Received March 21, 2022, accepted April 4, 2022, date of publication April 8, 2022, date of current version April 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3165821

Online Safety Zone Estimation and Violation Detection for Nonstationary Objects in Workplaces

HYUNJOONG CHO[1], KYUIYONG LEE[1], NAKKWAN CHOI[1], SEOK KIM[2], JINHWI LEE[2], AND SEUNGJOON YANG[1]

[1]Department of Electrical Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan 44919, South Korea
[2]Process and Engineering Research Laboratory, AI Research Group, POSCO, Pohang 37763, South Korea

Corresponding author: Seungjoon Yang (syang@unist.ac.kr)

This work was supported by POSCO.

ABSTRACT This study presents a deep neural network (DNN)-based safety monitoring method. Nonstationary objects such as moving workers, heavy equipment, and pallets were detected, and their trajectories were tracked. Time-varying safety zones (SZs) of moving objects were estimated based on their trajectories, velocities, proceeding directions, and formations. SZ violations are defined by set operations with sets of points in the estimated SZs and the object trajectories. The proposed methods were tested using images acquired by CCTV cameras and virtual cameras in 3D simulations in plants and on loading docks. DNN-based detection and tracking provided accurate online estimation of time-varying SZs that were adequate for safety monitoring in the workplace. The set operation-based SZ violation definitions were flexible enough to monitor various violation scenarios that are currently monitored in workplaces. The proposed methods can be incorporated into existing site monitoring systems with single-view CCTV cameras at vantage points.

INDEX TERMS Safety monitoring, safety zone estimation, safety zone violation detection, nonstationary objects, deep learning, morphology.

I. INTRODUCTION

Safety monitoring is an important part of workplace safety that prevents accidents by issuing alarms at critical moments and enforcing safety rules and regulations. However, it is unrealistic for safety officers to monitor large dynamic sites with multiple operations performed by many workers. There have been various developments for assisting or automating site monitoring. Detecting workers and equipment in workplaces and tracking their trajectories for possible accidents are integral parts of automated and continuous safety monitoring. In [1]–[5], a real-time tracking system was used to track workers. An additional advantage of wearable sensors is that they can also be used to measure other data, such as workers' activities and health conditions. However, these require all workers, possibly from many different organizations, to be equipped with compatible and calibrated sensors. When there are many reflective metal structures in a workplace, accurate localization of sensors that utilize radio frequencies may be difficult. Sensitive issues, such as the disclosure of personal information, may also arise. CCTV cameras are commonly used to monitor safety in the workplace [6]–[8]. Computer vision techniques were applied to achieve an understanding and visualization of situations in images acquired by the cameras. Objects in images are detected using object models based on pixel, color, and other feature information [9], [10]. Trajectories of objects can be tracked through pixels, segmentation, contours, kernels, and graph-based tracking algorithms [11]–[14]. Physical locations in the 3D space of detected and tracked objects cannot be specified through a single CCTV camera [15]. In [16]–[19], two or multiple cameras were used to locate objects in 3D space. Alternatively, range data from distance sensors, such as LIDAR, can be used to determine locations of objects in 3D spaces [20].

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Sharif.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Deep learning-based object detection enables accurate detection of multiple objects in the workplace, such as workers and heavy equipment. In [21]–[25], personal protective equipment was detected using DNNs. In [26]–[28], heavy equipment in images was detected using DNNs. In [29] and [30], images from multiple cameras were monitored using DNNs to detect workers in near distances of heavy equipment in operation. In [31], support structures at construction sites were segmented using a deep network to detect workers standing on the structures. The safety rule violations in these processes require safety zones (SZs) around stationary objects, such as heavy equipment operating at fixed locations or concrete and steel structures in the sites.

Tracking the trajectories of moving objects has been studied in various areas [32], [33]. Once an object is detected, its locations over time can be tracked using deterministic [34], [35], probabilistic [36], [37], and deep network models [38]. Recently, understanding of the trajectories of multiple objects were studied in sports events [39], [40] or in clouded spaces [41]–[44] is being studied. These approaches try to model the general movement of objects, for example, players and humans, in a given situation. For safety monitoring purposes, we cannot expect heavy equipment and workers to follow general and safe trajectories. A simple tracking method based on a simple deterministic motion model without complicated assumptions is more suitable for our purpose.

This study presents online SZ estimation and detection of safety rule violations involving nonstationary equipment. In particular, we consider collisions of moving workers, moving heavy equipment in plants, and accidental falls of moving workers from elevated platforms arranged with moving pallets in loading docks. Images from a single CCTV camera at the vantage point, which is often already available in workplaces, were used to monitor the site. Objects, such as workers, helmets, forklifts, trucks, pallets, and staircases, were detected and segmented using DNNs. Object locations were mapped to the top plane, where trajectories of objects, as well as the distances between moving objects, were estimated. The SZs of moving workers and heavy equipment are defined by circular sectors adaptive to the time-varying trajectories and speeds. The SZs of elevated platform formations are defined via morphological operations [45]. Various safety rule violations are defined and detected using set operations involving SZ sets. The proposed zone estimation and zone violation detection algorithms, which involve the use of neural networks for detection and segmentation, were trained using images from CCTV cameras installed in a plant and a loading dock. The trained algorithms were implemented in the current site monitoring systems for field testing. We also prepared images acquired from virtual cameras in 3D worlds created using the 3D simulation software Unity 3D [46]. The proposed methods were trained and tested using Unity 3D images, which allowed us to validate the concepts and quantitatively evaluate the detection, segmentation, and tracking accuracy. With DNN-based object detection, segmentation and trajectory tracking and morphological operation-based online zone estimation, we were able to detect zone violations and issue alarms for collision and fall accidents.

The contributions of this work can be summarized as follows. i) The DNN-based object detection and segmentation methods provided accurate detection and segmentation of multiple objects in the workplace with small false negatives by a single camera so that they can be used in safety monitoring. ii) Time-varying SZs are estimated based on the trajectories, velocities, headings, and formations of objects so that safety violations involving moving objects in workplaces can be considered. iii) SZ violations are defined as set operations with SZs and trajectories in the top-view plane so that violation scenarios in various workplaces can be expressed easily and detected accurately. iv) The proposed methods can be readily incorporated into existing site monitoring systems with single-view CCTV cameras at vantage points.

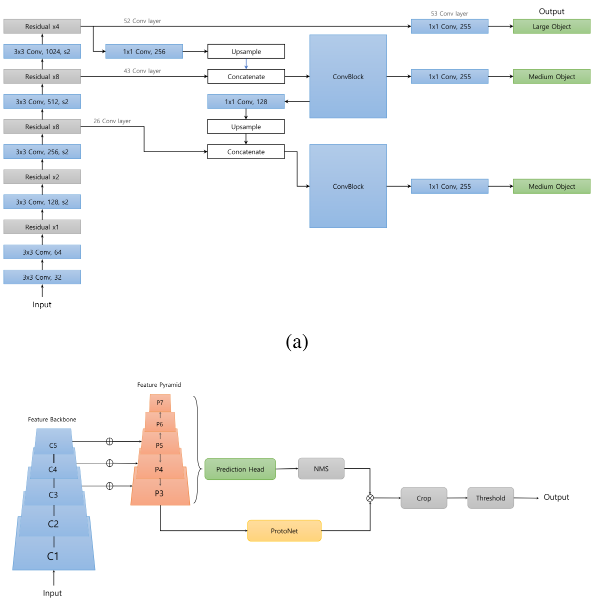
The remainder of this study is organized as follows. Section II-B presents the perspective transformation of acquired images to the top-view plane, where trajectories of objects were estimated. In Sections II-C and II-D, the time-varying SZs of moving heavy equipment and moving elevated platforms are defined. Sections III-A and III-B address SZ violation detection using set operations. Section IV provides the experimental results and discussion. Section V concludes the study.

II. ONLINE SAFETY ZONE ESTIMATION

A. OBJECT DETECTION AND SEGMENTATION

DNNs are utilized for object detection and segmentation. Fig. 1 shows the network architectures of YOLOv3 [47] and YOLACT [48] used for object detection and segmentation, respectively. YOLOv3 extracts features with deep convolutional layers. The features at three different resolutions are used to predict the types and locations of objects. YOLACT utilizes a fully convolutional layer (FCN) [49] to produce prototype masks and then refines the prototype masks based on the object detection results to find pixels that belong to each detected object.

