MirrorVision: Light-Weight Floor Detection System for an Autonomous Robot in a Crowded Elevator

Azimbek Khudoyberdiev and Jihoon Ryoo*

Department of Computer Science, State University of New York, Korea, Incheon, Korea
Email: {azimbek.khudoyberdiev, jihoon.ryoo}@stonybrook.edu
*Corresponding author

***Abstract*—Delivering robots impact many facets of our life, including food delivery and restaurant services, with advancements enabling obstacle overcome, faster delivery, and minimizing human intervention. However, delivering robots remained to experience poor vertical mobility-elevator usage in multi-floor buildings. Incorporating new elevator models into the robot's elevator usage capabilities involves a long process of manufacturer approval and authentication. Furthermore, strict *fire-code* regulations pose communication barriers between the robot and the elevator. In this paper, we introduce *MirrorVision*-a novel approach designed for accurate floor detection during vertical mobility, regardless of obstructions blocking the robot's direct line of sight to the elevator number panel. First, we collected and pre-processed a dataset of direct and reflective views of elevator number panels via the pre-installed mirrors. Then, we trained mirrored images in various possibilities to accomplish accurate floor detection. *MirrorVision* provides a solid mechanism to understand floor numbers at the level of distorted images. Extensive evaluations show that *MirrorVision* achieves 98.8% accuracy for floor detection in a crowded elevator, while state-of-the-art EfficientDet and YOLOv5 achieved 90.8% and 93.3%, respectively.**

***Keywords*—autonomous robots, floor detection, indoor navigation, *MirrorVision*, faster Region Convolution Neural Network (R-CNN), EfficientDet, YOLOv5**

I. Introduction

Over the past decades, a number of robots have been deployed in several public places, including airports, hospitals, hotels, museums, etc., to perform a wide range of services (e.g., security, cleaning, delivery, and guidance) [1, 2]. These robots are fitted with several state-of-art algorithms and technologies, including computer vision, outdoor navigation, embedded systems, and just to name a few. For instance, Airstar, the first regularly operating robot assistant at Korea international airport, provides multilingual assistance and guides passengers using advanced functionalities like facial recognition and accurate navigation [3]. Another real-life example is Camello, an autonomous robot in Singapore that delivers groceries from markets to customers [4].

Robots require complete autonomous abilities to achieve the highest reliability and performance in accurately accomplishing tasks. One of the essential capabilities of autonomous robots is accurate indoor navigation that helps to localize the robot accurately and perform tasks in the right place [5]. However, their indoor navigation system has several limitations in terms of the multi-floor indoor environment and elevator control. The successful movement of robots in multi-floor indoors requires accurate floor detection in elevators [6]. Floor detection can help robots to move from floor to floor, just as people can move another floor using visual eyesight abilities in a multi-store building.

Current research progress has offered several methodologies for floor detection in elevators for multi-floor buildings. These approaches applied Wi-Fi signal strength [7, 8], air pressure calculations [9], and computer vision-based floor recognition techniques [10–18] to accurately detect the current floor status. However, there are several challenges to apply the above-mentioned approaches.

Wi-Fi-based methodologies are often grounded in the concept of Radio Frequency (RF) fingerprinting. They leverage the signal strength and angle of arrival properties of RF signals to determine the current floor. However, due to the elevator's closed environment and its metal composition, it is difficult to receive external signals in elevators. Furthermore, numerous factors can influence Wi-Fi signals, such as interference from other devices, signal multipath, and environmental changes [14].

Floor positioning using air pressure computations take advantage of the fact that air pressure decreases with altitude parameters. These techniques use built-in pressure sensors to compute the current floor level based on the measured air pressure. However, such methods are a highly unreliable solution for floor detection. The issue lies in the nearly uniform characteristics of air pressure within elevators due to their relatively confined and controlled environment. Thus, the differences in air pressure between floors are often so minute that they fall

Manuscript received September 6, 2023; revised September 30, 2023; accepted October 15 2023; published January 4, 2024.

within the sensor's error range, leading to a high probability of failures in floor detection.

Computer vision-based techniques are the most promising approach. As represented in the literature review section, they can provide advanced solutions for the robots to recognize the floor number from elevator number panels in elevators. However, if the elevator is crowded and occupied with people, the robot's line of sight to the elevator number panel can be blocked. This creates a significant obstacle for computer vision algorithms, which rely on a clear view of the target object to function effectively. Therefore, the robot may struggle to correctly identify the floor number from the elevator panel due to obstruction from people, luggage, or other objects. This, in turn, can negatively impact the robot's ability to accurately localize itself within the building and navigate effectively to its destination.

This paper suggests a low-cost mirror reflection based on a floor detection approach—*MirrorVision* to solve the blockage problem between the elevator number panel and the robot. As a result, even the robot cannot see the elevator number panel directly. Instead, the robot can use the reflected numbers to detect the current floor or upcoming floors, as shown in Fig. 1. The proposed floor-detecting system consists of three stages, firstly, the robot, which is equipped with a camera, enters the elevator, and starts to record a video from the inside of the elevator. In the second stage, a video is used as input to the video segmentation module, and the FFmpeg-based video segmentation model divides the video into frames by decreasing the size of the video. In the final stage, segmented frames are applied to the pre-trained Faster R-CNN-based floor detector module to detect and recognize the current floor number. We validate our system via various tests and confirm its accuracy in various state-of-the-art computer vision-based algorithms.

