

A Design and Implementation of Real-time Automatic Traffic Violation Detection and Reporting System based on Edge Devices

Minsoo Sun*, Jeonghak Nam**, Hyunseok Song***, Eunseo Jang****,
Yoonji Jung*****, Sejong Lee†, Ho-Young Kwak‡

*Student, Dept. of Computer Engineering, Jeju National University, Jeju, Korea

**Student, Dept. of Computer Engineering, Sejong University, Seoul, Korea

***Student, Dept. of Artificial Intelligence, Daegu University, Kyeongsan, Korea

****Student, Dept. of Computer Science & Engineering, Kangnam University, Gyeonggi, Korea

*****Student, Dept. of Geoinformatic Engineering, Inha University, Incheon, Korea

†Assistant Professor, School of Computer Science and Engineering, Yeungnam University, Gyeongsan, Korea

‡Professor, Dept. of Computer Engineering, Jeju National University, Jeju, Korea

[Abstract]

In this paper, we propose an edge device-based automated traffic law violation detection and easy reporting system. This system automatically detects traffic law violations such as signal violations, reckless driving, not wearing a motorcycle helmet, and vehicles driving with their tail lights off at night in a road traffic environment. It collects images in real time from edge devices and analyzes the images using a high-performance deep learning model (YOLOv8) and a lane detection model (UFLDV2) to detect various traffic law violations. In particular, to reduce false detection rates, vehicles driving without taillights are detected through multi-frame continuous detection, while helmet violations on two-wheeled vehicles are processed immediately in a single frame. Additionally, the system integrates vehicle license plate recognition and an automated reporting procedure, allowing for easy reporting of traffic law violations. The proposed system demonstrated an accuracy of 83.3% and a recall of 71.4% in detecting traffic violations. Future work will focus on validation in diverse environments and model optimization to enhance commercialization potential.

▶ **Key words:** Traffic Violation Detection, Real-Time Object Detection, Automated Reporting System, Edge Computing, Yolov8

-
- First Author: Minsoo Sun, Corresponding Author: Sejong Lee
 - *Minsoo Sun (tjsalstn33@gmail.com), Dept. of Computer Engineering, Jeju National University
 - **Jeonghak Nam (njh110024@gmail.com), Dept. of Computer Engineering, Sejong University
 - ***Hyunseok Song (ridersong0723@gmail.com), Dept. of Artificial Intelligence, Daegu University
 - ****Eunseo Jang (ccolornyong@gmail.com), Dept. of Computer Science & Engineering, Kangnam University
 - *****Yoonji Jung (kellyice@naver.com), Dept. of Geoinformatic Engineering, Inha University
 - †Sejong Lee (kingsaejong@yu.ac.kr), School of Computer Science and Engineering, Yeungnam University
 - ‡Ho-Young Kwak (kwak@jejunu.ac.kr), Dept. of Computer Engineering, Jeju National University
 - Received: 2025. 04. 01, Revised: 2025. 04. 27, Accepted: 2025. 05. 05.

[요 약]

본 논문에서는 엣지 디바이스 기반의 교통 법규 위반 자동 단속 및 간편 신고 시스템을 설계하고 구현한다. 이 시스템은 도로 교통 환경에서 신호 위반, 난폭 운전, 이륜차 안전모 미착용, 야간에 후미등을 끄고 주행하는 차량 등의 교통 법규 위반 행위를 자동으로 탐지한다. 엣지 디바이스에서 영상을 실시간으로 수집하여 고성능 딥러닝 모델(YOLOv8)과 차선 검출 모델(UFLDV2)을 활용하여 영상을 분석함으로써 다양한 교통 법규 위반 행위를 탐지한다. 특히 후미등을 끄고 주행하는 차량은 다중 프레임 연속 검출로 오탐지율을 낮추었으며, 이륜차 헬멧 미착용은 단일 프레임에서 즉시 위반 처리가 가능하도록 구현한다. 그리고 자동차 번호판 인식 및 자동 신고 절차를 연동하여 교통 법규 위반 차량을 간편하게 신고할 수 있도록 구현한다. 본 논문에서 제안하는 시스템은 83.3% 정밀도와 71.4% 재현율로 교통 법규 위반 행위를 탐지하는 결과를 보였다. 향후 다양한 환경에서 검증하고, 모델 경량화함으로써 상용화 가능성을 높일 수 있다.

