

A Study on the Impact of Noise on YOLO-based Object Detection in Autonomous Driving Environments

Ra Yeong Kim*, Hyun-Jong Cha**, Ah Reum Kang***

*Student, Dept. of Cyber Security, Pai Chai University, Daejeon, Korea

**Professor, Dept. of Software Engineering, Pai Chai University, Daejeon, Korea

***Professor, Dept. of Cyber Security, Pai Chai University, Daejeon, Korea

[Abstract]

Noise caused by adverse weather conditions in data collected during autonomous driving can lead to object recognition errors, potentially resulting in critical accidents. While this risk is widely acknowledged, there is a lack of research that quantitatively and systematically analyzes it. Therefore, this study aims to examine and quantify the extent to which noise affects object detection in autonomous driving environments. To this end, we utilized the YOLO v5 model trained on unprocessed datasets. The test data were divided into noise ratios of 0% (Original), 20%, 40%, 60%, and 80%, and the detection results were evaluated by constructing a Confusion Matrix. Experimental results show that as the noise ratio increases, the True Positive (TP) rate decreases, and the F1-score also significantly drops across all noise levels, specifically from 0.69 to 0.47, 0.29, 0.18, and 0.14. These findings are expected to contribute to enhancing the stability of autonomous driving technology. Future research will focus on collecting real datasets that include naturally occurring noise and developing more effective noise removal techniques.

▶ **Key words:** YOLO, Deep Learning, Object Detection, Data Quality, Computer Vision, Noise

[요약]

자동차의 자율주행 시 수집되는 데이터에 악천후 등의 원인에서 발생하는 노이즈는 객체 인식의 오류를 초래하여 자율주행에 각종 사고 등 치명적인 결과를 낳을 가능성이 크다. 이러한 위험성은 널리 인식되고 있지만, 이를 정량화하고 체계적으로 분석한 연구는 부족한 실정이다. 따라서 본 연구에서는 자율주행 환경에서 노이즈가 적용되는 비율에 따라 객체를 인식하는 것에 어느 정도의 영향을 미치는지 살펴보고 수치화하고자 한다. 이를 위해 가공을 하지 않은 데이터셋으로 학습한 YOLO v5 모델을 사용한다. 이를 통해 노이즈 비율을 0%(Original), 20%, 40%, 60%, 80%로 나눈 Test Data를 detection 하여 결과를 확인하고 Confusion Matrix를 구성하여 실험 결과를 평가했다. 실험 결과, 노이즈의 비율이 높아질수록 TP(True Positive)의 비율이 낮아지는 것을 확인할 수 있었으며, F1-score 또한 0.69, 0.47, 0.29, 0.18, 0.14로 적은 노이즈 비율뿐 아니라 모든 비율에서 수치가 큰 폭으로 떨어진다는 것을 알 수 있다. 이를 통해 자율주행 기술의 안정성을 높일 수 있는 기반이 될 것이라 기대되며 향후에는 자연적으로 발생한 노이즈가 포함된 실제 데이터셋을 수집하고 발생한 노이즈를 더욱 효과적으로 제거할 수 있는 연구를 진행할 예정이다.

▶ **주제어:** YOLO, 딥러닝, 객체 탐지, 데이터 품질, 컴퓨터 비전, 노이즈

- First Author: Ra Yeong Kim, Corresponding Author: Ah Reum Kang
- *Ra Yeong Kim (yoooong_@naver.com), Dept. of Cyber Security, Pai Chai University
- **Hyun-Jong Cha (hjcha@pcu.ac.kr), Dept. of Software Engineering, Pai Chai University
- ***Ah Reum Kang (armk@pcu.ac.kr), Dept. of Cyber Security, Pai Chai University
- Received: 2024. 08. 01, Revised: 2024. 09. 19, Accepted: 2024. 09. 25.

I. Introduction

자율주행은 운전자가 직접 조작하지 않아도 센서와 카메라 등을 이용하여 주행 환경을 인식해 위험을 판단하고 스스로 운전하는 기술이다[1]. 자율주행의 기본적인 원리는 인지, 판단, 제어 단계로 나눌 수 있다. 인지 단계는 여러 종류의 센서들을 통해 주행 중 차량, 보행자, 차선 등의 데이터를 수집하는 단계이며 자율주행 시 가장 중요한 요소이다[2]. 이때 사용하게 되는 센서는 대표적으로 레이더, 라이다, 카메라가 있다. 카메라는 차량 주변의 환경을 촬영하고 이를 기반으로 주행 환경을 판단하는 데 사용된다. 하지만 강한 비나 안개, 눈과 같은 악천후 조건 등에서 오는 신호의 흡수 또는 반사 강한 태양광을 받는 환경에 있는 경우 카메라에서 얻은 데이터의 왜곡이 일어날 가능성이 크다.