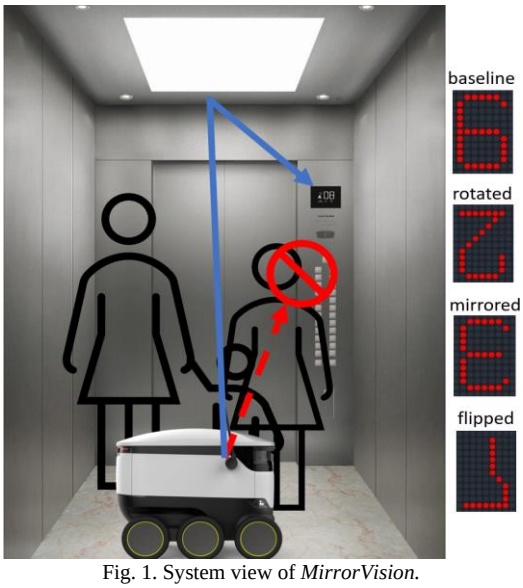


Fig. 1. System view of *MirrorVision.*

The rest of the paper is organized as follows. Related studies and highlights of the relevant literature in floor detection are presented in Section II. Section III illustrates the proposed system design, the detailed description of data processing, and applied computer vision algorithms for floor detection. Section IV depicts performance evaluation results. Section V presents the comparison and significance of the proposed Faster R-CNN module with various state-of-the-art computer vision techniques. Finally, the conclusion is given in Section VI.

II. Literature Review

Compared to the vast literature on conventional robot navigation systems, only a limited number of publications are available for floor detection systems in elevators, and authors mainly paid attention to elevator button recognition. In this section, we only pay attention to the intelligent floor detection systems in elevators for autonomous robots. This work proposed a computer vision-based robust elevator button recognition approach with a robot arm to control the elevator buttons [15]. A robot arm was able to click the target button and recognize the clicked button among other floor number buttons. The authors used contour-based object segmentation and feature point re-ordering algorithms to solve the ambiguity issue and increase the performance in object segmentation. One of the main issues of that work was that the robot had to stay near the elevator buttons. In real-life, elevators are crowded and used by many people. People can be a blockage between the number panel and the robot and click multiple buttons at a time. As a result, the robot cannot detect the required elevator button and fails in floor recognition.

This work presented the novel elevator button recognition approach based on OCR-RCNN [16]. Optical Character Recognition (OCR) network and Faster Region-Based Convolutional Neural Networks (RCNN) architecture were combined with a single neural network and used to recognize elevator buttons. The authors paid attention to several factors to increase the accuracy of button recognition, including various light effects, perspective distortion, and different button content, which made the task complicated. Although the proposed system's accuracy 94.6% was remarkable, computational efficiency was not high enough as they expected.

Zhu and Liu *et al.* [17] presented a novel algorithm to autonomously correct perspective distortions of elevator number panel images. First, the Gaussian Mixture Model (GMM) provided the grid fitting procedure using the results of button recognition. Then the estimated grid center was utilized to calculate camera motions to correct the image distortions. The authors used only 50 images to prove the efficiency of their proposed algorithm. However, more experiments were required to accurately remove perspective distortions for a valid comparison.

Several other methodologies have been suggested for button detection using buttons' visual features, for instance, texture and color. Yu and Dong *et al.* [18] introduced Hough transform, multi-symbol, and structural inference-based elevator button detection systems. García-Domínguez *et al.* [19] suggested a technique that applies shape, size, and color to identify buttons.

However, brightness and background illuminations lead to failure in button detection in those works.

III. Proposed System Architecture and Implementation Details

In this section of the paper, detailed proposed system design, data pre-processing, and Faster R-CNN-based floor detection mechanism were described briefly in subsections.

A. System Design

The detailed proposed system design is presented in Fig. 2. It can be clearly seen that the proposed approach includes four main stages, namely elevator-in, video segmentation, Faster R-CNN-based floor detection, and elevator-out. First, as the robot enters the elevator, it starts to record direct and reflection views of the elevator number panel from pre-installed mirrors using the camera. After that, the video streaming is sent to the video segmentation module, and the video is cut into individual frames in specific periodical thumbnails to decrease the video size to avoid dataset duplication. Segmented frames are applied to the pre-trained Faster R-CNN-based floor detection module. Afterward, the floor detection module detects the floor number with its classification score and bounding box. According to the detected floor number, the robot can localize itself and decide to go our wait on the required floor in the last stage. Accurate detection of the mirror-reflected elevator number panel has allowed the robot to watch and localize itself, even if there was a blockage between the robot and the elevator number panel.