▶주제어: 교통 법규 위반 탐지, 실시간 객체 탐지, 자동 보고 시스템, 엣지 컴퓨팅, YOLOv8

I. Introduction

국내 교통사고 발생 건수와 교통 법규 위반 사례는 매년 꾸준히 증가 추세를 보이고 있으며, 이에 따른 사회·경제적 비용 역시 매우 큰 비중을 차지하고 있다. 특히 신호 위반, 난폭 운전, 안전모 미착용 이륜차 운전 등의 위반 행위들은 도로 위 안전을 위협하는 주요 원인이다[1]. 그러나 도로 위 곳곳을 일일이 단속하기에는 경찰·행정기관 인력의 물리적 한계가 존재하고, 기존 무인단속 카메라나 블랙박스 제보 시스템은 그 설치 및 운영 범위가 제한적이어서 즉각적인 대응이 어려운 실정이다.

그림 1과 같이 사고가 발생한 뒤에야 교통 법규 미준수 문제가 드러나는 사례가 많으며, 단속 인프라 확충 및 공익 신고 활성화에도 불구하고 실제 위반 행위를 효율적으로 억제하기 위한 기술적·운영적 대안이 여전히 요구되고 있다[2]. 따라서 이러한 교통 법규 위반 사례를 실시간으로 자동 탐지하고, 신고할 수 있는 시스템이 필요하다.

본 논문에서는 교통 법규 위반 상황을 실시간으로 자동 검출하고, 신고 프로세스를 간소화하는 엣지 디바이스 기반 교통 법규 위반 자동 단속 및 신고 시스템을 제안한다. 제안 시스템은 블랙박스 혹은 라즈베리파이[3], Jetson[4] 등의 엣지 디바이스를 활용하여 도로 주행 차량 영상을 실시간으로 수집하고, 인공지능 모델을 적용하여 신호 위반, 급차선 변경, 헬멧 미착용 이륜차, 야간에 후미등을 끄고 주행하는 차량 등 다양한 교통 법규 위반 유형을 자동으로 탐지한다. 탐지 결과에 따라 운전자에게 경고 메시지를 제공함과 동시에, 위반 차량의 번호판 인식 및 위치 정보 파악을 통해 안전신문고 등 관련 플랫폼에 신고할 수 있도록 구현한다.

이 논문의 구성은 다음과 같다. 2장에서는 기존 교통 법규 위반 단속 방식 및 유사 시스템의 문제점을 분석한다. 3장에서는 제안 시스템의 전체 아키텍처와 엣지 디바이스 구성, AI 모델 설계 및 학습 과정을 설명한다. 4장에서는 제안하는 시스템 구현하고, 실제 도로 환경에서 실험한 결과와 성능을 평가한다. 마지막으로 5장에서는 결론 및 향후 연구 방향을 제시한다.

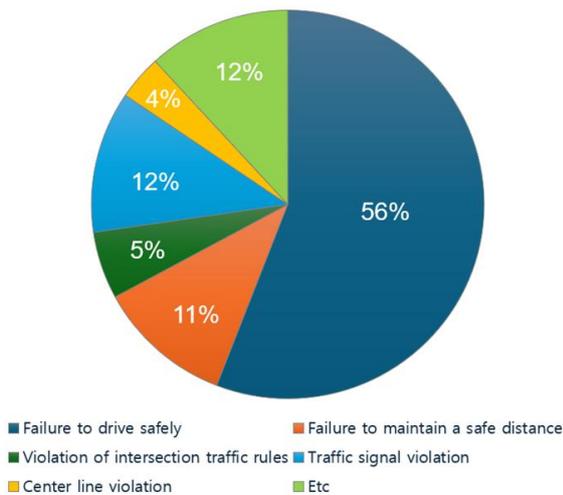


Fig. 1. Traffic accidents by violation of laws in 2023[1]

II. Related Works

1. Deep-Learning based lane recognition and detection

최근 자율주행 및 운전자 보조 시스템(ADAS)의 발전에 따라 딥러닝을 활용한 차선 인식 알고리즘 연구가 활발히 진행되고 있다. 기존에는 영상처리 기반의 엣지(Edge) 검출 및 허프 변환(Hough Transform) 같은 전통적인 기법이 주로 사용되었으나, 복잡한 도로 환경이나 차선 훼손, 조명 변화 등에 대응하기 위해서는 학습 기반 모델이 유리하다는 점이 부각되고 있다.

