https://doi.org/10.7840/kics.2025.50.12.1822

A MobileViT-Based Detection System for Motorcycle Traffic Violations

In Gon Kim◆, Soo Young Shin°

ABSTRACT

This paper proposes a real-time detection system for motorcycle traffic violations, which pose a significant threat to road safety. The system defines and detects three types of violations: signal violations, centerline crossing, and crosswalk violation. Road elements and motorcycles are detected using YOLO (You Only Look Once), followed by MobileViT (Mobile Vision Transformer)-based time-series analysis to interpret movement patterns over time. The system is built on ROS 2 (Robot Operating System 2) and operates in real time on embedded platforms such as Jetson Orin. Detection results are stored in JSON and CSV formats for further use. Experimental validation using actual blackbox driving footage demonstrated over 93% accuracy across various conditions. The integration of MobileViT effectively compensated for missed detections by capturing temporal patterns.

Key Words : Traffic violation detection, MobileViT, time-series analysis, object detection, real-time detection, lightweight model

Ⅰ. Introduction

Traffic violations by motorcycles significantly threaten road safety, increasing the demand for technologies capable of real-time detection[1]. Existing traffic detection systems primarily rely on stationary CCTV or single-frame-based object detection, limiting their ability to accurately capture context-dependent behaviors such as signal violations or centerline crossing[2,3].

Due to their smaller size and higher maneuverability compared to vehicles, motorcycles present unpredictable and irregular trajectories, making conventional vehicle detection methods unsuitable. Single-frame position data alone often leads to false positives or missed detections, necessitating time-series analysis[2]. This paper proposes a motorcycle traffic violation detection system combining YOLOv11-N(a lightweight variant of the YOLO family) for object detection and MobileViT-based time-series analysis.

The proposed system processes real-time footage from motorcycle-mounted blackbox cameras independently, without external infrastructure, to ana-

※ "This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government(MSIT) (IITP-2025-RS-2020-II201612, 30%)."
※ "This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2025-RS-2022-00156394, 40%) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation)"
※ "This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2018R1A6A1A03024003, 30%) and the Gyeongsangbuk-do RISE (Regional Innovation System & Education) project (Idea Start-up Valley unit).
◆ First Author : Kumoh National Institute of Technology, ingon4359@gmail.com, 학생회원
° Corresponding Author : Kumoh National Institute of Technology ,wdragon@kumoh.ac.kr, 종신회원
논문번호 : 202505-120-A-RU, Received May 22, 2025; Revised June 11, 2025; Accepted June 26, 2025

lyze violations during driving. The system recognizes key road elements such as traffic lights, stop lines, centerlines, and crosswalks to determine three types of violations: signal violations, centerline crossings, and crosswalk violation. Violation conditions are set based on the Road Traffic Act, utilizing traffic signal status, vehicle position, and movement trajectory.

YOLO rapidly detects objects, while MobileViT analyzes temporal visual changes across frames, reducing false positives caused by instantaneous detection failures or temporary scenarios. The modular system is based on ROS 2 and operates in real-time on embedded environments like Jetson Orin. Each function runs as an independent node optimized for embedded scenarios, with violation information stored in standard JSON and CSV formats for subsequent analysis or integration with cloud-based monitoring systems[4].

Real motorcycle blackbox video data was used for training and evaluation, enhancing detection generalizability across various environments including day/night and urban/rural conditions.

The paper is structured as follows: Section 2 provides an overview of the system and its components; Section 3 describes the ROS 2 environment, hardware setup, and data training processes; Section 4 details implementation of the violation detection algorithm using YOLO and MobileViT; and Section 5 presents experimental results and future tasks.

