KIEAE Journal, Vol. 25, No. 6, Dec. 2025, pp.5-14.

KIEAE Journal

Korea Institute of Ecological Architecture and Environment

136

Improving Object Classification Accuracy for Detecting Obstacles on Indoor Evacuation Paths Using YOLOv8

Hyo-Ju Shin* · Sang-Pil Jung** · Jin-Wook Kim***

* *Main author, Master's Course Student, Dept. of Architecture, Seoul National Univ., of Science and Technology, South Korea (shj918@seoultech.ac.kr)*
** *Coauthor, Associate Professor, Dept. of AI Convergence, Sehan Univ., South Korea (citta@sehan.ac.kr)*
*** *Corresponding author, Professor, Dept. of Architecture, Seoul National Univ. of Science & Technology, South Korea (Jinwook@seoultech.ac.kr)*

ABSTRACT

Purpose: This study proposes a foundational framework for estimating the effective evacuation path width in complex building corridors through real-time obstacle detection. The feasibility of a low-cost high-efficiency system was assessed by leveraging an existing closed-circuit television infrastructure to eliminate the need for additional hardware deployment. **Method:** A pretrained YOLOv8n model was used to detect objects on static frames from corridor surveillance footage. To address the challenges specific to indoor environments, a custom dataset was created via data augmentation and class relabeling using Roboflow. Performance was enhanced after training by applying class unification and parallel detection-based post-processing strategies. **Result:** After model customization and data augmentation, the system achieved improved detection metrics: precision of 0.9925, recall of 1.000, and mean average precision@0.5 of 0.9950. The results revealed the robustness of the model against occlusion and illumination variations and validate the technical feasibility of object detection as a core component of automated evacuation path recognition.

KEYWORD

Object Detection
Evacuation Route
Data Augmentation

ACCEPTANCE INFO

Received Oct. 10, 2025
Final revision received Oct. 22, 2025
Accepted Oct. 28, 2025

1. Introduction

1.1. Research Background and Purpose

In recent years, high-rise and complex buildings that have evolved to encompass a range of functions, including business, commercial, residential, and cultural use, are becoming increasingly prominent. The spatial structure of these complex buildings involves not only basic vertical and horizontal connections but also multiple movement flows and evacuation paths, with occupant distribution flexibly changing depending on time and purpose. This spatial and personnel complexity requires even more precise and timely information for securing and determining evacuation paths in the event of an emergency [1].

However, the current evacuation design system mostly relies on fixed structural information based on architectural drawings and preliminary simulations, thus failing to sufficiently reflect dynamic elements that can influence evacuation paths in real time [2]. In real environments, furniture, cleaning supplies, and temporary loads can be randomly placed and moved within corridors or hallways, which serve as evacuation paths, thereby causing the actual effective evacuation width to differ from the planned dimensions. These environmental changes can become an important factor that delays evacuation processes during emergencies, even if they are not perceived as such under normal conditions.

Recent research has proposed the feasibility of digital twin technology for monitoring the status of internal spaces in real time and providing evacuation information that reflects changing environmental factors [3]. Digital twin technology can reflect real-time changes in actual spaces, in addition to enabling high-precision spatial analysis when integrated into 3D scanning equipment, high-resolution sensors, and object location detection systems [4]. However, it requires expensive hardware and specialized software that present financial and technical challenges for its adoption in existing buildings [5,6].

Therefore, the aim of this study was to explore object detection and spatial analysis potential without additional infrastructure by utilizing closed-circuit television (CCTV) systems already installed in existing buildings. The majority of large-scale buildings have CCTVs installed across a wide range of areas, enabling the collection of real-time image data. Recently developed lightweight object detection algorithms are suitable for real-time image processing owing to their fast processing speed and low computational resource consumption.

This study investigates whether single-view images captured by existing CCTV systems are sufficient for detecting obstacles

pISSN 2288-968X, eISSN 2288-9698
http://dx.doi.org/10.12813/kieae.2025.25.6.005

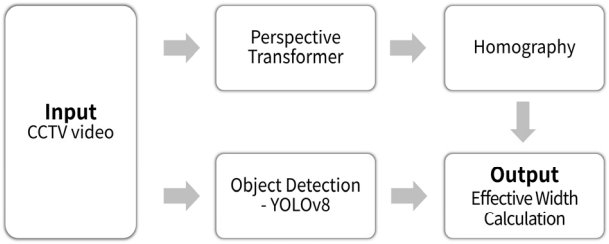


Fig. 1. Overall process for estimating usable corridor width

along indoor evacuation paths without additional sensing equipment. These results demonstrate the feasibility of providing indoor evacuation path information in real time and establishing efficient evacuation plans based on a low-cost automated object detection system.

1.2. Research Methods and Scope

The purpose of this study was to identify loads and obstacles in corridors using existing CCTV footage in a complex and diverse indoor evacuation path setting, to automatically calculate the effective evacuation path width based on the occupied area of identified objects and quantitatively evaluate the adequacy of the calculated evacuation path (Fig. 1.).

Accordingly, the actual locations of the objects, areas they occupy, and unit conversion are necessary in determining actual evacuation paths. A series of calculations first convert pixel-sized objects into physical distances to calculate the effective evacuation width. A homography-based perspective transform technique[1)] is then used to correct perspective distortion in converting to distances on the reference plane.

However, the scope of this study was limited to evaluating the accuracy and consistency of an object detection algorithm in indoor corridor settings with multiple obstacles, focusing specifically on the "object detection" step in the overall process.

