

YOLOv8n에서 2인 탑승 전동 킥보드 탐지 개선을 위한 새로운 라벨링 방법 제안

남 궁 동 혁¹ · 김 동 회^{2*}

¹강원대학교 IT대학 전기전자공학과 학사과정

^{2*}강원대학교 IT대학 전기전자공학과 교수

Proposal of New Labeling Method for Detection Improvement of Two-People Riding on E-Scooter in YOLOv8n

Dong-Hyeok NamGung¹ · Dong-Hoi Kim^{2*}

¹Undergraduate, Electrical and Electronic Engineering, IT College, Kangwon National University, Chuncheon, Korea

^{2*}Professor, Electrical and Electronic Engineering, IT College, Kangwon National University, Chuncheon, Korea

[요 약]

편의성과 경제성을 앞세워 개인형 이동장치인 전동 킥보드의 사용이 증가되고 있다. 전동 킥보드는 자전거, 오토바이와 같은 이륜 이동장치이나 바퀴의 크기가 훨씬 작으며, 서서 탑승하는 특수성으로 인해 다른 이륜 이동장치 대비 동일 사고 발생 시 위험이 더 크다. 이러한 이유로 전동 킥보드는 2인 이상의 탑승이 금지되어 있으나 단속이 현실적으로 어려운 상황이다. 이에 본 논문은 카메라를 이용하여 모니터링하는 화면에서 2인 이상이 탑승한 킥보드가 감지되면 알람을 발생시키는 YOLO 기반 시스템을 구성하는 과정에서 학습 타겟이 되는 라벨링의 방식을 새롭게 제안하며 YOLOv8n 기반 학습을 진행하여 제안한 방법을 적용한 후 학습 데이터 외 이미지로 실험 결과 정확도가 약 18% 상승하였음을 확인하였다.

[Abstract]

The use of an electric kickboard, which is a personal mobile device, is increasing due to its convenience and economic feasibility. An electric kickboard has a much smaller size of wheels than other two-wheeled vehicles such as bicycles and motorcycles, and due to the specificity of riding while standing, there is a greater risk than other two-wheeled vehicles in the same accident. For this reason, an electric kickboard is prohibited from boarding two or more people, but in reality, this practice is difficult to crack down on. Therefore, this paper proposes a new labeling method that becomes a learning target in the process of configuring a YOLO-based system that generates a notification when a kickboard with two or more people is detected on a monitoring screen using a camera. Through the tests with images, which are not included in the training dataset, we confirmed that the proposed method had an approximately 18% improvement in accuracy.

색인어 : YOLO, 객체 탐지, 라벨링, 전처리, 딥러닝

Keyword : YOLO, Object Detection, Labeling, Preprocessing, Deep-Learning

<http://dx.doi.org/10.9728/dcs.2023.24.7.1575>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 13 June 2023; Revised 26 June 2023

Accepted 27 June 2023

*Corresponding Author; Dong-Hoi Kim

Tel: +82-33-250-6349

E-mail: donghk@kangwon.ac.kr

I. 서론

전동 킥보드는 개인형 이동장치로서, 간편한 조작과 경제성과 더불어 대기오염, 교통 혼잡 등 도시 문제 개선에 기여할 수 있어 활발히 사용되는 교통수단이다. 이에 국내에서는 킥고잉을 선두로 지쿠터, 고고쟁 등의 공유 킥보드 서비스가 서비스 중이다[1].

전동 킥보드는 다른 개인형 이동장치와 달리 바퀴가 작으며, 서서 탑승하는 특수성을 가지기에 돌발상황 등으로 사고 발생 시 자전거나 오토바이와 같은 타 이륜 이동장치에 비해 운전자의 신변에 더 큰 위협으로 다가올 수 있다는 단점이 있다. 이러한 문제로 도로교통법은 개인형 이동장치에 대해 원동기 면허, 승차 정원 제한 등의 제약을 두고 이를 어길 시 범칙금이 부과된다. 그러나 공유 전동 킥보드는 접근 및 이용이 용이하며 상시 단속이 힘들기에 승차 정원을 초과한 탑승 및 헬멧 미착용 등의 도로교통법 위반 사례의 증가 및 사고의 발생 사례가 보고되었다[2].