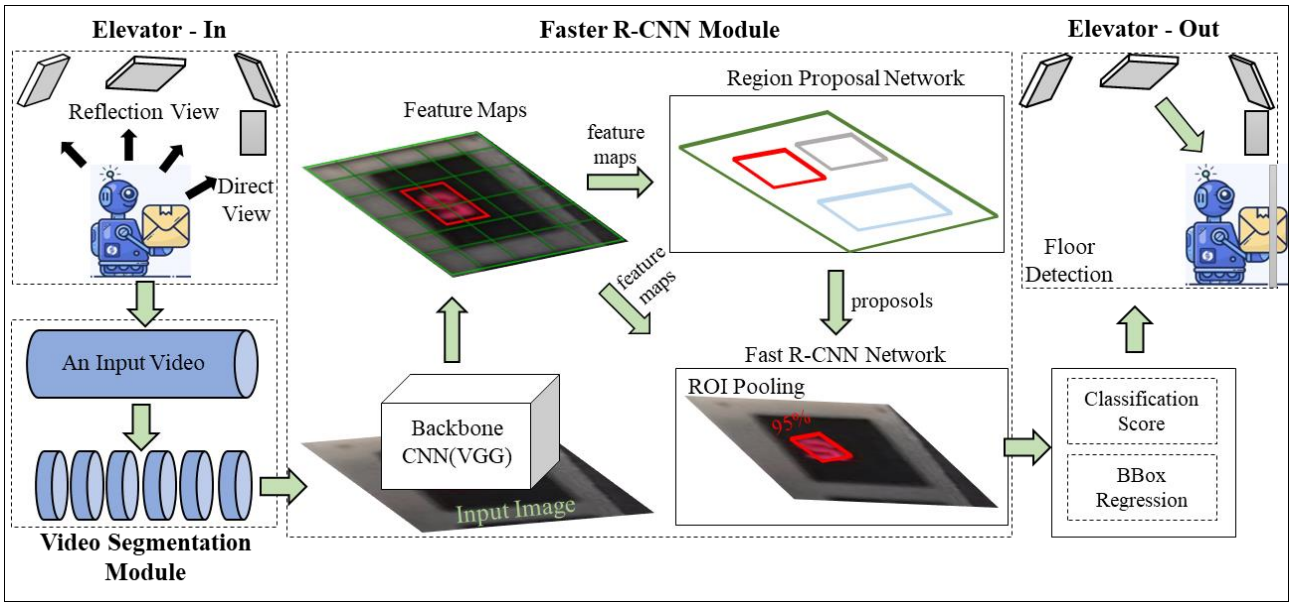


Fig. 2. Proposed system design.

B. Data Collection and Pre-Processing

The data collection and pre-processing stages are described in detail in this subsection of the paper. For the data collection, there were already three mirrors on the elevator's left, right, and back sides from the manufacturer. However, the left and right-side mirrors cannot directly reflect the elevator number panel. Therefore, we installed three mirrors on the elevator's back wall, which could reflect floor numbers from six different coordinates, as shown in Fig. 3. These three mirrors allow a robot to obtain reflections of the elevator number panel from the different angles and elevator coordinates. If there is a blockage between the robot and the elevator number panel, the robot can see the elevator number panel through the mirrors.

To increase the floor detector's accuracy by considering the elevator's size, the elevator floor is divided into six coordinates. As the robot enters the elevator, it can occupy one of the coordinates of the elevator from the given six coordinates each time. The view of the elevator number panel reflections can be different from different coordinates. Thus, all possible scenarios are collected from the mirrors based on six coordinates to increase the detection accuracy.

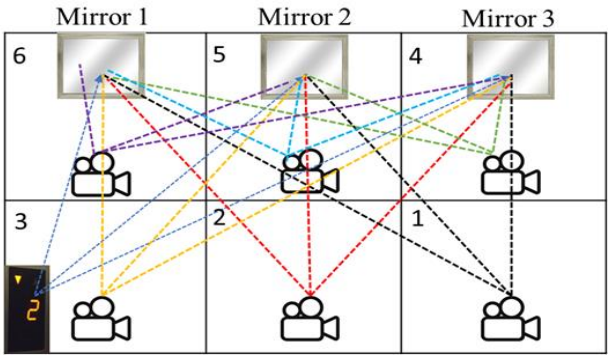


Fig. 3. Installation of mirrors and data collection from different angles and elevator coordinates.

The reflected floor number videos were collected from all six coordinates at different angles using a camera attached to the laptop from the ground floor to the seventh floor. Overall, 18 min of video were recorded

from the ground floor to the seventh floor. The frame rate of the collected video was 30 Frames per Second (FPS), and more than 32,000 raw images could be collected from the recorded video. However, the collected raw images could not be applied to the proposed system because of computational cost. Thus, the recorded video was inputted into the FFmpeg-assisted video segmentation module to avoid duplications and decrease video size. This allowed us to optimize the computational time by analyzing input frames faster. More precisely, the collected video was forwarded to the video segmentation module, which segmented the video into frames every 0.25 s (I = 0.25 s). It allowed extracting four images from a one-second video instead of duplicating 30 images. Overall, around 4300 individual frames were collected in the video segmentation phase from the basement to the seventh floor of the building to train the proposed system.

Several examples of the individual frames segmented from the reflected elevator number panel recordings are shown in Fig. 4. However, we cannot directly use these images to train our proposed Faster R-CNN algorithm-based floor detection model. The reason is that the robot must recognize and detect the floor number; the surroundings of the number panel are not important for detecting floors in crowded elevators. In addition, if the surroundings of the elevator number panels are considered in each training and testing, the accuracy of floor detection decreases, which leads to failures in accurate floor detection. Therefore, elevator number panels were cropped, and cropped images were annotated to train the Faster R-CNN-based floor detection module.

Fig. 4. Examples of floor number reflected images.

Annotation maps an object to its respective label by drawing a rectangular box (bounding box) over the object. Bounding boxes are a series of values or coordinates that present the position of the floor number in an image. Several annotation formats are widely used to create annotation files, including YOLO, Pascal Visual Object Classes (VOC), COCO, and others. For example, in the annotation phase, the first elevator number panel images were annotated in the Pascal VOC XML format to prepare the dataset for the training models because the Pascal Visual Object Classes (VOC) format allows creating of single XML annotation files for each image with the image details, bounding box coordinates, classes, rotation, and other essential values [20]. Therefore, it became easier to label floor number images separately in eight classes, namely, *Base 1*, *First*, *Second*, *Third*, *Fourth*, *Fifth*, *Sixth*, and *Seventh*, as represented in Fig. 5 with their respective bounding boxes.

Fig. 5. Examples of annotated images with bounding boxes in eight classes.

