

중간보고서

자동차 센서 데이터를 위한 Sim2Real 기술 구현
및 자동차 상태 분석 시스템

2020.07.31(금)

부산대학교 전기컴퓨터공학부

정보컴퓨터공학전공

2020전기 졸업과제 분과D(H/W)

지도 교수: 백윤주 교수님

팀 38: 여기에 팀 이름 입력

201524582 정희석

201524473 방형진

201524527 이석준

목차

1. 과제 목표
2. Sim2Real
3. 세부 요구 사항
4. 진행 상황
5. 진행 예정 사항
6. 개발 일정
7. 구성원 별 역할

1. 과제 목표

(1) 시뮬레이터 데이터의 운전 상태 분석

데이터 수집에는 Euro-Truck Simulator2 프로그램, Euro-Truck Simulator2에서 주행 데이터를 기록하는 프로그램, 기록된 주행데이터를 해석 및 가공이 용이한 CSV 파일로 변환하는 프로그램 이렇게 3개의 프로그램이 사용된다. 이 프로그램들을 통해 얻어낸 자료를 통해서 각각의 운전자가 여러가지 경우에 어떤 식으로 운전하는지 패턴을 파악한다. 예를 들어 센서 데이터 값들의 종합적인 변화를 파악하여 해당 운전 데이터는 커브에 해당하는지, 차선 변경에 해당하는지 파악할 수 있을 것이다.

운전 데이터 수집에는 팀원 3명이 모두 참여한다. 각자 자신의 평소 습관대로 운전을 하여 데이터를 수집한 후, 충분한 양의 데이터가 모였으면, 이를 토대로 데이터 분석과 운전 상태확인 작업에 들어간다. 앞서 말한 데이터의 스티어링 휠 움직임이나, 가/감속, 방향 표시등 점등과 같은 데이터를 통해 현재 차량이 차선을 따라 진행하는지, 좌회전을 하는지 등의 상태를 확인한다.

1	Timestamp	Odometer	Gear(dash)	RPM	Fuel_Level	Avg_fuel_c	Velocity	Accelerati	Accelerati	Accelerati	Coordinate	Coordinate	Coordinate	Rotation_	Rotation_	Rotation_	Steering_v	Accelerate	Brake	Clutch	Winker(left)	Winker(right)
482	20200518_	132842.9	0	1034.726	313.2924	0.314411	93.01067	-0.00617	0.015184	-0.66724	-30196.69	11.24101	-4663.18	0.249722	-0.00176	2.93E-05	-0.00233	0.0000	0	0	0	0
483	20200518_	132842.9	0	1035.476	313.2924	0.314411	93.07794	-0.00919	0.014446	-0.63064	-30197.55	11.241	-4663.18	0.249656	-0.00176	2.79E-05	-0.0023	0.0000	0	0	0	0
484	20200518_	132842.9	0	1036.182	313.2924	0.314411	93.14133	-0.00985	0.014831	-0.59553	-30198.41	11.241	-4663.18	0.249592	-0.00177	2.71E-05	-0.0023	0.0000	0	0	0	0
485	20200518_	132842.9	0	1036.847	313.2924	0.314411	93.20097	-0.00829	0.017338	-0.56186	-30199.27	11.24099	-4663.18	0.249526	-0.00177	2.74E-05	-0.0023	0.0000	0	0	0	0
486	20200518_	132842.9	0	1037.164	313.2924	0.314411	93.22948	-0.007	0.018712	-0.54558	-30199.7	11.24099	-4663.18	0.249492	-0.00177	2.77E-05	-0.0023	0.0000	0	0	0	0
487	20200518_	132842.9	0	1038.058	313.2924	0.314411	93.30967	-0.00149	0.01678	-0.49834	-30201	11.24098	-4663.19	0.249392	-0.00177	2.82E-05	-0.0023	0.0000	0	0	0	0
488	20200518_	132842.9	0	1038.101	313.2924	0.314411	93.33475	0.00027	0.016656	-0.48328	-30201.43	11.24097	-4663.19	0.249359	-0.00177	2.83E-05	-0.0023	0.0047615	0	0	0	0

(그림 1. 시뮬레이터로부터 추출한 자동차 데이터)

