**IoT시스템 설계 및 실습 2조 프로젝트 보고서**

**정보융합학부/2020204032/김진화**

**정보융합학부/2020204068/손호성**

**정보융합학부/2020204066/송승규**

**정보융합학부/2020803066/신의철**

|  |
| --- |
| **< 목차 >**  **1. 서론**  **2. 학습 데이터 취득 방법**  **3. 사용한 학습 모델 및 학습 방법**  **4. 학습 및 주행 결과에 대한 discussion**  **5. 조원의 역할**  **6. 조원의 기여도**  **7. 참고문헌** |

**1. 서론**

자율 주행 기술은 최근 몇 년간 급격히 발전하여 수년 내에 자율주행 자동차가 상용화 될 것이라 예측되고 있다[1]. 이러한 자율 주행 기술의 핵심 요소 중 하나는 차량이 스스로 주행 환경을 인식하고, 경로를 계획하며, 사고 없이 목적지까지 이동할 수 있는 능력이다. 이를 위해서는 딥러닝 알고리즘과 컴퓨터 비전 기술을 활용한 주행 환경의 인식과 판단이 필수적이다.

본 실습에서는 라즈베리 파이(Raspberry Pi)를 기반으로 한 주행 자동차를 설계하고 구현하였다. 이를 통해 자율 주행의 기본적인 개념과 기술을 실험적으로 검증하고, 실제 환경에서의 적용 가능성을 탐구하고자 한다. 특히, 본 실습에서는 ResNet(Residual Network) 모델을 이용하여 차량의 주행 경로를 예측하고, 주행 환경을 인식하고자 하였다. 라즈베리 파이를 기반으로 한 주행 자동차는 복잡한 센서 및 제어 시스템을 갖추지 않고도 자율 주행의 핵심 개념을 이해하고 실험할 수 있는 좋은 도구가 된다. 본 연구에서는 라즈베리 파이에 카메라 모듈을 장착하고, 이미지 데이터를 실시간으로 처리하여 주행 경로를 예측하는 시스템을 구축하였다.