Moreover, the data augmentation technique was applied for the dataset to crop randomly, flip (horizontal and vertical), rotate (clockwise, counterclockwise, and upside down), bright (from −25% to 25%), blur, generate images (in 2 copies), and to resize 416×416 pixels. At the end of the data preparation, more than 8000 trainable images were ready to use for training (70%), validation (15%), and testing (15%) with eight-floor labels.

C. Implementation and Training

The data collection and development of the proposed system were conducted at the State University of New York, Korea campus, Block C from April 2022 to August 2022. The development of the proposed system comprises three main phases: data collection, implementation, and deployment. In the implementation phase, we deployed and tested several state-of-the-art object detection algorithms for accurate floor detection for autonomous robots in the elevator using mirror-reflected floor numbers. We implemented the Faster R-CNN, EfficientDet, and YOLOv5-based floor detection systems, and pre-trained COCO weights were applied to all models.

The dataset of over 8000 mirrored elevator number panel images is categorized into eight classes, each representing a distinct floor level. Annotating these images was streamlined using the Pascal VOC XML format. The dataset is divided into 5600 training images, 1200 for validation, and 1200 for testing, facilitating a robust and well-rounded evaluation of our model's effectiveness. Training the Faster R-CNN, EfficientDet, and YOLOv5 models required about 5 h, 6.5 h, and 5.5 h for 100 epochs, respectively.

1) Faster R-CNN: We used Detectron2 [20] framework to implement our Faster R-CNN model. Detectron2 was developed based on PyTorch by Facebook's AI Research team, and using this framework, we can easily design and deploy our object detection, recognition, and segmentation models. The initial version of Detectron was implemented in Caffe2, but the current version was written on PyTorch. Detectron2 requires datasets in the Common Objects in Context (COCO) JSON format. After data preparation and pre-processing, VGG16 is the backbone for image feature extraction.

Then the extracted image feature maps are shared between Region Proposal Network (RPN) and Fast R-CNN. Based on extracted feature maps, the RPN generates region proposals. The region proposal encompasses the object's location in the images with various scales and aspect ratios. Then region proposals are inputted to the ROI Pooling, and the responsibility of ROI Pooling is extracting fixed-length feature vectors from all region proposals and feature maps. After that, Fast R-CNN classifies the extracted feature vectors to detect the floor number with their classification score and bounding box [21]. We applied the weight attenuation to 0.0005 for the training network model, and the momentum was 0.937. The training iterations were 5000. The learning rate was initialized as 0.01, and the learning rate decay was 0.00001. The batch size was 64, with 100 epochs.

2) EfficientDet: EfficientDet [22] is one of the widely used methods of convolutional neural networks, and this technique can efficiently detect objects by combining layer width, layer depth, and resolution parameters. EfficientDet was initially implemented in Tensorflow and Keras but currently has implementations using PyTorch. PyTorch-based implementation of EfficientDet performs faster and more accurate in debugging capabilities compared with Tensorflow and Keras-based implementations. EffecientNet [23] is the backbone of the EfficientDet, and it has classification and a custom detection network. EfficientDet also requires datasets in the COCO JSON format. The same number of learning rates, iterations, batch sizes, and epochs were employed in the EfficientDet training process. The EfficientDet-based classification model accurately classified (90.8%) the floor numbers based on reflected elevator number panel images.

3) YOLOv5: To test and prove the effectiveness of our mirror reflection-based floor detection model, we also used YOLOv5 [24], one of the widely used object detection algorithms, because of its accuracy and speed. YOLO is an acronym for "You Only Look Once," and it is for detecting objects. The network architecture of YOLOv5 comprises three main parts: Model Backbone, Model Neck, and Model Head. The Model Backbone extracts essential features from the given input images. CSP (Cross Stage Partial Networks) is used as a Backbone in YOLOv5. Model Neck is applied to generate feature pyramids to identify the same object in various sizes and scales. YOLOv5 calls PANet a model neck to obtain feature pyramids. Model Head mainly performs the final object detection layer by generating final output vectors with objectness score, class probability, and bounding box. The original Pascal VOC XML dataset was exported to the YOLOv5 PyTorch format because training and test data must be in the YAML file in YOLOv5 [25]. The same number of learning rates, iterations, batch sizes, and epochs were also employed in the YOLOv5 training process. The YOLOv5-based classification model accurately classified (93.3%) the floor numbers based on reflected elevator number panel images.

D. Used Technologies

Table I presents the technologies used on a general-purpose machine for the implementation environment.

TABLE I. SPECIFICATION OF THE IMPLEMENTATION ENVIRONMENT

System Parameter	Description
Operating System	Windows 10
CPU	Intel Core (TM) i7-7700K CPU @4.20 GHz
GPU	NVIDIA GeForce GTX 2060 Ti
Primary Memory	24 GB DDR4
Framework	FFmpeg, Detectron2
Libraries	Torch 1.5, Torchvision 0.6, CUDA 10.1
CNN models	Detectron2, EfficientDet, YOLOv5
Programming language	Python 3.9

IV. EVALUATION RESULTS

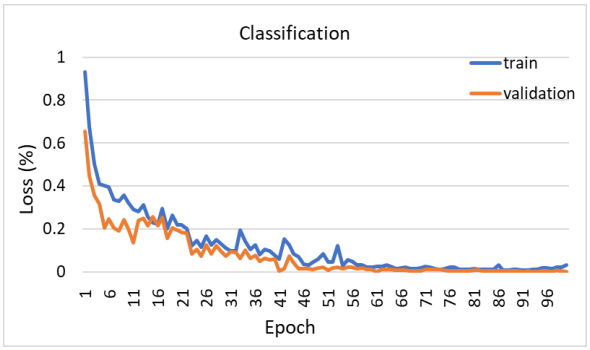
A. Loss Analysis

As discussed, the proposed *MirrorVision* network model architecture contains two fully connected network layers, the RPN and Fast R-CNN layers. Both layers define independent loss functions for floor number classification and bounding-box regression. Thus, a combination of the network losses is considered a multitask loss: classification loss and regression loss. The former is employed to classify the target floor number among all other floors, and the latter is applied for regressing a bounding-box to locate the classified floor number.