Object detection in real evacuation situations occurs in a dynamic, real-time environment; however, this study used images captured in a static setting and performed post-processing to examine feasibility based on basic data. Detection performance was analyzed by applying a pre-trained YOLOv8 model alongside custom training, using image data obtained by controlling the imaging angle, lighting intensity, and the number of objects within the field of view. To account for undetected object classes not included in previously available datasets, a pilot-level custom object detection experiment was conducted using labeling and augmentation on a limited amount of experimental data.[2)]

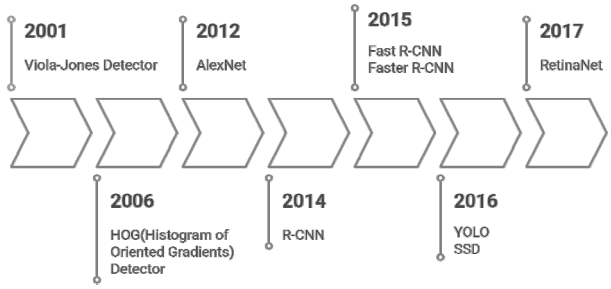


Fig. 2. Object detection research trends

2. Literature Review

2.1. Trends in Object Detection Research

The development of object detection technology before and after the introduction of deep learning in 2012 is depicted in Fig. 2. In the early years, traditional image-processing-based methods such as the Viola-Jones detector (2001) and histogram of oriented gradients (2006) were frequently used, but their performance was limited in adapting to complex backgrounds and varying object shapes. Furthermore, the sliding-window-based approach searched large areas of an image that were mostly background, resulting in lower computational efficiency, processing speed, and detection accuracy [7]. However, research on object detection based on convolutional neural networks (CNNs) accelerated after the AlexNet excelled in the ImageNet competition in 2012.

Deep-learning-based object detection techniques are divided based on two-and one-stage detectors depending on their structure. Two-stage object detection methods sequentially perform region proposal followed by classification, with representative models including the region-based convolutional neural network (R-CNN) (2014), Fast R-CNN (2015), and Faster R-CNN (2015). They offer high detection accuracy but can be unsuitable for real-time processing due to their computationally complex architecture. In contrast, one-stage methods have a simpler structure and faster inference speed by performing region proposal and classification simultaneously. YOLO (2016), single-shot detector (2016), and RetinaNet (2017) are commonly used one-stage models [8]. Specifically, YOLO is well suited for real-time monitoring as it processes an input image in a single batch using one CNN in an end-to-end manner, providing high inference speed and detection accuracy [9].

2.2. You Only Look Once (YOLO)

The YOLO algorithm is a representative one-stage deep-learning model commonly used for object detection. It predicts object classes and locations simultaneously by processing

the entire image at once. This algorithm divides an image into S×S grids and simultaneously outputs each cell's bounding box, confidence score, and class probability, thereby providing the advantage of high-speed and real-time object detection. YOLO outperforms other two-stage series object detectors in terms of computational efficiency and speed because the structure omits region proposal and simultaneously executes classification and regression in a single neural network.

Beginning with YOLOv1, which was introduced in 2016, YOLO has demonstrated improved accuracy through continuous structural enhancements. YOLOv1 performs real-time detection by processing the entire frame end-to-end using a single CNN; however, it still exhibits low detection performance for small objects and generates errors in complex backgrounds. YOLOv2 introduced anchor boxes and employed batch normalization, which enhanced model stability and accuracy. YOLOv3 further improved the detection of objects at multiple scales through predictions on multiscale feature maps. In YOLOv4, the balance between accuracy and speed was further improved by integrating advanced feature aggregation techniques, such as spatial pyramid pooling and the path aggregation network built on the cross-stage partial-Darknet-53 backbone. YOLOv5 is a version developed by Ultralytics based on PyTorch. It provides user friendliness and flexibility while offering lightweight models of various sizes (n, s, m, l, x). The YOLOv7 implementation is based on the extended efficient layer aggregation network architecture; it enhances training efficiency and auxiliary output head, thereby improving accuracy without increasing computational costs. YOLOv8 introduces an anchor-free detection head and an efficient computational structure, such as the C2f block in the backbone, achieving both a lightweight model and improved accuracy. In addition, inference speed is improved by simplifying the post-processing steps [10].

Table 1. presents a comparison of the detection performance and computational characteristics of YOLOv8 models in terms of five indicators. Mean average precision (mAP)@50-95 is the average precision calculated by changing the intersection over union (IoU) threshold between 0.50 and 0.95, thus indicating the overall accuracy of object location and class prediction. Central processing unit (CPU) open neural network exchange (ONNX) latency indicates the time required to infer a single frame on a CPU without graphics processing unit (GPU), whereas A100 TensorRT latency represents the inference speed in an NVIDIA A100 GPU environment optimized with TensorRT. The number of parameters indicates the total trainable weights of the model (unit: 1 million), with GFLOPs, representing the number of floating-point operations (unit: gigabyte), required for a single inference. These indicators enable estimating the model's memory requirements and computational complexity. YOLOv8n, which is a lightweight model, has the lowest computational requirements with 3.2 million parameters and 8.7 GFLOPS, and inference time of 80 ms with CPU ONNX and 1 ms with A100 TensorRT. Thus, object detection was performed with the ultra-lightweight YOLOv8n, considering its high training stability and balanced real-time obstacle detection without the need for additional hardware (Table 1.).[3)]