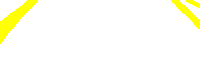
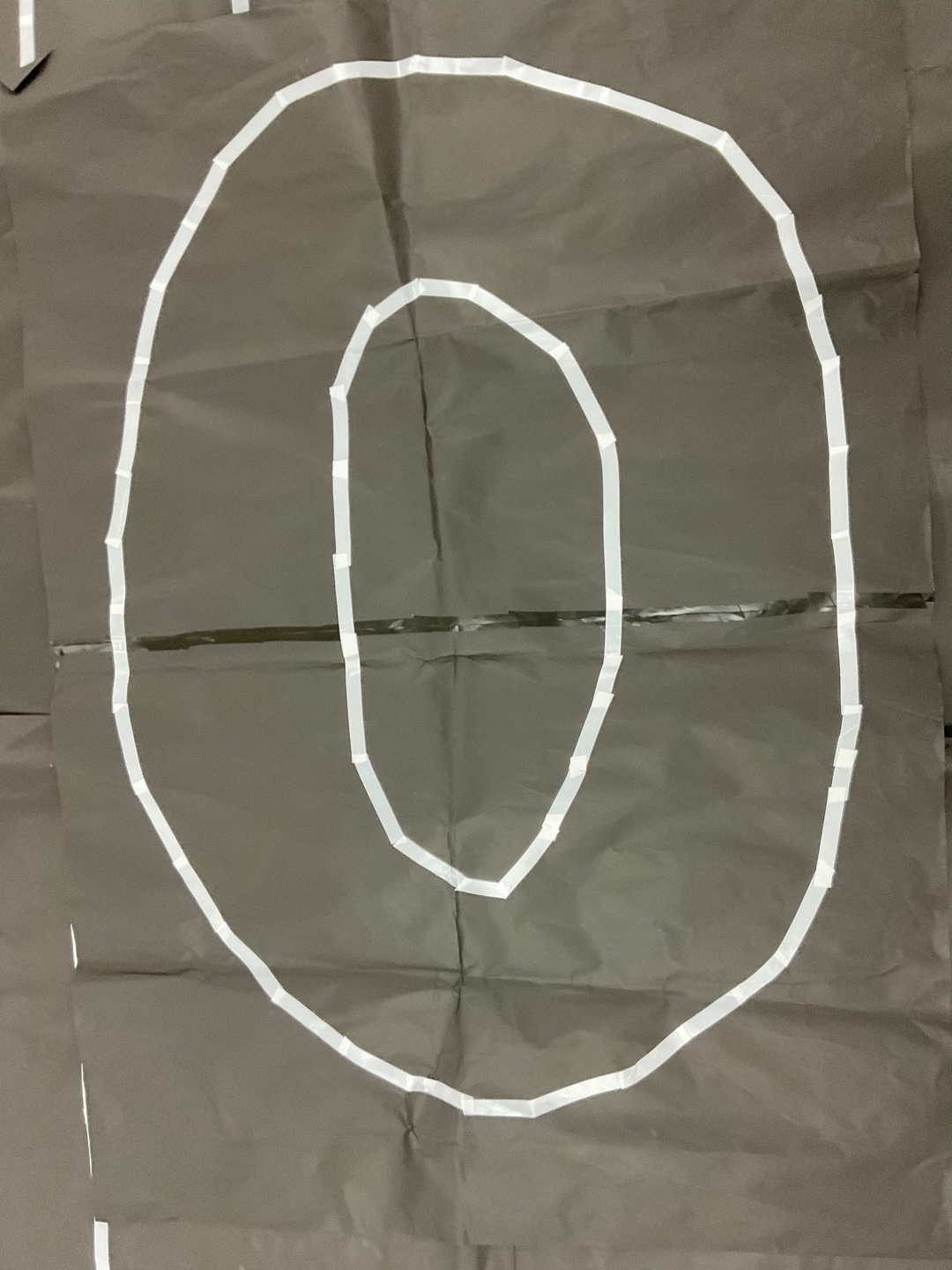
ResNet 모델은 이미지 인식 분야에서 높은 성능을 보이는 딥러닝 모델로서, 주행 경로 예측에 효과적으로 사용될 수 있다. 본 연구에서는 ResNet 모델을 이용하여 주행 경로를 예측하고, 이를 기반으로 차량의 모터를 제어하여 자율 주행을 구현하였다. 이를 통해 딥러닝 모델을 활용한 자율 주행의 가능성과 한계를 탐구하고, 향후 연구 방향을 제시하고자 한다.