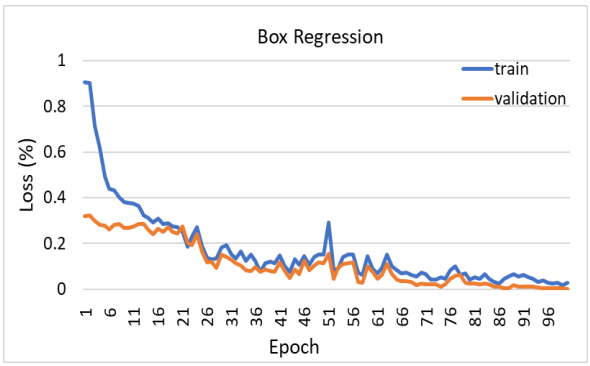
Fig. 6 depicts the loss metrics for the classification and regression results, respectively. In the first epoch of classification, the training and validation loss remains at around 90% and 65%, respectively. Both loss figures decline gradually between training epochs 2 and 21. There are small fluctuations from epoch 21 to epoch 40, and the classification loss is around 15%. After 55 epochs, both classification losses maintain the same level over the training and validation sets, and the last loss scores are just above 0.001%.

Box regression validates how well the proposed model can localize the center of the floor number panel and how well the predicted bounding-boxes fit a floor number. Fig. 6(b) compares the bounding-box localization errors around floor numbers over the training and validation sets. At the beginning of the epochs, the training loss was around 90% and decreased remarkably until epoch 20, and in this epoch, training and validation errors reached the same level (loss = 20%).

To sum up, although classification loss has reached nearly the same level for both training and validation results, training loss of box regression is moderately higher than validation loss. Compared with training loss, validation loss is less by around 2% loss after epoch 50. At the end of the training and validation epochs, bounding-box regression achieved 2% and 0.2% errors, respectively.

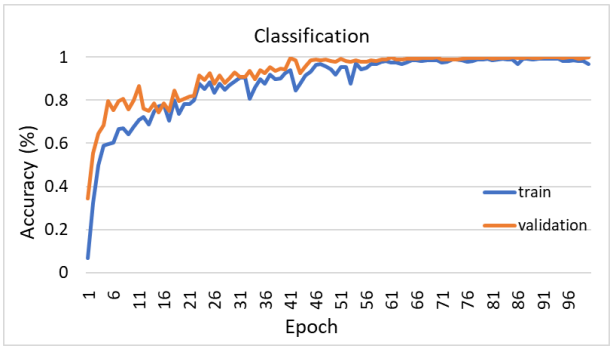


(a) Classification loss

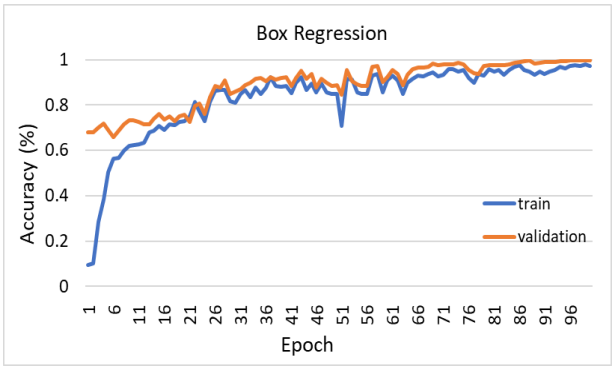


(b) Regression loss

Fig. 6. (a) Classification loss and (b) Regression loss results over the training and validation sets.



(a) Classification accuracy



(b) Regression accuracy

Fig. 7. (a) Classification accuracy and (b) Regression accuracy results over the training and validation sets.

B. *Accuracy Analysis*

Fig. 7 presents the classification and regression accuracy results over the training and validation sets. Classification accuracy is one of the most common metrics for summarizing the performance of classification models. More precisely, classification accuracy requires employing the classification model to predict each example in the given dataset. Then, predicted outcomes are compared to the known labels for those examples in the dataset. The accuracy of the proposed floor number detection model is the proportion of predicted correct floor numbers divided by all floor number predictions over the given dataset. As can be seen, in the initial epochs, the classification accuracy of the training and validation sets are about 6% and 34%, respectively. After epoch 50, the training and validation accuracy is around 99% for both training and validation cases. Regression accuracy shows how well the faster R-CNN algorithm can locate the bounding-boxes around the floor numbers. The training and validation accuracy plots show that the proposed floor number detection system achieved 99.8% accuracy at the end of the given epochs.

To summarize, the performance of the *MirrorVision* is remarkably accurate in localizing the autonomous robot in the elevator in terms of floor number classification and bounding-box regression. The former achieved nearly 100% accuracy in classifying the floor number after epoch 60, whereas the latter reached 99.6% accuracy in bounding-box localization after epoch 85 over the validation set.