Table 1. Performance comparison of YOLOv8 variants on COCO

Model	YOLO v8n	YOLO v8s	YOLO v8m	YOLO v8l	YOLO v8x
Input size (px)	640	640	640	640	640
mAP @50–95 (%)	37.3	44.9	50.2	52.9	53.9
CPU ONNX Latency (ms)	80.4	128.4	234.7	375.2	479.1
A100 TensorRT Latency (ms)	0.99	1.20	1.83	2.39	3.53
Parameters (M)	3.2	11.2	25.9	43.7	68.2
GFlops	8.7	28.6	78.9	165.2	257.8

2.3. Data Augmentation

The real evacuation path environment, subject to constrained space, varied lighting, occlusion, and visual similarities between objects, causes difficulty for an object detection model to achieve its expected generalization performance. In particular, a limited amount of training data and training based on single-time-point images can reduce detection accuracy, necessitating even more stable performance in a CCTV-based environment that demands real-time processing.

Chen et al. (2025) conducted an experiment on detecting crowd density in a smart library using a YOLOv8n-based lightweight model, to study these limitations. They achieved detection stability in the indoor environment by augmenting data to reflect a wide range of perspectives, lighting conditions, and behaviors. Additionally, they used a learning structure that reflects a real-time streaming environment, which contributed to improved object classification accuracy and a reduced false positive rate [11].

A. Istiak et al. (2024) constructed a custom automated license plate recognition model utilizing YOLOv8 and Roboflow for a traffic environment of Bangladesh. They experimentally verified detection performance in real-world environments using images with low resolution, occlusions, or varying license plate formats, and proved that Roboflow-based augmentation effectively enhances the model's generalization performance [12]. They

demonstrated that augmentation strategies can significantly contribute to detection performance improvement in unstructured environments, which is also directly related to the augmented-learning approach in this study.

2.4. Trends in Evacuation Path Research

Studies on pathway obstacles have shown that obstacle placement in corridors plays a critical role in evacuation efficiency. C. Siyuan et al. (2019) conducted an experiment to compare three layouts—parallel, convex, and concave—and reported that convex and concave obstacles mitigated the bottleneck, thereby increasing the average pedestrian speed by 19% and reducing passage time by 17% [13]. F. Claudio et al. (2020) experimentally proved that forming a vestibule region directly in front of an exit reduces the congestion near the exit and improves overall evacuation flow in both competitive and non-competitive situations [14].

Sticco et al. (2021) performed numerical simulations to verify that adjusting the distance between obstacles and the exit, as well as the friction coefficient, can optimize evacuation efficiency by controlling the density within the vestibule [15]. Kim & Quaini (2019) used a dynamic model to numerically analyze the location and shape of obstacles, which can induce significant changes in pedestrian path selection and flow characteristics [16]. These previous studies imply that the placement and shape of obstacles must be strategically considered when designing evacuation paths.

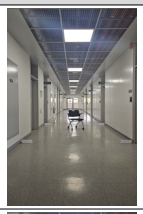
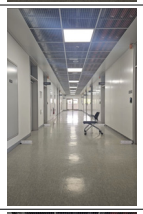
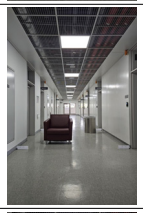
Based on these findings, obstacle-placement scenarios were developed featuring convex and concave configurations, previously demonstrated to mitigate corridor bottlenecks. An experimental framework was constructed to assess object detection performance under different placement conditions.

3. Object Detection Experiment

3.1. Experimental Setup and Sample Configuration

The experiment was conducted in a straight corridor of a university building in Gongneung-dong, Seoul, from 14:00 to 16:00 on June 6, 2025, to evaluate the feasibility of detecting obstacles in the corridor. An Android smartphone (1080p, 30 fps) was mounted on a tripod fixed at the center of the corridor to capture images vertically. This experiment did not precisely replicate a CCTV environment, but it did reflect the visual complexity of indoor spaces and partial occlusion by objects from the perspective of evacuees (1.2 – 1.4 m). Six objects were used in the experiment, considered loads during evacuation (Table 2.). The number of objects (single/multiple), direction (front/side), location (center/leaning toward left or right), and occlusion were incorporated to generate various detection conditions.

Table 2. Object composition

Type	Purpose	Image
Chair	Standard sample	
Chair	Structural deformation	
Couch+Table	Object occlusion	
Couch+Bin	Corridor narrowing	
Bin	Unlearn object	
Boxes	Intentional FP (false positive)	

3.2. Object Detection Results

Object detection was performed using a pre-trained YOLOv8n model. The analysis was conducted by extracting middle frames from each experimental sample and then performing YOLO inference on each image. The confidence threshold was set to 0.4. The experimental results are presented in Table 3. and Fig. 3.