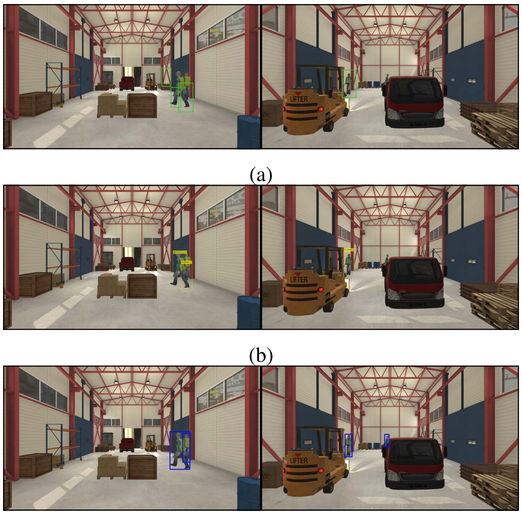
Both YOLOv3 and YOLACT learn features using deep convolutional layers from data. DNN-based object detection shows improved performance over detection methods based on domain-specific hand-selected features. For example, features such as histogram of oriented gradients (HOG) [50]–[52] and aggregated channel features (ACF) [53], [54] can be extracted using a sliding window on various scales, and then objects can be detected using a classifier such as a support vector machine (SVM) [55] based on the extracted features. Fig. 2 shows examples of DNN-based and specific feature-based object detection. Images from a frontal view camera in workplaces are used to detect workers. Both HOG-based and ACF-based detection showed difficulties in detecting workers partially occluded by other objects and multiple workers who reside in close proximity.



(a)

(b)

FIGURE 1. **Schematics of DNNs used for objection and segmentation: (a) YOLOv3; (b) YOLACT.**



(a)

(b)

(c)

FIGURE 2. **Comparisons of DNN-based and particular feature-based object detection: (a) HOG-based detection; (b) ACF-based detection; (c) DNN-based detection (YOLOv3).**

In comparison, DNN-based detection showed robust detection of the workers in these cases. In general, DNN-based detection showed better performance for corner cases where specific feature-based algorithms may suffer. Similar trends in performance comparisons were reported in other applications [56], [57].

The recall, which is defined as

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}, \tag{1}$$

was measured for the images used in the evaluation. The recall values of the HOG-based and ACF-based detection methods were 85.29 and 82.50, respectively, while that of the DNN-based method was 98.95. The DNN-based method reported significantly higher recall than the HOG and ACF-based methods. In safety monitoring, the false negative should be kept as small as possible to prevent missing an object or an occasion that may be involved in an accident. Hence, this work employs DNN-based methods for object detection and segmentation.

B. PERSPECTIVE TRANSFORM

A single camera is used to monitor a workplace. Objects in the scene are detected using a DNN. Locations of the detected objects are provided as pixel locations of bounding boxes that contain the objects. We transferred the object locations to a physical plane. In particular, we transferred the object locations to a plane from a top-view camera via a perspective transform [29], [30]. Let $(\tilde{x}, \tilde{y})$ and (x, y) be the pixel locations in an image acquired by the camera and in an image transformed to the top-view plane by the perspective transformer, respectively. Furthermore, the pixel locations are related using

$$[\tilde{x} \quad \tilde{y} \quad 1]\,\mathbf{M} = \gamma[x \quad y \quad 1], \tag{2}$$

where $\mathbf{M}$ is a 3×3 matrix and γ is a scalar quantity. We locate a square or a rectangular structure at the workplace ground and use the pixel locations of the quadrangle vertices to be mapped to the square or rectangular vertices to determine $\mathbf{M}$ and γ.

In various cases, a monitoring camera is installed at a high vantage point with the camera angled downward. Pointing the camera downward allows it to capture a wide angle of the workplace. One disadvantage of this installation is that the distances between objects in an acquired image are different depending upon the locations of the images. Transforming the object locations to the top-view plane via the perspective transformer makes the same distances appear the same, regardless of the object locations. Moreover, if the quadrangle vertices of a structure with a known dimension are used to find the perspective transformer, the physical dimension of a pixel can be found and used to set up safety distances or SZ dimensions.

C. SAFETY ZONE FOR MOVING HEAVY EQUIPMENT

SZs for moving heavy equipment, such as forklifts and trucks, were estimated. First, heavy equipment was detected using a DNN, which provided bounding boxes that contained the objects in the current frame. Furthermore, the centers of the bases of the bounding boxes were compared to those detected in the previous frame. Based on the Euclidean distance between the current and previous bounding boxes, the objects were assigned identification numbers. A newly appearing object was registered and assigned an i.d. number, and an object that disappeared was de-registered and de-assigned the i.d. number. Trajectories of objects were recorded in the top-view plane as

$$\{(x_t^e, y_t^e), (x_{t-1}^e, y_{t-1}^e), \cdots, (x_{t-K}^e, y_{t-K}^e), \} \tag{3}$$

where t is the current time index, K is the number of previous frames that we tracked, and $e = 1, 2, \cdots, E$ is the i.d. number of the moving heavy equipment.

The SZ of moving heavy equipment is estimated based on the trajectory. The velocity of the eth heavy equipment is estimated by

$$\mathbf{v}_t^e = (x_t^e - x_{t-K}^e, y_t^e - y_{t-K}^e) \tag{4}$$

The safety zone of the eth object is set up as a circular sector with the radius

$$r_t^e = \alpha_e \|\mathbf{v}_t^e\| \tag{5}$$

and the angle between $\angle \mathbf{v}_t^e - \theta_e$ and $\angle \mathbf{v}_t^e + \theta_e$. The estimated SZ is placed at the current location (x_t^e, y_t^e). The parameter α_e controls how far the SZ extends in front of an object, and the parameter θ_e controls how wide the safe zone spreads in front of an object. The safety zone of the eth equipment in the tth frame, denoted by z_t^e, is defined by the set of pixel indices (x, y)'s inside the circular sector of the eth heavy equipment. Note that the SZ of moving heavy equipment changes frame-by-frame depending on the locations and the velocity of the heavy equipment. For visual monitoring, the estimated SZ of the moving equipment was mapped back to the acquired images via inverse perspective transformation.

SZ estimation for moving heavy equipment can be extended to objects whose locations are provided by other methods. For example, the crane operating in a plant is a considerable factor in plant safety. Unlike heavy equipment such as forklifts and trucks, the location of a crane head can be provided by a crane control system. Let (x_t^e, y_t^e, z_t^e) be the location of the crane head. Then, the trajectory of the crane in the top-view plane is given by

$$\{(x_t^e, y_t^e), (x_{t-1}^e, y_{t-1}^e), \cdots, (x_{t-K}^e, y_{t-K}^e)\}. \tag{6}$$

The SZ of the crane is set as the union of circles

$$z_t^e = \bigcup_{\tau=t+1}^{t+T} C(x_\tau^e, x_\tau^e; r_t^e) \tag{7}$$

where $C(x_\tau^e, x_\tau^e; r_t^e)$ is a circle centered at (x_τ^e, x_τ^e) with radius r_t^e. The future locations (x_τ^e, y_τ^e) for $\tau = t+1, \cdots, t+T$ are predicted using the velocity of the crane head, $\mathbf{v}_t^e$, which is estimated by subtracting the Kth previous location from the current location. The radius r_t^e is set as a function of the crane height by

$$r_t^e = \beta_e + \alpha_e z_t^e \tag{8}$$

where β_e and α_e are the parameters that determine the size of the height-dependent SZ.

D. SAFETY ZONE ESTIMATION FOR ELEVATED PLATFORMS

Edges of elevated platforms are estimated as the SZ. First, elevated platforms, such as pallets and stairs, were detected using a DNN, which segmented pixels belonging to the detected objects. Furthermore, the binary mask that represented the segmented pixels of the elevated platform equipment was mapped to the top-view plane using the perspective transformer. We applied a series of morphological operations [45] to the binary mask to estimate the SZ. The closing operation was applied to fill small gaps between multiple pieces of equipment:

$$m_t^c = ((m_t \oplus s^c) \ominus s^c) \tag{9}$$

where m_t and m_t^c are the binary masks before and after closing, respectively, and $\oplus$ and $\ominus$ are the dilation and erosion operations, respectively. s^c is the structural element for the closing operation. The size of the gaps to close is controlled by the structural element s^c. The edge of the closed masks is determined by

$$m_t^e = m_t^c - (m_t^c \ominus s^e) \tag{10}$$

m_t^e is the binary mask that represents the edge, and s^e is the structural element. The width of the edge region is controlled by the structural element s^e. The binary mask of the edge served as the SZ of the elevated platform. The SZ for the elevated platform, denoted by z_t^p, is the set of pixel indices where m_t^e is one. Note that the SZ changes frame-by-frame and hence can handle the changes in SZ due to the formation of multiple pallets. For visual monitoring, the estimated SZ of the moving equipment was mapped back to the acquired image via an inverse perspective transformer.