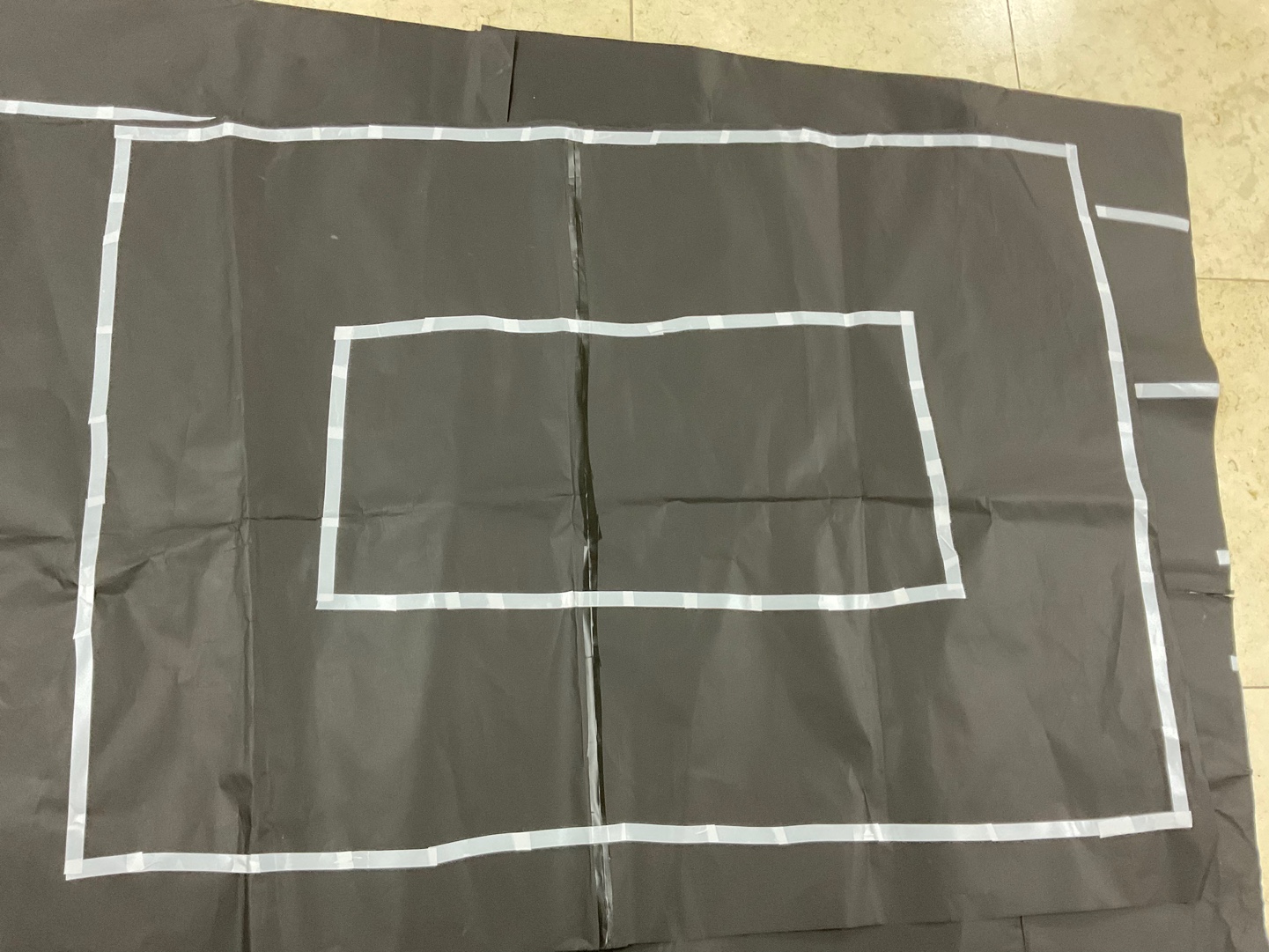
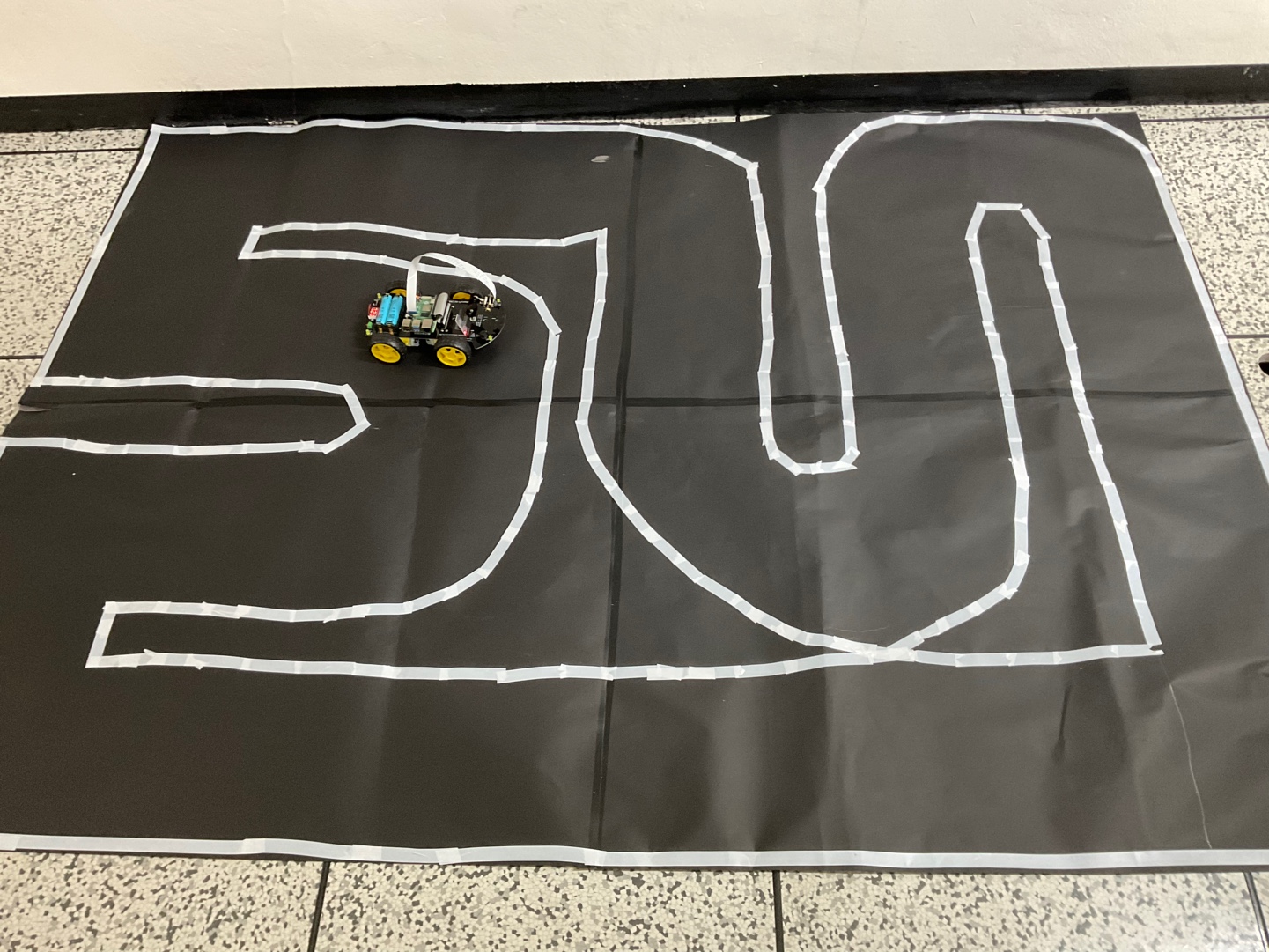
본고는 다음과 같은 구조로 진행된다. 먼저, 학습 데이터 취득 방법을 설명하고, 사용한 학습 모델과 학습 방법을 소개한다. 그 다음으로 학습 및 주행 결과를 논의하였다. 이를 통해 본 연구의 결과와 의의를 종합적으로 평가하고, 향후 연구 방향을 제안한다.