YOLOv2와 같은 딥러닝 세그멘테이션 모델을 통해 차선 영역을 추출하고, 후처리 단계(시점 변환, 클러스터링, 3차 곡선 피팅 등)를 적용함으로써 차선 인식 정확도와 처리 속도를 모두 향상시키는 방법이 제안되었다[6]. 이 방법은 BDD100K 등 대규모 오픈 데이터셋을 활용하였으며, 약 200ms 처리 속도 및 86%에 달하는 차선 인식 성공률을 보고하였다. 한편, 차선 검출 모델로 주로 거론되는 Ultra Fast Lane Detection(UFLD)[7], LaneNet[8] 등은 차선 포인트 자체를 직접 검출하는 방식으로 빠른 처리 속도와 간단한 파이프라인 구성을 장점으로 내세운다. 그러나 도메인 차이가 큰 실제 도로 환경에서는 검출 성능이 낮아지는 문제가 있어, 차선 영역을 분할(segmentation)한 후, 후처리 과정으로 보완하는 방식이 주목받고 있다[9][10].

2. Automatic detection of traffic violations

교통량이 많은 도시에서의 혼잡과 사고의 주요 원인은 급차선 변경, 과속 등과 같은 난폭 운전과 신호 위반, 이륜차 안전모 미착용, 불법 주정차 등의 안전 규정 위반이다. 이를 효율적으로 단속하기 위해 영상 기반 AI 모델을 도입하는 사례가 늘어나고 있다.

YOLOv8을 적용하여 CCTV 등에서 취득한 영상으로부터 차량 및 보행자를 실시간 검출하고, 속도 초과 등 특정 위반 행위를 판별하는 모델이 제시되었다[11]. COCO 데이터셋을 사용하여 학습한 뒤 실도로 환경에서 테스트한 결과, 높은 바운딩 박스 정밀도(Precision)는 확보했으나, 특정 교통 상황(이륜차량 다수, 조도 변화 등)에서 Recall이 낮아지는 현상을 보고하였다. 또한, 기존 YOLOv3-v5보다 개선된 YOLOv8 알고리즘이 직관적 인터페이스, 빠른 추론 속도 측면에서 우수하다는 점을 실험적으로 입증하였다. 다만 범용 COCO 데이터셋으로 학습하다 보니, 지역별·환경별 특수 위반 행위(인도 주행 이륜차, 야간 미등 주행 등)에 대해선 오인식이 발생하기 쉬우며, 실제 교

통 법규 위반 사례에 특화된 데이터셋 구축이 필요함을 강조하였다.

III. Design of Pickcarchu Application

1. Pickcarchu application architecture

본 논문에서는 엣지 디바이스 기반의 교통 법규 위반 자동 단속 및 간편 신고 시스템인 피카추(Pickcarchu) 애플리케이션을 제안한다. 블랙박스 영상을 이용한 자동 신고 기능을 제공하는 피카추 애플리케이션의 구조는 그림 2와 같다.

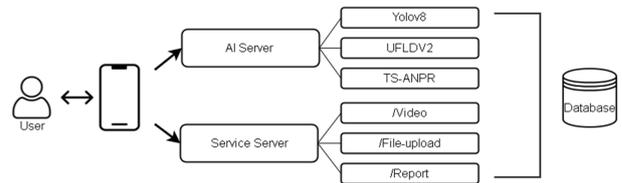


Fig. 2. Pickcarchu application architecture

그림 2에서 사용자의 블랙박스 영상에서 위반 차량이 포함된 장면을 AI 기술로 자동 탐지하고, 위반 행위, 시간, 날짜, 차량 번호를 전달한다. 사용자에게 해당 영상을 직접 확인할 수 있도록 하고, 신고를 원하는 경우 한 번의 클릭만으로 간편하게 안전신문고에 자동 신고할 수 있는 서비스를 지원한다.

1) Service Server

서비스 서버는 사용자의 요청을 처리하고 데이터를 관리하기 위해 세 가지 주요 엔드 포인트를 제공한다.

첫째, 영상 관련 엔드 포인트는 사용자가 동영상을 확인하고 위반 유형별로 검색할 수 있는 기능을 제공한다. video-stream 엔드 포인트를 이용하여 사용자의 블랙박스에서 추출된 영상을 스트리밍할 수 있도록 하며, detected-videos type은 사용자가 특정 위반 유형에 해당하는 동영상을 선별해서 볼 수 있도록 한다.

둘째, 데이터베이스를 담당하는 엔드 포인트는 AI 서버에서 전달하는 결과를 저장하고 관리하는 기능을 제공한다. files upload 엔드 포인트는 AI가 분석한 JSON 형식의 데이터를 데이터베이스에 저장하고, 분석 결과와 연관된 비디오 파일을 함께 저장하는 기능을 제공한다.