SZ estimation for the elevated platforms can be extended to consider fixed areas. The SZ defined by a user from the image can be transformed to the top view to be considered to be the SZ.

III. SAFETY ZONE VIOLATION DETECTION

Violations of SZ can be expressed in terms of the relations between the locations, trajectories, velocities, heading, and formations of objects. We consider the following violation scenarios: i) future trajectories of workers and heavy equipment that collide, ii) workers staying in dangerous areas for a longer period, iii) future trajectories of workers entering dangerous areas, and iv) workers stepping backward to dangerous areas. Violation scenarios are written as set operations with SZs and trajectories defined in the top-view plane for violation detection.

A. SAFETY ZONE VIOLATION FOR MOVING HEAVY EQUIPMENT

Workers are detected using a DNN, which provides bounding boxes that contain the workers in the current frame. Following the same procedure presented in Section II-C, the trajectories of workers are recorded as

$$\{(x_t^w, y_t^w), (x_{t-1}^w, y_{t-1}^w), \cdots, (x_{t-K}^w, y_{t-K}^w), \} \tag{11}$$

for the K previous frames, where $w = 1, 2, \cdots, W$ is the i.d. number of workers. The velocity of the wth worker, $\mathbf{v}_t^w$, is estimated by subtracting the Kth previous location from the current location. The SZ of the wth worker is set as a circular sector with the radius

$$r_t^w = \alpha_w \|\mathbf{v}_t^w\| \tag{12}$$

and the angle between $\angle \mathbf{v}_t^w - \theta_w$ and $\angle \mathbf{v}_t^w + \theta_w$. The estimated SZ is placed at the current location (x_t^w, y_t^w). The parameters α_w and θ_w control how far and wide the SZ extends in front of a worker. The SZ of the wth worker in the tth frame, z_t^w, is defined by the pixel indices (x, y)'s inside the circular sector of the wth worker.

SZ violation occurs when the future trajectories of a worker and heavy equipment collide. In terms of the SZs of the heavy equipment and workers, z_t^e and z_t^w, respectively, an SZ violation occurs for the wth worker when the intersection of the two sets is not empty:

$$z_t^w \cap z_t^e \neq \emptyset \quad \text{for } e = 1, \cdots, E \tag{13}$$

where E is the number of heavy equipment detections in the frame. When an SZ violation is detected, the wth worker in the acquired images is highlighted for monitoring, and an alarm is issued.

B. SAFETY ZONE VIOLATION FOR ELEVATED PLATFORMS

The same DNN that provides segmentation of pixels for the platform equipment was used to detect workers. A bounding box that contains segmented pixels of a detected worker is found, whose center of the base is used as the location, trajectory, and velocity of the worker.

SZ violation occurs when a worker stays at the edge of the elevated platform for a long duration. In terms of the SZs of platform z_t^p and the locations of workers (x_t^w, y_t^w), an SZ violation occurs for the wth worker when the worker's future locations belong to the current set z_t^p:

$$(x_\tau^w, y_\tau^w) \in z_t^p \quad \text{for } \tau = t, t-1, \cdots, t-T \tag{14}$$

where T is a parameter that determines the duration of a worker staying at the edges of the platform.

SZ violation occurs when the future trajectory of a worker intersects the edge of the platform. In terms of the SZs of the platform and workers, z_t^p and z_t^w, respectively, an SZ violation occurs for the wth worker when the intersection of the two sets is not empty:

$$z_t^w \cap z_t^p \neq \emptyset \tag{15}$$

Another violation that we detected was when a worker stepped backward on the platform. To determine whether a worker is walking forward or backward, the gaze direction is estimated. Workers in the workplace are required to wear helmets, which are white on the front and blue on the back. The helmets are detected using the same DNN. The detected helmets are assigned to the worker i.d. number based on the Euclidean distance between the locations of helmets and workers. The image of the helmet is resized to a fixed dimension. Furthermore, binary maps for the white and blue parts of the helmet are found using thresholding in the HSV color space. The gaze direction of the wth worker, $\tilde{\mathbf{g}}_t^w$, is defined by subtracting the center of mass of the white mask from that of the blue mask. To compare the walking direction and the gaze direction, the trajectory of the worker is recorded with the locations in the acquired images as

$$\{(\tilde{x}_t^w, \tilde{y}_t^w), (\tilde{x}_{t-1}^w, \tilde{y}_{t-1}^w), \cdots, (\tilde{x}_{t-K}^w, \tilde{y}_{t-K}^w), \} \tag{16}$$

The velocity of the wth worker, $\tilde{\mathbf{v}}_t^w$, is computed. Backstepping is detected when the angle between the gaze and walking directions is greater than a threshold, or

$$\angle(\tilde{\mathbf{g}}_t^w, \tilde{\mathbf{v}}_t^w) > \phi \tag{17}$$

where ϕ is a threshold.

When an SZ violation is detected, the wth worker in the acquired images is highlighted for monitoring, and an alarm is issued.

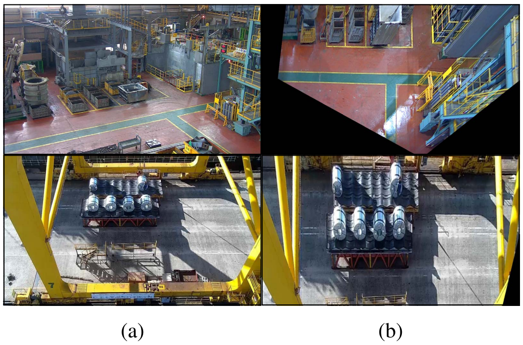


FIGURE 3. Examples of perspective transforms: (a) Images from CCTV cameras in vantage points; (b) Top-view plane images. Top: plant; Bottom: loading dock.

IV. EXPERIMENTS AND RESULTS

A. SAFETY ZONE ESTIMATION

1) PERSPECTIVE TRANSFORM

Fig. 3 shows examples of the perspective transformers. Images in (a) were acquired using CCTV cameras installed at vantage points in a plant and a loading dock. Because the cameras are pointed downward, objects showcase perspective with a vanishing point. In the images, objects in the front appear larger than those in the back. We cannot determine distances between objects by simply measuring the distance between pixels. Images in (b) show the results of mapping the images to the top-view plane via the perspective transforms. Quadrangle vertices of rectangular structures with a known dimension are used to find the perspective transform in (2). In particular, we measured the dimensions of four points in the pathways. Objects may appear stretched out in some directions in the top-view image. However, the footings of the objects bear correct locations in the ground.

We evaluated the accuracy of the perspective transformer through images created using the 3D simulation software

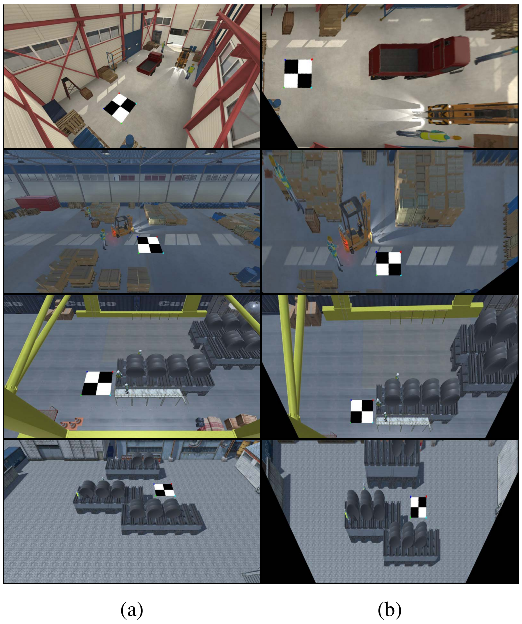


FIGURE 4. Examples of perspective transforms using Unity 3D simulation: (a) Images from virtual cameras at vantage points; (b) Top-view plane images.