Ⅱ. Related Work

2.1 Motorcycle Violation Detection

Early traffic-violation studies rely on fixed roadside cameras; red-light or stop-line violations are detected with Faster R-CNN or YOLOv3[2]. Such static viewpoints miss violations occurring outside the field of view and are prone to occlusion. Motorcycle detection is even harder because of a narrow silhouette, sudden lane changes, and lane-splitting. Lim et al[3]. used stationary CCTV to analyse two-wheeler behaviour, but their system cannot capture rider-centric violations in a first-person view (e.g., crossing a centerline in dense traffic). RideSafe-400 and similar datasets focus on helmet use; they supply single frontal images only, contain no sequential context, and therefore cannot model time-dependent offences such as prolonged crosswalk occupancy. In-vehicle, first-person datasets for motorcycles remain scarce, creating a clear need for an onboard solution that continuously tracks the rider's trajectory without roadside infrastructure.

2.2 Lightweight Temporal Reasoning

Detecting violations that unfold over several seconds (e.g., lingering on a crosswalk or late signal passage) requires temporal reasoning. Heavy 3D-CNNs (approximately 6 – 15 GFLOPS for 224×224 clips) and ConvLSTM stacks incur hundreds of megabytes of parameters, exceeding the 10 – 15 W power and 30 fps latency budgets of edge devices such as Jetson Orin NX. Corsel et al[5]. combine YOLOv5 with a ConvLSTM head for tiny-object tracking, but their inference speed drops below 5 fps on an RTX 2080 and remains unreported on embedded hardware.

MobileViT[6] introduces Transformer self-attention at only ~300 MFLOPS and 5 MB of weights, enabling frame-wise features to be fused across time with mobile-level cost. When paired with extreme-lightweight detectors such as YOLOv5s or YOLOv11-N — each operating at <10 ms per frame on Jetson-class devices[7] — the full pipeline sustains >25 fps while processing 4 – 6-frame clips in a sliding window, making real-time, on-device temporal reasoning practical without additional training epochs or large memory overhead.

Despite these advances, published evaluations remain limited to static traffic scenes or car-mounted front-view datasets. No prior work, to our knowledge, benchmarks a MobileViT-YOLO hybrid on first-person motorcycle footage or validates generalisation on publicly released dashcam videos. Consequently, the effectiveness of embedded temporal models in highly dynamic, rider-centric scenarios is still largely unverified — an empirical gap addressed by the present study.

Ⅲ. System

3.1 System Overview

The system aims for real-time detection and analy-

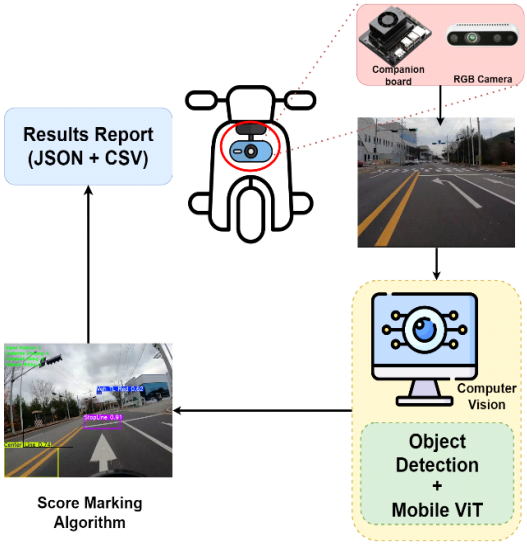


Fig. 1. Overall structure of the proposed system

sis of motorcycle traffic violations, divided into two key processes: acquiring driving information and evaluating violations. Fig. 1 illustrates the overall structure.

An RGB camera (Intel RealSense D435i) installed on motorcycles captures real-time video, sending it to a Jetson Orin NX embedded board, which performs object detection and violation analysis.

Upon receiving video data, YOLO detects key road elements such as traffic lights, stop lines, crosswalks, and centerlines. Vehicle and pedestrian traffic lights are classified separately to avoid detection confusion. Objects relevant to each violation type are detailed in Table 1.