따라서 본 연구는 다양한 상황에서 발생할 수 있는 노이즈가 자율주행을 위한 객체 인식에 어느 정도의 영향을 미치는지 알아보려 한다. 실험을 위해 어떤 전처리도 하지 않은 Train data를 사용하여 YOLO 모델을 학습하고 각각 0%(Original), 20%, 40%, 60%, 80%의 노이즈를 합성한 Test data로 객체 탐지를 진행했다. 이후 객체 탐지 결과로 IoU, Confusion Matrix를 구성하여 노이즈가 미치는 영향을 수치화하고 최종적으로 실험 결과를 정리한다.

본 논문의 구성으로 2장에서는 객체 탐지에 사용되는 YOLO 모델의 개념과 각 버전의 주요 특징에 대해 기술하고 본 연구와 관련된 선행 연구를 살펴본다. 3장에서는 전반적인 실험 구조를 소개하고 실험에 사용된 데이터셋, 데이터 전처리 및 실험 환경을 구체적으로 서술한다. 4장은 실험 환경을 바탕으로 생성한 YOLO 모델과 노이즈를 추가한 데이터셋으로 객체 탐지를 진행하고 노이즈 비율별 탐지 결과를 도출하고 분석 및 평가한다. 마지막으로 5장에서는 연구 결과를 종합하여 정리하고, 향후 연구 방향을 제시한다.

II. Preliminaries

1. Related works

1.1 YOLO

YOLO는 2015년 Joseph Redmon 외[3]의 논문 "You Only Look Once: Unified, Real-Time Object Detection"에서 처음으로 소개되었다. YOLO는 객체 탐지를 이미지 픽셀에서 바운딩 박스의 좌표와 클래스 확률까지 하나

의 회귀 문제로 재구성하기 때문에 탐지하려는 객체를 한번만 학습하면 모든 객체의 위치와 클래스를 예측할 수 있다. 또한 기존에 객체 탐지를 위해 많이 사용되었던 R-CNN과 DPM 보다 월등히 빠른 실시간 처리 속도를 보인다.

본 연구에서는 탐지 정확도가 높고 초당 최대 140프레임의 빠른 탐지 속도를 보이는 YOLO v5 버전을 채택하여 실험했다. YOLO v5의 경우, 이전 버전인 YOLO v4보다 가중치 파일의 크기가 90% 정도 낮아 실시간으로 객체를 탐지하는 것에 적합하다[4]. 또한 YOLO v5 이후에 발표된 버전인 YOLO v6, v7, v8의 경우 연구와 적용 사례가 활발하게 이루어지지 않는 실정이다[5].

YOLO는 다양한 버전이 있지만 모델의 특성에 따라 적합한 버전의 차이가 있는데 Table 1.은 버전별 적합한 모델의 특성에 관한 표이다. Joon-Yong Kim[5]의 YOLO의 버전별 특징을 살펴보면 YOLO v1과 v2는 탐지율이 중요한 모델에 적합하고 v3과 v4는 탐지 속도가 중요한 요인이 되는 모델에 적합하다고 기술했다. v5, v6에서는 탐지 속도가 빠르고 탐지율이 높아야 하는 모델에 적합하고 v7, v8은 적은 자원으로 객체 탐지를 진행하여 객체 추적 및 포즈 추정을 진행할 모델에 적합하다고 서술했다.

Table 1. Features of YOLO Version-Specific Suitable Models[5]

Version	Features
v1, v2	Accuracy-Critical Models
v3, v4	Speed-Critical Models
v5, v6	Models Needing a Balance Between Accuracy and Speed / Real-time Object Detection Model
v7, v8	Pose Estimation and Object Tracking Models

1.2 Previous Research

최근 비디오 및 이미지 왜곡에 따른 객체 탐지에 대한 연구가 꾸준히 진행되고 있다. 연구들은 주로 일반적인 객체 탐지의 향상과 영상 품질 및 왜곡에 따른 정확도나 인식률에 대해 발표되었다.