Unity 3D [46]. 3D worlds that are similar to the plant and the loading dock in Fig. 3 were created. Virtual cameras were placed at vantage points, and images from the virtual camera were acquired. The perspective transformers are found using four points in the ground, with which the images are mapped to the top-view planes. Fig. 4 shows examples of the acquired and transformed images. For evaluation, we placed checkerboard patterns on the ground in the virtual worlds and measured the dimensions of the checkerboard patterns in the top-view images. A 2 m × 2 m square and a 4 m × 4 m square were used in the plant and the loading dock, respectively. The average angle between the adjacent sides of the squares mapped to the top-view plane was 89.34 degrees. The average aspect ratio of the square was 1.00:1.07. The physical dimension of a pixel can be calculated from the known dimensions of the checkerboard pattern. A pixel in the top-view images corresponds to 0.015 × 0.014, 0.016 × 0.017, 0.018 × 0.017, and 0.026 × 0.020 m in Fig. 4.

2) ONLINE SAFETY ZONE ESTIMATION FOR MOVING EQUIPMENT

Objects in the acquired images are detected using a DNN. YOLOv3 [47] is used to detect workers, forklifts, and trucks. Images in a plant were acquired while workers performed various tasks over several days, and the objects in the images were labeled. A total of 1443 images with 4198 labeled objects were used for the training. Data augmentation with scaling by x0.5 and x1.5 and flipping in both directions of the images was implemented. The network was implemented with Keras and TensorFlow using two NVIDIA GTX 2080 Ti GPUs. Adam [58] was used as the optimizer. The learning rate was set to 1.0 × 10-3 with decay. Random batches were used with batch sizes of 8. The accuracy of object detection was evaluated with 350 images that included 925 labeled objects prepared separately for testing. For evaluation, we also prepared images in two 3D worlds similar to the plant. The movement of the workers, forklifts, and trucks was simulated, while a view from a virtual camera at vantage points was acquired. Objects in the images were labeled, and YOLOv3 was trained using the labeled images.

TABLE 1. Accuracy of object detection by YOLO for moving heavy equipment, average and STD of Fourfold cross-validation.

	mAP		recall	
Class	Unity 3D	CCTV	Unity 3D	CCTV
worker	99.49 (0.13)	95.24 (0.76)	99.60 (0.35)	96.13 (0.38)
forklift	99.97 (0.04)	98.04 (0.38)	99.78 (0.31)	95.28 (0.24)
truck	99.79 (0.25)	99.65 (0.44)	99.62 (0.58)	99.73 (0.27)
Total	99.75 (0.11)	97.64 (0.37)	99.67 (0.32)	97.04 (0.24)

TABLE 2. Difference between distance measures for the YOLO dataset, average of Fourfold cross-validation.

Dataset	KSD	Kuiper	ADD	WD	WAD
Unity 3D	0.011439	0.010592	0.000027	0.024100	0.000028
CCTV	0.069525	0.057025	0.000110	0.107639	0.000116

Table 1 shows the accuracy of object detection in terms of the mean average precision (mAP) [59] and recall. Fourfold cross-validation [60] is used for the evaluation. The training set is divided into four folds. The images in three of the folds are used for training, and the images in the remaining fold are used for evaluation. The process is repeated for each fold to acquire the average and the standard deviation of the detection rates. The recall is the ratio between the detected objects and all the objects. The recall is not 100%; hence, there are false negatives, i.e., some objects are not detected by the network. Since we are detecting objects in video sequences, there is no case where an object is missed for an entire appearance. A rare case of a frame with a missed object is handled by the tracking algorithm, where only objects that were not detected for consecutive frames were deregistered.

Several methods are used for evaluating the safety and robustness of deep learning-based systems in [61]–[64]. We also evaluated the applicability of safety-security monitoring based on significant difference measures of our systems by SafeML in [62]. Table 2 shows the difference between various distance measures for the dataset used in YOLO. Five methods were selected for evaluation: Kolmogorov-Smirnov Distance (KSD), Kuiper Distance, Anderson-Darling Distance (ADD), Wasserstein Distance (WD), and a combination of ADD and Wasserstein-Anderson-Darling Distance (WAD). Fourfold cross-validation [60] is used for the evaluation. Error values are sufficiently small such that the results are acceptable in both the Unity 3D and CCTV datasets for YOLO. In all cases, WAD estimated the least error.

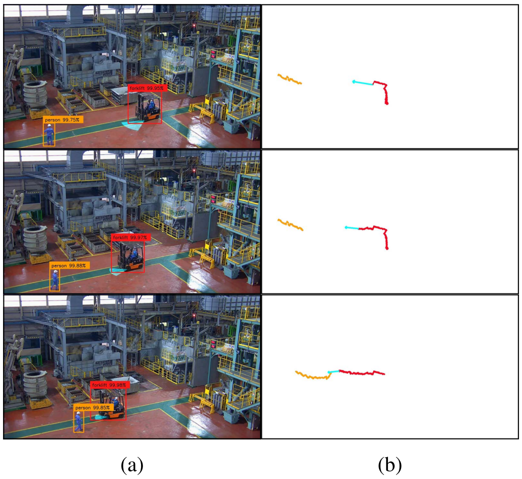


FIGURE 5. Examples of safety zone estimation for moving vehicles at various speeds: (a) Images; (b) Trajectories in the top view.

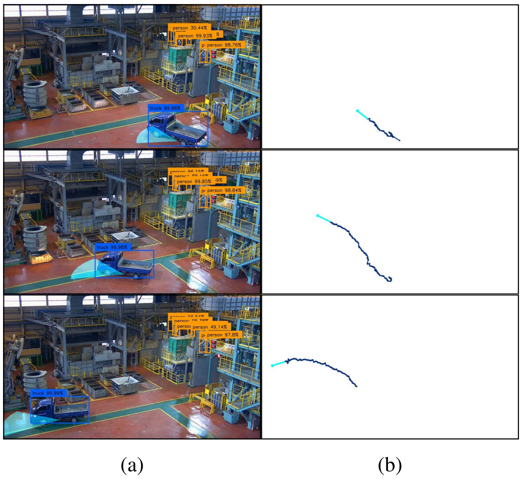


FIGURE 6. Examples of safety zone estimation for moving vehicle turning direction: (a) Images; (b) Trajectories in the top view.

The detection, tracking, and trajectory estimation allowed us to obtain the SZ of moving heavy equipment, which is adaptive to the trajectories and speeds. Fig. 5 shows examples of online SZ estimation for moving heavy equipment. The SZ of heavy equipment is a circular section. The radius of the circular section depends on the speed of the heavy equipment. The forklift in the images (a) decelerated spotting a worker in front of it. It can be observed that the SZ shrinks as the forklift decelerates. The trajectories and velocities of the forklifts are shown as red and cyan lines, respectively, in (b). Fig. 6 shows another example, where a truck was turning left. It can be observed that as the truck in the images (b) was turning left, the circular section was directed in the turning direction. The trajectories and velocities of the truck are shown as blue and cyan lines, respectively, in (b). The spreads of the circular sections are controlled by the parameters α_w, α_e, θ_w, and θ_e, which were determined through experiments.

TABLE 3. Performance of object detection by YOLACT for an elevated platform, average and STD of Fourfold cross-validation.

	mAP		recall	
Class	Unity 3D	CCTV	Unity 3D	CCTV
person	99.88 (0.08)	99.18 (0.86)	99.57 (0.28)	99.26 (0.29)
helmet	99.79 (0.16)	96.85 (0.40)	99.22 (0.78)	99.31 (0.53)
pallet	99.66 (0.42)	99.75 (0.14)	99.74 (0.19)	99.63 (0.28)
stairs	99.79 (0.23)	99.18 (0.51)	99.80 (0.35)	99.74 (0.13)
Total	99.78 (0.23)	98.74 (0.37)	99.58 (0.36)	99.48 (0.27)

TABLE 4. Difference between distance measures for the YOLACT dataset, average of Fourfold cross-validation.