셋째, 신고 엔드 포인트는 사용자가 원하는 영상을 신고하고 그 결과를 조회할 수 있는 기능을 제공한다. report

apply 엔드 포인트는 사용자가 신고하는 기능을 담당하며, report results 엔드 포인트를 통해 사용자는 본인이 제출한 신고 내역을 조회할 수 있다.

2) AI Server

AI Server는 엣지 디바이스에서 1분 간격으로 전송되는 영상을 저장하고 분석하는 기능을 제공한다. 분석 단계에서는 헬멧 미착용, 과적, 신호 위반과 같은 위반 유형을 판별하기 위해 단일 프레임 검출 모델을 적용하고, 급차선 변경과 같은 연속 동작 판단이 필요한 경우에는 다중 프레임 검출 모델을 활용한다. 이를 통해 각 위반 유형을 violation 항목으로 저장하며, 위반이 감지된 시점 전후 10초간의 영상은 별도로 보존한다. 또한 해당 차량에 대한 번호판 인식을 수행하여 결과를 car_num 항목으로 기록한다. 분석을 마친 후에는 순번.json 파일과 관련 영상을 Service Server로 전송함으로써, 실시간 교통 위반 데이터를 효율적으로 공유하고 처리한다.

2. Scheme of Cutting in Maneuver Detection

이 기법은 도로에서 발생하는 급속 차선 변경 행위를 감지한다. 이 기법은 도로 영상 스트림을 입력으로 받아 차량의 위치와 차선을 분석하여 급 차선 변경 패턴을 식별한다. 이 기법은 Video stream, Object detection, Lane detection, Lane assignment, Rapid lane change detection의 5개 주요 모듈로 구성한다.

1) Object detection module

Object Detection 모듈은 영상 내에서 차량을 검출하는 기능을 제공한다. YOLOv8을 통하여 차량을 탐지한다. YOLOv8은 단일 단계 객체 탐지 모델로, 높은 정확도와 빠른 처리 속도를 제공한다. YOLOv8은 내부적으로 객체 추적 기능을 포함하고 있으며, 이를 활용하여 프레임 간 차량 추적을 수행한다. 본 논문에서는 속도와 정확도의 중요성을 고려하여 YOLOv8(large)을 사용한다. 각 차량의 중심 좌표는 바운딩 박스의 하단을 기반으로 계산된다.

2) Lane detection module

Lane Detection 모듈은 차량을 인지한 프레임의 차선을 검출하는 기능을 제공한다. UFLDV2(Ultra Fast Lane Detection v2)를 활용하여 차선을 감지한다. UFLDV2는 경량화된 구조로 실시간 차선 검출에 특화된 모델이며 CULane 데이터셋(133,235장 이미지)을 활용하여 차선 검출 모델을 학습한다. 이 모델은 차선의 경계선을 식별하

고, 차량 바닥의 y 좌표와 차선의 교차점을 계산한다. UFLDV2는 차선을 픽셀 단위로 제공하며, 이는 (x,y) (x,y) 좌표의 집합으로 나타난다. 예를 들어, 각 차선은 $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ 형태의 좌표 목록으로 주어진다.

$$rapid\ change = \begin{cases} 1 & lane_{start} \neq lane_{end} \text{ and } \Delta t < t_f \text{ and stable for } t_s \\ 0 & otherwise \end{cases}$$

$lane_{start}$: The lane at the starting point of the lane change

$lane_{end}$: The lane at the completion point of the lane change

Δt : The number of frames taken for the lane change (from start to end)

t_f : The maximum number of frames within which a lane change is considered a rapid change(in this paper, $t_f = 40$)

t_s : The minimum number of frames during which the lane change must remain stable after lane change(in this paper, $t_s = 8$)

(수식 1)

차량 바운딩 박스의 하단 y 좌표(C_y)에 해당하는 수평선 ($y=C_y$)을 그린다. 이 수평선과 각 차선의 픽셀들이 만나는 점이나 그러한 점이 없다면 가장 가까운 좌표를 선택한다. 이를 수식으로 표현하면 아래 수식 2와 같다.

$$i_{min} = \arg \min_i |y_i - C_y|, \quad i = 1, 2, \dots, n \quad (\text{수식 2})$$