(2) 실제 데이터의 운전 상태 분석

우리가 수집한 시뮬레이터 운전 데이터를 sim2real을 통해 실제 데이터로 바꾸는 것이 아니라 시뮬레이터 데이터를 대상으로 운전 상태를 분석하는 모델을 구현하고 실제 자동차 ECU로부터 얻은 데이터들을 넣어서 동작 확인 후 시뮬레이터 데이터와 실제 데이터를 비교, 실제 데이터와의 차이를 파악하고 시뮬레이터의 센서 데이터를 정규화 함으로써 실제 데이터와의 GAP을 줄이는 것으로 Sim2Real 기술을 구현한다. 이 과정을 거치면서 모델 학습 및 개량을 계속 진행하여 최종적으로 실제 차량 데이터를 모델에 넣었을 때 운전 상태를 분석할 수 있도록 하는 것이 목표이다.

1	ms	s	Steering	Brake	Speed	FL_Speed	FR_Speed	RL_Speed	RR_Speed	Throttle_Pos.	Accel	Long Acc.	yaw Deg.	Later Acc.	Pred Label	Real Label
2	37	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
3	187	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
4	287	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
5	387	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
6	488	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
7	588	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
8	689	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7

(그림 2. 실제 자동차 데이터)

2. Sim2Real

(1) 개요

Sim2Real 기술은 현재 각종 이미지 처리, 이미지 분석 분야에서 연구 및 사용하고 있는 기술이다. 이 기술을 자동차 센서 데이터에 대해 진행하려고 한다. 프로젝트의 목표인 실제 차량의 센서 데이터를 우리가 만든 데이터 처리 모델에서 사용할 수 있도록 개량하기 위해서는 Sim2Real 기술을 적용해야 하는데 현재 Sim2Real을 적용하기 위한 데이터를 만들어 내는 방식은 크게 3가지로 나뉘어 있다. System Identification과, Domain Randomization, 그리고 Data Adaptation이 대표적이다.

(2) System Identification

가상의 데이터를 추출해내는 프로그램 자체를 개량하여 처음부터 현실과 매우 유사한 테스트 데이터를 만들어 내는 것이다. 이는 그렇게 해서 추출해낸 데이터 자체에 별다른 후처리를 하지 않아도, 모델 개량이 간단하다는 장점이 있다. 이 경우 시뮬레이션 프로그램 자체를 좀 더 사실적이고 많은 데이터를 제공하는 프로그램으로 대체하거나 시뮬레이션에서 데이터를 추출하는 프로그램을 수정하여야 한다. 그러나 우리 팀의 프로젝트의 경우 시뮬레이션 프로그램으로는 상용화된 EuroTruckSimulator2를 사용하고, 데이터 추출 프로그램도 이미 제작이 완료된 프로그램을 사용하기 때문에 우리가 변경할 수 있는 부분이 매우 제한적이다.

(3) Domain Randomization

일정 개수의 데이터를 추출해 낸 뒤, 그 데이터를 알고리즘을 통해 랜덤하게 조정된 수많은 데이터를 자동으로 생성하고 이렇게 생성된 데이터를 후처리를 통해서 모델에 학습시키는 방식이다. 이 경우 데이터를 직접 추출하는 시간이 크게 단축된다는 장점이 있으나, 추출해낸 데이터를 Randomizing하기 위한 알고리즘을 따로 제작하는 데에 시간이 소비된다.

(4) Data Adaptation

가장 간단하게 시뮬레이션에서 많은 수의 데이터를 직접 추출해 내고 이후에 후처리를 통해서 실제와 비슷하게 만들어 낸 후, 데이터를 학습시켜서 모델의 정확도를 높이는 것이다. 우리 팀의 경우 시뮬레이터와 데이터 추출 프로그램이 이미 준비되어 있었기 때문에 가장 간단하게 실행할 수 있는 Data Adaptation을 선택하였다.