Jaewon Ahn 외[6]는 COCO 데이터셋으로 객체 탐지 정확도에 따른 수학적 모델링을 진행하여 각기 다른 이미지 해상도에 따른 객체 탐지율에 대해 분석했다. Sungick Kong 외[7]는 도로에서 직접 데이터를 수집하고 수집한 데이터를 바탕으로 YOLO 네트워크 구조를 활용하여 탐지 정확도를 높이기 위한 방법을 제안한다. Jisoo Tak과 Sang-Woong Lee[8]는 YOLO의 실시간 탐지를 포함해 영상의 전송과 처리 과정에서 발생하는 노이즈를 학습 데이터에 포함한 뒤 학습하여 노이즈가 발생한 이미지에서

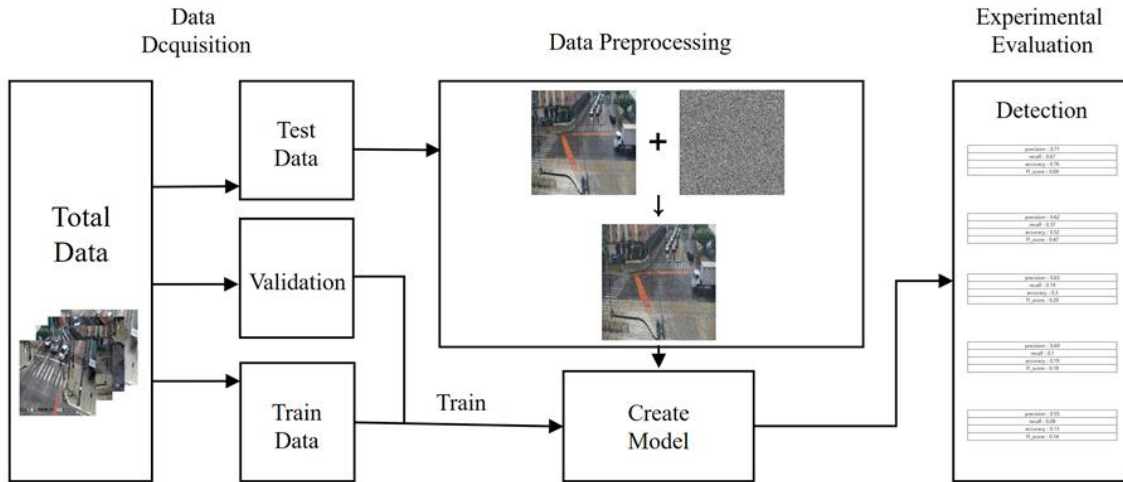


Fig. 1. Experimental Process

탐지 정확도를 높이기 위한 연구를 진행했다.

Junnan Li 외[9]는 PASCAL VOC와 MS-COCO 데이터셋을 사용해 객체 레이블과 바운딩 박스 좌표 및 모델 매개변수를 교대로 노이즈 수정 및 모델 훈련을 수행했다. Hew-soo Park 외[10]는 이미지 노이즈, 해상도 등의 다양한 영상 왜곡과 정도에 따르는 정확도에 대해 조사했다. Eunjin Jeon 외[11]는 이미지의 Blur 처리와 객체 탐지를 위해 OpenCV와 YOLO v3을 이용하여 Blur의 비율이 증가할수록 결괏값이 도출되지 않거나 객체 정확도가 하락했다고 서술했다. Nguyen Thi My Xuyen[12]은 어두운 조명 환경에서의 흐림과 노이즈를 최소화하기 위해 딥러닝 기반 CDAE(Convolutional Denoising Autoencoder) 모델을 사용하여 노이즈를 제거하여 객체 인식률을 크게 향상했다. Na-Hyoun Kim 외[13]는 야간에서의 이미지 객체 탐지 성능을 향상시키기 위한 Denoising Autoencoder를 통해 다양한 노이즈를 학습하고 이를 처리할 수 있는 프레임워크를 제안했다. Jieun Kim 외[14]는 자율주행 시 충돌 위험 인식 및 경고를 위하여 수집한 도로 주행 데이터로 주행 상황에서 발견할 수 있는 주요 객체의 검출 방법 및 결과를 정리했다. Kana Kim과 Hakil Kim[15]는 신뢰성 있는 데이터 정제 프레임워크를 사용하여 이상 데이터를 제거함으로써 모델의 인식 성능을 개선하는 전략을 소개했다. Choi, Yoonjo 외[16]는 효율적으로 고품질의 데이터를 구축하기 위해 자율주행 AI 학습 데이터 어노테이션 자동화 프로그램을 개발했다. Seongho Son과 Changsun Ahn[17]은 악천후에 취약한 라이더에 발생하는 노이즈를 줄이기 위해 특이값 분해(Singular Value Decomposition)를 활용하는 방안을 제시했다. Taresh Sarvest Sharan 외[18]는 DnCNN의 경우

Residual Learning과 Batch Normalization을 이용하여 학습 속도를 높이고 노이즈 제거 성능을 향상시킬 수 있다고 서술했다.