수식 2에서 y_i 는 차선 픽셀의 y 좌표이고 C_y 는 차량 바운딩 박스의 하단 y 좌표를 의미한다. 이렇게 찾은 i_{min} 은 $(x_{i_{min}}, y_{i_{min}})$ 이 된다. 만약 $y_{i_{min}} = C_y$ 라면 교차점의 x 좌표는 $x_{i_{min}}$ 이다. 하지만, 그렇지 않은 경우 C_y 에 가까운 두 픽셀을 선택하여 선형 보간을 적용한다. $y_{i_{min}}$ 과 가장 가까운 픽셀을 사용하여 보간한다.

$$x_{intersect} = x_1 + \frac{(y_{target} - y_1)(x_2 - x_1)}{y_2 - y_1} \quad (\text{수식 3})$$

수식 3에서 y_{target} 은 C_y 를 의미한다. 각 차선에 대해 위 과정을 반복하여 모든 차선과의 교차점을 구한다.

3) Lane assignment module

Lane Assignment 과정에서는 탐지된 차량의 중심 좌표를 기반으로 수식 4와 같이 현재 차선을 판단한다.

$$lane = \begin{cases} lane_0 & \text{if } C_x < x_{intersect,1} \\ lane_i & \text{if } x_{intersect,i} \leq C_x < x_{intersect,i+1} \text{ for } i = 1, 2, \dots, k-1 \\ lane_k & \text{if } C_x \geq x_{intersect,k} \end{cases} \quad (\text{수식 4})$$

정렬된 교차점들은 차선의 경계를 나타내기 때문에 차량 바운딩 박스의 중심 x좌표(C_x)를 이 경계들과 비교하여

어떤 차선에 위치하는지 결정할 수 있다. 이 과정을 통해 현재 차량의 차선 위치를 파악할 수 있다.

4) Rapid lane change detection module

이 모듈은 여러 과정을 통해 얻은 결과로 급차선변경 차량을 찾아내는 과정이다. 차량의 차선 변경 패턴을 분석하여 급차선변경을 감지한다. 차선 변경이 짧은 시간 내에 발생하고, 변경 후 일정 프레임 동안 안정적으로 유지되면 이를 급차선변경으로 분류한다. 차선 변경의 시작과 끝이 같지 않으며, 수직 1과 같이 차선을 변경하는데 걸리는 총 시간인 Δt 가 기준인 t_b 보다 짧고, t_s 만큼의 시간 동안 변경된 차선을 유지하는 경우 급속 차선 변경으로 인지한다.

5) Model Evaluation module

급차선변경에 대한 대량 데이터 셋을 찾을 수 없어서 소량의 데이터로 평가를 진행한다. 표 1과 같이 10개의 테스트 영상(급차선변경 7개, 미포함 3개)를 사용하여 모델을 평가한다.

Table 1. Data set.

True Positive(TP)	5
True Negative(TN)	2
False Positive(FP)	1
False Negative(FN)	2

3. Scheme of traffic violation detection

이 기법은 스텔스 차량과 이륜차 헬멧 미착용을 동시에 탐지하여 교통 법규 위반 여부를 식별한다. 이 기법의 데이터셋 구성, 모델 아키텍처, AI 학습 방법, 그리고 탐지 로직에 대하여 설명한다.

1) Dataset

스텔스 차량(후미등을 끄고 주행하는 차량) 탐지와 이륜차 헬멧 미착용 탐지를 위해 두 가지 데이터셋을 구성한다. 스텔스 차량 탐지용 데이터셋은 총 10,000장의 차량 이미지로 구성한다. 이미지 중 약 2,000장은 스텔스 차량을 포함한 이미지이고, 나머지 8,000장은 일반 차량 이미지로 구성한다. 스텔스 차량은 실제 환경에서 데이터 수집이 매우 어렵고, 낮과 밤, 가로등 존재 유무 등 다양한 조명 조건을 포함한다. 또한, 승용차, SUV, 트럭 등 여러 차종과 다양한 도로 상태를 반영하여 데이터셋의 다양성을 확보한다.

이와 같은 클래스 간 불균형 문제를 완화하기 위해 Mosaic(1.0), MixUp(0.2), 회전($\pm 10^\circ$), 이동(0.1), 크기 조

정(0.5), 전단(2.0), 수직 및 수평 뒤집기(0.5) 등 여러 이미지 증강 기법을 적용하였다. 이를 통해 원본 데이터셋의 약 2배 규모로 학습용 이미지를 확보함으로써 모델의 일반화 성능을 향상한다.