While the YOLO-based detection method enables rapid identification of individual objects on a per-frame basis, it is limited in its ability to accurately determine violation scenarios that require temporal continuity[5]. To address this limitation, the proposed study employs MobileViT, a lightweight version of the Vision Transformer (ViT) known for its spatio-temporal feature analysis capabilities. MobileViT receives sequences of Regions of Interest (ROIs) extracted from the bounding boxes detected by YOLO across consecutive frames, and efficiently analyzes object state transitions and movement trajectories. This approach facilitates the detection of temporally-dependent violations such as signal violations and crosswalk violation[8].

Table 1. Object classes used for type of violation

Violation Type	Detected Classes
signal violation	Traffic lights (red, green, yellow, left-turn), Stop line, Pedestrian light
centerline crossing	Centerline
crosswalk violation	Crosswalk (horizontal, vertical)

In particular, the use of a computationally optimized MobileViT model ensures real-time performance and high-accuracy violation detection, even in embedded environments such as the Jetson Orin NX. The final violation results are recorded in both JSON and CSV formats, including timestamp, location, and type of each violation event. These records can be further utilized for downstream analysis, integration with web services, or cloud-based monitoring systems[7].

The system is built upon the ROS 2 framework and performs real-time data processing and violation determination through inter-node message communication. Each module is designed to operate independently, enhancing maintainability and scalability.

3.2 ROS 2-Based System Flow and Architecture

The system is built upon the publisher-subscriber architecture of ROS 2, with each function implemented as an independent node running in parallel. The */camera* node publishes real-time driving video, which is received by the *yolo_node* for object detection. The detection results are then published to the */yolo/detections* topic.

The outputs from object detection are processed in parallel by the *violation_detector* and *tracking_node*. The *violation_detector* handles frame-based violations that can be immediately determined, such as signal violations and stop line overruns. In contrast, the *tracking_node* detects cumulative violations over time, such as crosswalk violation and centerline crossing, based on the positional tracking of objects across consecutive frames.

Detected violations are forwarded to the debug_node, where they are visualized, logged in JSON

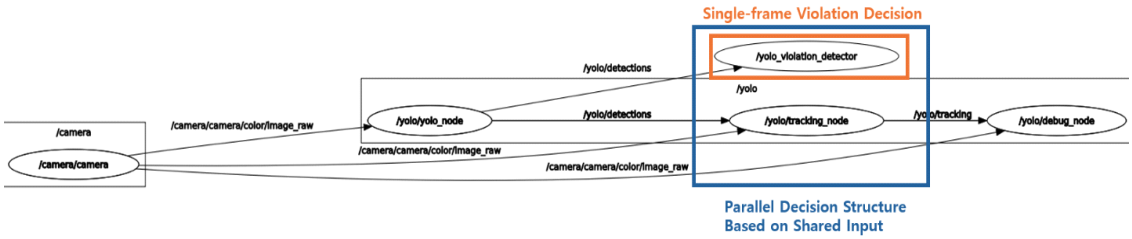


Fig. 2. ROS node and topic architecture used in the system

and CSV formats, or passed on for further processing. All nodes are interconnected via message topics, and the data flow is designed to maintain stable real-time processing. The complete system architecture in Fig. 2.

3.3 Dataset Construction and Training

A dataset for traffic violation detection is constructed using real motorcycle driving footage. A total of 40 minutes of blackbox video is divided into individual frames at 30 fps, and brightness adjustment-based augmentation is applied to improve robustness under varying lighting conditions. This process yields 14,392 image samples.

YOLOv11-N is used as the object detection model. Training runs for 100 epochs with a batch size of 8, using the training and validation sets. The dataset is split into training, validation, and test subsets in a 7:2:1 ratio, as summarized in Table 2.

The trained model achieves 91% precision, 88% recall, and 90.5% mAP@0.5. Most classes are detected reliably, but some misclassifications occur in visually similar background elements. Fig. 3 shows the class-wise confusion matrix, where most categories exceed 0.9 recall, except for traffic light yellow (0.50) and green (0.78), which show lower performance due to class imbalance and visual similarity.