기존의 연구에서는 노이즈가 포함된 데이터를 학습 데이터에 추가하거나 데이터에 발생한 노이즈를 제거했다 [8][12]. 하지만 본 연구에서는 데이터를 변형시키지 않은 Original 데이터로 모델을 학습하고 Test Set에 노이즈를 비율별로 추가하여 detection을 진행하여 실험했다.

III. The Proposed Scheme

1. System Overview

1.1 Experimental Process

Fig. 1은 제안 시스템의 구조를 보여준다. 연구에 필요한 데이터셋은 Alhub, 대전공공데이터, Kaggle, Roboflow 등에서 교통상황과 관련된 데이터 위주로 수집한다. 하지만 Roboflow를 제외한 데이터셋은 라벨링이 충분이 되어 있지 않아 Roboflow에서 수집한 데이터셋을 중심으로 사용하고 데이터셋의 Train Data, Test Data, Validation을 8:1:1의 비율로 나누어 실험한다.

모델 생성을 위해 Train Data와 Validation Data를 사용하여 YOLO v5 모델을 학습하고 검증하며, Python의 OpenCV 라이브러리로 전처리 과정을 거친 Test Data로 객체 탐지율과 탐지 개수를 측정한다. YOLO v5 모델의 하이퍼파라미터와 데이터 전처리 과정에 대한 자세한 설명은 1.2절과 1.3절에 서술한다. 본 실험에서 모델 학습을 위한 데이터는 전처리 과정을 거치지 않은 Original Data를 사용한다. 이는 비교적 높은 품질로 학습되어 탑재된 기존 자율주행

모델과 비슷한 환경을 조성하기 위함이다. 이후 생성된 모델을 사용하여 노이즈가 추가된 Test Data를 Detection 한 후 도출된 결과를 바탕으로 IoU(Intersection Over Union)를 계산하고 Confusion Matrix를 구성하며 실험 결과를 정리한다.

1.2 Experiment Environment

본 연구를 진행하기 위한 실험 환경은 Table 2와 같다. 노이즈를 추가하고 해상도를 Resizing 하기 위해 사용한 Python은 3.10.9 버전을 사용한다. 또한 Python 코드에 영상 데이터를 불러오기 위한 OpenCV 라이브러리는 4.7.0.72 버전을 적용하며, 모델을 생성하는 과정에 YOLO v5 버전을 통해 모델을 학습하여 진행했다. YOLO v5의 하이퍼파라미터는 epochs 200, optimizer SGD(Stochastic Gradient Descent), batch size 16으로 조정하여 실험했다.

Table 2. System Environment

Item	Value
OS	Ubuntu 22.04.4 LTS
CPU	13th Gen Intel(R) Core(TM) i7-13700KF
GPU	NVIDIA GeForce RTX 4090
Python Version	Python 3.10.9
OpenCV Version	4.7.0.72
YOLO Version	YOLO v5

1.3 Data Preprocessing

Test Data에 적용한 가우시안 노이즈는 주로 자연환경에서 발생하는 갑작스러운 변동이나 오차를 모델링하는 데 사용되는 노이즈이다. 가우시안 노이즈는 가우시안 분포를 따르며 중앙값, 최빈값, 평균이 동일하다는 특성을 가진다. 가우시안 노이즈의 강도가 커질수록 데이터의 시각적 품질이 저하되고, 데이터를 처리하는 데 부정적인 영향을 입힌다. 이때, 합성 정도에 따라 노이즈의 비율이 결정된다. 본 실험에서 노이즈는 Python으로 Albumentations 라이브러리의 GaussNoise 클래스를 사용하여 Test Data에 적용하여 실험한다. 가우시안 노이즈의 분산은 아무 객체도 탐지되지 않는 구간인 5,000을 100%로 가정하고 0(Original), 1,000(20%), 2,000(40%), 3,000(60%), 4,000(80%)로 나누어 적용한다. Fig. 2.는 데이터셋에 노이즈를 비율별로 적용한 모습을 보여준다.