V. Comparison and Significance

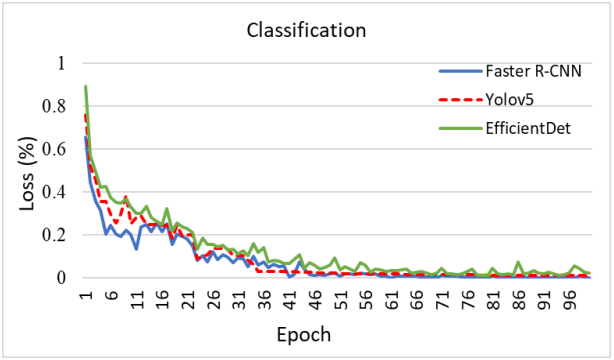
This section of the paper briefly describes the significance of the proposed Faster R-CNN-based floor number detection results by comparing EfficientDet and YOLOv5-based floor number detection results in terms of loss and detection accuracy.

A. *Comparative Analyses of Detection Loss Results*

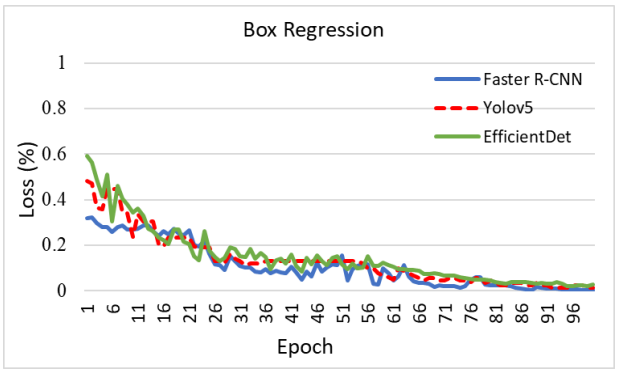
Fig. 8 compares the floor number classification and regression loss results of the Faster R-CNN, YOLOv5, and EfficientDet over the validation set. The error rate of the EfficientDet-based floor number classification is greater than the remaining models. The average classification loss of Faster R-CNN, YOLOv5, and EfficientDet-based floor number detection equals 0.07%, 0.09%, and 0.12%, respectively.

In the first epoch of the validation set, the error rates of the Faster R-CNN, YOLOv5, and EfficientDet models based on bounding box regression are approximately 0.32%, 0.48%, and 0.59%, respectively. On the other hand, in the last epoch of the validation set, the classification loss results are nearly equal.

Generally, there is no significant difference between YOLOv5 and EfficientDet's error rates. In contrast, our proposed Faster R-CNN-based floor detection model's error rate in bounding box regression is slightly less than the remaining models. The average bounding box regression loss of Faster R-CNN, YOLOv5, and EfficientDet-based floor detection equals 0.09%, 0.12%, and 0.14% over the validation epochs, respectively.



(a) Classification loss



(b) Regression loss

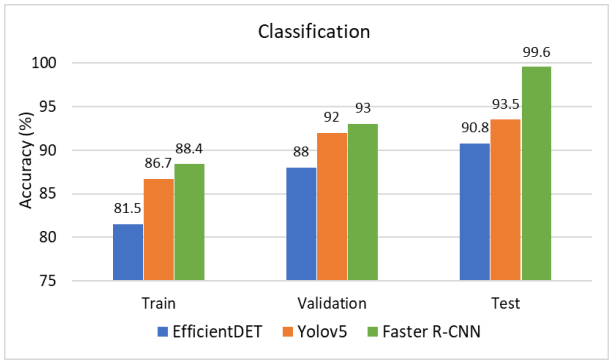
Fig. 8. Comparative analysis of (a) Classification loss and (b) Regression loss results.

B. Comparative Analyses of Detection Accuracy Results

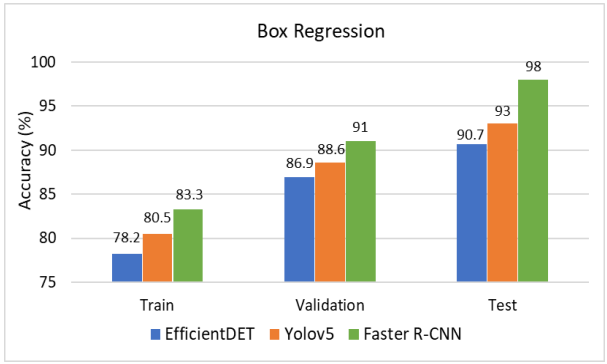
In this subsection of the paper, we present a detailed comparison of the proposed Faster R-CNN-based floor detection results with two other state-of-the-art YOLOv5, EfficientDet models over the training, validation, and test sets.

Fig. 9 compares the floor number classification and regression ac-curacy results of Faster R-CNN, YOLOv5, and EfficientDet-based floor detection over the training, validation, and test sets. The training accuracy of the EfficientDet, YOLOv5, and Faster R-CNN-based floor detection models are 81.5%, 86.7%, and 88.34%, respectively. For the validation set, the classification accuracy score is about 6% higher for all models compared with the training set. The test accuracy results show that the proposed Faster R-CNN-based floor detection model achieved 99.6% classification accuracy. In contrast, the YOLOv5 and EfficientDet-based floor detection models performed 93.5% and 90.8% classification accuracy, respectively.

Compared to the other two modules, the EfficientDet-based floor detection model has achieved the lowest bounding box regression accuracy: 78.2%, 86.9%, and 90.7% over the training, validation, and test sets, respectively. While the regression accuracy of the YOLOv5-based model has had 80.5%, 88.6%, and 93% over the mentioned data set. Our proposed Faster R-CNN model achieved the highest accuracy over the data sets, and the final bounding box regression accuracy in the new and unseen floor number panels is 98%.



