

# PilotNet 성능향상을 위한 SLAM과 YOLO 활용

고강현 \*김성민 유준호 박건우 임수현 황성수

한동대학교 전산전자공학부

e-mail : [21800017@handong.ac.kr](mailto:21800017@handong.ac.kr), [winwin27@handong.ac.kr](mailto:winwin27@handong.ac.kr), [handong18@handong.ac.kr](mailto:handong18@handong.ac.kr),  
[21901023@handong.ac.kr](mailto:21901023@handong.ac.kr), [22000612@handong.ac.kr](mailto:22000612@handong.ac.kr), [sshwang@handong.edu](mailto:sshwang@handong.edu)

## Using SLAM and YOLO to Improve Pilotnet Performance

Kang-Hyun Ko, \*Sung-Min Kim, Jun-Ho Yoo, Geon-Woo Park, Su-Hyun Rim,  
Sung-Soo Hwang

Computer Science and Electrical Engineering, Handong Global University

### Abstract

PilotNet, based on end-to-end deep learning, enables self-driving for a designated driving section without additional programming by imitating the driver's driving process.

However, since PilotNet cannot detect a pedestrian, an accident may occur if it encounters a pedestrian while driving. In addition, there is also a problem when the vehicle is out of a predetermined driving section because the current location is not recognized. In conclusion, the existing PilotNet lacks stability in driving.

Therefore, in this study, we propose a method to improve the safety and accuracy of autonomous driving by recognizing pedestrians and locating vehicles using YOLO and SLAM for the better performance of PilotNet.

With YOLO, the vehicle will make an emergency stop if a pedestrian approaches. With SLAM, based on the map stored in advance, the vehicle will make an emergency stop if the vehicle deviates from the designated driving section.

PilotNet with improved performance was executed with a model vehicle. As for the test results of this study, autonomous driving was possible with a success rate of 100% for emergency stops due to pedestrians approaching dangerous distances and 28.57% for emergency stops when they deviated from the

designated section. These results in the study suggest that it is possible to improve the performance of PilotNet by using YOLO and SLAM.

### I. 서론

Nvidia에서 개발한 PilotNet은 운전자의 주행을 모방하여 차량의 조향각을 실시간으로 예측하는 방법을 학습하는 자율주행을 위한 딥러닝 기반 알고리즘이다. 모델을 학습할 때 PilotNet은 주행 영상과 조향각을 입력으로 사용하고 경사 하강법과 같은 최적화 알고리즘을 사용하여 손실 함수를 최소화한다. 자세한 구조와 학습의 방식은 “End-to-end learning for lane keeping of self-driving cars”[1]의 내용을 따르고 있다. 그러나 PilotNet은 보행자를 인식하지 못하고 위치를 탐지하지 못하는 한계가 있다.

본 연구에서는 이와 같은 PilotNet의 문제점을 해결하기 위해 2가지 기술을 활용하여 자율주행 성능을 향상시키고자 한다.

1) YOLO(You Only Look Once)는 실시간 객체 탐지를 수행하는 CNN을 활용한 딥러닝 기반 알고리즘이다. 주행시 객체 검출을 통해 보행자를 인식하여 더 안전한 주행을 가능하게 한다.

2) ORB-SLAM3는 실시간 3D 지도 작성 및 카메라 위치 추정을 수행하는 컴퓨터 비전 기반 SLAM(Simultaneous Localization and Mapping) 알고리즘이다. 이미 생성한 지도를 바탕으로 주행시 차량의 현재 위치 추정을 통해 경로 이탈을 확인할 수 있으며 이는 안정적인 주행을 가능하게 한다.