이륜차 헬멧 미착용 탐지 데이터셋은 약 10,000장의 이미지로 구성하며, 헬멧 착용 및 미착용 상태를 모두 포함하고 있다. 해당 데이터는 낮과 밤, 실내 주차장 등 다양한 조명 조건과 혼잡한 도로, 한적한 도로 등 교통 상황의 변화를 반영하여 수집하였고, 일부는 착용 여부를 직관적으로 구분하기 어려운 경우도 포함되어 있어 실제 환경에 가까운 데이터로 구성한다. 이러한 데이터셋 또한 앞서 언급한 이미지 증강 기법들을 동일하게 적용하여 데이터 불균형 문제를 완화하였고, 보다 강건한 탐지 모델 학습이 가능하다.

두 데이터셋 모두 Roboflow[14] 플랫폼을 통해 수작업으로 라벨링 작업을 수행한다. 그림 3의 이미지는 스텔스 차량과 이륜차 헬멧 미착용 여부를 동시에 탐지할 수 있다.



Fig. 3. Data set sample

2) Model Architecture

스텔스 차량 및 이륜차 헬멧 미착용 탐지를 위해 YOLOv8x 모델을 활용한다. YOLOv8x는 고성능 객체 탐지 모델로, CSPDarknet53[12] 백본과 PANet[13] 구조를 기반으로 다양한 크기의 객체를 효율적으로 탐지할 수 있는 아키텍처를 갖추고 있다. YOLOv8x의 백본으로 사용된 CSPDarknet53은 CSPNet 구조를 기반으로 하여 연산 효율성을 확보하면서도 풍부한 특징 표현이 가능하다. 이와 함께, PANet 구조를 통해 다중 스케일의 피쳐 맵을 효과적으로 결합함으로써 작은 객체부터 큰 객체까지 다양한 크기의 객체에 대해 강인한 탐지 성능을 발휘할 수 있다. 또한, 해당 모델은 자동 앵커 박스 생성 기능을 내장하고 있어 주어진 데이터셋의 분포에 맞춰 최적의 앵커 크기를 자동으로 추정 및 적용하며, 탐지 성능과 시각화 효율

을 동시에 향상시킨다. 모델의 입력 해상도는 640×640으로 통일하여 처리하고, 출력 결과는 각 객체(스텔스 차량, 일반 차량, 헬멧 착용 이륜차, 헬멧 미착용 이륜차)에 대한 클래스 정보와 바운딩 박스 좌표를 포함한다.

3) Deep Learning methods

모델 학습에는 Adam 옵티마이저를 사용하며, 학습률(Learning Rate)은 0.001, 배치 크기(Batch Size)는 16, 총 학습 에폭(Epoch)은 100으로 설정한다. 데이터셋 구성 단계에서 적용된 다양한 이미지 증강 기법은 학습 시에도 함께 사용되어, 모델이 조명 변화, 도로 환경, 객체 크기 등의 다양한 상황에 적응할 수 있도록 한다. 이러한 학습 전략은 모델의 일반화 성능을 높이고, 실제 환경에서의 탐지 안정성을 확보한다.

4) Detection Logic

스텔스 차량과 이륜차 헬멧 미착용의 특성과 탐지 목적의 차이를 고려하여, 각각의 탐지 로직을 다음과 같이 차별화한다.

스텔스 차량의 경우, 단일 프레임 기준으로는 조도나 노이즈 등에 의해 오탐지(False Positive)가 발생할 가능성이 높다. 이를 방지하기 위해 30프레임으로 구성된 영상 내에서 동일 객체가 20프레임 이상 연속으로 스텔스 차량으로 검출될 경우에만 최종적으로 스텔스 차량으로 판단하였다. 이러한 기준은 신뢰성을 확보하고 실제 단속 시스템에서의 오경보를 최소화하는 데 목적이 있다. 반면, 이륜차의 헬멧 미착용은 단일 프레임에서도 비교적 명확하게 식별하며, 운전자의 안전과 직결된 요소이므로, 해당 객체가 프레임 내에서 헬멧 미착용 상태로 탐지되면 즉시 위반으로 간주하는 방식으로 처리한다.

이러한 탐지 기법의 차별화는 각 객체의 특성과 사회적 중요도를 반영한 설계이며, 실제 교통 환경에서의 운영 상황을 고려한 전략적 판단에 기반하고 있다.

IV. Implementation of Pickcarchu Application

피카추 애플리케이션 구현을 위한 소프트웨어 및 프레임워크는 표 2와 같다.

Table 2. Development Language & Framework

Usage	Framework
Frontend	Flutter
Backend	FastAPI
Database	MySQL
Deep learning	Python3, Pytorch

1. Main page

피카추 애플리케이션의 메인 화면은 그림 4과 같다.

