, Hostikka S, Hokkanen J, Kaario O, Karvinen A, Kivistö I, Korhonen M, Österberg M. Modelling aerosol transport and virus exposure with numerical simulations in relation to SARS-CoV-2 transmission by inhalation indoors. Safety Science, 2020, 130: 104866
DOI:10.1016/j.ssci.2020.104866

9 Sharma S. Social distancing in the workplace: the new norm, Buro Happold. [2020-8-7] Available from: https://www.burohappold.com/articles/social-distancing-in-the-workplace/#

10 Fort J, Crespi A, Elion C, Kermanizadeh R, Wani P, Lange D. "Simulation + Coronavirus", Unity Technologies White Paper 2020.

11 Google Sodar n.d., GoogleInc. [2020-8-7] Available from: https://sodar.withgoogle.com

12 BradPorter, Amazon introduces 'Distance Assistant', The Amazon Blog. [2020-8-9] Available from: https://blog.aboutamazon.com/operations/amazon-introduces-distance-assistant

13 Papageorgiou C, Poggio T. A trainable system for object detection. International Journal of Computer Vision, 2000, 38 (1): 15–33
DOI:10.1023/a: 1008162616689

14 Mohan A, Papageorgiou C, Poggio T. Example-based object detection in images by components. IEEE transactions on pattern analysis and machine intelligence, 2001, 23(4): 349–361
DOI: 10.1109/34.917571

15 Viola P, Jones M J, Snow D. Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision, 2005, 63(2): 153–161
DOI: 10.1007/s11263-005-6644-8

16 Wu B, Nevatia R. Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. Beijing, China, IEEE, 2005, 90–97
DOI:10.1109/iccv.2005.74

17 Wang X Y, Han T X, Yan S C. An HOG-LBP human detector with partial occlusion handling. In: 2009 IEEE 12th International Conference on Computer Vision. Kyoto, Japan, IEEE, 2009, 32–39
DOI:10.1109/iccv.2009.5459207

18 Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. SanDiego, CA, USA, IEEE, 2005, 886–893
DOI:10.1109/cvpr.2005.177

19 Girshick R, Felzenszwalb P, McAllester D. Object detection with grammar models. Advances in Neural Information Processing Systems, 2011, 24: 442–450

20 Zhu Q, Yeh M C, Cheng K T, Avidan S. Fast human detection using a cascade of histograms of oriented gradients. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. NewYork, NY, USA, IEEE, 2006, 1491–1498
DOI:10.1109/cvpr.2006.119

21 Felzenszwalb P F, Girshick R B, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence, 2009, 32(9): 1627–1645
DOI:10.1109/tpami.2009.153

22 Sadeghi M A, Forsyth D. 30Hz object detection with dpm v5. European Conference on Computer Vision. Springer, Cham, 2014, 65–79

23 Hosang J, Omran M, Benenson R, Schiele B. Taking a deeper look at pedestrians. In: 2015 IEEE Conference on

Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA, IEEE, 2015, 4073−4082
DOI:10.1109/cvpr.2015.7299034

24 Zhang L, Lin L, Liang X, He K. Is faster R-CNN doing well for pedestrian detection? European conference on computer vision. Springer, Cham, 2016, 443−457

25 Cao J L, Pang Y W, Li X L. Learning multilayer channel features for pedestrian detection. IEEE Transactions on Image Processing, 2017, 26(7): 3210−3220
DOI:10.1109/tip.2017.2694224

26 Tian Y L, Luo P, Wang X G, Tang X O. Pedestrian detection aided by deep learning semantic tasks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA, IEEE, 2015, 5079−5087
DOI:10.1109/cvpr.2015.7299143

27 Xu D, Ouyang W, Ricci E, Wang X, Sebe N. Learning cross-modal deep representations for robust pedestrian detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, 5363−5371

28 Wang X L, Xiao T T, Jiang Y N, Shao S, Sun J, Shen C H. Repulsion loss: detecting pedestrians in a crowd. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 7774−7783
DOI:10.1109/cvpr.2018.00811