## II. 본론

### 2.1 소프트웨어 구성

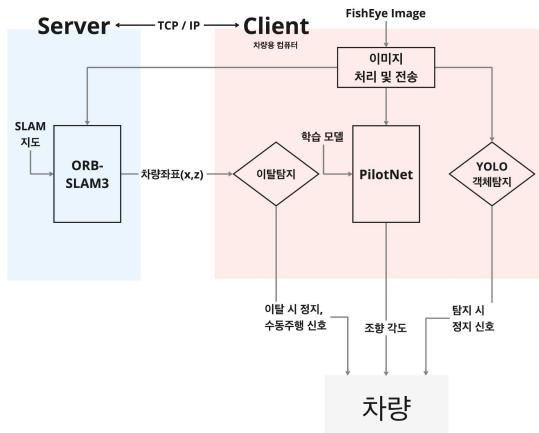


그림 1. 시스템 개요도

앞서 ORB-SLAM3와 YOLO는 PilotNet의 자율주행 성능을 향상시키기 위해서 필요하다고 판단하였다.

ORB-SLAM3, YOLO, PilotNet을 동시에 실행하면 연산량이 많아져 처리 속도가 느려진다. 이러한 문제를 해결하기 위해 TCP/IP 통신을 사용하였다. ORB-SLAM3은 서버 컴퓨터에서 실행하고 YOLO와 PilotNet은 클라이언트인, 차량에 탑재된 컴퓨터에서 실행한다. 이때 보행자 탐지에 사용한 YOLO모델로는 YOLOv4-tiny 버전[2]을 선택하여 사람에 대한 인식 정확도를 높이고 보행자 탐지 속도를 빠르게 하였다.

차량의 어안렌즈 카메라로부터 실시간으로 입력받는 이미지를 클라이언트에서 YOLO와 PilotNet의 입력으로 사용하여 TCP/IP 통신을 통해 서버 컴퓨터로 전송하여 ORB-SLAM3의 입력으로도 사용한다. YOLO는 보행자를 탐지하고, 이 따라 정지 혹은 다시 주행하는 제어신호가 차량에 전달된다. PilotNet은 입력받은 이미지에 대한 조향각을 학습된 모델을 통해 예측한다. 그리고 차량의 현재 위치는 ORB-SLAM3를 통해 인지할 수 있다. 지정된 경로를 유지하기 위해 서버에서 차량으로 차량의 위치 정보를 전송한다. 차량이 주행 전 미리 설정해 둔 경로를 벗어나면 정지하라는 신호를 보내고 수동 주행으로 전환한다.

### 2.2 하드웨어 구성

본 연구에서 HENES사의 유아용 전동 차량인 HENES-T870<sup>1</sup>을 사용한다. 카메라는 어안렌즈<sup>2</sup>를

사용하였고, 차량의 전면 430mm 높이에 부착되어 있다.



그림 2. 실험용 모형 차량과 어안렌즈의 사진

연구에서 차량과 PC가 통신하기 위해서 하드웨어가 추가적으로 필요하다. 따라서 ARM Cortex-M3 기반의 32bit 프로세서인 Nucleo - F103RB를 사용하였다. 이는 차량을 제어하는 메인보드와 PC 사이에 Serial 통신을 한다. PC에서 신호를 보내면 뉴클레오 보드에서 차량에 맞는 신호로 변환하는 과정이 필요한데, 이 과정은 뉴클레오 보드의 MBED OS에서 사용되는 MBED 코드를 이용하여 수행해주었다. MBED 코드를 통하여 변환된 신호는 차량으로 보내져 주행 명령을 수행하는 방식이다.

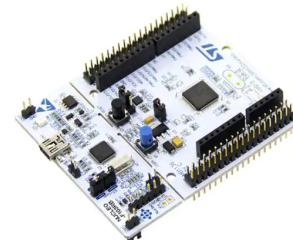


그림 3. 뉴클레오 보드(Nucleo-F103RB)

## III. 실험 및 결과

### 3.1 학습 단계 및 환경 세팅(환경세팅)

#### 3.1.1 데이터셋 획득 및 구성

PilotNet학습을 위한 입력 정보는 주행영상과 영상 별 주행 각도이다. 여기서, 주행 각도는 헤네스 차량의 앞바퀴 각도이고 정면을 0도로 하여 오른쪽은 양수(+) 각도값을 왼쪽은 음수(-) 각도값을 가진다.



그림 4. 주행 경로 장소(좌), 주행 경로 평면도(우)

학습을 진행하는 장소는 그림 4와 같이 직진과 코너 구간이 포함된 40m 거리의 공간으로 지정하였다.