3. 세부 요구 사항

(1) 요구조건 분석

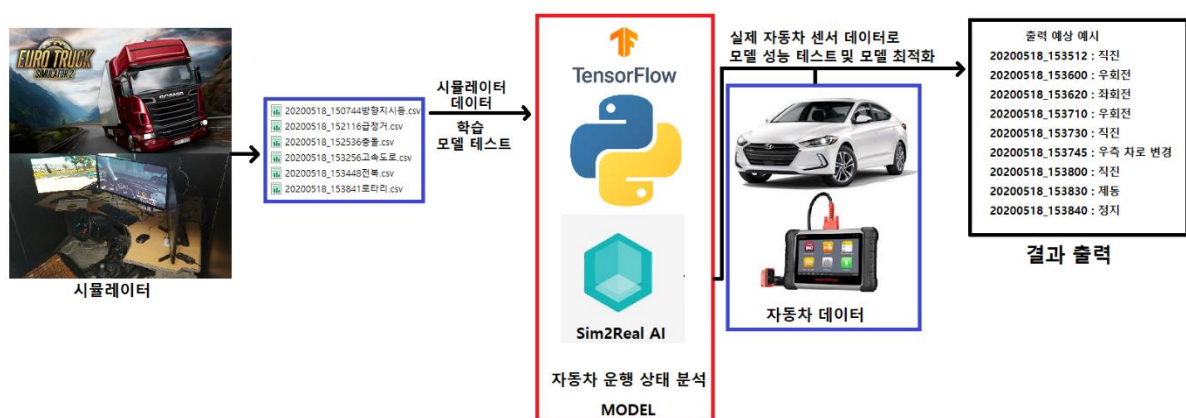
시뮬레이터 프로그램을 통해 얻어낸 데이터를 해석하고 해당 데이터가 차량의 어떤 상태를 나타내는 데이터인지 구별할 수 있는 프로그램을 구현해야 한다. 차량의 상태에는 가장 기본으로 직진, 좌측 커브 길, 우측 커브 길, 좌 회전, 우 회전, 왼쪽 차선 변경, 오른쪽 차선 변경여부를 확인할 수 있어야 한다.

이 조건이 달성되면 그 다음으로는 실제 자동차 데이터를 해석하고 데이터를 분석 프로그램에 넣어서 실행했을 때 직진, 좌측 커브 길, 우측 커브 길, 좌 회전, 우 회전, 왼쪽 차선 변경, 오른쪽 차선 변경여부를 확인할 수 있어야 한다.

(2) 향후 연구 가능 방향

이번 졸업 과제를 통해 차량의 센서데이터를 성공적으로 가져오고 이를 기반으로 차량의 현재 상태를 알아낼 수 있다면, 주행시에 차량의 상태를 기반으로 현재 차량이 안전운전을 하고 있는지 여부를 측정할 수 있다. 이 정보는 현재 T-map에서 GPS 기반 데이터를 통해 시행중인 안전운전 점수제와 같은 서비스에 적용하여 차선 변경 횟수와 같은 한층 더 자세한 점수측정을 가능케 할 수 있을 것이라 예상한다. 또한 자동차 운행 상태를 파악하여 여러 운전자의 운전 패턴을 분석, 학습하여 운전자 인식으로도 연구를 진행할 수 있으며 운전자 인식으로부터 졸음운전 패턴을 시뮬레이터로부터 추출하여 실제 차량을 사용하여 발생하는 리스크를 줄이고 차량의 센서데이터를 가지고 졸음운전 여부를 추론할 수 있도록 연구를 진행할 수 있을 것이라 예상한다.

(3) 설계 구상도



(그림 3. 시스템 예상도)