(a) Classification accuracy



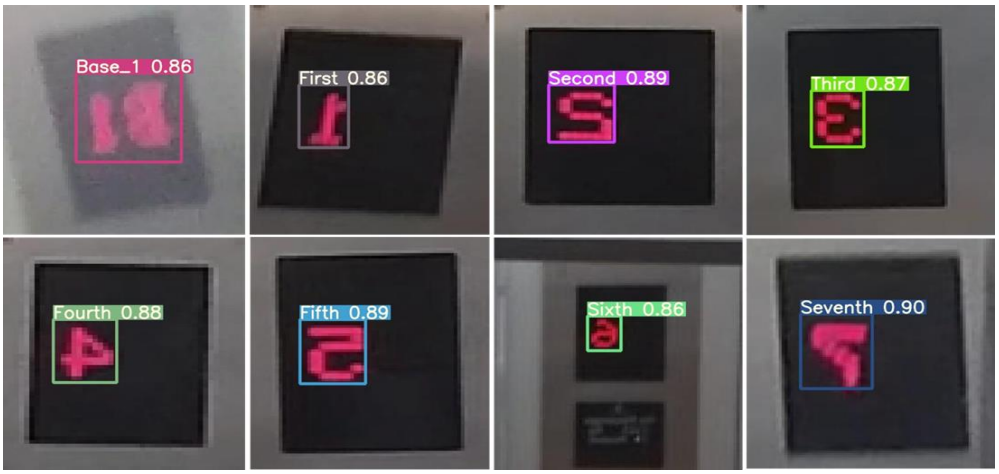
(b) Regression accuracy

Fig. 9. Comparative analysis of (a) Classification accuracy and (b) Regression accuracy results.

Fig. 10 represents the comparative analyses of Efficient-Det, YOLOv5, and Faster R-CNN-based floor detection results over the test set. As can be seen from the EfficientDet-based floor detection results in Fig. 10(a), *base 1*, the *fifth*, and *sixth* floors were detected with higher than 90% accuracy. In contrast, the *third*, *fourth*, and *seventh* floors achieved 89%, 86%, and 88% accuracy, respectively. However, the *first* and the *second* floors had the lowest accuracy, 74%, and 75%, respectively, compared with the remaining floors.

Fig. 10(b) depicts the YOLOv5-based floor detection results; the *basement*, *first*, and *sixth* floors achieved 86% accuracy in the output examples, while the *third* and *fourth* floors had 87% and 88% accuracy, respectively. The *second* and *fifth* floors' recognition achieved 89% accuracy, whereas the *seventh* floor had the highest accuracy (90%) among other floor numbers. Output result examples of the Faster R-CNN-based floor detection Fig. 10(c) illustrate that the model achieved 100% accuracy in detecting the *basement* floor over the test dataset, while the classification accuracy was from 99% to 100% in the remaining floor numbers. Rotated, mirrored, and flipped floor numbers with various brightness and blurred levels could not affect the accuracy of the floor number detection model.

(a) EfficientDet-based floor detection results



(b) YOLOv5-based floor detection results

(c) Our Faster R-CNN-based proposed floor detection results

Fig. 10. Some test results over the test set to comparatively analyze EfficientDet, YOLOv5, and Faster R-CNN-based floor number detection results. The bounding boxes refer to the detected floor numbers annotated by their classification scores.

VI. Conclusion

In this work, we proposed MirrorVision, an accurate floor detection system in crowded elevators for autonomous robots. Our system can accurately detect floor numbers from the elevator number panels using mirror-reflected images even though there is an obstacle between the robot and the elevator number panel. The mirror-reflected videos for each floor were collected from the experimental environment, and the gathered videos were divided into frames in periodical thumbnails using the FFmpeg-based video segmentation module to decrease video size and increase computational efficiency. Those frames were used to train, validate, and test

modern state-of-the-art object detection algorithms, including Faster R-CNN, EfficientDet, and YOLOv5. The comparative analysis of the floor detection algorithms shows that the Faster R-CNN achieved an average of 98.8% accuracy in classifying floor labels with respected bounding boxes, whereas the EfficientDet and YOLOv5-based floor detection have achieved 90.8% and 93.3% accuracy, respectively.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

JR and AK conceived the idea of the paper, conducted the research, and wrote the paper; AK analyzed the data and prepared the original draft; JR supervised the manuscript; all authors have approved the final version.

FUNDING

This work is supported by the Korean National Research Foundation grant NRF-2021R1F1A1052489, the MSIT, ICT CC Program (IITP-2019-2011-1-00783), and the Korea Radio Promotion Association (RAPA) XR-LAB grant.

ACKNOWLEDGMENT

The authors are thankful to the anonymous reviewers for their valued comments and suggestions.

REFERENCES

[1] W. Jochen, W. Kunz, and S. Paluch, "The service revolution, intelligent automation and service robots," *European Business Review*, vol. 29, no. 5, 909, Feb. 2021.

[2] E. Chang, *Museums for Everyone: Experiments and Probabilities in Telepresence Robots*, 1st ed. Routledge, New York: Taylor Francis Group, 2019, pp. 65–76.

[3] M.J. Kim, S. Kohn, and Shaw, T. "Does long-term exposure to robots affect mind perception? An exploratory study," in *Proc. the Human Factors and Ergonomics Society Annual Meeting*, Dec 2020, vol. 64, no. 1, pp. 1820–1824.

[4] R. Bogue, "Robots in a contagious world. Industrial robot," *The International Journal of Robotics Research and Application*, vol. 47, no. 5, pp. 673–642, 2020.