<sup>1</sup> 전장 1,400mm, 전폭 780mm, 전고 530mm인 소형 유아용 전동차량

<sup>2</sup> 어안렌즈(ELP-USB8MP02G-L180)의 가로 시야각은 180도

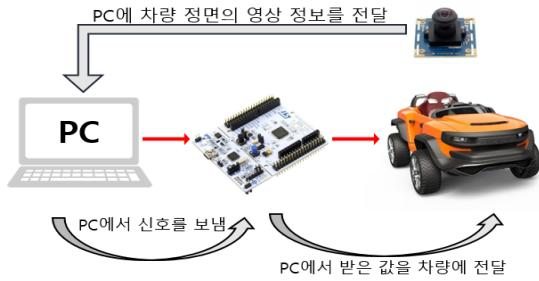


그림 5. 차량과 PC의 단방향 연결

그림 5와 같이 학습에 필요한 주행 데이터셋을 만들 때에는 PC와 간의 Serial 통신을 활용하였다.

본 연구에서는 총 50,880장의 주행 이미지와 조향각을 수집하여 학습을 진행하였다.

### 3.1.2 PilotNet 학습

데이터셋에 구성에 맞게 데이터셋들을 수집 후, PilotNet에 맞게 변환하여 학습을 진행하였다. 이때 epoch 수는 60으로 지정하고, batch size는 12로 지정하였다. 학습 후 나온 validation loss, training loss값의 그래프는 다음과 같다.

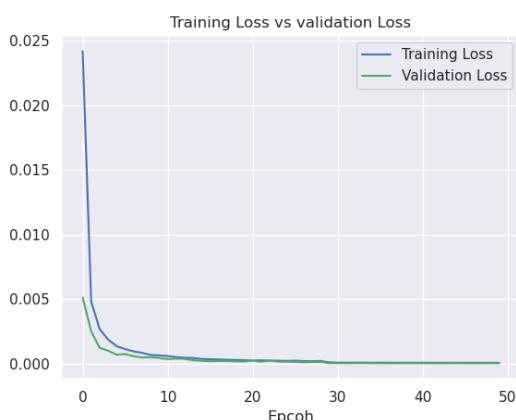


그림 6. Validation Loss &amp; Train Loss Graph

그림 6에서 그래프의 x축은 epoch수, y축은 training loss, validation loss 값을 나타낸다. epoch이 증가할수록 loss 값이 0으로 수렴하는 모습으로 보아 학습이 잘 되었음을 알 수 있다.

### 3.1.3 ORB-SLAM3 지도 생성

주행 중에는 차량의 위치를 추정하는 과정에서 어려움을 겪지 않도록 견고한 지도를 생성하는 것이 중요하다. 여기서 견고한 지도란, 최대한 많은 특징점들을 담고, 많은 keyframe을 포함하여 SLAM에서 tracking과 localization을 하는데 용이한 지도를 말한다.

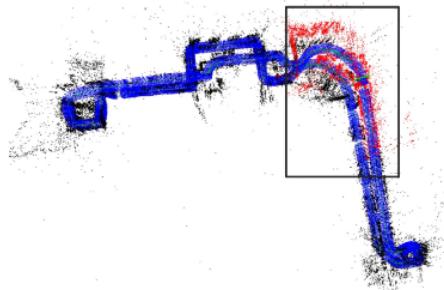


그림 7. SLAM 지도

이를 위해 SLAM 실행시 지도 상에 담을 특징점과 지도 정보, 카메라 내부 파라미터 등의 세부조건을 수정할 수 있는 파일을 생성하였다. 이를 활용해 지도에 특징점과 키프레임들을 여러번 업데이트 하여 최종 주행시 사용가능한 지도(그림 7)를 생성하였다. 그림 7에서 직사각형으로 표시한 구간은 실험에서 자율주행 테스트한 구간이다.