Fig. 4 illustrates the training process All loss values, including box loss and classification loss, steadily

Table 2. Dataset composition

Subset	Number of Images
Training Set	10,074 (70.0%)
Validation Set	2,878 (20.0%)
Test Set	1,440 (10.0%)
Total	14,392 (100%)

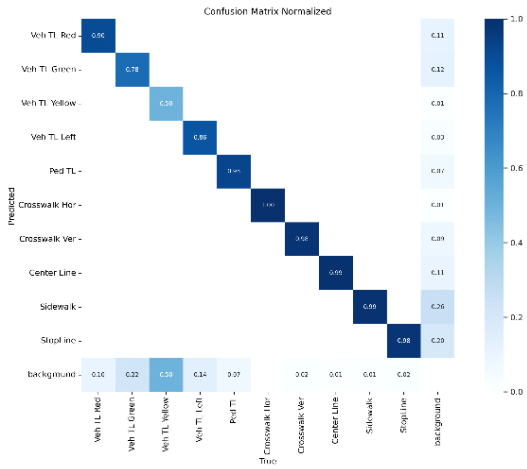


Fig. 3. Normalized confusion matrix

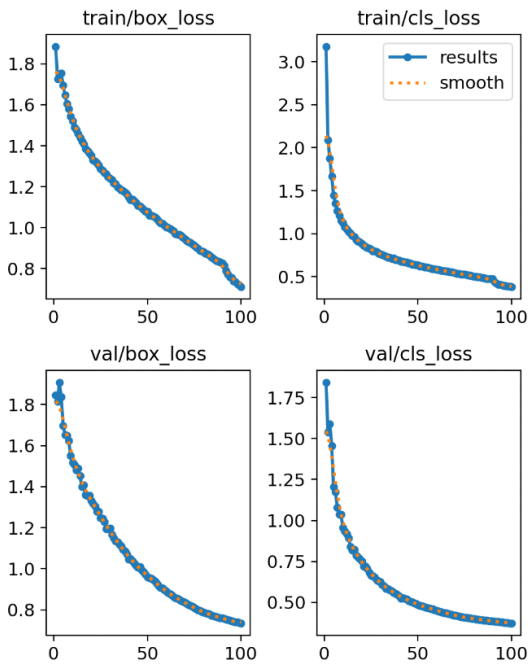


Fig. 4. Loss and mAP curves over epochs

decrease across 100 epochs. No significant overfitting is observed, as validation losses closely follow the training trends. The mAP@0.5 and mAP@0.5:0.95 metrics also show consistent improvement, confirming the model's stable convergence and learning effectiveness.

3.4 Algorithm Implementation

The violation determination algorithm of the proposed system is structured as shown in the pseudocode. The system first detects objects in each frame using YOLO and extracts the bounding box regions (Regions of Interest, ROIs) of the detected objects. These ROIs are organized into sequences of a fixed length and passed as input to the MobileViT model. MobileViT analyzes the changes in object states and movement trajectories across consecutive frames to determine whether a traffic violation has occurred. The final violation results are stored in both JSON and CSV formats.

The algorithm detects objects in each frame and performs time-series analysis using MobileViT based on the ROI information extracted from the detected objects. MobileViT continuously learns changes in object positions and states across frames to determine traffic violations by considering temporal context, such as the relative position changes between traffic lights and vehicles. When the computed violation probability exceeds a predefined confidence threshold, the system immediately records the violation. These records are stored in both JSON and CSV formats for subsequent processing.