[5] W. Guan, S. Chen, S. Wen, Z. Tan, H. Song, and W. Hou, "High-accuracy robot indoor localization scheme based on robot operating system using visible light positioning," *IEEE Photonics Journal*, vol. 12, no. 2, pp. 1–16, March 2020. doi: 10.1109/JPHOT.2020.2981485

[6] J. Huang, H. Luo, W. Shao, F. Zhao, and S. Yan, "Accurate and robust floor positioning in complex indoor environments," *Sensors*, vol. 20, no. 9, 2698, Jan 2020. doi:10.3390/s20092698

[7] H. Qi, Y. Wang, J. Bi, H. Cao, and S. Xu, "Research on har-based floor positioning," *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, 437, June 2021. doi: 10.3390/ijgi10070437

[8] C. De Cock, W. Joseph, L. Martens, J. Trogh, and D. Plets, "Multi-floor indoor pedestrian dead reckoning with a backtracking particle filter and Viterbi-based floor number detection," *Sensors*, vol. 21, no. 13, 4565, July 2021. doi: 10.3390/s21134565

[9] M. Yu, F. Xue, C. Ruan, and H. Guo, "Floor positioning method indoors with smartphone's barometer," *Geo-Spatial Information Science*, vol. 22, no. 2, pp. 138–148, June 2019. doi: 10.1080/10095020.2019.1631573

[10] J.-G. Kang, S.-Y. An, and S.-Y. Oh, "Navigation strategy for the service robot in the elevator environment," in *Proc. 2007 International Conference on Control, Automation and Systems*, IEEE, Seoul, 2007, pp. 1092–1097. doi: 10.1109/ICCAS.2007.4407062

[11] K. T. Islam, G. Mujtaba, R. G. Raj, and H. F. Nweke, "Elevator button and floor number recognition through hybrid image classification approach for navigation of service robot in buildings," in *Proc. International Conference on Engineering Technology and Technopreneurship (ICE2T). IEEE*, Kuala Lumpur, 2017, pp. 1–4. doi: 10.1109/ICE2T.2017.8215992

[12] H. Zhang, W. Tao, J. Huang, and R. Zheng, "Development of an in-building transport robot for autonomous usage of elevators," in *Proc. 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, IEEE, Shenyang, China, 2018, pp. 44–49. doi: 10.1109/IISR.2018.8535646

[13] J. G. Kang, S. Y. An, W. S. Choi, and S. Y. Oh, "Recognition and path planning strategy for autonomous navigation in the elevator environment," *International Journal of Control, Automation and Systems*, vol. 8, no. 4, pp. 808–821, Aug 2010.

[14] N. Jain, H. Shrivastava, and A. A. Moghe, "Production-ready environment for HLS player using FFMPEG with automation on s3 bucket using ansible," in *Proc. 2nd International Conference on Data, Engineering and Applications (IDEA). IEEE*, Bhopal, India, 2020, pp. 1–4. doi: 10.1109/IDEA49133.2020.9170694

[15] H. H. Kim, D. J. Kim, and K. H. Park, "Robust elevator button recognition in the presence of partial occlusion and clutter by specular reflections," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, pp. 1597–1611, Jul 2011.

[16] D. Zhu, T. Li, D. Ho, T. Zhou, and M. Q. Meng, "A novel OCR-RCNN for elevator button recognition," in *Proc. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Madrid, Spain, 2018, pp. 3626–3631. doi: 10.1109/IROS.2018.8594071

[17] D. Zhu, J. Liu, N. Ma, Z. Min, and M. Q. H. Meng, "Autonomous removal of perspective distortion for robotic elevator button recognition," in *Proc. 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, Dali, China, 2019, pp. 913–917. doi: 10.1109/ROBIO49542.2019.8961720

[18] X. Yu, L. Dong, L. Li, and K. E. Hoe, "Lift-button detection and recognition for service robot in buildings," in *Proc. 2009 16th IEEE International Conference on Image Processing (ICIP)*, IEEE, Cairo, Egypt, 2009, pp. 313–316. doi: 10.1109/ICIP.2009.5413667

[19] M. García-Domínguez, C. Domínguez, J. Heras, E. Mata, and V. Pascual, "UFOD: An AUTOML framework for the construction, comparison, and combination of object detection models," *Pattern Recognition Letters*, vol. 145, pp. 135–140, May 2021. doi: 10.1016/j.patrec.2021.01.022

[20] F. Utaminingrum, Renaldi P. Prasetya, and Rizdania, "Combining multiple feature for robust traffic sign detection," *Journal of Image and Graphics*, vol. 8, no. 2, pp. 53–58, June 2020. doi: 10.18178/joig.8.2.53-58

[21] R. Hasegawa, Y. Iwamoto, and Y. Chen, "Robust Japanese road sign detection and recognition in complex scenes using convolutional neural networks," *Journal of Image and Graphics*, vol. 8, no. 3, pp. 59–66, Sept. 2020. doi: 10.18178/joig.8.3.59-66

[22] R. Khan, T. F. Raisa, and R. Debnath, "An efficient contour based fine-grained algorithm for multi category object detection," *Journal of Image and Graphics*, vol. 6, no. 2, pp. 127–136, Dec. 2018. doi: 10.18178/joig.6.2.127-136

[23] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10781–10790.

[24] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. International Conference on Machine Learning. PMLR*, Long Beach, California, 2019, pp. 6105–6114.

[25] Y. Liu, L. Geng, W. Zhang, Y. Gong, and Z. Xu, "Survey of video based small target detection," *Journal of Image and Graphics*, vol. 9, no. 4, pp. 122–134, Dec 2021. doi: 10.18178/joig.9.4.122-134