29 Hagras H. Toward human-understandable, explainable AI. Computer, 2018, 51(9): 28−36
DOI:10.1109/mc.2018.3620965

30 Mukhopadhyay A, Mukherjee I, Biswas P. Decoding CNN based object classifier using visualization. In: 12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. Virtual Event DC USA, New York, NY, USA, ACM, 2020, 50−53
DOI:10.1145/3409251.3411721

31 Nikolakis N, Alexopoulos K, Xanthakis E, Chryssolouris G. The digital twin implementation for linking the virtual representation of human-based production tasks to their physical counterpart in the factory-floor. International Journal of Computer Integrated Manufacturing, 2019, 32(1): 1−12
DOI:10.1080/0951192x.2018.1529430

32 Ray tracing n.d., UnityTechnologies. [2020-5-10] Available from: https://unity.com/ray-tracing

33 Path tracing n.d., UnityTechnologies. [2020-5-10] Available from: https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@7.1/manual/Ray-Tracing-Path-Tracing.html

34 Murthy L R D. Multimodal interaction for real and virtual environments. In: International Conference on Intelligent User Interfaces, Proceedings IUI. Association for Computing Machinery, 2020, 29−30
DOI:10.1145/3379336.3381506

35 Mukhopadhyay A, Mukherjee I, Biswas P. Comparing CNNs for non-conventional traffic participants. In: Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications: Adjunct Proceedings. Utrecht Netherlands, New York, NY, USA, ACM, 2019, 171−175
DOI:10.1145/3349263.3351336

36 Punn N S, Sonbhadra S K, Agarwal S. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. 2020

37 Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, IEEE, 2016, 779−788
DOI:10.1109/cvpr.2016.91

38 Zeiler M D, Fergus R. Visualizing and understanding convolutional networks. European conference on computer vision. Springer, Cham, 2014, 818−833

39 Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy, IEEE, 2017, 618−626
DOI:10.1109/iccv.2017.74

40 Volk R, Stengel J, Schultmann F. Building Information Modeling (BIM) for existing buildings—Literature review and future needs. Automation in Construction, 2014, 38, 109−127
DOI:10.1016/j.autcon.2013.10.023

41 Kupriyanovsky V, Klimov A, Voropaev Y, Pokusaev O, Dobrynin A, Ponkin I, Lysogorsky A. Digital twins based on the development of BIM technologies, related ontologies, 5G, IoT, and mixed reality for use in infrastructure projects and IFRABIM. International Journal of Open Information Technologies, 2020, 8(3): 55–74

42 Lu Q C, Xie X, Heaton J, Parlikad A K, Schooling J. From BIM towards digital twin: strategy and future development for smart asset management. In: Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future. Cham: Springer International Publishing, 2019, 392–404
DOI:10.1007/978-3-030-27477-1_30

43 Tridify 2020, Tridify Ltd. [2020-8-6] Available from: https://www.tridify.com

44 PiXYZ Plugin n.d., PiXYZSoftware. [2020-8-6] Available from: https://www.pixyz-software.com/plugin/

45 Unity Reflect n.d., UnityTechnologies. [2020-8-6] Available from: https://unity.com/products/unity-reflect

46 Tao F, Zhang H, Liu A, Nee A Y. Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics, 2018, 15(4): 2405–2415

47 ProBuildern.d., UnityTechnologies. [2020-8-6] Available from: https://unity3d.com/unity/features/worldbuilding/probuilder

48 ProGridsn.d., UnityTechnologies. [2020-8-6] Available from: https://docs.unity3d.com/Packages/com.unity.progrids@3.0/manual/index.html

49 TurboSquid 2020. [2020-8-6] Available from: https://www.turbosquid.com/

50 Sketchfab 2020. [2020-8-6] Available from: https://sketchfab.com/

51 LeBlanc H. What are PBR Materials, E-on Software Blog. [2020-8-6] Available from: https://info.e-onsoftware.com/learning_vue/what-are-pbr-materials