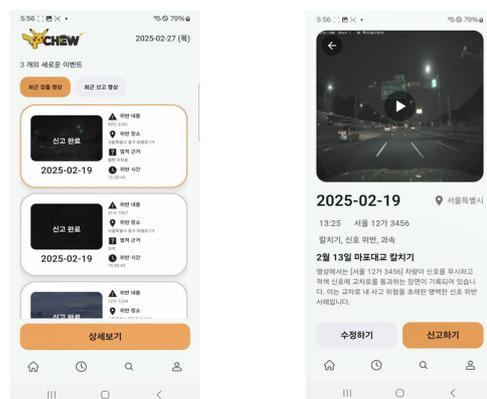


Fig. 4. Main Page

그림 4에서는 실시간으로 포착된 최근 검출 영상과 사용자가 신고한 과거 신고 영상을 쉽게 확인할 수 있다. 또한, 사용자의 ESG 점수를 배치하여 신고를 통해 기여한 ESG 실천 점수를 확인할 수 있고, 신고 건수를 시각화하여 이용 현황을 한눈에 파악할 수 있도록 구성한다.

2. Log Page

그림 4에서 [더보기] 및 하단 네비게이션 바의 backward 버튼을 클릭하면 그림 5의 과거 기록 화면이 나타난다.



(a) Past log

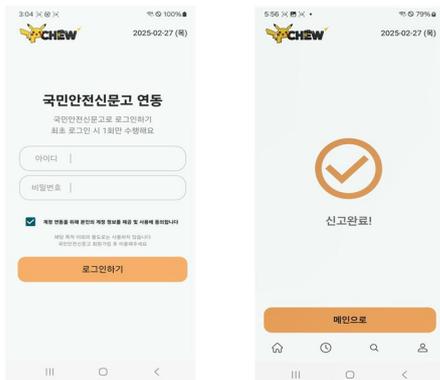
(b) Detail show

Fig. 5. Past log page

그림 5 (a)Past Log는 과거 기록을 볼 수 있는 화면으로 최근 검출 영상과 최근 신고 영상으로 분류하여 영상들을 확인할 수 있다. 로그는 각 영상의 이미지, 날짜, 위반 내용, 위반 장소, 법적 근거, 위반 시간을 나타내어 이용자가 쉽게 식별할 수 있는 기능을 제공하며, 신고한 영상일 경우 신고 완료 문구를 출력하여 용이성을 준다. 하단의 [상세보기] 버튼을 클릭하면 (b)Detail Show 화면으로 넘어가 신고 리포트의 제목 및 세부 내용을 확인할 수 있다. 해당 화면에서 이용자는 직접 수정 또는 신고를 할 수 있다.

3. Report

그림 6 (a)System Login은 앱 회원가입 후 최초 1회 진행되는 국민안전신문고 연동 과정이다. 이러한 연동 과정을 통해 국민안전신문고를 활용한 신고가 가능해진다.



(a) System login (b) Report page
Fig. 6. 4-cut image generated by prompt

그림 6 (a)System login에서 연동 과정 후, 그림 5 (b)Detail show 화면에서 [신고하기] 버튼을 클릭하면 검출 영상, 날짜, 위반 내용, 위반 장소, 법적 근거, 위반 시간, 신고 제목, 신고 내용 등 로그의 세부 내용과 이용자의 정보를 바탕으로 자동 신고를 진행한다. 이 과정에서 약간의 로딩 시간이 소요되고, 신고 완료 시 그림 6 (b)Report page 화면을 출력하여 신고가 완료되었음을 알린다.

V. Conclusions

본 논문에서는 엣지 디바이스 기반의 교통 법규 위반 차량 탐지 및 신고 간편화 시스템인 피카추 애플리케이션을 설계하고 구현하였다. 이 애플리케이션은 YOLOv8과 UFLDV2를 결합하여 실시간 급차선변경 검출 모듈을 구현했다. 이 모듈은 83.3%의 정확도와 71.4%의 재현율을 달성

함으로써 ADAS 및 자율주행 시스템에 실제 적용할 수 있는 상용화 가능성을 보여주었다. 향후 연구에서는 다양한 조명 및 날씨 등의 환경에서의 성능을 검증하고, 대규모의 데이터 셋을 활용한 검증으로 일반화하는 것이 필요하다.

ACKNOWLEDGEMENT

이 논문은 2025년도 K-디지털 트레이닝 사업의 일환으로 고용노동부, 직업능력심사평가원과 함께 SK텔레콤이 진행하는 FLY AI Challenger 프로그램의 지원을 받아 수행된 결과입니다.