Dataset	KSD	Kuiper	ADD	WD	WAD
Unity 3D	0.013108	0.016658	0.000325	0.014118	0.000022
CCTV	0.016483	0.025233	0.000448	0.034743	0.000054

3) ONLINE SAFETY ZONE ESTIMATION FOR ELEVATED PLATFORMS

YOLACT [48] was used to segment pixels for workers, pallets, and stairs. Images in the loading dock were acquired while workers performed loading and unloading in various pallet formations over several days. Pixels belonging to the objects were labeled. A total of 2007 images, including 25433 labeled objects, were used for the training. Data augmentation was implemented as well. The network was implemented with Keras and PyTorch using an NVIDIA GTX 2080 Ti GPU. Adam was used as the optimizer. The learning rate was set to 1.0 × 10-3 with decay. Random batches were used with batch sizes of 5. The accuracy of object detection was evaluated using 863 images with 7691 labeled objects prepared separately for testing. For evaluation, we also prepared images in two 3D worlds similar to the loading dock. The movement of workers and pallets was simulated, while a view from a virtual camera at vantage points was acquired. YOLACT was trained with the labeled images. YOLACT provides both bounding boxes and segmentation of pixels of the detected objects. Table 3 shows the accuracy of object detection in terms of the mean average precision (mAP) and the recall based on the bounding boxes of the fourfold cross validation.

Table 4 shows the difference between various distance measures for the dataset used in YOLACT. Five methods used in YOLO evaluation were also selected for YOLACT. Fourfold cross-validation [60] is used for the evaluation. Similar to the YOLO case, the error values are sufficiently small such that the results are acceptable in both the Unity 3D and CCTV datasets for YOLACT. In all cases, WAD also estimated the least error.

We used the segmentation results to estimate the SZ, which was determined to be the edge of the segmented pallets. Hence, how accurately YOLACT segments the pixels is important for accurate estimation of the SZ. The pixelwise segmentation accuracy was evaluated with the intersection over union (IoU) [65], which is given by

$$\text{segmentation accuracy} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}}, \tag{18}$$

TABLE 5. Segmentation accuracy by YOLACT for an elevated platform, average and STD of Fourfold cross-validation.

Classes	Unity Dataset	CCTV Dataset
pallet	98.84 (0.20)	98.58 (0.27)
stairs	96.56 (0.20)	94.96 (0.33)
Total	97.70 (0.19)	96.77 (0.19)

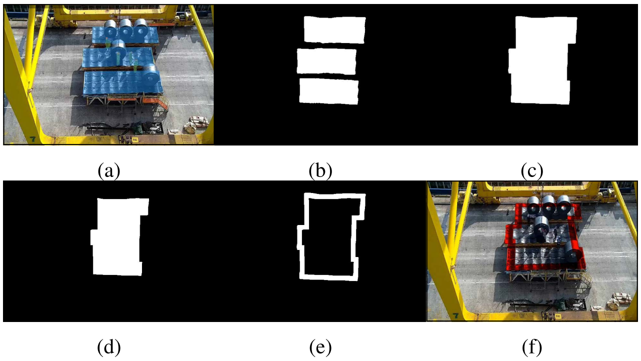

(a) (b) (c)
(d) (e) (f)

FIGURE 7. Examples of safety zone estimation for loading zone pallets: (a) Object segmentation in the image; (b) Binary mask in the top-view plane; (c) Map after closing; (d) Map after erosion; (e) Edge regions; (f) Safety zone overlaid on the image.

where TP, FN, and FP are the number of pixels in the true positive, false negative, and false positive segmentation, respectively. Table 5 shows the segmentation accuracy. It can be observed that the estimated segmentation accurately overlaps the ground truth segmentation.

Morphological operations were applied to obtain SZ for the elevated platform. For morphological dilation and erosion, a 5×5 size rectangle was used as the structural element. Fig. 7 illustrates the procedure of obtaining the SZ. The segmentation of the pallets in (a) was mapped to the top-view plane to form a binary mask in (b). The iterations of dilation followed by the iterations of erosion close the narrow gaps between the pallets, as given in (c). The number of iterations determined the sizes of the gaps to be closed. The map in (c) was shrunk by the iterations of erosion, the result of which is shown in (d). The shrunk map was subtracted from the closed map in (c) to determine the edges around the combined pallets. The width of the edges was determined by the iterations of the erosion. The SZ for the elevated platform formed by the three pallets is shown in (e). It was mapped to the acquired image using the inverse perspective transformer. The SZ overlay on the image is shown in red in (f).

Fig. 8 shows examples of SZ estimation for the elevated platform. In (a) and (b), the pallet in the front moved in and formed the workspace shown in (c). Following unloading, the middle pallet moved out in (d). The SZ is estimated for each frame to accommodate the changing formation of the elevated platform.

B. SAFETY ZONE VIOLATION

1) SAFETY ZONE VIOLATION FOR MOVING HEAVY EQUIPMENT

Workers and heavy equipment were detected by YOLO. The center of the base of the bounding box of each worker and heavy equipment was tracked in time to find the trajectory. The ground truth trajectory was difficult to obtain for the CCTV images. We use Unity 3D images, in which we were able to obtain the exact locations of objects that we placed in the 3D world, to evaluate the accuracy of the estimated trajectories.

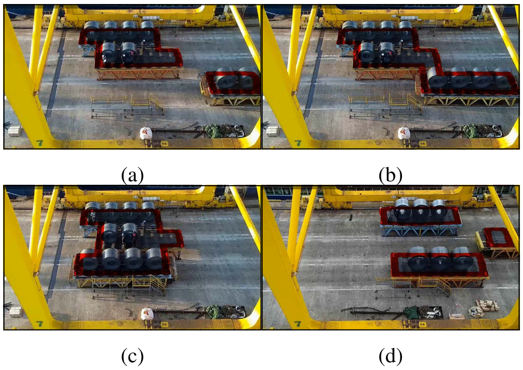

(a) (b)
(c) (d)

FIGURE 8. Examples of safety zone estimation for an elevated platform: (a), (b) Front pallet moving in; (c) Pallet set up for unloading; (d) Middle pallet moving out.

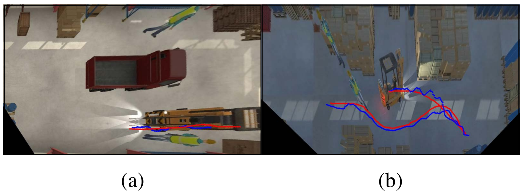

(a) (b)

FIGURE 9. Examples of trajectory estimation with Unity 3D images. Blue: estimated trajectory, Red: ground truth.

TABLE 6. Average error of trajectory estimation of Fourfold cross-validation [pixels].

video	forklift	truck	worker 1	worker 2	worker 3	Ave	Ave [m]
#1	20.92		1.59	0.95		7.82	0.22
#2	13.10		3.23	1.32	3.41	5.27	0.21
#3	7.55		2.68	1.32	2.68	3.57	0.15
#4	5.77	6.29	8.38	1.16		5.40	0.22
#5	9.29	7.86	17.56	3.10		9.45	0.38
Ave	15.70	7.07	6.69	1.57	3.04	6.82	0.23
STD	(5.53)	(0.85)	(6.07)	(0.81)	(0.42)	(2.74)	(0.08)

Fig. 9 shows examples of trajectory estimation using Unity 3D images. The trajectories in the top-view plane are shown, where the estimated trajectories and the ground truth are marked with blue and red lines, respectively. Table 6 shows the average error between the estimated and true trajectories of the fourfold cross-validation. Five video sequences from two virtual cameras at two plants were used. The average error was 6.82 pixels. By using the physical dimension of pixels obtained through the checkerboard patterns in Fig. 4, the average error of the trajectory estimation converted to meters was 0.24 meters.

SZ violation occurs when the SZs of a worker and heavy equipment collide. Fig. 10 shows examples of SZ violations for moving heavy equipment using CCTV images. Images

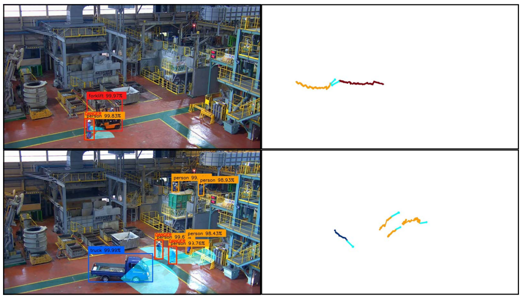


FIGURE 10. Examples of safety zone violations for moving vehicles using CCTV images: (a) Images, (b) Trajectories in the top-view plane.