이러한 부적절한 전동 킥보드 사용으로 인한 사고를 예방하고자 유동인구가 많은 시간대에 도로경찰이 단속을 하는 등의 노력을 기울이고 있으나 인력의 부족, 광범위한 이용 등으로 한계가 있다. 이를 도로의 CCTV(Closed-circuit Television)를 통해 실시간 모니터링하여 빠르게 2인 이상 탑승한 킥보드를 탐지하여 경찰의 출동 명령을 내릴 수 있는 시스템을 개발하기 위해 본 논문에서는 딥러닝 알고리즘 중 객체 탐지 알고리즘(Object Detection Algorithm)을 이용한다.

객체 탐지는 사람이 가장 많은 정보를 받아들이는 시각 정보를 컴퓨터가 분석하고 해석할 수 있도록 하는 연구 분야로, 영상 감시, 얼굴 인식 등 산업 전반에서 핵심 기술로 사용 및 연구되고 있다[3]. 객체 탐지 알고리즘 중 대표적으로 YOLO(You Only Look Once)가 있으며 YOLOv8은 2020년 발표된 YOLOv5의 개선된 알고리즘으로 다른 객체 탐지 알고리즘 대비 빠른 수행 속도와 정확도를 특징으로 한다.

본 논문에서는 카메라로 촬영하는 화면에서 1인 탑승 킥보드와 2인 탑승 킥보드를 구분하는 시스템을 제안한다. 전동 킥보드 1인 탑승과 2인 탑승에 대한 이미지 데이터를 객체 탐지 알고리즘 YOLOv8에 학습시켜 계산된 가중치 파일(모델)을 활용하여 탐지를 수행한다. 이를 CCTV를 통한 실시간 모니터링에 적용하여 안전수칙 위반으로 발생할 수 있는 사고를 사전에 방지할 수 있을 것으로 기대한다.

본 논문의 II장의 관련 연구에서는 객체 탐지 기술 중 YOLO에 관한 내용을 설명하고 III장에서는 객체 탐지 정확도를 높이기 위한 제안 방법을 기술한다. IV장에서는 제안 방법을 적용한 학습 모델의 성능 개선을 검증하고 V장에서 결론을 맺는다.

II. YOLO의 기존 라벨링 방법

2-1 본 연구에서 사용한 YOLOv8

YOLO[4]는 2015년 첫 번째 버전이 발표된 이후 각기 다른 학자들에 의해 개선된 버전들이 발표되고 현재도 연구가 계속되고 있는 대표적인 1-stage Detector 모델이다[5]. 가장 널리 활용되어 온 YOLO 버전은 Ultralytics의 YOLOv5이며, 본 논문에서는 Ultralytics가 2023년 1월 발표한 YOLOv8을 사용한다. YOLOv5는 이전 버전까지 사용하던 DarkNet Framework에서 PyTorch Framework로 사용자 친화적으로 바뀌면서 영상 처리 분야에서 활용되어 왔으며 YOLOv8은 YOLOv5의 네트워크 구조를 일부 변경하고 학습 및 추론에 필요한 파라미터의 수를 줄이며 정확도 및 속도의 개선을 가져오면서 API(application programming interface) 형식으로 배포하여 PIP(Pip Install Package)를 통해 설치가 가능해 개발 환경 구축 및 사용이 용이해 YOLOv5를 대체할 것으로 기대되고 있다.

2-2 기존의 라벨링 방법

본 장에서는 단순히 전동 킥보드를 감지하는 것이 아닌, 전동 킥보드에 탑승한 인원의 수에 따라 달리 구분하기 위해 적용한 시스템을 설명한다.

1명 탑승과 2명 탑승 구별을 위해 각각의 이미지를 각각 80장, 70장 가량을 인터넷을 통해 수집하고, 그림 1과 같이 Labellmg 프로그램을 통해 정답 데이터를 라벨링하여 2진 분류가 가능하도록 클래스를 지정하여 데이터셋을 준비한다[6].



그림 1. Labellmg를 이용한 데이터 라벨링
Fig. 1. Data labelling with Labellmg

YOLO는 MS COCO(Microsoft Common Objects in Context) 기준 80개의 클래스가 기본값으로 지정되어 있으므로 학습에 사용할 데이터셋에 맞는 클래스를 지정한 후 매 epoch에서 객체의 위치를 결정짓는 경계 박스의 위치를 예측하고 해당 객체의 종류를 구하기 위한 가중치를 조정하며 학습을 진행한다.