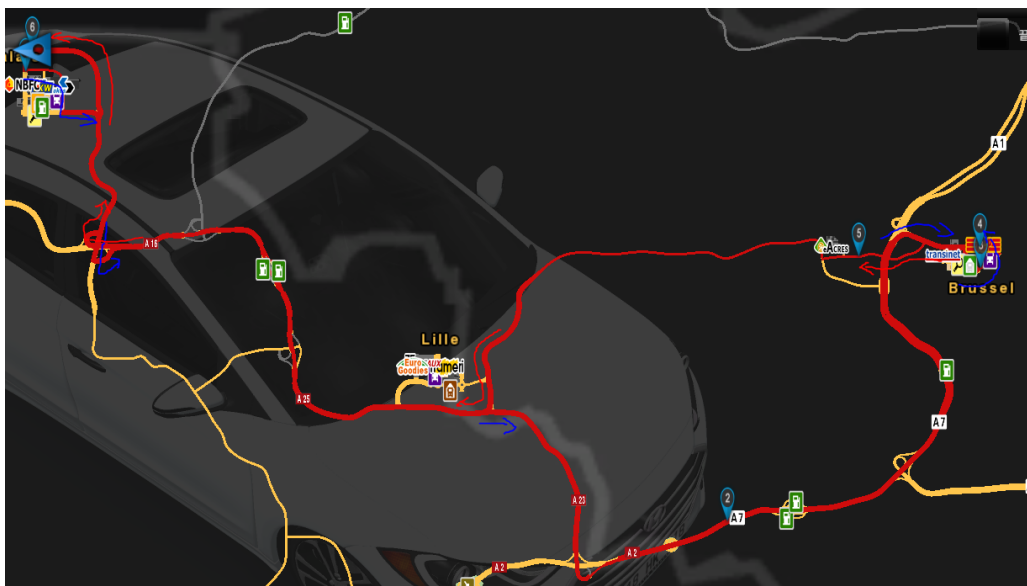
4. 진행 상황

(1) 공통: 시뮬레이터 차량 센서 데이터 수집 및 Labeling.

- 좌/우회전 단편 데이터 80개
- 좌측/우측 커브 단편 데이터 각각 20개
- 좌/우측 차선 변경 단편 데이터 각각 20개
- 직진, 정지 데이터 13분 분량
- 시내 주행 43분 분량 데이터
- 시외 주행 왕복 32분 분량 데이터



(그림 4. 학습 데이터 시내주행 맵)



(그림 5. 학습 데이터 시외주행 맵)



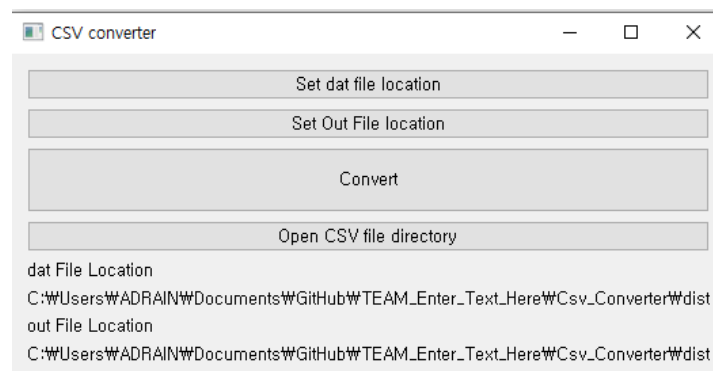
(사진 1. 시뮬레이터에 앉아서 데이터 모으는 사진)

- Labeling 상태: (0: 직진, 1: 좌측 커브, 2: 우측 커브, 3: 좌회전, 4: 우회전, 5: 좌측 차선 변경, 6: 우측 차선변경, 7: 정지)

-> 현재 정지는 정확성을 위해서 후 처리 과정에서 처리하는 것을 고려 중

(2) 이석준: CSV 변환 프로그램 GUI 제작

- 기존의 CSV converter 프로그램은 python을 사용한 파일로써 따로 exe파일을 사용하지 않는 소스코드 파일임
- 이로 인해 CSV converter 실행 시 CMD 환경에서 직접 타이핑을 통해 실행해야 함
- GUI기반의 exe 파일을 제작하여 실행 시간 단축
- 추후 CSV converter에 변동 사항 있을 시 지속적으로 업데이트



(그림 6. Converter GUI)

(3) 이석준: 운전 영상 및 차량 상태 출력 프로그램(GUI) 제작(구현 중)



(그림 7. 영상 및 예측 결과 출력 프로그램)

- 현재 프로그램에서는 자동차 영상(그림 7은 자동차 시뮬레이터 녹화영상)을 띄울 수 있다.
- CSV파일을 읽어서 학습된 모델을 통해 상태 예측을 실시(예정)
- 예측 결과를 영상과 같이 확인할 수 있는 프로그램을 제작(진행 중)

(3) 방형진: 시뮬레이터 차량 데이터 분석 및 가공

- 시뮬레이션에서 데이터 추출 프로그램을 통해 추출해낸 dat 데이터를 CSV 파일로 변환하며 실제 데이터의 형태와 유사하고 측정 모델에 학습시키기 용이한 형태로 변경
- 사용할 데이터 종류를 설정하여 필요한 데이터만 추출
- 변경된 데이터를 측정 모델에 학습시키기 위한 Label 설정
- 데이터 변경 시 핸들 각도 등의 센서 데이터를 토대로 기본적인 자동 Labeling
- 자동 Labeling된 데이터를 2차적으로 기록 동영상을 통해 직접 Labeling