REFERENCES

- [1] <https://www.police.go.kr/www/open/public/public0204.jsp>
- [2] <http://www.engjournal.co.kr/news/articleView.html?idxno=3001>
- [3] Raspberry Pi, <https://www.raspberrypi.com/>
- [4] Jetson, <https://www.nvidia.com/ko-kr/autonomous-machines/embedded-systems/>
- [5] Jocher, G., Chaurasia, A., Qiu, J., & Ultralytics. (2023). YOLO by Ultralytics. arXiv:2301.09313
- [6] J.Y. Seo, I. S. Kim, A. R. Kim and J. H. Park, "Study on Deep Learning-Based Lane Recognition and Lane Information Recognition Algorithms," KICS Fall Conference 2024, pp. 572-575 Nov. 2024
- [7] Z. Qin, H. Wang and X. Li, "Ultra Fast Structure-aware Deep Lane Detection," ECCV 2020, pp. 276-291 Aug. 2020 <https://doi.org/10.1007/978-3-030-58586-0>
- [8] Z. Wang, W. Ren and Q. Qiu, "LaneNet: Real-time lane detection networks for autonomous driving," arXiv preprint arXiv:1807.01726, July 2018 <https://doi.org/10.48550/arXiv.1807.01726>
- [9] N. J. Zakaria, M. I. Shapiai, R. A. Ghani, M. N. M. Yassin, M. Z. Ibrahim, N. Wahid, "Lane Detection in Autonomous Vehicles: A Systematic Review," IEEE Access, vol. 11, pp.3729-3765 Jan. 2023 <https://doi.org/10.1109/access.2023.3234442>
- [10] C. Han, Q. Zhao, S. Zhang, Y. Chen, Z. Zhang, J. Yuan, "YOLOPv2: Better, Faster, Stronger for Panoptic Driving Perception," arXiv preprint arXiv:2208.11434, Aug. 2022 <https://doi.org/10.48550/arXiv.2208.11434>
- [11] D. Tangamus, N. A. Thai, J. Shan, E. F. A. Sihotang, E. Irwansyah, "Method For Traffic Violation Detection Using Deep Learning," 2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS), pp. 16-21 Nov. 2023 <https://doi.org/10.1109/icimcis60089.2023>

10349009

[12] Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I-H. "CSPNet: A new backbone that can enhance learning capability of CNN," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15754-15763. June 2020 <https://doi.org/10.1109/cvprw50498.2020.00203>

[13] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J., "Path aggregation network for instance segmentation," arXiv:1803.01534 <https://doi.org/10.48550/arXiv.1803.01534>

[14] Roboflow, <https://roboflow.com/>



Sejong Lee received a B.S. degree in Computer Science and Engineering from Jeju National University, South Korea, in 2018. He received a Ph.D. degree in Computer Science and Engineering from Hanyang

University, South Korea, in 2024. Dr. Lee joined the faculty of College of Digital Convergence, School of Computer Science and Engineering at Yeungnam University, Korea, in 2025. His research interests include IoT security and Blockchain-based medical data-sharing systems, Artificial intelligence, Cloud platform.



Ho-Young Kwak received the B.S., M.S., and Ph.D. degrees in Computer Science from Hong-Ik University, Korea, in 1983, 1985, and 1990, respectively. Dr. Kwak joined the Department of Computer Engineering at Jeju

National University, Jeju, Korea, in 1990. He is currently a Professor in the Department of Computer Engineering, Jeju National University. He is interested in IT-Medical convergence, Companion Animal Healthcare systems, and Software systems.

Authors



Minsoo Sun will receive the B.S. degree in Computer Engineering from Jeju National University in 2026. His research interests include multimodal representation learning, Vision Transformers (ViT), and natural

language processing.



Jeonghak Nam will receive the B.S. degree in Computer Engineering from Sejong University, Seoul, in 2026. His research interests include time-series data, computer vision, and natural language processing, and

data analysis.



Hyunseok Song is expected to receive his bachelor's degree in artificial intelligence from Daegu University in 2025. His research interests include computer vision, deep learning, and scene text recognition.



Eunseo Jang received the B.S degree in Computer Science and Engineering from Kangnam University, Korea, in 2025. Her research interests include Bigdata Analysis and Deep Learning



Yoonji Jung will receive the B.S. degree in Geoinformatic Engineering from Inha University, Korea, in 2025. Her research interests include Spatial Data, Geographic Information Systems, Computer Programming,

Data Engineering and Deep Learning.