III. 라벨링 방법 제안

전동 킥보드에 탑승한 이미지는 사람의 눈으로는 직관적으로 탑승한 인원수를 구별할 수 있으나 컨볼루션을 통해 특징을 추출해 객체의 종류를 구별해 내는 객체 탐지 알고리즘에서는 사물의 형태, 윤곽 등 각종 특징을 자동으로 추출해 특징 맵(Feature Map)을 만들어 낸다.

본 논문에서 사용된 데이터셋의 경우 이미지 내 킥보드를 탑승한 사람의 크기가 충분히 크지 못할 경우 한 명만 탑승한 경우와 두 명이 탑승한 경우의 특징 맵이 그림 2와 같이 유사하게 나타날 수 있다. 추출된 특징 맵에서 색이 밝은 부분은 객체의 종류를 추측하는 신경망에서 민감하게 반응하는 중요한 특징으로 해석이 가능하다. 그림 2를 보면 한 명만 탑승한 이미지와 두 명이 탑승한 이미지의 특징 맵 중 일부가 매우 유사한 형태를 보임을 확인할 수 있고, 이는 각 클래스를 잘못 분류해낼 수 있는 위험이 있다고 해석 가능하다. 이를 해결하기 위해 YOLO 학습을 위해 사용된 데이터셋의 경계 박스 정보를 결정하는 방법을 그림 3과 같이 제안한다.

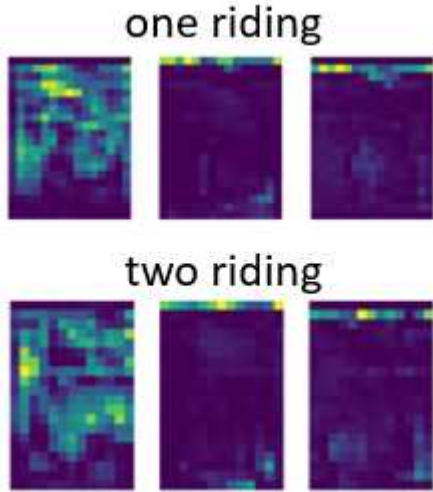


그림 2. 1명 탑승과 2명 탑승의 특징 맵 일부
 Fig. 2. Partial of feature map of one riding and two riding

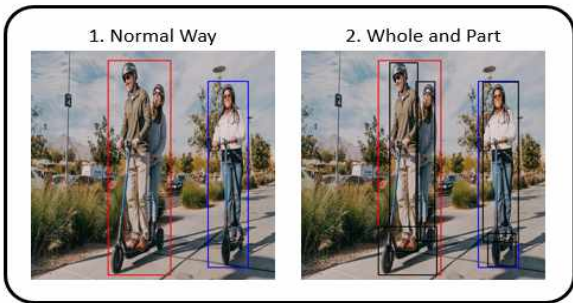


그림 3. 부분별 라벨링 제안
 Fig. 3. Suggestion of partial labeling



그림 4. 제안 방법의 라벨링
 Fig. 4. Labelling of proposed way

그림 3의 1번은 일반적인 객체 탐지 시 사용하는 라벨링 방법을 나타냈으며, 2번은 사람과 킥보드를 라벨링한 후, 전체를 감싸는 라벨링을 추가로 하여 사람 객체와 킥보드 객체 존재에 따른 검출이 가능하도록 기대할 수 있는 방법을 나타낸다. 킥보드에 탑승해 있는 인원의 수에 따른 분류가 목적이기에 사람을 탐지할 수 있도록 킥보드 외에 배경의 사람도 라벨링 하였으며, 킥보드 사진에서 공통적으로 보이는 킥보드 하부를 탐지할 수 있도록 하였다. YOLO의 탐지가 수행되면 학습을 통해 계산된 가중치를 바탕으로 그림 4와 같이 감지된 객체를 감싸는 경계 박스와 함께 객체의 종류와 그 확률이 표시된다.



그림 5. YOLO 탐지 예시
 Fig. 5. Sample of YOLO detection

IV. 실험 결과 및 성능 평가

4-1 실험 환경

표 1은 본 논문에서 YOLO 학습 및 실험을 위해 사용한 PC 환경의 사양이다. PyTorch의 병렬 연산을 위한 NVIDIA사의 CUDA(Compute Unified Device Architecture) 코어가 탑재된 GPU(Graphic Processing Unit)를 사용하였으며, 통합

개발 환경(IDE: Integrated Development Environment) VSCode를 사용하여 Python 코드를 작성 및 실행하였다.