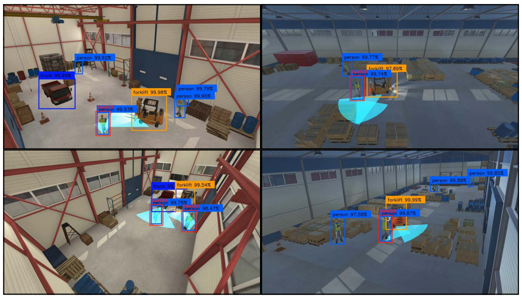


FIGURE 11. Examples of safety zone violations for moving vehicles using Unity 3D images.

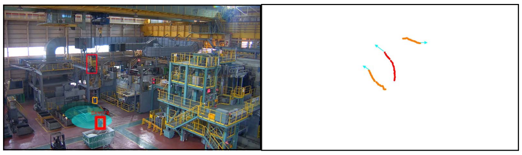


FIGURE 12. Examples of safety zone violations for moving cranes using CCTV images: (a) Images, (b) Trajectories in the top-view plane.

with overlaid SZs are shown in (a), and the trajectories of objects in the top-view plane are shown in (b). We also showcased examples of SZ violations using Unity 3D images in Fig. 11.

SZ violation for moving equipment is extended to cases where the locations of moving equipment are provided from outside. Fig. 12 and 13 show examples of SZ violations for moving cranes using CCTV and Unity 3D images, respectively. The locations of the crane head are provided from crane control systems. The SZ is the union of the circles under the current and future crane locations with the radius proportional to the height of the crane. SZ violation occurs when a worker's SZ overlaps with the crane's SZ. The trajectories and velocities of the crane head are shown in red and cyan lines, respectively, in (b).

2) SAFETY ZONE VIOLATION FOR ELEVATED PLATFORMS

YOLACT returns both the bounding boxes and segmentation of detected objects. The base of the bounding boxes is tracked in the top-view plane to obtain the trajectories. Fig. 14 shows examples of trajectory estimation using the Unity 3D images. Blue and red lines indicate the estimated and ground truth trajectories, respectively. Table 7 shows the average error of trajectory estimation of the fourfold cross-validation, which was 13.00 pixels, or equivalently 0.16 meters.

To evaluate the accuracy of a worker's gaze direction estimation, we prepared test videos of a worker wearing a helmet with different colors on the front and back. The front half of the helmet is colored white, while the back half is colored blue. Fig. 15 illustrates the procedure of the gaze direction estimation. From the helmet image inside the bounding box in (a), binary maps for the white and blue parts of the helmet are found by the thresholding in the HSV color space, which are (b) and (c), respectively. Then, gaze direction is estimated by subtracting the center of mass of the white mask from that of the blue mask. The estimated gaze direction is overlaid in (d).

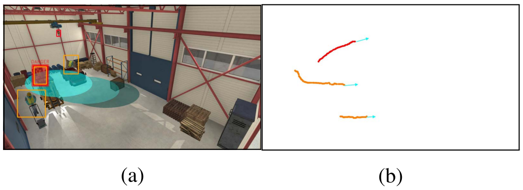


FIGURE 13. Examples of safety zone violations for moving cranes using Unity 3D images: (a) Images, (b) Trajectories in the top-view plane.

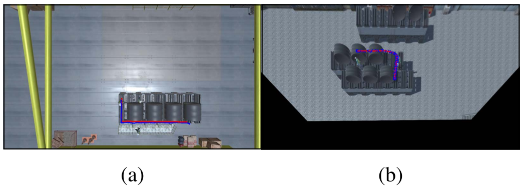


FIGURE 14. Examples of trajectory estimation using Unity images. Blue: estimated trajectory, Red: ground truth.

TABLE 7. Average error of trajectory estimation of Fourfold Cross-validation [pixels].

Video	worker 1	worker 2	worker 3	Ave	Ave [m]
#1	1.02	22.10	21.13	14.75	0.18
#2	12.14	10.49	20.11	14.25	0.17
#3	13.19	13.45	1.08	9.24	0.12
#4	2.34	24.47	14.45	13.75	0.17
Ave	7.17	17.62	14.19	13.00	0.16
STD	(6.38)	(6.72)	(9.22)	(2.54)	(0.03)

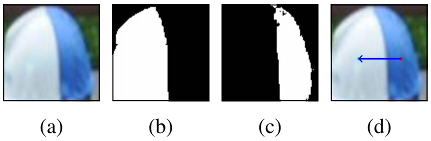
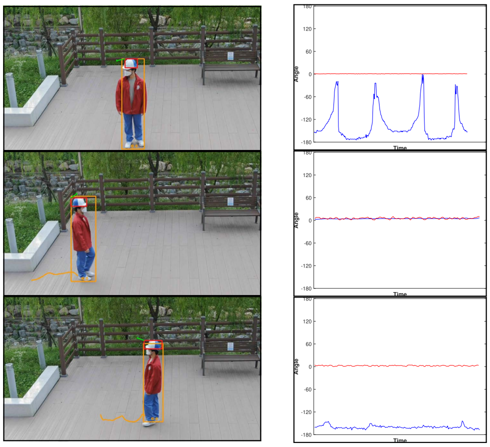
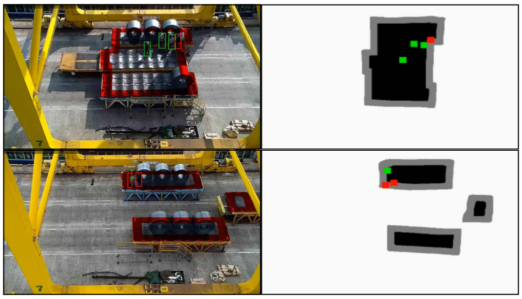


FIGURE 15. Gaze direction estimation via morphological operations: (a) Helmet, (b) Thresholding for the white part of the helmet, (c) Thresholding for the blue part of the helmet, (d) Center of mass and gaze direction.



(a) (b)

FIGURE 16. Examples of backstepping detection: (a) Image, (b) Moving direction in red, and gaze direction in blue. Top: looking around, Middle: stepping forward, Bottom: stepping backward.

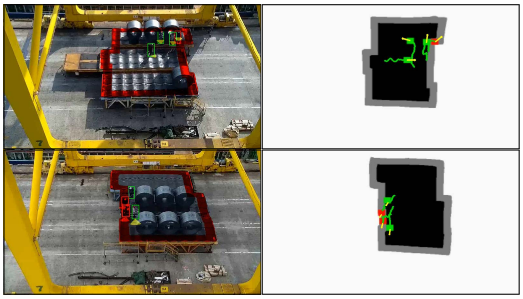


(a) (b)

FIGURE 17. Examples of SZ violations for elevated platforms, workers staying at the edge of the platform for a long duration; (a) CCTV images, (b) Violation situation in the top view.

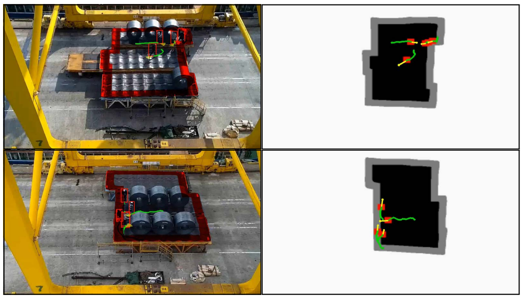
Examples of videos prepared to evaluate the detection of backstepping workers are shown in Fig. 16. Three scenarios were shown: workers standing while looking around, moving forward, and moving backward. The worker's movement and gaze directions are plotted in red and blue, respectively. The direction was between −180 degrees and 180 degrees, where the right horizontal direction was at zero degrees. While looking around in (a), the gaze direction changes from −180 to 0 degrees, while the worker looks left and right facing front. In the case of moving workers, the differences between the moving and gazing directions are small while stepping forward and large while stepping backward, as shown in (b) and (c).

We had three SZ violations for the elevated platform: a worker staying at the edge of the platform for a long duration, a worker walking toward the edge of the platform, and a worker backstepping on the platform. Fig. 17 shows examples of workers staying at the edge for a long duration. The images and the corresponding violation situation in the top-view plane are shown in (a) and (b), respectively. The workers with and without SZ violations are shown in red and green, respectively. The duration of the stay was set to 3 seconds.