In S2V6, the “chair” object is detected with a confidence level

Table 3. Detection result - YOLOv8n

Test No.	Object type	Detection label	Confidence
S2V6	Chair	Chair	0.77
S2V7	Chair	Chair	0.66
S2V8	Couch	Chair	0.78
	Table	Bench	0.46
S2V9	Couch	Chair	0.8
	Bin	Fail	-
S2V10	Bin	Fail	-
S2V11	Boxes	Fail	-

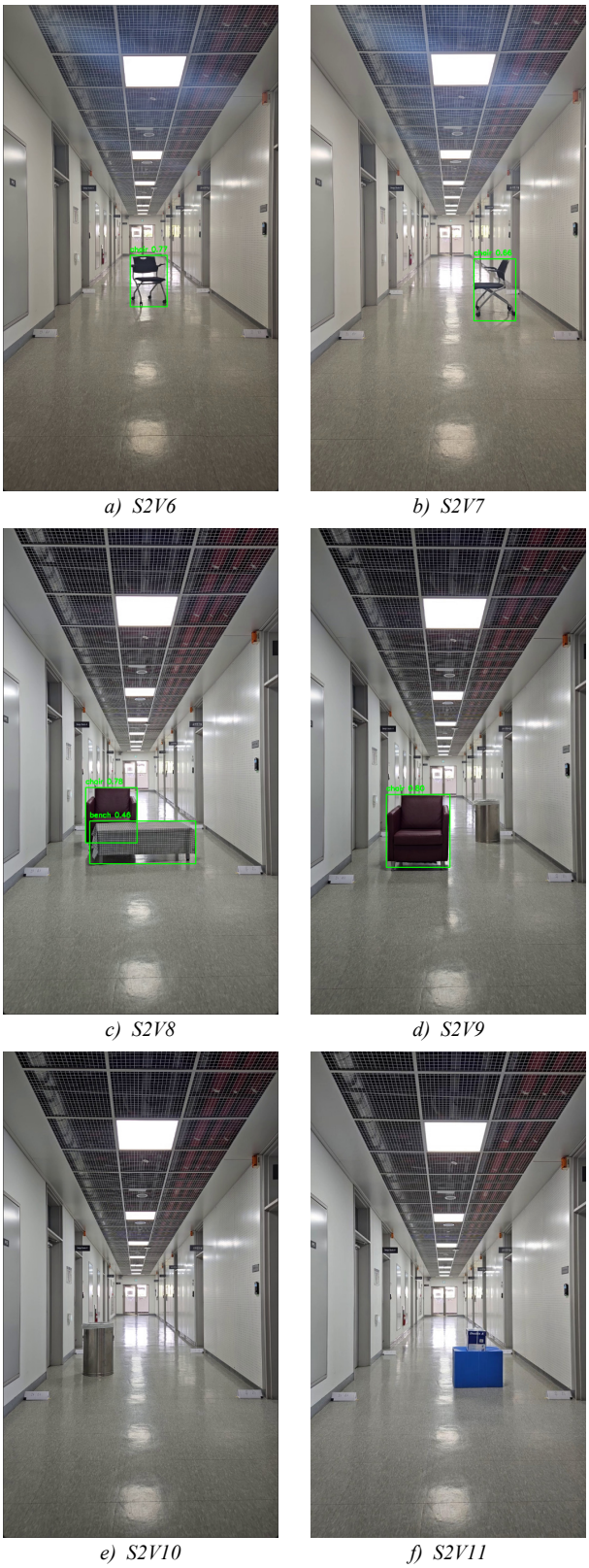
of 0.77, which is relatively high. This object is placed facing forward in the center of the corridor, and its bounding box is clearly defined. Conversely, in S2V7, the same "chair" object is present, but the confidence level decreases to 0.66. The object placed toward the right side of the corridor is assumed to have contributed to the decreased confidence level. In S2V8, the "chair" and "table" objects overlap with occlusion. Only "chair" is detected, whereas "table" is detected as "bench." Unlike "chair," which is detected with a confidence level of 0.78, "bench" is detected with a relatively lower confidence of 0.46, as occluded objects affect both its confidence and classification accuracy. In S2V9, only one object, recognized as a "chair," is detected, whereas the adjacent, "bin," which is not included in the recognizable classes, goes undetected. This object is not detected because it is not included in the pre-trained model's recognized classes. For the same reason, no detections are associated with S2V10, which only has trash bins, and S2V11, which includes only two boxes.

3.3 Result and Error Analysis

The object detection experiment using YOLOv8n showed that some objects were either missed or detected with reduced confidence. Accordingly, we analyzed the errors for each component, along with the limitations of the dataset structure and class definitions.

First, the placement angle of objects directly affected detection confidence. Comparing cases S2V6 and S2V7 revealed that although both contained the same object, "chair," the first case had a higher confidence level of 0.77 because the object was placed in the center, whereas the second case had a lower confidence level of 0.66 because the object was positioned toward the right side of the corridor. This result indicates that the YOLOv8n training data lack sufficient coverage for accurate detection across different fields of view.

Second, occlusion was a key factor that reduced detection reliability and classification accuracy. In S2V8, "chair" and "table" overlapped; the "chair" object had a high confidence level of 0.78, whereas the "table" object was misidentified as "bench"



a) S2V6 *b) S2V7*

c) S2V8 *d) S2V9*

e) S2V10 *f) S2V11*

Fig. 3. YOLOv8n object detection output

with a low confidence level of 0.46. This signifies that when certain parts of an object are occluded, YOLO cannot perform classification accurately solely based on the shape information.

Third, missed detections were observed for objects not

included in the recognizable classes. In S2V9, the image included trash bins, which were not part of the pre-trained classes in the existing model, and they were not detected at all. Detection also failed in S2V10, where trash bins were placed alone, and in S2V11, which contained stacked boxes. These objects were neither labeled nor included in the pre-training dataset (COCO) and, therefore, were not detected by the model. Thus, the model's restricted scalability is a limitation in detecting various types of obstacles that may appear in real-world environments.