표 1. 실험용 pc의 스펙 시트

Table 1. Spec sheet of simulation PC

CPU	AMD Ryzen 5600g
RAM	64GB
GPU	NVIDIA RTX 3070
VRAM	8GB
Pythone	3.10
PyTorch	1.21.1+CU116

YOLOv8은 PIP를 통해 설치한 라이브러리를 통해 Python 파일에서 YOLO 클래스를 import해 사용이 가능하다. 그림 6과 같이 Python 코드를 작성하여 기존 YOLO의 CLI(Command Line Interface)로만 사용이 가능하던 방식에 비해 용이하게 훈련 및 탐지 수행이 가능하다.

```
def train():
    model = YOLO("ymls/yolov8n_label2.yaml")
    model = YOLO("yolov8n.pt")

    model.train(data = "ymls/kb_2.yaml",
                epochs = 20,
                patience = 3,
                batch = 8,
                imgsz = 640, # img-size
                device = 0,
                workers = 8,
                name = "8n_label2_0517_adam",
                pretrained = False,
                optimizer = "Adam", # ['SGD', 'Adam', 'AdamW', 'RMSProp']
                verbose = True,
                cos_lr = True,
                lr0 = 0.001, # SGD = 1E-2, Adam = 1E-3
                lrf = 0.01, # final lr (lr0 * lrf)
                momentum = 0.937,
                weight_decay = 0.0005,
                warmup_epochs = 3.0,
                warmup_momentum = 0.8,
                warmup_bias_lr = 0.1,
                box = 7.5,
                cls = 0.5,
                dfl = 1.5,
                label_smoothing = 0.0,
                nbs = 64,
                val = True)
```

그림 6. YOLOv8 train 함수 작성 예시

Fig. 6. Example of defining YOLOv8 train function

학습에 사용할 이미지 데이터셋은 인터넷을 통해 수집하였으며, 전동킥보드에 2명 이상이 탑승한 이미지의 수가 약 70 장 정도로 그 수가 머신러닝 학습에서 높은 신뢰성을 얻기에는 충분히 많지 않아 데이터셋의 크기를 늘리기 위해 그림 7과 같이 데이터 증강기법[7]을 적용하여 전체 데이터셋의 크기를 32배 증가하여 인위적으로 데이터셋을 확보하여 모델 학습을 수행하였다.



그림 7. 데이터 증강을 적용한 이미지

Fig. 7. Image with data augmentation

그림 8은 본 논문에서 제안하는 방법을 검증하기 위해 실험을 진행한 순서도이다. 인터넷을 통해 수집한 데이터셋에 일반적인 방법으로 라벨링을 하여 YOLO 모델을 학습시킨 후 데이터셋에 포함되어 있지 않은 별개의 이미지를 이용하여 실제 성능을 확인한 후, 제안 방법을 적용하여 라벨링을 새롭게 한 후 YOLO 모델을 학습시켜 실제 성능을 2차로 확인하였다. 이때 검증에 사용한 이미지는 국내 기사에 첨부된 사진으로 좌측과 우측에 한 명이 탑승, 두 명이 탑승한 이미지가 포함되어 있다. 전체 이미지 크기 대비 객체의 크기가 작기에 카메라를 이용한 실제 시스템 구성 시 타 이미지 대비 실제 환경과 유사성이 더 높아 신뢰성이 높다고 판단되어 선택하였다.