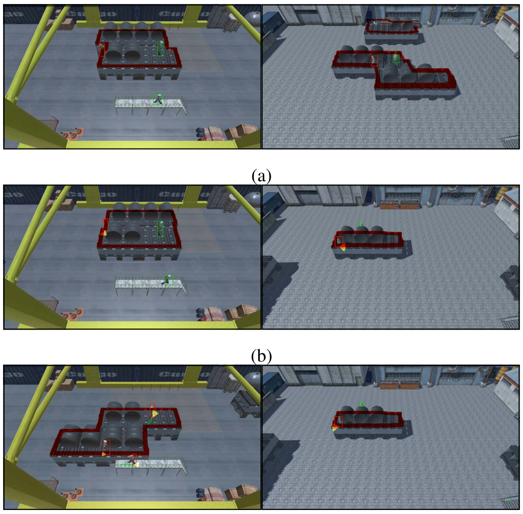
(a) (b)

FIGURE 18. Examples of SZ violations for elevated platforms, workers entering the edge of the platform; (a) CCTV images, (b) Violation situation in the top view.



(a) (b)

FIGURE 19. Examples of zone violations for elevated platforms, workers backstepping on the platform; (a) CCTV images, (b) Violation situation in the top view.



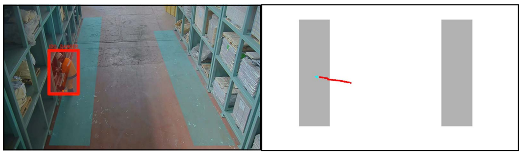
(a)

(b)

(c)

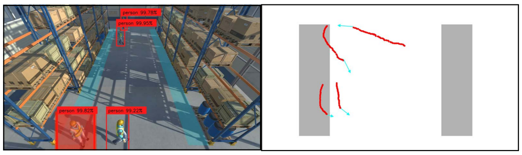
FIGURE 20. Examples of zone violations for elevated platforms using Unity 3D images; (a) Workers staying for a long duration, (b) Workers entering the edge of the platform, (c) Workers backstepping on the platform.

Fig. 18 shows examples of the SZ violation where workers are walking onto the edge of the platform. The images and the corresponding violation situation, along with the trajectories of workers in the top-view plane, are shown in (a) and (b), respectively. Workers reaching the edges of the platform were



(a) (b)

FIGURE 21. Examples of safety zone violations for fixed areas using CCTV images: (a) Images, (b) Trajectories in the top-view plane.



(a) (b)

FIGURE 22. Examples of safety zone violations for a fixed area using Unity 3D images: (a) Images, (b) Trajectories in the top-view plane.

detected and marked in red. Fig. 19 shows examples of the zone violation where workers are backstepping on the platform. Accidents have been reported in which workers have stepped backward and fallen down. However, this is a rare incident and is also dangerous to enact for evaluation. Hence, we played the recorded videos backward to envision all walkers stepping backward. The images and the corresponding violation situations along with the trajectories of workers in the top-view plane are shown in (a) and (b), respectively. All workers were detected as backstepping and marked in red.

Fig. 20 shows examples of the three cases of SZ violation using Unity 3D images. For the backstepping cases, the videos are also played backward for evaluation. All the SZ violations were appropriately detected in the simulation using Unity 3D images.

SZ violations for elevated platforms can be extended to consider fixed areas. Fig. 21 and 22 show examples of SZ violations for fixed areas in front of the part storage area. The selected fixed areas are transformed to the top-view plane, which are shown in gray in (b). SZ violation occurs when the workers stay in the SZ for longer than a predefined period.

V. CONCLUSION

The DNN-based object detection and segmentation methods provided accurate and efficient detection and segmentation of multiple objects that were adequate for safety monitoring in workplaces. Time-varying SZs involving moving objects and set operation-based SZ violation definitions allowed us to monitor various SZ violation scenarios that are currently monitored by safety monitoring teams in workplaces. Safety and robustness measures for object detection are also provided by the evaluation method. The proposed methods are currently incorporated into existing site monitoring systems, from which feedback is being collected. The proposed methods can be easily extended to various workplaces with their own safety monitoring requirements. However, object detection using a single CCTV camera has a limitation in detecting obscured objects. In future work, we will improve detection and localization performance by incorporating multiple views of a workplace.

REFERENCES

[1] T. Cheng, M. Venugopal, J. Teizer, and P. A. Vela, "Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments," *Autom. Construct.*, vol. 20, no. 8, pp. 1173–1184, Dec. 2011.

[2] S. Chae and T. Yoshida, "Application of RFID technology to prevention of collision accident with heavy equipment," *Autom. Construct.*, vol. 19, no. 3, pp. 368–374, May 2010.

[3] P. Li, R. Meziane, M. J.-D. Otis, H. Ezzaidi, and P. Cardou, "A smart safety helmet using IMU and EEG sensors for worker fatigue detection," in *Proc. IEEE Int. Symp. Robotic Sensors Environments (ROSE)*, Oct. 2014, pp. 55–60.

[4] H. Li, G. Chan, J. K. W. Wong, and M. Skitmore, "Real-time locating systems applications in construction," *Autom. Construct.*, vol. 63, pp. 37–47, Mar. 2016.

[5] K. Yang, C. R. Ahn, and H. Kim, "Deep learning-based classification of work-related physical load levels in construction," *Adv. Eng. Informat.*, vol. 45, Aug. 2020, Art. no. 101104.

[6] H. Guo, Y. Yu, and M. Skitmore, "Visualization technology-based construction safety management: A review," *Autom. Construct.*, vol. 73, pp. 135–144, Jan. 2017.

[7] W. Fang, L. Ding, H. Luo, and P. E. D. Love, "Falls from heights: A computer vision-based approach for safety harness detection," *Autom. Construct.*, vol. 91, pp. 53–61, Jul. 2018.

[8] M.-W. Park, A. Makhmalbaf, and I. Brilakis, "Comparative study of vision tracking methods for tracking of construction site resources," *Autom. Construct.*, vol. 20, no. 7, pp. 905–915, Nov. 2011.

[9] S. Du, M. Shehata, and W. Badawy, "Hard hat detection in video sequences based on face features, motion and color information," in *Proc. 3rd Int. Conf. Comput. Res. Develop.*, vol. 4, Mar. 2011, pp. 25–29.

[10] M. Memarzadeh, M. Golparvar-Fard, and J. C. Niebles, "Automated 2D detection of construction equipment and workers from site video streams using histograms of oriented gradients and colors," *Autom. Construct.*, vol. 32, pp. 24–37, Jul. 2013.

[11] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[12] P. A. Vela, M. Niethammer, G. D. Pryor, A. R. Tannenbaum, R. Butts, and D. Washburn, "Knowledge-based segmentation for tracking through deep turbulence," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 3, pp. 469–474, May 2008.

[13] D. Freedman and T. Zhang, "Active contours for tracking distributions," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 518–526, Apr. 2004.

[14] J. Malcolm, Y. Rathi, and A. Tannenbaum, "Multi-object tracking through clutter using graph cuts," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–5.

[15] H. S. Coxeter, *Projective Geometry*. Springer, 2003.

[16] S. Chi and C. H. Caldas, "Image-based safety assessment: Automated spatial safety risk identification of earthmoving and surface mining activities," *J. Construct. Eng. Manage.*, vol. 138, no. 3, pp. 341–351, Mar. 2012.

[17] M.-W. Park, C. Koch, and I. Brilakis, "Three-dimensional tracking of construction resources using an on-site camera system," *J. Comput. Civil Eng.*, vol. 26, no. 4, pp. 541–549, Jul. 2012.

[18] I. Brilakis, M.-W. Park, and G. Jog, "Automated vision tracking of project related entities," *Adv. Eng. Informat.*, vol. 25, no. 4, pp. 713–724, Oct. 2011.

[19] Z. Zhu, M.-W. Park, C. Koch, M. Soltani, A. Hammad, and K. Davari, "Predicting movements of onsite workers and mobile equipment for enhancing construction site safety," *Autom. Construct.*, vol. 68, pp. 95–101, Aug. 2016.

[20] J. Teizer, C. H. Caldas, and C. T. Haas, "Real-time three-dimensional occupancy grid modeling for the detection and tracking of construction resources," *J. Construct. Eng. Manage.*, vol. 133, no. 11, pp. 880–888, Nov. 2007.

[21] M. Z. Shanti, C.-S. Cho, Y.-J. Byon, C. Y. Yeun, T.-Y. Kim, S.-K. Kim, and A. Altunaiji, "A novel implementation of an AI-based smart construction safety inspection protocol in the UAE," *IEEE Access*, vol. 9, pp. 166603–166616, 2021.