**2. 학습 데이터 취득 방법**

본 실습에서 학습 데이터는 교안 10\_2.py를 통해 취득했다. 학습과정 중 선이 제대로 보이지 않아서 thresholding값을 조절해서 카메라가 선을 제대로 인식하게 하였다. [사진 1]은 직진 데이터 사진을 가져왔다.

학습 데이터는 아래와 같은 학습 맵 [사진2], [사진3], [사진4]를 좌우로 돌려가면서 수집했다. 후에 횡단보도에 대한 정지 데이터를 각도를 0으로 맵핑시켜 추가시켰다. [사진 5]와 같이 횡단보도와 정지선이 보이는 데이터를 학습시켰다.

[사진1]  [사진 2]

[사진 3] [사진4]

[사진5]

**3. 사용한 학습 모델 및 학습 방법**

본고에서 사용한 학습 모델은 Resnet18 모델이다. 실습 초기에는 교안을 참고하여 엔비디아 모델[2]을 사용하려 했으나, 후에 더 나은 성능을 위해 ResNet18[3] 모델로 전환하게 되었다. ResNet은 합성곱 신경망 (convolutional neural network)의 한 종류로, 은닉 계층의 출력에 이전 계층의 입력 값을 더해주는 “residual block” 이라는 특수한 인공지능 모듈을 사용한다[4]. 이는 깊은 네트워크를 효과적으로 학습할 수 있는 잔여 연결(residual connection)을 사용하여 이미지 분류와 같은 비전 태스크에서 우수한 성능을 발휘할 수 있다. 잔여 연결은 각 레이어의 출력에 입력값을 더해주는 방식으로, 네트워크가 더 깊어지더라도 정보 손실을 최소화하고 학습을 촉진한다. 특히, 본 실습에서는 ResNet-18 모델을 채택했는데, 이는 18개의 레이어로 구성된 경량화 된 버전으로, 작은 데이터셋에서도 강력한 성능을 보인다. 연구에 사용된 코드는 Colab을 통해 작성하였으며, ResNet-18 모델을 구현하고 학습시키는 과정을 포함하고 있다.

ResNet18 모델을 학습시키기 위해, 수집된 영상을 이미지로 변환하고 필요한 부분만 자른 후, YUY 색상 공간을 변환하여 특징을 강조하였다. 전처리된 이미지는 모델의 입력에 사용되었다. 전처리된 이미지를 Training data (학습 데이터), Validation data (검증 데이터), Test data(테스트 데이터)으로 나누었다. 학습 데이터는 모델을 훈련하는 데 사용되며, 검증 데이터는 모델의 성능을 평가하고 하이퍼파라미터를 조정하는 데 사용된다. 테스트 데이터셋은 최종 모델의 성능을 평가하는 데 사용된다. 최적화 기법으로 Adam Optimizer를 사용하여 모델을 학습시켰다. 학습 과정에서는 다양한 데이터 증강 기법을 적용하여 모델의 일반화 성능을 높였다. 학습이 완료된 모델은 지정된 경로에 저장되게 하였으며, 테스트 데이터셋을 사용하여 모델의 성능을 평가하였다. 성능 평가는 평균 제곱 오차(MSE)와 결정 계수(R-squared)로 측정하였다. MSE는 예측값과 실제 값의 차이를 제곱하여 평균한 값으로, 값이 작을수록 모델의 예측 성능이 좋음을 의미한다. R-squared는 모델의 설명력을 나타내며, 1에 가까울수록 모델의 설명력이 높음을 의미한다.

이러한 모델을 바탕으로, 본 실습에서는 다양한 데이터셋을 활용해 보고자 하였다. 처음에는 위의 [사진4]를 통해 만든 모델을 통해 베이스 모델을 만들고, [사진2], [사진3]의 데이터를 추가로 학습하여 모델에 전이학습을 통해 추가 학습시키고자 하였다. 또한, 여러 모델을 만들어서 모델들을 병합하고자 하는 시도도 있었으나, 모델의 크기가 일치하지 않아 병합에 실패하였다. 데이터 양 조절 및 각 모델의 크기 조절하는 과정에 어려움이 있었다. 그래서 모델을 만들 때 여러 데이터를 한 번에 넣는 방법을 선택하였다.

**4. 학습 및 주행 결과에 대한 discussion**

프로젝트 첫 단계에서는 교안에 있는 엔비디아 모델로 모델링을 진행하였다. 데이터 수집 과정에서 우측 데이터가 과도하게 포함되어 차량이 우측으로만 가려는 경향을 보였다. 이로 따라 데이터 균형의 중요성을 깨닫고 이후 데이터 셋에는 좌회전, 우회전, 직진 데이터 균형을 맞추려고 노력했다.

본 실습에서는 더 나은 성능을 위해 ResNet-18 모델을 사용해서 모델링을 진행했다. 그러나 Resnet-18 모델로 나온 pth파일의 크기가 엔비디아 모델에 비해 무거웠고, 그로 인해 카메라 반응속도가 저하되는 문제가 발생했다. 카메라 반응속도가 느려지면서 차선을 이미 이탈한 후 늦게 회전하는 경우가 많아졌다. 카메라 반응 속도를 개선하기 위해 멀티쓰레드를 사용해서 병목현상을 해결해 보려고 노력하였으나 큰 효과가 없었다. 결국, 차량의 속도를 줄이는 방향으로 문제를 해결하였다. 속도를 줄인 후에는 기본적인 코스를 안정적으로 돌 수 있었다.