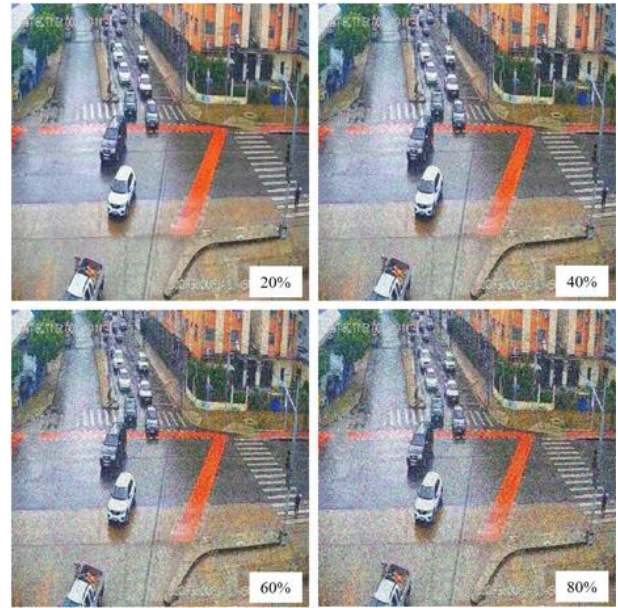


Fig. 2. Data Preprocessing

IV. Experiment

1. Model Construction

제안 시스템에서 객체 탐지에 사용할 학습모델을 생성하기 위해 Roboflow에서 수집한 데이터셋을 Ubuntu 환경에서 YOLO v5 버전으로 학습시켰다. 데이터셋은 도로 위를 주행하는 개체를 포함하며 학습 데이터의 개수는 Train Data 5,313개, Test Data 660개, Validation Data 660개로 총 6,633개이며 크기는 Labeling 된 Bounding Box의 좌표를 나타낸 txt 파일을 포함하여 총 603MB이다. 식별할 클래스는 'Bicycle', 'Bus', 'Car', 'Motorbike', 'Person' 총 5개로 나누어져 있으며, 학습 수행시간은 약 10,184초가 소요되었다. Table 3.은 Total Data의 클래스별 객체 수를 나타낸다.

Table 3. Number of Objects per Class

Class	Total	Train	Valid	Test
Bicycle	3,150	2,595	334	221
Bus	1,181	1,032	108	41
Car	47,231	35,253	5,294	6,684
Motorbike	15,883	11,868	1,955	2,060
Person	12,183	10,626	886	671

2. Experiment

	A	B	C	D	E	F
1	image	x_center	y_center	Width	Height	Class
2	aguanambi-1000_png.jpg.f.7179a0df8ad6448028bc5bc21dca41e.jpg	0.57578125	0.20078125	0.0421875	0.0671875	car
3	aguanambi-1000_png.jpg.f.7179a0df8ad6448028bc5bc21dca41e.jpg	0.6359375	0.27109375	0.015625	0.0671875	motorbike
4	aguanambi-1000_png.jpg.f.7179a0df8ad6448028bc5bc21dca41e.jpg	0.6359375	0.2046875	0.0453125	0.05625	car
5	aguanambi-1000_png.jpg.f.7179a0df8ad6448028bc5bc21dca41e.jpg	0.61484375	0.12890625	0.0328125	0.0546875	car
6	aguanambi-1000_png.jpg.f.7179a0df8ad6448028bc5bc21dca41e.jpg	0.62734375	0.10390625	0.0234375	0.0421875	car
7	aguanambi-1000_png.jpg.f.7179a0df8ad6448028bc5bc21dca41e.jpg	0.596875	0.1625	0.034375	0.05625	car
8	aguanambi-1000_png.jpg.f.7179a0df8ad6448028bc5bc21dca41e.jpg	0.6390625	0.08046875	0.025	0.0390625	car
9	aguanambi-1000_png.jpg.f.7179a0df8ad6448028bc5bc21dca41e.jpg	0.6409375	0.0546875	0.0234375	0.034375	car
10	aguanambi-1095_png.jpg.f.4d9f0370f1c09fb2a1d1666b155911e3.jpg	0.5921875	0.4921875	0.090625	0.1375	car
11	aguanambi-1095_png.jpg.f.4d9f0370f1c09fb2a1d1666b155911e3.jpg	0.52421875	0.334375	0.0515625	0.103125	car
12	aguanambi-1095_png.jpg.f.4d9f0370f1c09fb2a1d1666b155911e3.jpg	0.6125	0.265625	0.053125	0.096875	car
13	aguanambi-1095_png.jpg.f.4d9f0370f1c09fb2a1d1666b155911e3.jpg	0.61015625	0.13984375	0.0390625	0.0671875	car

Fig. 3. Bounding Box Coordinates Predicted by the Model

생성한 YOLO 모델과 전처리한 데이터셋을 사용하여 객체 탐지를 진행하고 예측한 바운딩 박스의 좌표를 x_center, y_center, Width, Height로 나누어 csv에 저장한다. Fig. 3.은 예측된 바운딩 박스 좌표와 예측 클래스를 csv에 정리한 결과이다.