Fourth, ambiguous class definitions between object groups also caused detection confusion. Distinguishing between a "chair" and a "couch," which have similar functions, can be ambiguous depending on their shape or size, leading to inconsistent classification or low confidence levels in the actual experiment. This implies that the current classification system relies more on morphological criteria than on functional criteria, indicating the need to redefine classes based on function by integrating related object groups.

According to this error analysis, limitations in the YOLOv8n pre-training dataset and inconsistencies in class composition can adversely affect detection performance during the experiment. Thus, additional training based on custom labeling is required, along with an object class composition appropriate for the experimental environment. In particular, actual CCTV-based evacuation environments contain various object groups that appear at different angles, with partial occlusion and complex placements. Ensuring that the training data reflect these conditions and improving the class-definition system can be key strategies for enhancing performance in the future.

4. Accuracy Improvement

4.1. Redefining Class and Custom Labeling

The pre-trained YOLOv8 model is designed to perform object detection of 80 classes based on the COCO dataset. However, this classification system poses clear limitations for the specific environment of indoor corridors and intended application of analyzing effective evacuation width.

In the experiments, "chair," "table," and "bin" were visually distinguishable, but in terms of function, they were recognized as identical based on the occupying width in the corridor.

Accordingly, the classes of objects used in the experiment were redefined as a single class, eliminating the existing class granularity, to establish a function-based class integration system. Consequently, the labeling structure was redefined (custom labeling) to ensure detection consistency and simplify the analysis.

This class redefinition process involved labeling all objects as a single "obstacle" class, considering their appearance, location, orientation, and occlusion, based on image data obtained from the existing experimental environment and using Roboflow. Training the model to include various fields of view (front, side, and diagonal), placement locations (center, right, and left), and occlusion conditions (partial or overlapping) allows a wider range of obstacle scenarios to be represented in real-world CCTV environments.

4.2. Process for Improving Detection

The training process, conducted after redefining the custom class, was based on the YOLOv8n model and aimed to enhance detection accuracy for the integrated "obstacle" class. A total of 106 images were manually labeled and augmented by horizontal flips, color distortion, and blurring (Table 4.). Training was conducted on the labeled data for 100 epochs with a confidence threshold of 0.4, a batch size of 8, and an image size of 640×640.

To improve detection performance, the pre-trained YOLOv8n model and the custom-trained model were applied independently, and their inference results were fused through IoU-based non-maximum suppression, rather than modifying the model structure (Fig. 4.). Accordingly, the models were designed to complement the generality of the pre-trained model with the environment-specific responsiveness of the custom model. The parallel inference results were applied using non-maximum suppression (NMS) during post-processing to remove duplicate objects and derive the final detection results.

Table 4. Data augmentation configuration used for training

Augmentation type	Description	Range / Parameters
Flip	Horizontal image flip	Applied
Rotation	90° rotation (clockwise, counter-clockwise)	±90°
Shear	Horizontal and vertical shear	±10° (both directions)
Brightness	Brightness adjustment	−15% to +15%
Blur	Gaussian blur	Up to 3 pixels
Noise	Random pixel noise	Up to 1.05% of pixels

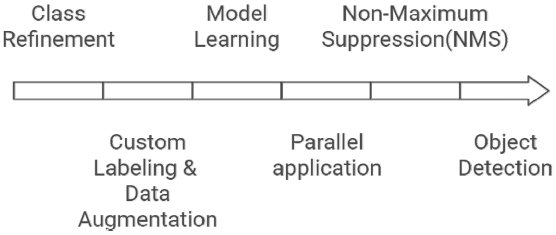


Fig. 4. Detection improvement process

4.3. Performance Improvement Results

The performance of the model custom-trained on augmented data is presented in Table 5., Fig. 5. and Fig. 6. Table 5. presents the performance comparison between the YOLOv8n model pre-trained on the COCO dataset and the custom-trained model of this study. Compared with the pre-trained model, the custom model exhibits numerical improvements across all areas, including Precision, Recall, mAP@0.5, and mAP@0.5:0.95, with mAP@0.5 showing an extremely high value of 0.995.

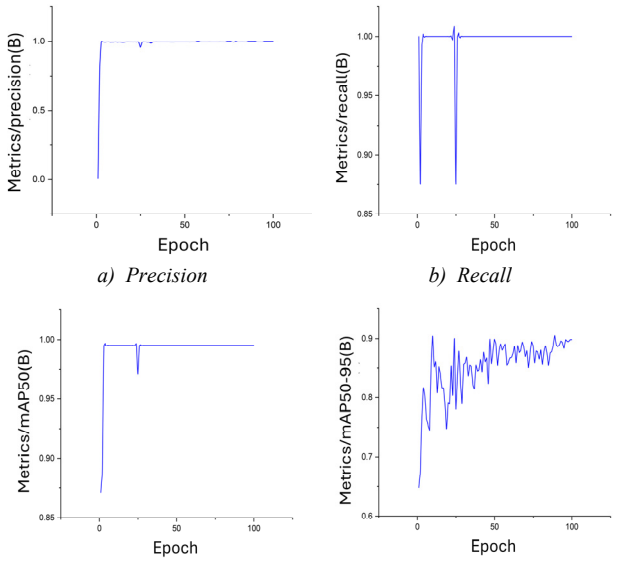
In Fig. 5., the changes in performance indicators are compared for training and validation losses by epoch between box loss, classification loss, and distribution focal loss (DFL). All three losses decrease sharply at the beginning, then gradually decline around 40 – 50 epochs and eventually converge at 100 epochs to 0.02 for box loss, 0.015 for classification loss, and 0.025 for DFL, marking the end of training (Fig. 5. a) – Fig. 5. c)). Specifically, the validation loss (Fig. 5. d) – Fig. 5. f)) also displays a similar decreasing curve, indicating that the model generalized stably based on the slight gap between the training and validation losses.