Timestamp	Velocity	accel_X	Steering_wheel_x	Accelerate	Brake	Winker(left)	Winker(right)	Label	Timecheck
20200724_	28	0.433068	-40	6	0	1	0	3	0:00:06
20200724_	28	0.45375	-37	6	0	1	0	3	0:00:06
20200724_	28	0.466464	-33	6	0	1	0	3	0:00:06
20200724_	28	0.507631	-30	6	0	1	0	3	0:00:06
20200724_	28	0.477962	-28	6	0	1	0	3	0:00:06
20200724_	28	0.455488	-24	6	0	1	0	3	0:00:06
20200724_	28	0.471062	-23	6	0	1	0	3	0:00:06
20200724_	28	0.441253	-20	6	0	1	0	3	0:00:06
20200724_	28	0.343013	-19	6	0	1	0	3	0:00:06
20200724_	28	0.318845	-18	6	0	1	0	3	0:00:06

(그림 8. 시뮬레이터 데이터)

- Timestamp: 기록 Timestamp
- Velocity: 차량 속도
- accel_X: 좌/우 가속도(Lateral Acceleration)
- Steering_wheel: 차량 스티어링 휠 각도(출력 값: -1 ~ 1 변환: -450 ~ 450) (1+1/4회전)
- Acceleration: 가속 페달 압력(출력 값: 0 ~ 1 변환: 0 ~ 100%)
- Brake: 브레이크 페달 압력(출력 값: 0 ~ 1 변환: 0 ~ 100%)
- Winker(left, right): 좌/우 방향지시등 점등 여부
- Label: 상태 표시

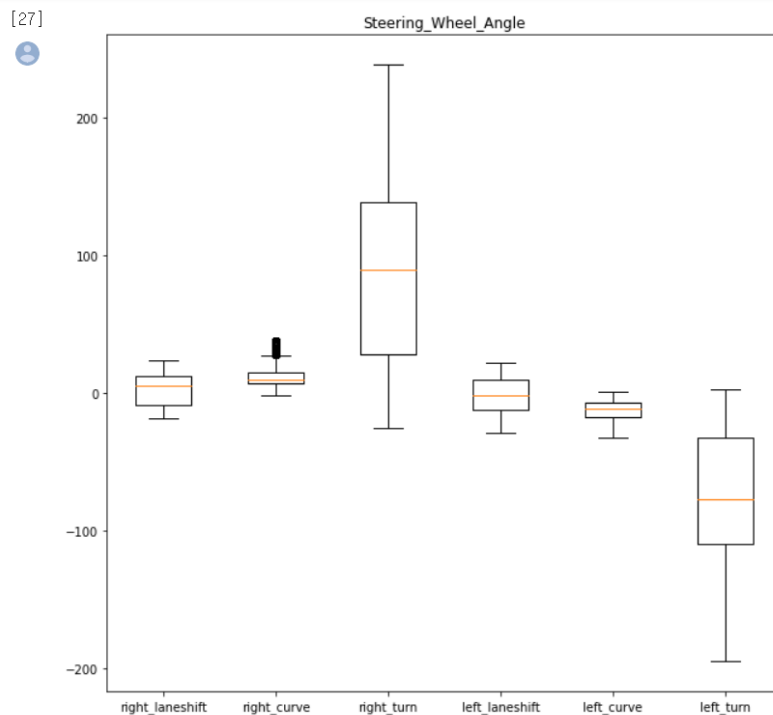
(4) 방형진: 실제 차량 데이터 분석

- 연구실의 실제 차량 사용 센서데이터 습득
- 시뮬레이션의 차량 데이터와 차이점 분석
- 데이터 간의 차이를 줄이기 위해 정규화 작업 수행 필요

1	ms	s	Steering	Brake	Speed	FL_Speed	FR_Speed	RL_Speed	RR_Speed	Throttle Pos.	Accel	Long Acc.	yaw Deg.	Lateral Acc.	Pred Label	Real Label
2	37	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
3	187	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
4	287	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
5	387	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
6	488	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7
7	588	00:00	280	73	0	0	0	0	0	3.137255	0	-0.00333	0	0	7	7

(그림 9. 실제 차량 OBD-II 데이터)