Algorithm 1 violation detection algorithm

```
Require: VideoFrames, YOLO, MobileViT
Ensure: ViolationLog (JSON, CSV)
for each frame in VideoFrames do
    objects ← YOLO.detect(frame)
    ROIs ← ExtractROIs(objects)
    violations ← MobileViT.predict(ROIs sequence)
    for each violation in violations do
        if violation.confidence > threshold then
            logViolation(violation, timestamp, location)
        end if
    end for
end for
Save ViolationLog as JSON and CSV
```

3.4.1 signal violation detection

According to Article 5 of the Road Traffic Act, it is considered a violation if a vehicle passes beyond the stop line during a red signal. YOLO is used to detect vehicles, traffic lights, and stop lines, and the nearest traffic light relative to the vehicle is used as the reference for judgment. Even in cases of momentary detection failure or when the traffic light is outside the frame, MobileViT performs time-series analysis of the vehicle's position relative to the stop line and the signal state to determine whether a violation has occurred.

3.4.2 crosswalk violation detection

According to Article 27 of the Road Traffic Act, prolonged driving within a crosswalk zone is a violation. Temporary or vertical crossings are excluded. MobileViT determines violations by analyzing how long the vehicle occupies the crosswalk across frames.

3.4.3 centerline crossing detection

According to Article 13 of the Road Traffic Act, crossing over a yellow solid centerline into the opposite lane constitutes a violation. The system immediately determines a violation when a vehicle enters this restricted zone, and MobileViT ensures accurate detection even for brief incursions.

3.5 Application of MobileViT

While YOLO enables rapid object detection, its reliance on single-frame information limits its effectiveness in determining violations that require temporal context[5]. For instance, a red traffic light may be successfully detected immediately after a vehicle passes the stop line; however, if the stop line is not detected in that specific frame, the system may fail to identify the signal violation.

To address this limitation, MobileViT, a lightweight version of the Vision Transformer (ViT), is applied. MobileViT combines the local feature ex-

Fig. 5. Example of a red light being detected while the stop line is missed after vehicle passage

traction capabilities of CNNs with the spatiotemporal context modeling of Transformers to learn changes in object states and movement patterns across consecutive frames. When a fixed-length sequence of ROIs detected by YOLO is provided as input, MobileViT can make precise judgments by jointly considering vehicle movements and traffic signal changes[6].

To ensure real-time operation in embedded environments such as Jetson Orin, a pretrained MobileViT model is used without additional fine-tuning. The model operates with an input resolution of 256×256 and processes a fixed-length sequence of five frames. Each frame is cropped to include YOLO-detected ROIs before being passed into MobileViT. This setup minimizes computational overhead while enabling robust temporal reasoning in violation scenarios such as signal and crosswalk violations. In particular, crosswalk violations are only determined when a vehicle remains in the crosswalk area for a certain duration, and MobileViT helps reduce false positives by capturing sustained behavioral patterns while ignoring transient events

Ⅳ. Experiments

4.1 Hardware and System Environment

The proposed system was implemented in an embedded environment for real-time traffic violation detection. The hardware specifications are summarized in the table below. An Intel RealSense D435i camera is mounted on the vehicle to capture driving video, and real-time analysis is performed on a Jetson Orin NX embedded board. The system operates on Ubuntu 22.04 with ROS 2 Humble, based on JetPack 6.

Table 3. Hardware specifications

Device	Name
Companion board	Jetson Orin NX
RGB camera	Intel RealSense D435i
Operating System	Ubuntu 22.04 + JetPack 6
Middleware	ROS 2 Humble

4.2 Detection Performance Analysis

To evaluate detection performance, two types of data sources were used: self-collected blackbox footage and publicly available online dashcam videos. The self-collected data consists of 75 minutes of driving footage, including 30 minutes in urban areas and 45 minutes in rural areas, recorded under both day and night conditions. Due to the inherent limitations in collecting real-world violation data such as legal constraints and safety concerns, publicly available dashcam videos were additionally used to supplement the evaluation and validate the generalizability of the system[10].