52 NavMesh Agents n.d., UnityTechnologies. [2020-8-6] Available from: https://docs.unity3d.com/Manual/class-NavMeshAgent.html

53 Biswas P, Saluja K S, Arjun S, Murthy L, Prabhakar G, Sharma V K, Dv J S. COVID-19 data visualization through automatic phase detection. Digital Government: Research and Practice, 2020, 1(4): 25
DOI:10.1145/3411756

54 Woldstad J C. Digital human models for ergonomics. 2000

55 Mixamo. Adobe. [2020-8-27] Available from: https://www.mixamo.com/#/?page=1&type=Motion%2CMotionPack

56 Li X, Wang K F, Tian Y L, Yan L, Deng F, Wang F Y. The ParallelEye dataset: a large collection of virtual images for traffic vision research. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(6): 2072–2084
DOI:10.1109/tits.2018.2857566

57 Martinez-Gonzalez P, Oprea S, Garcia-Garcia A, Jover-Alvarez A, Orts-Escolano S, Garcia-Rodriguez J. UnrealROX: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. Virtual Reality, 2020, 24(2): 271–288
DOI:10.1007/s10055-019-00399-5

58 Paisley J, Blei D, Jordan M. Variational Bayesian inference with stochastic search. arXiv preprint arXiv:1206.6430, 2012

59 Burda Y, Grosse R, Salakhutdinov R. Importance weighted autoencoders. arXiv preprint arXiv:1509.00519, 2015

60 Chen R T Q, Li X, Grosse R, Duvenaud D. Isolating sources of disentanglement in variational autoencoders. arXiv preprint arXiv:1802.04942, 2018

61 Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. Advances in neural information processing systems, 2014, 27, 2672–2680

62 Shaham T R, Dekel T, Michaeli T. SinGAN: learning a generative model from a single natural image. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South), IEEE, 2019, 4569–4579
DOI:10.1109/iccv.2019.00467

63 Xian W Q, Sangkloy P, Agrawal V, Raj A, Lu J W, Fang C, Yu F, Hays J. TextureGAN: controlling deep image synthesis with texture patches. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 8456–8465
DOI:10.1109/cvpr.2018.00882

64 Zhang K, Zuo W M, Chen Y J, Meng D Y, Zhang L. Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Transactions on Image Processing, 2017, 26(7): 3142–3155
DOI:10.1109/tip.2017.2662206

65 Zontak M, Irani M. Internal statistics of a single natural image. CVPR 2011, 2011, 977−984
DOI:10.1109/cvpr.2011.5995401

66 Zontak M, Mosseri I, Irani M. Separating signal from noise using patch recurrence across scales. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR, USA, IEEE, 2013, 1195−1202
DOI:10.1109/cvpr.2013.158

67 Denton E, Chintala S, Szlam A, Fergus R. Deep generative image models using a laplacian pyramid of adversarial networks. Advances in neural information processing systems, 2015, 1486−1494

68 Huang X, Li Y X, Poursaeed O, Hopcroft J, Belongie S. Stacked generative adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, IEEE, 2017, 1866−1875
DOI:10.1109/cvpr.2017.202

69 Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196, 2017

70 Barnett S A. Convergence problems with generative adversarial networks (gans). arXiv preprint arXiv:1806.11382, 2018

71 Park S W, Huh J H, Kim J C. BEGAN v3: avoiding mode collapse in GANs using variational inference. Electronics, 2020, 9(4): 688
DOI:10.3390/electronics9040688

72 Kingma D P, Welling M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013

73 Open Images Dataset V4. [2020-8-27] Available from: https://github.com/openimages/dataset

74 Bertoni L, Kreiss S, Alahi A. Perceiving humans: from monocular 3D localization to social distancing. IEEE Transactions on Intelligent Transportation Systems, 2021, (99): 1−18
DOI:10.1109/tits.2021.3069376