The changes in Precision, Recall, mAP@0.5, and mAP@0.5:0.95

Table. 5. Performance comparison: Pretrained COCO YOLOv8n and custom-trained YOLOv8n (indoor augmentation+parallel detection)

Metric	YOLOv8n*	Custom-trained YOLOv8n
Training dataset	COCO	COCO+ augmented indoor data
Detection class	80 classes	1 class
Precision	~0.900	0.9925
Recall	~0.880	1.000
mAP@0.5	~0.915	0.9950
mAP@0.5:0.95	~0.670	0.8998
F1 score (Peak)	~0.89	1.0000
GFlops	8.7	8.2
Training epochs	N/A (pretrained)	100 epochs
Detection strategy	Single detection	Parallel detection with post-processing

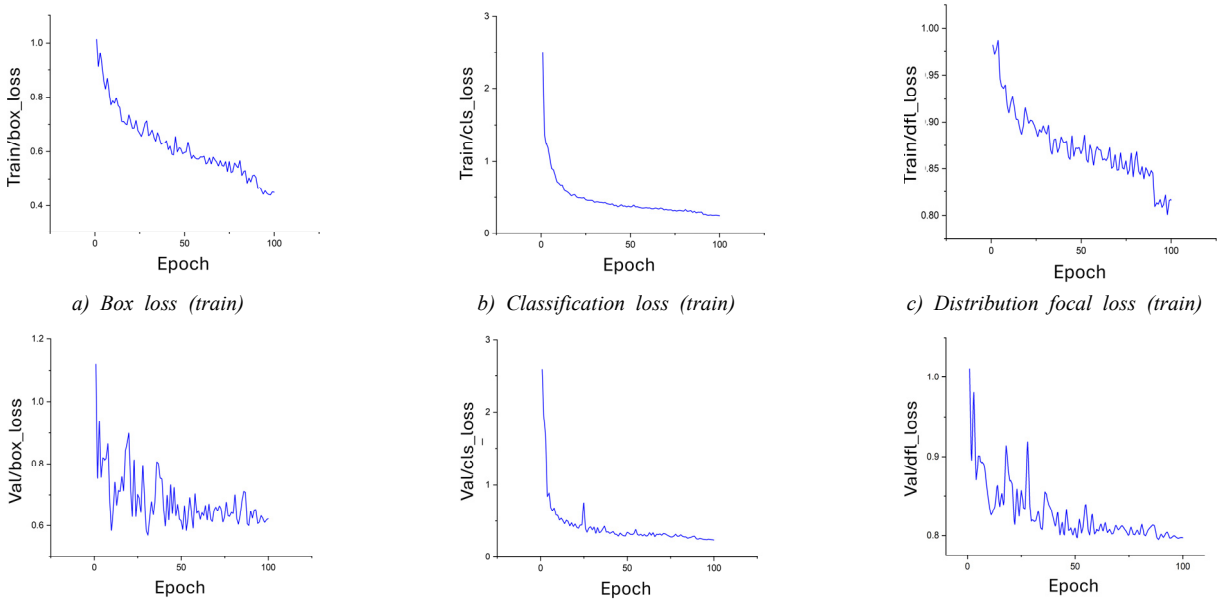
*COCO pretrained metrics are based on average performance benchmarks provided by Ultralytics



a) Precision *b) Recall*

c) mAP@0.5 *d) mAP@0.5:0.95*

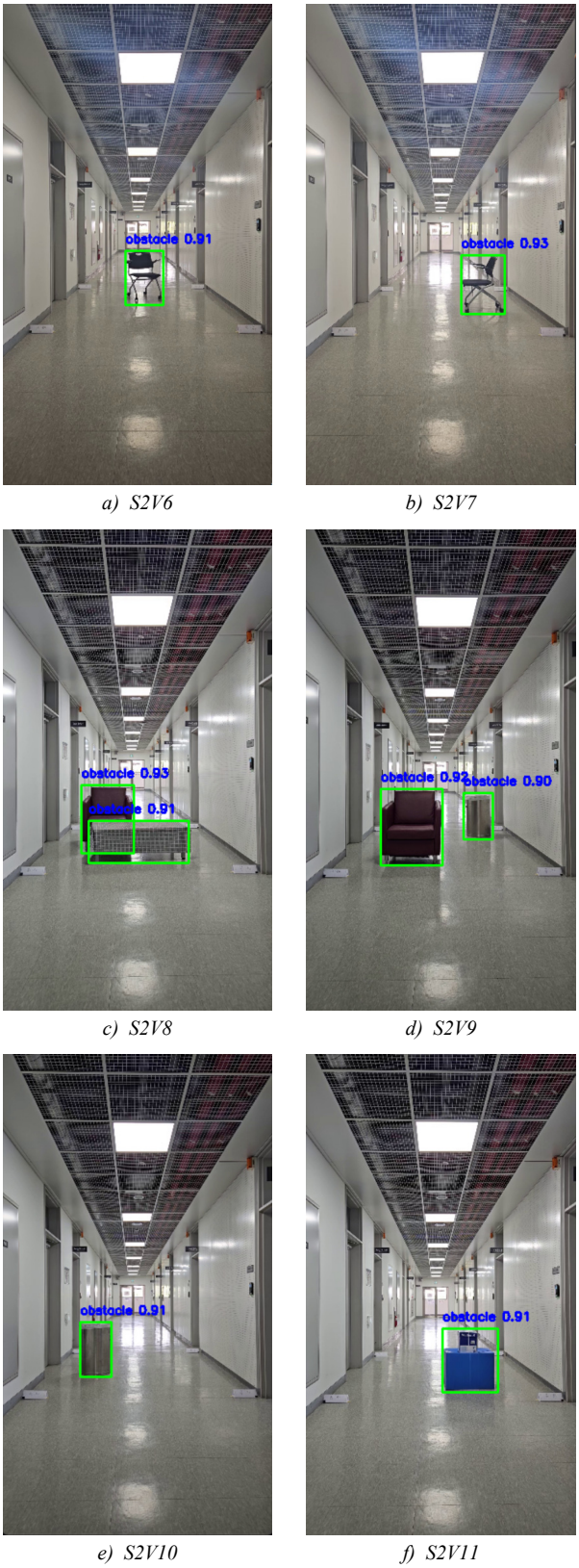
Fig. 6. Performance metric trends over epochs during model training