Traffic violation detection was conducted using the YOLO and MobileViT-based system, and the detection performance was assessed by comparing the

Fig. 6. Example of violation detection result on real driving footage

number of actual violations with the number of violations detected by the system.

Table 4 summarizes the number of actual traffic violations identified during the dataset construction process. These values represent the ground truth occurrences used to evaluate the detection performance.

Table 5 presents the comparative detection results for both the self-collected and external datasets, using the standalone YOLO and the combined YOLO+MobileViT models.

The YOLO+MobileViT model outperformed the standalone YOLO model across both dataset categories. For the self-collected footage, YOLO+MobileViT reached an average detection accuracy of 94.1% versus 78.8% for YOLO. On publicly available dashcam videos, the combined model achieved 91.0% accuracy, compared with 74.6% for YOLO. Overall, across all 185 ground-truth violations, YOLO+MobileViT attained an average accuracy of 93.5%, reducing missed detections by more than 50% relative to YOLO. Accordingly, it outperformed the standalone YOLO model in all major

Table 4. Ground truth violation counts

violation type	signal violation	centerline crossing	crosswalk Driving
Self-collected Footage	24	69	25
External Videos	19	27	21
Total	**43**	**96**	**46**

Table 5. Detection results by environment and model type

Dataset Type	Violation Type	Ground Truth	YOLO	YOLO+MobileViT
Day (Urban)	signal violation	8	6	7
	crosswalk violation	5	3	5
	centerline crossing	23	20	22
Day (Rural)	signal violation	6	5	5
	crosswalk violation	3	2	3
	centerline crossing	20	17	19
Night (Urban)	signal violation	6	4	5
	crosswalk violation	9	7	8
	centerline crossing	14	11	13
Night (Rural)	signal violation	4	3	5
	crosswalk violation	8	6	8
	centerline crossing	12	9	11
External Video	signal violation	19	14	17
	Crosswalk Violation	21	16	20
	centerline crossing	27	20	24
Total	–	**185**	**143**	**172**

Table 6. Comparison of Precision, Recall, and F1-score

Model	Precision (%)	Recall (%)	F1-score (%)
YOLO-only	89.2	85.3	87.2
YOLO + MobileViT	92.4	90.0	91.2

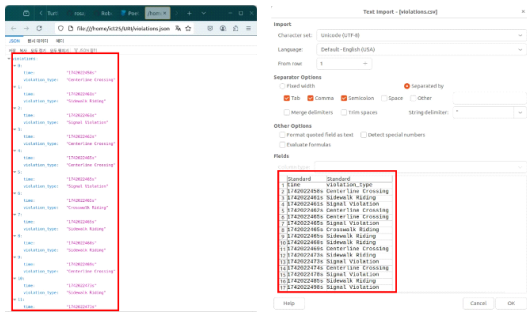


Fig. 7. Example of saved detection results: JSON format (left) and CSV format (right)

detection metrics, including precision, recall, and F1-score, as summarized in Table 6.

As shown in the figure below, the detection results are stored in both JSON and CSV formats, including information such as the timestamp, violation type, and location of each event.

The proposed system demonstrated robust detection performance under various conditions. The incorporation of temporal information through MobileViT significantly contributed to reducing missed detections compared to the YOLO-only model.

V. Conclusion

This paper proposed a real-time traffic violation detection system for motorcycles, based on object detection and temporal analysis. The system combines YOLO-based detection of key road elements with MobileViT-based time-series analysis to identify violations such as stop line overruns, centerline crossings, and crosswalk violation.

Each functional component of the system is implemented as a separate ROS 2 node to ensure real-time operation, and the architecture is designed to run reliably on embedded platforms such as Jetson Orin NX. The violation detection results are stored in JSON and CSV formats, enabling subsequent analysis and integration with real-time services.

Through experiments on both self-collected black-box footage and publicly available dashcam videos, the proposed system achieved an average detection accuracy of 93.5% (94.1% on self-collected data and 91.0% on external footage). The inclusion of

MobileViT for temporal context analysis reduced missed detections by more than 50% compared with single-frame YOLO inference and maintained robust performance across day-night and urban-rural scenarios.