기본적인 주행을 완성한 후, 횡단보도 인식을 추가로 구현했다. 처음에 횡단보도 패턴을 카메라가 인식해서 블록 개수가 6개 이상일 때 자동차가 멈추도록 설계했다. 그러나 대부분의 경우 정지선을 지나친 후 정지하거나 횡단보도가 없음에도 차량이 정지하는 문제가 발생했다. 이를 해결하기 위해 정지 데이터를 학습 데이터에 추가하고, 데이터셋을 학습시켜 더 정확하게 횡단보도에 멈추도록 개선하였다.

최종적으로 만들어진 모델은 기본적인 주행 코스를 안정적으로 돌 수 있었고, 횡단보도에서 정지하는 기능도 구현되었다. 그러나 몇 가지 개선 사항이 필요하다.

1. 카메라 반응 속도 개선

Resnet - 18 모델의 파일 크기가 커서 카메라 속도에 과부하가 걸리는 문제가 있었다. 차량 속도를 줄이는 방법으로 이를 해결했지만 이로 인해 주행 시간이 오래 걸린다는 단점이 생겼다. 해결책으로는 모델 자체를 최적화해서 파일 크기를 줄이거나 카메라 구동 방식을 효율적으로 바꿔서 해결하는 방법이 있을 것이다.

1. 모델링 전이 학습

Resnet-18 모델로 데이터셋을 학습시킬 때 결과물이 나오려면 약 1시간 30분 정도의 시간이 소요되었다. 그래서 매번 데이터셋을 개선할 때마다 오랜 시간이 걸려서 전이학습을 통해 데이터 병합을 구상해 보았지만 결과적으로는 실패하였다. 전이 학습을 통해 데이터 병합이 성공하면 작업 효율이 올라갔을 것 같고 좀 더 좋은 모델을 만들 수 있을 것이다.

1. 정확한 횡단보도 인식

횡단보도를 인식하기 위해 두 가지 방법을 사용해 보았지만, 실제 시험에서는 제대로 작동하지 못하였다. 그 이유에 대해 분석한 결과. 먼저, 학습된 횡단보도 사진이 정확하지 않았다. 카메라 입장에서는 정지선의 일자나 직각 회전에서의 일자가 크게 다르지 않게 인식되었을 것이다. 그래서 데이터 수집 과정에서 정지선 옆의 길까지 한 번에 인식하는 '∏'와 같은 형태의 데이터를 수집했더라면 더 정확하게 정지선에 멈출 수 있었을 것으로 보인다.

**5. 조원의 역할**

조원의 역할은 다음과 같다

김진화 : 자율주행 자동차 모델 조사 및 소스코드 작성

손호성 : 자율주행 자동차 학습 데이터 수집 및 소스코드 작성

송승규 : 자율주행 자동차 학습용 소스코드 작성

신의철 : 자율주행 자동차 주행용 소스코드 작성

**6. 조원의 기여도**

조원의 기여도는 다음과 같다.

김진화 : 10%

손호성 : 30%

송승규 : 30%

신의철 : 30%

참고문헌

[1] 김여경, 정윤서, 황소영 (2020). Design of Shared Autonomous Vehicle Model Using Raspberry Pi, Arduino and OpenCV, 46-48.

[2] 8. (Colab+PyTorch) 카메라를 이용한 자율주행 자동차 제작-3(모델\_학습) 강의안

[3] Kaming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep Residual Learning for Image Recognition”, in Proc. IEEE CVPR, 2016

[4] 김예린, 곽지은, 안나현, 이훈 (2022). Artificial intelligence-Based Autonomous Driving System for Privacy Protection, 1238-1239.