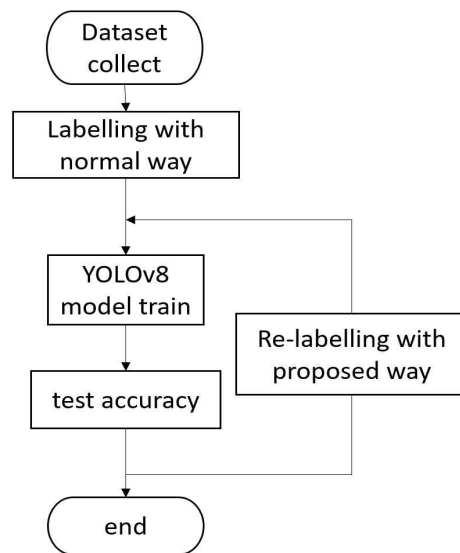


그림 8. 제안된 방법의 실험 순서도

Fig. 8. Simulation flowchart of proposed method

학습에 사용할 데이터셋을 준비한 후 객체 탐지로 구별할 객체의 라벨링 정보가 담긴 텍스트 파일을 함께 저장한 후 이에 대응하는 클래스 정보 등이 저장된 yaml 파일과 모델의 기반이 될 사전 학습된 모듈을 불러온 후, 이미지 크기, 배치 크기, epoch 수 등 하이퍼파라미터를 개인이 지정한 코드를 작성하고 해당 함수를 동작시킨다. 본 논문에서는 하이퍼파라미터로 20 epoch, Adam optimizer, learning rate 등을 설정하였다.

4-2 YOLO의 성능 지표

본 논문에서는 YOLOv8n의 모델에 킥보드 1인 탑승과 2인 탑승 이미지를 학습시켜 킥보드 2인 이상 탑승 탐지 모델을 추출한 후 킥보드 탐지를 수행 가능하도록 하였다. YOLO 모델 학습에는 탐지할 대상 이미지를 둘러싸는 경계 박스를

표시한 라벨링 정보를 정답 데이터로 활용하며 진행된다. 일반적인 YOLO 모델 학습에서 사용되는 라벨링 방법은 3장에서 서술한 라벨링 방법 중 1번의 Normal Way로 서로 다른 객체를 단순히 다른 클래스로 나누어 각 클래스의 특징을 추출한 후 사물의 위치를 나타내는 경계 박스와 사물의 종류를 예측한 후 정답 데이터와 비교하여 객체의 위치 정확도와 분류 정확도를 높이는 방향으로 가중치를 학습해 나간다. 객체의 위치는 모델이 예측한 경계 박스와 정답 데이터의 경계 박스의 영역이 겹치는 정도를 계산한다.

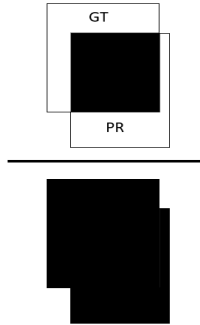


그림 9. IoU의 개념
Fig. 9. Concept of IoU

IoU(Intersection over Union)는 그림 9를 이용하여 식 (1)처럼 정의한다.

$$IoU = \frac{\text{겹치는 영역}}{\text{전체 영역}} \quad (1)$$

그림 9는 사물의 위치를 예측하기 위해 YOLO 모델이 예측한 경계 박스의 정확도를 계산하기 위해 사용하는 IoU 지표를 나타낸 것으로 식 (1)과 같이 정답 영역(Ground Truth)와 예측 영역(Prediction Area)의 겹치는 정도를 계산하여 IoU가 높을수록 사물의 위치를 정확하게 예측하는 모델이 된다.

IoU는 사물의 위치를 예측하는 척도이므로 해당 사물이 어떤 종류인지를 분류(classification)하는 것의 정확도 지표는 AP(Average Precision)를 이용한다. AP는 각 사물 클래스에 대한 정밀도(Precision)와 재현율(Recall)을 구해 계산 가능하며 정밀도와 재현율은 각각 식 (2)와 (3)를 통해 계산된다[8].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

여기에서, TP = 실제 Positive, 예측 Positive, FP = 실제 Negative, 예측 Positive, FN = 실제 Positive, 예측

Negative, TN = 실제 Negative, 예측 Negative 값이다.

클래스 A에 대해 정밀도는 A으로 예측했을 때 실제로 A일 확률이며 재현율은 전체 클래스 A인 사물 중 클래스 A로 예측한 비율로 이해할 수 있으며, 이를 통해 도출된 정밀도-재현율 그래프(P-R Curve)를 이용하여 각 클래스의 AP의 평균인 mAP(Mean Average Precision)를 구해 객체 탐지 모델의 성능 지표를 보인다[9].

4-3 기존 방법의 성능

표 2는 3장에서 서술한 라벨링 방법 중 일반적으로 YOLO 모델 학습에 사용되는 라벨링 방법을 사용하여 가장 가벼운 모델인 8n(nano)을 통해 20 epoch 학습을 진행한 후 검증 데이터셋을 이용하여 정확도를 AP로 측정하여 나타낸 표이다. AP는 예측한 경계 박스가 정답과 50% 이상 겹치는 경우 (IoU > 0.5)에 재현율과 정밀도를 비교하여 그래프로 표시한 후 그래프에 표시된 선 아래를 차지하는 공간을 계산하면 구할 수 있다.