Future work will focus on expanding the system to support a wider variety of road types and integrating it with online visualization platforms to enhance applicability in broader operational scenarios.

References

[1] *Korea Transportation Safety Authority, Half of delivery motorcycles violate traffic laws* (2021), Retrieved Jul., 27, 2021, from https://m.news.nate.com/view/20210727n14656

[2] *Maeil Business Newspaper, Two-way unmanned surveillance cameras only achieve partial enforcement of motorcycle violations* (2025), Retrieved Feb., 20, 2025, from https://www.mk.co.kr/en/society/11245535.

[3] J. B. Lim, K. M. Kim, and J. T. Park, "An empirical study for the introduction of two-wheeled vehicle control technology and behavior analysis," in *Proc. KICS Int. Conf. Commun.* 2010 (KICS ICC 2010), pp. 145-148, Busan, Korea, Nov. 2020. (https://doi.org/10.5762/KAIS.2022.23.6.343)

[4] A. N. Hakim, *YOLO for traffic violence detection* (2024), Retrieved Feb. 5, 2025, from https://github.com/abdurrahmannurhakim/AI-Trafic-Violence

[5] C. W. Corsel, M. van Lier, L. Kampmeijer, N. Boehrer, and E. M. Bakker, "Exploiting temporal context for tiny object detection," in *Proc. IEEE/CVF WACV Wkshps.)*, pp. 361-370, Waikoloa, HI, USA, Jan. 2023. (https://doi.org/10.1109/WACVW58289.2023.00013)

[6] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," in *Proc. ICLR,* Apr. 2022. (https://arxiv.org/abs/2110.02178)

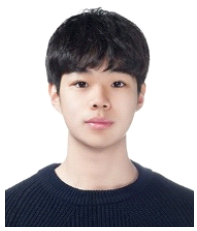
[7] Y. Wang, S. Xu, P. Wang, K. Li, Z. Song, Q. Zheng, Y. Li, and Q. He, "Lightweight vehicle detection based on improved YOLOv5s," *Sensors*, vol. 24, no. 4, Art. 1182, Feb. 2024. (https://doi.org/10.3390/s24041182)

[8] C. Yu, X. Shi, W. Luo, J. Feng, Z. Zheng, A. Yorozu, Y. Hu, and J. Guo, "MLG-YOLO: A model for real-time accurate detection and classification of winter jujube objects," *Plant Phenomics*, vol. 6, 2024. (https://doi.org/10.34133/plantphenomics.0258)

[9] Korea Ministry of Government Legislation, *Road Traffic Act (current version)* (2024), Retrieved Jul. 2, 2025, from https://www.law.go.kr

[10] Korea Transportation Safety Authority and Smart *Mobility AI Lab, Motorcycle Accident Driving Dataset* (2023), Retrieved Jul. 2, 2025, from https://huggingface.co/datasets/smart-dashcam/motorcycle-accident-driving-datasets.

In Gon Kim



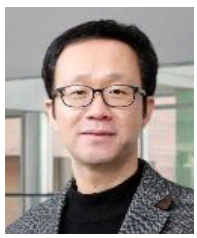
Feb. 2024 : B.S. degree, Korea National University of Transportation

Mar. 2024~Current : M.S. student, Kumoh National Institute of Technology

<Research Interests> Computer Vision, Large Language Models, AI Agents, Autonomous Systems

[ORCID:0009-0007-0027-9557]

Soo Young Shin



Feb. 1999 : B.S. degree, Seoul University

Feb. 2001 : M.S. degree, Seoul University

Mar. 2010~Current : Professor Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do, South Korea

<Research Interests> Wireless communications, Deep learning, Machine learning, Autonomous driving

[ORCID:0000-0002-2526-2395]