예측 데이터를 바운딩 박스 좌표로 저장한 것은 모델에서 예측된 데이터와 실제 데이터를 매칭시키기 위해 IoU를 계산하기 위함이다. 본 실험에서 IoU의 threshold는 0.5로 설정되었다.

IoU는 객체 탐지에서 예측된 바운딩 박스와 실제 바운딩 박스 간의 겹침 정도를 측정하는 지표이다. 이는 수식 1과 같이 2개의 바운딩 박스의 교집합 영역에서 합집합 영역을 나누어 계산한다. IoU는 0에서 1 사이의 값을 가지며 1에 가까워질수록 높은 예측값을 가진다. 이 단계에서 예측값과 실제값을 매칭한 후 Confusion Matrix를 구성해 실험 결과를 분석한다.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (1)$$

3. Experimental Evaluation

실험 중 객체 탐지한 결과에서 한 이미지 내에 있는 여러 객체의 예측값과 실제값을 IoU를 이용하여 매칭했다. 이를 바탕으로 Fig. 4.와 같이 예측값과 실제값을 비교하여 성능을 평가할 수 있는 Confusion Matrix를 구성했다.

본 실험에서는 정확한 값을 예측한 TP(True Positive, 정답을 정답으로 예측)를 중심으로 탐지 개수를 분석했다. 실험 결과, 'Bicycle' 클래스에서 171개, 139개, 49개, 13개, 8개, 'Bus' 클래스에서 36개, 2개, 1개, 1개, 'Car' 클래스에서 5,990개, 4,023개, 2,260개, 1,281개, 711개로 감소했다. 또한 'Motorbike' 클래스에서 1,323개, 648개, 231개, 92개, 29개, 'Person' 클래스에서 543개, 424개, 222개, 106개, 41개로 거의 모든 클래스가 노이즈 비율이 증가할수록 TP 값이 점차 감소하는 모습을 보였다.

Original		Actual					
		None	Bicycle	Bus	Car	Motorbike	Person
Pred	None	0	16	8	405	328	136
	Bicycle	50	171	0	0	0	0
	Bus	5	0	36	0	0	0
	Car	690	0	0	5990	0	0
	Motorbike	740	0	0	0	1323	0
	Person	126	0	0	0	0	543
20%		Actual					
		None	Bicycle	Bus	Car	Motorbike	Person
Pred	None	0	44	3	136	97	153
	Bicycle	80	139	0	0	0	0
	Bus	36	0	2	0	0	0
	Car	2615	0	0	4023	0	0
	Motorbike	1404	0	0	0	648	0
	Person	239	0	0	0	0	424
40%		Actual					
		None	Bicycle	Bus	Car	Motorbike	Person
Pred	None	0	28	1	64	25	60
	Bicycle	150	49	0	0	0	0
	Bus	22	0	1	0	0	0
	Car	4103	0	0	2260	0	0
	Motorbike	1625	0	0	0	231	0
	Person	429	0	0	0	0	222
60%		Actual					
		None	Bicycle	Bus	Car	Motorbike	Person
Pred	None	0	17	0	38	7	26
	Bicycle	151	13	0	0	0	0
	Bus	15	0	1	0	0	0
	Car	4232	0	0	1281	0	0
	Motorbike	1422	0	0	0	92	0
	Person	464	0	0	0	0	106
80%		Actual					
		None	Bicycle	Bus	Car	Motorbike	Person
Pred	None	0	13	3	24	5	7
	Bicycle	118	8	0	0	0	0
	Bus	7	0	1	0	0	0
	Car	3510	0	0	711	0	0
	Motorbike	1089	0	0	0	29	0
	Person	401	0	0	0	0	41

Fig. 4. Confusion Matrix by Noise

Table 4. Confusion Matrix Estimation

Noise	Pre.	Recall	Acc.	F1-score
Original	0.71	0.67	0.76	0.69
20	0.62	0.37	0.52	0.47
40	0.63	0.19	0.3	0.29
60	0.69	0.1	0.19	0.18
80	0.55	0.08	0.13	0.14

Table 4.는 Confusion Matrix를 바탕으로 노이즈별 Precision, Recall, Accuracy, F1-score를 정리한 표이다. 데이터셋의 클래스 간 개수가 불균형할 때 Precision과 Recall만 고려하는 경우 정확한 평가가 어렵기 때문에 Precision과 Recall의 조화평균인 F1-score를 사용해 성능을 최종적으로 평가한다. Table 4.를 살펴보면 노이즈 비율이 20%에서 40%로 증가했을 때 Precision은 0.62에서 0.63으로 약간 증가했지만, Recall은 0.37에서 0.19로 절반 가까이 감소하여 F1-score가 0.47에서 0.29로 급격히 하락하는 결과를 확인할 수 있다. 전체적으로

F1-score는 노이즈 비율이 증가할수록 0.69, 0.47, 0.29, 0.18, 0.14로 꾸준히 감소하는 모습을 볼 수 있다.