### 3.1.4 OBR-SLAM3 지도 활용(바운더리 설정)

미리 생성한 지도를 불러와 주행 해야 할 경로의 경계를 함수로 구현하였다. 차량의 카메라로 얻어지는 이미지 프레임에서 경로의 방향이 바뀌는 임의의 두 지점을 각각  $(x_1, y_1)$ ,  $(x_2, y_2)$ 로 설정하여 다음과 같은 직선의 방정식을 활용하여 경로의 경계를 나타내는 아래의 식 1을 만들었다.

$$y - y_1 = (y_2 - y_1)/(x_2 - x_1) * (x - x_1)$$

식 1. 경로 설정 방정식

이를 통해 주행 경로 이탈 여부를 판단 할 수 있다.

## 3.2 실험 결과

실험은 기존 PilotNet의 자율주행과 YOLO와 SLAM의 활용으로 성능이 향상된 PilotNet의 자율주행으로 진행했다. 실험 결과는 아래에 표를 통해서 나타낸다.

표 1. 기존 PilotNet 자율주행 실험 결과

PilotNet	자율주행	전체주행	자율주행 /전체주행
Test 1	93 초	123 초	75.61%
Test 2	92 초	117 초	78.63%
Test 3	93 초	123 초	75.61 %
Test 4	98 초	123 초	79.67 %

Test 5	100 초	115 초	86.96 %
--------	-------	-------	---------

표 2. PilotNet + SLAM + YOLO 실험 결과

PilotNet + SLAM + YOLO	자율주행(초) /전체주행(초)	이탈탐지횟수 /전체이탈횟수	보행자 탐지횟수 /보행자 등장횟수
Test 6	72/72 (100%)	이탈없음	1/1 (100%)
Test 7	64/69 (93%)	1/2 (50%)	1/1 (100%)
Test 8	86/96 (90%)	0/1 (0%)	1/1(100%)
Test 9	100/110 (91%)	1/2 (50%)	2/2(100%)
Test 10	81/86 (94%)	1/2 (50%)	0/1(0%)

표 1은 기존 PilotNet의 자율주행 성능을 5회 테스트한 결과를 나타낸다. 성능평가 기준으로는 차량이 출발지부터 도착지까지 주행할 때 자율주행 시간(초)의 비율이다.

자율주행 중 차량이 구간을 이탈하는 경우가 있다. 이에 차량이 이탈 구간에서 가까운 정상 구간으로 옮기는 견인 시간이 고려되어야 한다. 견인 시간은 평균적으로 5초 소요되며 전체 소요 시간은 자율주행 시간과 견인 시간을 합한 시간이다. 따라서 기존 PilotNet의 자율 주행 성능은  $(\text{자율 주행 시간}) / (\text{자율 주행 시간} + \text{견인횟수} \times 5\text{초}) \times 100\%$ 로 평가하였다.

표 2는 PilotNet에서 YOLO와 SLAM을 활용한 자율주행 성능을 5회 테스트한 결과를 나타낸다. 해당 실험에서는 이탈 탐지와 보행자 탐지가 결합된 기능을 수행한다. 성능 평가 기준으로는 차량이 전체 구간에 대해 출발지부터 도착지까지 주행할 때 자율주행 시간과 SLAM의 이탈 탐지 횟수, YOLO의 보행자 탐지 횟수의 비율이다.

이탈 탐지에 대한 성능은  $(\text{이탈 탐지 횟수} / \text{전체 이탈 횟수}) \times 100\%$ 로 평가하였다.

차량이 주행 중 보행자가 위험 범위로 접근 시, 보행자 탐지가 정상적으로 이루어졌을 경우 차량은 정지하게 된다. 위험 범위 설정에는 YOLO를 통해 생성된 Bounding box 사이즈가 기준이 되었다. 위험 범위는 보행자에 대한 Bounding box의 width가 50(픽셀) 이상일 때로 설정하였다. width가 50(픽셀) 이상일 때로 설정한 이유는, 이 값이 차량과 보행자와의 거리가 차량이 안전하게 제동하기에 충분한 거리였기 때문이다.

보행자 탐지에 대한 성능은  $(\text{보행자 탐지 횟수}) / (\text{보행자 출현 횟수}) \times 100\%$ 로 평가하였다.