표 2. 기존 방법으로 학습된 모델의 mAP
Table 2. mAP of model trained with normal way

one_person	more_than_two	mAP
0.911	0.938	0.925

학습 데이터에 대해서는 매우 높은 성능을 보여주지만 그림 10과 같이 4-1장에서 서술한 검증용 이미지에 대해서 예측을 실시했을 때 의도한 대로 탐지를 수행하는 경우도 있는 반면 두 명이 탑승한 이미지에 한 명이 탐승한 것으로 오 판단하는 등 학습 데이터에 과하게 적응하여 새로운 데이터에 대해 잘못된 결과를 도출해내는 과적합(OverFitting)[10] 문제가 발생하였다.

탐지 수행 결과에 나타난 해당 객체로 예측한 확률을 이용하여 모델의 성능을 계산한 결과를 표 2를 통해 확인 가능하다. 4개의 이미지 중 1개의 오탐지로 모델의 정확도가 69%로 계산되어 신뢰도가 떨어지는 것을 표 3을 통해 확인 가능하다.



그림 10. 기존 방법의 검증용 이미지 탐지 수행 결과
Fig. 10. Detection result of existing method

표 3. 기존 방법의 탐지 정확도

Table 3. Detection accuracy of existing method

	Answer	Predict	Answer Rate	Mean Answer Rate
a	one	one	86%	69%
b	two	two	92%	
c	one	one	92%	
d	two	one	6%	

4-4 제안 방법의 성능

본 논문에서 제안한 방법은 키보드를 탑승한 이미지 구역 내 사람과 키보드를 추가로 학습시키며 키보드에 탑승하지 않은 사람들도 라벨링하여 동일 데이터셋에서 같은 수의 파라미터를 더 다양한 클래스에 적용해야 하므로 기존 방법보다 약간의 오차를 허용하며 학습을 진행하게 되어 과적합 개선을 기대해 볼 수 있다.

표 4. 제안한 방법으로 학습된 모델의 mAP

Table 4. mAP of model trained with proposed method

one person	more than two	person	kickboard	mAP
0.93	0.927	0.891	0.897	0.911

표 4는 3장에서 제안한 방법을 적용하여 데이터셋에 라벨링을 한 후 20 epoch 학습한 모델의 mAP이다. 학습 데이터에 대해서는 클래스의 증가 및 키보드에 탑승하지 않은 사람 이미지에 대해서도 라벨링을 진행하였기에 학습해야 할 대상이 늘어나 일반적인 방법의 0.92보다 소폭 낮아진 0.91의 mAP 점수를 보이지만 학습 데이터셋 외 이미지로 테스트 시 사람과 키보드 여부를 탐지하며 일반적인 방법으로 라벨링하고 학습을 진행했던 모델로 확인하였을 때 그림 11에서 두 명이 탑승하였으나 한 명이 탑승하였다고 오탐지한 이미지(d)에 대해서도 두 명이 탑승했다고 탐지해 낸 모습을 그림 10에서 확인 가능하고 탐지해 낸 각 클래스에 대한 확률을 확인한 결과 기존 방법 대비 학습 데이터셋 외 이미지에 대한 정확도가 약 87%로 상승하였음을 표 5에서 확인 가능하다. 따라서 정확도가 기존방법에서 평균 69% (표 3)에서 제안한 방법에서 86.875 (표 5)로 약 18% 상승하였음을 확인하였다.



그림 11. 제안한 방법으로 학습 데이터셋 외 이미지 탐지 결과
Fig. 11. Result of detection with proposed way

표 5. 제안한 방법의 탐지 정확도

Table 5. Detection accuracy of proposed method

	Answer	Predict	Answer Rate	Mean Answer Rate
a	one	one	87%	86.875%
b	two	two	89.5%	
c	one	one	92%	
d	two	two	79%	