V. Conclusions

본 논문에서는 자율주행 시 발생할 수 있는 노이즈가 주변 환경을 탐지하는 데 어떠한 영향을 미치는지 살펴보고 이를 수치화하기 위한 실험을 진행했다. 이를 위해 가공이 되지 않은 데이터셋으로 학습한 YOLO v5 모델을 사용하였고, detection은 노이즈 비율을 0(Original)%, 20%, 40%, 60%, 80%로 나누어 적용한 Test Set을 가지고 실험했다. 또한 탐지 결과를 바탕으로 IoU를 계산하여 예측 값과 실제값을 매칭하였으며 Confusion Matrix를 구성해 정답을 정답으로 예측한 TP 값을 중심으로 탐지 개수를 확인하고 Precision, Recall, Accuracy, F1-score를 계산하여 결과를 수치화했다.

실험 결과, 클래스 중 큰 비율을 차지하는 'Car'와 'Motorbike' 클래스를 살펴보면 'Car' 클래스에서 5,990개, 4,023개, 2,260개, 1,281개, 711개로 감소하였고 "Motorbike" 클래스에서는 1,323개, 648개, 231개, 92개, 29개로 감소한 것을 확인했다. F1-score 또한 노이즈 비율이 증가할수록 0.69, 0.47, 0.29, 0.18, 0.14로 평균적으로 약 32.58% 감소하는 것으로 파악됐다. 이는 육안으로 보았을 때 원본과 크게 다르지 않은 정도의 노이즈(20%)인 경우에도 객체 탐지에 치명적인 영향을 준다는 것을 보여준다.

실험을 통해 자율주행 시스템의 객체 탐지 성능이 수집되는 데이터 품질에도 영향을 크게 받을 수 있다는 것을 수치상으로 확인할 수 있었다. 자율주행에서 객체 탐지의 오차가 생기는 것은 각종 사고 등의 심각한 결과를 초래할 수 있기 때문에 수집되는 데이터의 품질은 일정 수준 보장되어야 함을 알 수 있다. 악천후와 같이 불가피하게 데이터의 변형이 생길 경우를 대비한 추가적인 기술 또한 지속적으로 개발되어야 할 것으로 보인다. 본 실험을 통해 향후 자율주행 기술의 안정성을 높일 수 있는 기반이 될 것으로 기대된다.

향후 연구에서는 자연적으로 발생한 노이즈가 포함된 실제 데이터셋을 수집하고 CNN 알고리즘을 사용해 수집되는 데이터의 노이즈를 제거하는 연구를 진행할 예정이다. 이를 통해 수집되는 데이터의 품질을 향상시켜 자율주행의 안정성을 높일 수 있을 것으로 기대된다.

ACKNOWLEDGEMENT

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program(IITP-2024-RS-2022-00156334) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation)

REFERENCES

- [1] Im. Y. Jung, "A Study on The Dangers and Their Countermeasures of Autonomous Vehicle," Vol. 20, no. 6, pp. 90-98, Jan. 2020, DOI: 10.5392/JKCA.2020.20.06.090.
- [2] Soonhong Kwon, and Jong-Hyoun Lee, "Self-driving Car Security Threats and Technology Trends," REVIEW OF KIISC, Vol. 30, No. 2, pp. 31-39, 2020.
- [3] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," arXiv:1506.02640, May 2016, DOI: 10.48550/arXiv.1506.02640
- [4] Yan, Bin, Pan Fan, Xiaoyan Lei, Zhijie Liu, and Fuzeng Yang, "A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5," Remote Sensing 13, no. 9: 1619, Apr. 2021, DOI: 10.3390/rs13091619
- [5] Joon-Yong Kim, "A Comparative Study on the Characteristics of Each Version of Object Detection Model YOLO," Proceedings of the Korean Society of Computer Information Conference, Vol. 31, No. 2, pp.75-78, Jeju, Korea, Jul. 2023.
- [6] Jaewon Ahn, DaeJin Kim, and Jun Kyun Choi, "A Study on Modeling the Object Detection Accuracy According to the Image Resolution," Proceedings of Symposium of the Korean Institute of communications and Information Sciences, pp.64-65, Youngpyoung, Korea, Feb. 2020.
- [7] Sungick Kong, Sang-Seol Lee, Sung-Joon Jang, and Byeongho Choi, "Korean Traffic-Sign Detection and Recognition using YOLO," Autumn Annual Conference of IEIE, pp.832-834, Incheon, Korea, Aug. 2018.
- [8] Jisoo Tak, and Sang-Woong Lee, "Noise-Augmented Object Detection: Improving Object Detection Accuracy in Noisy Image," Korean Institute of Next Generation Computing, pp.141-144, Kyungnam University, Korea, Jun. 2023.
- [9] Junnan Li, Caiming Xiong, Richard Socher, and Steven Hoi, "Towards Noise-resistant Object Detection with Noisy Annotations," arXiv preprint arXiv:2003.01285, Mar. 2020. DOI: 10.48550/arXiv.2003.01285