따라서 전체 구간에 대해 자율주행 성능, SLAM의 이탈 탐지 성능, YOLO의 보행자 탐지 성능을 확인할 수 있다.

### 3.2 결과 분석

본 연구에서 사용한 모델은 총 50,880장의 주행 이미지와 조향각을 학습하였다. 기존 PilotNet(표 1)은 전체 주행 중 평균 79.2%를 자율주행 하였다. 같은 모델을 사용하여 PilotNet에 SLAM과 YOLO를 결합(표 2)한 후 성능을 평가해보았을 때 전체 주행 중 평균 93.1% 자율주행을 보였고 약 17.6%의 성능 향상을 보였다. ORB-SLAM3를 활용한 경로이탈탐지는 약 42.9%의 성공률을 보였으며, YOLO를 활용한 보행자탐지는 83%의 성공률을 보였다.

42.9%로 이탈탐지 성공률이 낮은 이유는 차량 가까이에서 객체가 탐지될 경우 Localization의 오차가 생기는 것을 확인할 수 있었고, 이에 따라 정상적인 이탈탐지가 어려운 경우들이 생긴 것으로 확인되었다. 또한 YOLO의 처리속도 한계로 인해 차량 앞으로 보행자가 빠르게 지나갈 경우에는 객체를 탐지하지 못할 경우가 있음을 확인할 수 있었다.

## IV. 결론 및 향후 연구 방향

본 연구에서 제안하는 방법은 기존의 PilotNet의 자율주행에서 2가지 기능이 추가된 것이다. 첫번째 기능은 YOLO를 활용하여 보행자가 위험 범위 안으로 들어오면 차량을 정지, 위험 범위를 벗어나면 다시 주행한다. 두번째 기능은 SLAM을 활용하여 주행하는 경로의 지도 정보를 먼저 생성한 후, 생성된 지도 정보를 사용하여 차량이 주행할 때 현재 위치를 받아와 이탈시 차량을 정지한다.

제안한 방법으로 성능 향상에 대한 실험 결과를 분석하기 위해서는 일관성, 정확도 등 실험에 대한 무결성이 지켜져야 한다. 따라서 실외에서는 날씨, 조명의 제약조건으로 인해 안정된 주행환경인 실내에서 먼저 실험을 진행하였다.

실험 결과를 통해 모델 학습시 사용한 수동주행과 모델의 예측값으로 주행하는 자율주행이 유사함을 볼 수 있었고, 특히 ORB-SLAM3와 YOLO를 활용한 실험에서 자율주행률이 향상되었음을 확인하였다. 또한 3.2 결과 분석과 같이 이탈 탐지기능과 보행자 탐지기능이 각각 42%와 83% 확률로 동작함을 확인하였다. 하지만 해당 이탈탐지와 보행자 탐지의 성공률이 상용화 단계에서 사용되기에는 부족하다. 따라서 첫째로 이탈탐지 기능을 보완하기 위해 이동하는 객체에도 강인한 SLAM 알고리즘 개발이 필요할 것으로 보인다. 둘째로 객체 탐지 기능을 보완하기 위해, 객체 탐지시 사용되는 YOLO의 연산량을 줄여, 빠르게 지나가는 객체에도 반응할 수 있어야 할 것으로 보인다.

이러한 결과를 기반으로 YOLO와 ORB-SLAM3를 통해 PIlotNet이 기존 PIlotNet보다 높은 주행안정성을 보이는 것을 확인하였다. 이탈 탐지와 보행자 탐지의 성공률이 보완된다면, 향후 연구에서는 충분한 데이터 수집과 모델 학습을 통해서 실외에서도 안정적인 자율 주행을 수행할

수 있을 것으로 기대된다.

### 참고문헌

- [1] Bojarski, M., Del Testa, D., Dworakowski, D., Firner,B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M.,Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K.(2016, April 25). End to end learning for self-drivingcars. arXiv.org. Retrieved June 7, 2022, from <https://arxiv.org/abs/1604.07316>
- [2] AlexeyAB. 2020. dartnet.  
<https://github.com/AlexeyAB/darknet.git>