[22] G. Peng, Y. Lei, H. Li, D. Wu, J. Wang, and F. Liu, "CORY-net: Contrastive res-YOLOv5 network for intelligent safety monitoring on power grid construction sites," *IEEE Access*, vol. 9, pp. 160461–160470, 2021.

[23] K. Han and X. Zeng, "Deep learning-based workers safety helmet wearing detection on construction sites using multi-scale features," *IEEE Access*, vol. 10, pp. 718–729, 2022.

[24] N. D. Nath, A. H. Behzadan, and S. G. Paal, "Deep learning for site safety: Real-time detection of personal protective equipment," *Autom. Construct.*, vol. 112, Apr. 2020, Art. no. 103085.

[25] A. M. Kamoona, A. K. Gostar, R. Tennakoon, A. Bab-Hadiashar, D. Accadia, J. Thorpe, and R. Hoseinnezhad, "Random finite set-based anomaly detection for safety monitoring in construction sites," *IEEE Access*, vol. 7, pp. 105710–105720, 2019.

[26] L. Ding, W. Fang, H. Luo, L. Peter, B. Zhong, and X. Ouyang, "A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory," *Autom. Construct.*, vol. 86, pp. 118–124, Feb. 2018.

[27] D. Kim, M. Liu, S. Lee, and V. R. Kamat, "Remote proximity monitoring between mobile construction resources using camera-mounted UAVs," *Autom. Construct.*, vol. 99, pp. 168–182, Mar. 2019.

[28] S. Arabi, A. Haghighat, and A. Sharma, "A deep-learning-based computer vision solution for construction vehicle detection," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 7, pp. 753–767, 2020.

[29] H. Son, H. Seong, H. Choi, and C. Kim, "Real-time vision-based warning system for prevention of collisions between workers and heavy equipment," *J. Comput. Civil Eng.*, vol. 33, no. 5, Sep. 2019, Art. no. 04019029.

[30] H. Luo, J. Liu, W. Fang, P. E. D. Love, Q. Yu, and Z. Lu, "Real-time smart video surveillance to manage safety: A case study of a transport mega-project," *Adv. Eng. Informat.*, vol. 45, Aug. 2020, Art. no. 101100.

[31] W. Fang, B. Zhong, N. Zhao, P. E. D. Love, H. Luo, J. Xue, and S. Xu, "A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network," *Adv. Eng. Informat.*, vol. 39, pp. 170–177, Jan. 2019.

[32] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[33] H. D. Najeeb and R. F. Ghani, "A survey on object detection and tracking in soccer videos," *Muthanna J. Pure Sci.*, vol. 8, no. 1, pp. 1–13, Jan. 2021.

[34] C. J. Veenman, M. J. T. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 1, pp. 54–72, Jan. 2001.

[35] K. Shafique and M. Shah, "A noniterative greedy algorithm for multiframe point correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 51–65, Jan. 2005.

[36] T. J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 1, pp. 90–99, Jan. 1986.

[37] G. Kitagawa, "Non-Gaussian state-space modeling of nonstationary time series," *J. Amer. Stat. Assoc.*, vol. 82, no. 400, pp. 1032–1041, Dec. 1987.

[38] S. Chen, Y. Xu, X. Zhou, and F. Li, "Deep learning for multiple object tracking: A survey," *IET Comput. Vis.*, vol. 13, no. 4, pp. 355–368, Jan. 2019.

[39] P. L. Mazzeo, P. Spagnolo, M. Leo, and T. D'Orazio, "Visual players detection and tracking in soccer matches," in *Proc. IEEE 5th Int. Conf. Adv. Video Signal Based Surveill.*, Sep. 2008, pp. 326–333.

[40] P. Garnier and T. Gregoir, "Evaluating soccer player: From live camera to deep reinforcement learning," 2021, *arXiv:2101.05388*.

[41] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun./Jul. 2004, pp. II–II.

[42] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.

[43] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.

[44] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 19, 2021, doi: 10.1109/TITS.2021.3069362.

[45] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 4, pp. 532–550, Jul. 1987.

[46] *Unity 3D*. [Online]. Available: http://unity3d.com

[47] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[48] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9157–9166.

[49] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[50] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.

[51] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "Architectural study of HOG feature extraction processor for real-time object detection," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 2012, pp. 197–202.

[52] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "HOGgles: Visualizing object detection features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1–8.

[53] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[54] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[55] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory (COLT)*, 1992, pp. 144–152.

[56] W. Lan, J. Dang, Y. Wang, and S. Wang, "Pedestrian detection based on Yolo network model," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2018, pp. 1547–1551.

[57] S. Ghosh and D. Das, "Comparative analysis and implementation of different human detection techniques," in *Proc. 5th Int. Conf. Image Inf. Process. (ICIIP)*, Nov. 2019, pp. 443–447.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[59] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland, Springer, Nov. 2016, pp. 198–213.

[60] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Stat. Soc., B (Methodol.)*, vol. 36, no. 2, pp. 111–133, Jan. 1974.

[61] J. Mena, O. Pujol, and J. Vitria, "Uncertainty-based rejection wrappers for black-box classifiers," *IEEE Access*, vol. 8, pp. 101721–101746, 2020.

[62] K. Aslansefat, I. Sorokos, D. Whiting, R. Tavakoli Kolagari, and Y. Papadopoulos, "SafeML: Safety monitoring of machine learning classifiers through statistical difference measures," in *Proc. Int. Symp. Model-Based Saf. Assessment*. Cham, Switzerland, Springer, Sep. 2020, pp. 197–211.

[63] S. Gerasimou, H. F. Eniser, A. Sen, and A. Cakan, "Importance-driven deep learning system testing," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng.*, Jun. 2020, pp. 702–713.

[64] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljic, and D. L. Dill, "The marabou framework for verification and analysis of deep neural networks," in *Proc. Int. Conf. Comput. Aided Verification*. Cham, Switzerland, Springer, Jul. 2019, pp. 443–452.

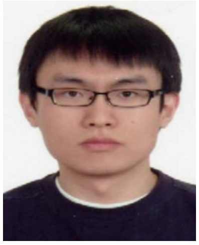
[65] T. T. Tanimoto, "An elementary mathematical theory of classification and prediction," Int. Bus. Mach. Corp., 1958. [Online]. Available: https://books.google.co.kr/books?id=yp34HAAACAAJ

HYUNJOONG CHO received the B.S., M.S., and Ph.D. degrees from the Ulsan National Institute of Science and Technology, Ulsan, South Korea, in 2015, 2017, and 2022, respectively. His research interests include image processing, human recognition, and deep learning.

KYUIYONG LEE was born in Seoul, South Korea, in 1995. She received the B.S. degree in electronic engineering from the Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea, in 2020. From 2020 to 2021, she studied computer vision and machine learning in the Signal Processing Laboratory, UNIST.

NAKKWAN CHOI received the B.S. degree in electronic engineering from Kyung Hee University, Seoul, South Korea, in 2021. He is currently pursuing the M.S. degree in electronic engineering with the Ulsan National Institute of Science and Technology, Ulsan, South Korea. His research interests include object detection and image processing using neural networks and the fundamental study of robotics.

SEOK KIM received the B.S. and M.S. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2008 and 2011, respectively. From 2011 to 2013, he was a Software Developer at the IT Development Team, WISOL, Osan-si, South Korea. Since 2013, he has been a Researcher with the Artificial Intelligence Research Group, POSCO, Pohang, South Korea. His research interests include virtual reality, digital twins, computer vision, AI, and smart manufacturing systems in the steelmaking industry.



JINHWI LEE received the B.S. and M.S. degrees in industrial engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2004 and 2007, respectively. From 2007 to 2009, he was a Researcher at the Institute of Industrial Management, KAIST. Since 2009, he has been a Researcher with the Artificial Intelligence Research Group, POSCO, Pohang, South Korea. His research interests include computer vision, continual learning, and smart manufacturing systems in the steelmaking industry.

SEUNGJOON YANG received the B.S. degree from Seoul National University, Seoul, South Korea, in 1990, and the M.S. and Ph.D. degrees from the University of Wisconsin–Madison, in 1993 and 2000, respectively, all in electrical engineering. He worked at the Digital Media Research and Development Center, Samsung Electronics Company Ltd., from September 2000 to August 2008, and is currently with the School of Electrical and Computer Engineering Ulsan National Institute of Science and Technology, Ulsan, South Korea. His research interests include image processing, estimation theory, and optimization theory.