- [10] Hwe-soo Park, Pil-joong Kim, and Seong-Joo Lee, "Change of Recognition Rate in YOLO v4 According to Image Quality," Summer Annual Conference of IEIE, pp.1984-1986, Jeju, Korea, Aug. 2020.
- [11] EunJin Jeon, YoonSeok Cha, Seongyeol An, and ByungRae Cha, "Comparison and Analysis of YOLO Object Detection According to Image Quality Clarity," 2021 KICS Fall Conference, pp.513-515, Yeosu, Korea, Nov. 2021.
- [12] Nguten Thi My Xuyen, "Object Detection and Speed Prediction in Night Environment Using Deep Learning", Master's thesis, National Pukyong National University, Pusan, Feb. 2024.
- [13] Na-Hyoun Kim, Donghoon Lee, Junwoo Kim, Kee-Young Kwahk, and Namgyu Kim, "Multiple Denoising Scheme for Effective Nighttime Object Detection", Proceedings of KIIT Conference, pp.88-90, Jeju, Korea, Nov. 2021.
- [14] Jieun Kim, Chan-young Jung, Woong Hwang, Don June Lim, and Hyeong-ju Noh, "YOLO based Object detection for Autonomous driving and Collision Warning", proceedings of HCI Korea 2023, pp. 1,107-1,110, Gangwon, Korea, Feb. 2023.
- [15] Kana Kim, and Hakil Kim, "AI Model-Based Automated Data Cleaning for Reliable Autonomous Driving Image Datasets", Journal of Broadcast Engineering, Vol. 28, No. 3, pp.302-313, May. 2023.
- [16] Choi, Yoonjo, Lee, Gyeong Gwan, & Hong, Seunghwan, "A study on Automatic Algorithm of 3D Dynamic Object Annotation for Autonomous Driving AI Training Data Construction", KSCE 2023 convention, pp.672-673, Oct. 2023.
- [17] Seongho Son, and Changsun Ahn, "Denoising LiDAR Point Cloud Using Higher Order Singular Value Decomposition in Adverse Weather Condition", KSAE 2021 Annual Spring Conference, pp. 435-435, Pyeongchang, Korea, Jun. 2021.
- [18] Taresh Sarvest Sharan, Romel Bahttacharjee, Shiru Sharma, and Neeraj Sharma, "Evaluation of Deep Learning Methods (DnCNN and U-Net) for Denoising of Heart Auscultation Signals", 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA), pp. 151-155, Mumbai, India, Jul 2020. DOI:10.1109/CSCITA47329.2020.9137813.

Authors



Ra Yeong Kim received the B.S degree in Cyber Security from Pai Chai University, South Korea, in 2023. She is currently pursuing M.S. degree in the Department of Cyber Security at Pai Chai University.

Her current research interests include artificial intelligence, object detection, and cyber security.



Hyun-Jong Cha received the M.S. and Ph.D. degree in Computer science and Defense Acquisition Program from Kwangwoon University, South Korea, in 2008 and 2014. He is a professor in the Department of

Software Engineering at Pai Chai University in Daejeon, South Korea. His current research interests include information security, artificial intelligence, IoT, and blockchain.



Ah Reum Kang received the M.S. and Ph.D. degrees in information security from Korea University, South Korea, in 2012 and 2016. She is a professor in the Department of Information Security at Pai Chai University

in Daejeon, South Korea. Her current research interests include security, artificial intelligence, malware, medical data analysis, and online game security.