a) Box loss (train) *b) Classification loss (train)* *c) Distribution focal loss (train)*

d) Box loss (validation) *e) Classification loss (validation)* *f) Distribution focal loss (validation)*

Fig. 5. Training and validation loss trends and performance metrics over epochs



a) S2V6 b) S2V7

c) S2V8 d) S2V9

e) S2V10 f) S2V11

Fig. 7. Custom-trained YOLOv8n object detection output

indicators are shown in Fig. 6. Precision and Recall quickly increase to 0.80 and 0.75, respectively, before 20 epochs, and then stably converge to above 0.99 after 60 epochs (Fig. 6. a) and Fig. 6. b)); mAP@0.5 also exceeds 0.95 around 30 epochs and finally reaches 0.9950 at 100 epochs, marking the end (Fig. 6. c)); mAP@0.5:0.95, which reflects the higher IoU condition, gradually increases to 0.80 and then eventually converges to 0.8998 (Fig. 6. d)). Thus, all performance indicators reveal high accuracy and consistent performance in convergence, which indicates stable detection performance of the custom trained model.

Table 6. Detection result - custom-trained YOLOv8n

Test No.	Object type	Detection label	Confidence
S2V6	Chair	obstacle	0.91
S2V7	Chair	obstacle	0.95
S2V8	Couch	obstacle	0.93
	Table	obstacle	0.91
S2V9	Couch	obstacle	0.92
	Bin	obstacle	0.90
S2V10	Bin	obstacle	0.91
S2V11	Boxes	obstacle	0.92

4.4 Re-detection Result of the Custom Model

Re-detection was performed on the middle frames of the same images from S2V6 to S2V11. The YOLOv8n and the custom pre-trained models were applied in parallel, and NMS was then applied based on the IoU to remove duplicate detections.

Eight objects were detected through parallel detection, and the bounding box clarity and confidence levels for all objects improved (Fig. 7.). The single YOLOv8n model resulted in non-detection and detection of certain objects with low confidence (S2V7 and S2V8), whereas custom-based parallel detection produced stable detection across all samples with a confidence score of 0.92, which is an improvement of approximately 20%. The confidence levels of the detected objects ranged between 0.90 and 0.95, and detection performance remained fairly consistent even when the objects were not facing forward or were partially occluded (Table 6.).

This performance improvement is attributed to custom labeling, in which “chair,” “bench,” and “stool,” previously distinguished in the COCO dataset, were integrated into a single “obstacle” class, thereby reducing model confusion and improving detection consistency. Furthermore, the issues of missed detections, lowered confidence levels and mitigated the class confusion observed in previous experiments through custom labeling and data augmentation, ultimately improving model detection accuracy and stability.

5. Conclusion

This study was conducted to explore the fundamental

feasibility of an automated system that recognizes possible obstacles in real evacuation paths and calculates the effective evacuation width based on that recognition. The significance of this study lies in combining existing CCTV installed in buildings with a lightweight object detection model, thereby enabling the detection of corridor obstacles and acquisition of real-time, useful information without additional equipment. The entire process involves object detection, distance conversion, and evacuation width calculation; however, this study experimentally examined only the object detection stage.

The experiment was conducted by creating static imaging data that simulated a real-world corridor setting, and object detection was performed using the pre-trained YOLOv8n model. The results showed that objects captured from the front were accurately detected with a high confidence level, whereas those that were partially occluded or placed on the side led to missed detection and lower confidence levels. In addition, objects not included in the COCO dataset, such as trash bins and boxes, were either not detected by the pre-trained model or were detected incompletely. These results reveal that using a universal dataset cannot guarantee sufficient performance in specialized environments such as real-world evacuation paths.

To overcome these limitations, object groups serving the same function were integrated into one label, and model retraining was then performed through custom labeling and data augmentation. Applying the re-trained and pre-trained models in parallel and then performing post-processing through NMS improved the detection rate of previously undetected objects, along with overall confidence levels and accuracy. Specifically, detection performance was stably maintained even under partial occlusion or side-placement conditions, and class confusion was significantly reduced. This implies that the drawbacks of pre-trained models can be overcome by adapting even a limited amount of data to the specific setting and modifying the training framework accordingly.