V. 결 론

본 논문에서는 YOLOv8n에 전동 키보드를 탑승한 사람의 이미지 데이터셋을 학습시키고 공통된 특징을 가진 서로 다른 객체 간의 분류 정확도를 높이기 위한 라벨링의 방식을 새롭게 제안하여 기존 방법과의 성능을 비교하였고 YOLOv8n 기반 학습을 진행하여 제안한 방법을 적용한 후 학습 데이터 외 이미지로 실험 결과 정확도가 약 18% 상승하였음을 확인하였다. YOLOv8은 기존의 YOLO 대비 개선된 알고리즘으로 학습 및 검증 데이터셋에 대해 향상된 정확도와 빠른 연산 속도를 보이거나 본 논문에서 활용한 데이터셋은 각 클래스 간의 유사성으로 인해 일반적인 경계 박스 설정으로는 과적합의 위험이 생긴다. 본 논문에서 제안한 객체 내 공통된 특징들을 세분화하여 인식시키는 방법은 본 논문에서 사용된 데이터셋과 같이 학습 데이터의 양이 충분히 확보되지 않은 상태에서 공통된 특징이 존재하는 서로 다른 객체를 구별해 내는 시스템을 개발하는 데 도움이 될 것으로 기대한다.

참고문헌

- [1] S. J. Kim, G. J. Lee, S. Choo, and S. H. Kim, "Study on Shared E-Scooter Usage Characteristics and Influencing Factors," *The Journal of the Korea Institute of Intelligent Transport Systems*, Vol. 20, No. 1, pp. 40-53, February 2021. <https://doi.org/10.12815/kits.2021.20.1.40>
- [2] Y. Choi, C. Ki, and H. Lee, "Deep Learning System that Determination Illegal Activities on Scooter," *The 2nd Korea Artificial Intelligence Conference*, pp. 281-282, 2021.
- [3] Y. H. Lee and Y. Kim, "Comparison of CNN and YOLO for Object Detection," *Journal of the Semiconductor & Display Technology*, Vol. 19, No. 1, pp. 85-92, March 2020.
- [4] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," in *Proceedings of the IEEE*, Vol. 111, pp. 257-276, January 2023. <https://doi.org/10.1109/JPROC.2023.3238524>
- [5] P. Jiang and D. Ergu, "A Review of YOLO Algorithm Developments," *Procedia Computer Science*, Vol. 199, pp. 1066-1073, February 2022. <https://doi.org/10.1088/1742-6596/1544/1/012033>

- [6] S. Y. Lee, S. Fu, and J. I. Park, "Detection of Helmet on Electric Scooter," *Korea Broadcasting Media Engineering Association Autumn Conference*, pp. 201-204, 2021.
- [7] M. G. Kim and H. J. Hae, "Data Augmentation Techniques for Deep Learning-Based Medical Image Analysis," *Journal of the Korean Society of Radiology*, Vol. 81, No. 6, pp. 1290-1304, November 2020. <https://doi.org/10.3348/jksr.2020.0158>
- [8] R. S. Shankar, L. V. Srinivas, P. Neelima, and G. Mahesh, "A Framework to Enhance Object Detection Performance by using YOLO Algorithm," *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, Erode, India, pp. 1591-1600, April 2022. <https://doi.org/10.1109/ICSCDS53736.2022.9760859>
- [9] Y. Jeong, S. H. Kim, and D. H. Kim, "Implementation of Detection and Classification System for Sudden pest using Object Detection Algorithm," *Journal of the Korean Institute of Communications and Information Sciences*, Vol. 81, No. 6, pp. 1290-1304, June 2023. <https://doi.org/10.7840/kics.2023.48.6.704>
- [10] X. Ying, "An Overview of Overfitting and Its Solutions," *Journal of Physics: Conference Series*, Vol. 1168, No. 2, 2019. <https://doi.org/10.1088/1742-6596/1168/2/022022>

남궁동혁
(Dong-Hyeok NamGung)



2016년~현 재: 강원대학교 IT대학
전기전자공학과 재학

※ 관심분야 : 인공지능(AI), 딥러닝, 객체 탐지

김동회(Dong-Hoi Kim)



2005년 : 고려대학교 전파공학과
(공학박사)

1989년 1월~1997년 1월: 삼성전자 전임연구원
2000년 8월~2005년 8월: 한국전자통신연구원 선임연구원
2006년 3월~현 재: 강원대학교 IT대학 전기전자공학과 교수
2020년 6월~2022년 8월: 강원대학교 정보화본부장 등
※ 관심분야 : 인공지능(AI), 무선 네트워크 및 사물인터넷 (IoT) 등