The findings of this study are significant because they highlight the potential of an automated detection system for monitoring indoor evacuation paths. A method for improving object detection accuracy using existing systems was examined, and the results can serve as key foundational data for determining the actual effective evacuation width. Furthermore, the proposed framework is low-cost and can be applied without expensive equipment or complex sensors; thus, it is widely adaptable to existing buildings.

This framework can be further enhanced in the future to collect real-time information on obstacles in evacuation paths by linking with building CCTVs and automatically notifying administrators or managers by evaluating the evacuation path width. Additionally, it can serve as a practical safety management tool when linked with an evacuation guidance system to enable resetting evacuation paths, guiding evacuees, and avoiding risk areas during emergencies.

However, this study did not fully reflect a dynamic environment resembling real emergency situations, as the experiment was conducted in a controlled, static setting. Imaging conditions were also limited. Therefore, performance in complex backgrounds requires further verification. The dataset was created at the pilot level; hence, training data reflecting more diverse conditions and larger quantities need to be acquired.

Future studies will incorporate distance conversion and evacuation width calculation, building on the object detection performance verified in this study. Specifically, pixel coordinates will be converted to actual distances using homography-based perspective transformation, and the error range will be systematically verified by comparing the calculated evacuation widths with actual measurements. Additionally, performance should be evaluated under various imaging conditions, and the sensitivity of parameter settings should be analyzed to enhance practical applicability. Furthermore, real-time images are required to examine whether detection can be performed stably in scenarios involving group movements.

Acknowledgement

This study was supported by the 2025 National Research Foundation of Korea (NRF) research grant(No.2021R1A2C1014274).

References

[1] C.G. Lee et al., A study on the evacuation performance of evacuation system using real-time IoT information, Journal of Korea Multimedia Society, 22(2), 2019.02, pp.281-291.
[2] M.S. Kim et al., A study on the development of smartphone-based real-time evacuation scenarios for large-scale buildings, Journal of the Architectural Institute of Korea Planning & Design, 36(1), 2020.01, pp.15-26.
[3] D. Yifei, Y. Zhang, X. Huang, Intelligent emergency digital twin system for monitoring building fire evacuation, Journal of Building Engineering, 77, 2023, 107416.
[4] H.J. Shin, S.P. Jung, J.W. Kim, Potential use of space scanners in evacuation plans, Proceedings of Annual Conference of Korea Institute of Ecological Architecture and Environment, 24(1), 2024.05, pp.124-125.
[5] Z. Hanjing et al., Building on digital twin: Overcoming barriers and unlocking success in the construction industry, Journal of Construction Engineering and Management, 150(10), 2024, 04024142.
[6] M. Saeed Reza et al., Determining the stationary digital twins implementation barriers for sustainable construction projects, Smart and Sustainable Built Environment, 14(5), 2024, pp.1538-1563.
[7] Z. Zhengxia et al., Object detection in 20 years: A survey, Proceedings of the IEEE, 111(3), 2023, pp.257-276.
[8] Z. Zixiao et al., ViT-YOLO: Transformer-based YOLO for object detection, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp.2799-2808.
[9] R. Joseph et al., You only look once: Unified, real-time object

detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[10] Y. Sun, Z. Sun, W. Chen, The evolution of object detection methods, Engineering Applications of Artificial Intelligence, 133, 2024, 108458.
[11] Z. Chen et al., Dense-stream YOLOv8n: A lightweight framework for real-time crowd monitoring in smart libraries, Scientific Reports, 15(1), 2025, 11618.
[12] A. Istiak, W. Feng, Enhancing Bangladeshi license plate recognition: A YOLOv8 approach with roboflow integration for accuracy and speed optimization, International Journal for Research in Applied Science and Engineering Technology, 12(5), 2024, pp.1686-1699.
[13] C. Siyuan et al., The effect of obstacle layouts on pedestrian flow in corridors: An experimental study, Physica A: Statistical Mechanics and Its Applications, 534, 2019, 122333.
[14] F. Claudio et al., Systematic experimental investigation of the obstacle effect during non-competitive and extremely competitive evacuations, Scientific Reports, 10(1), 2020, 15947.
[15] I.M. Sticco, G.A. Frank, C.O. Dorso, Improving competitive evacuations with a vestibule structure designed from panel-like obstacles in the framework of the Social Force Model, Safety Science, 146, 2022, 105544.
[16] K. Daewa, A. Quaini, A kinetic theory approach to model pedestrian dynamics in bounded domains with obstacles, arXiv preprint arXiv:1901.07620, 2019.

1) Homography-based perspective transformation enables the restoration of actual distances and angular relationships on a plane from a single camera image. Hartley et al. (2004) stated that perspective distortion can be mathematically corrected by modeling the projection relationship between the actual geometry of a reference plane (such as a floor or wall) and the image plane in a single camera image using a homography. Zhang (2000) proposed a method for correcting a camera's intrinsic parameters and lens distortion by computing homography from corresponding points in images of a plane captured from two or more different angles, followed by conversion of pixel coordinates into real-world physical distances.

2) YOLOv8 model is pre-trained based on the Common Objects in Context (COCO) dataset to perform detection for nearly 80 object classes (person, chair, desk, automobile, etc.). In this study, 106 images were extracted from indoor corridor scenes and labeled to serve as additional training data, to assess detection performance for five classes: chairs, tables, couches, bins, and boxes, of which the latter two are not included in the COCO dataset.

3) Ultralytics, https://docs.ultralytics.com/ko/models/yolov8/#overview