- ms, s: Timestamp
- Steering: 차량 스티어링 휠 각도
- Brake: 차량 브레이크 페달
- Speed: 차량의 속도
- FL/FR/RL/RR Speed: 각 4방향의 휠 속도 -> 시뮬레이터에서는 추출 불가 -> 미사용
- Throttle Position: 엔진 스로틀 위치 -> 시뮬레이터에서는 추출 불가 -> 미사용
- Accel: 차량 가속 페달
- Longitudinal Acceleration: 전/후 가속도 -> 시뮬레이터에서의 값이 불안정 -> 미사용
- Yaw rate: 요(각) 가속도 -> 시뮬레이터에서의 값을 확인 어려움 -> 분석 중
- Lateral Acceleration: 좌/우 가속도



(그림 10. 각 상황에서 시뮬레이터 핸들 회전 각도 데이터)

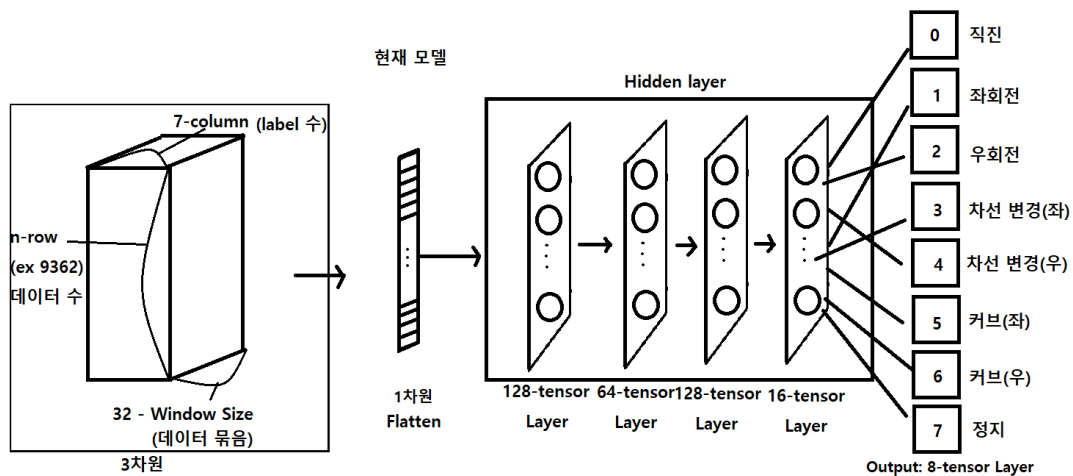
- 시뮬레이션 과정에서 차량 상태에 대한 각각의 데이터의 유사성을 분석
- 시뮬레이션의 차량 데이터와 차이점 분석
- 데이터 간의 차이를 줄이기 위해 시뮬레이션 데이터에 정규화 작업 수행
- 실제 데이터와의 GAP을 줄이는 데에 가장 중요한 과정

(5) 방형진, 정희석: 데이터 정규화

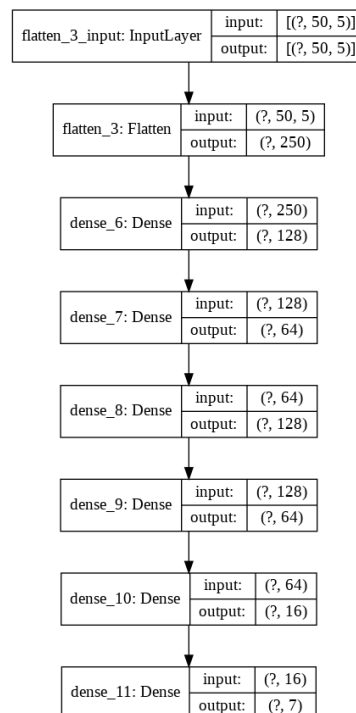
- Steering Wheel: 실제 data에서는 총 1080° , 좌/우로 540° -> 1/2 지점인 270° 를 기준으로 정규화
- Brake: 실제 data에서는 0 ~ 100% -> 50%를 기준으로 정규화
- Speed: 실제 data에서는 0 ~ 255km/h -> 고속도로 제한속도 100km/h를 기준으로 정규화
- Accelerator: 실제 data에서는 0 ~ 100% -> 70%를 기준으로 정규화
- Lateral Acceleration: 실제 data의 분포 범위가 -1.9~2 -> 그 중 가장 변화가 많은 구간인 -0.6 ~ 0.6 기준으로 정규화
- 실제 데이터를 기준으로 정규화를 진행 -> 최적화를 진행해야 하는 부분이다.

(6) 정희석: 상태 분석 모델 제작

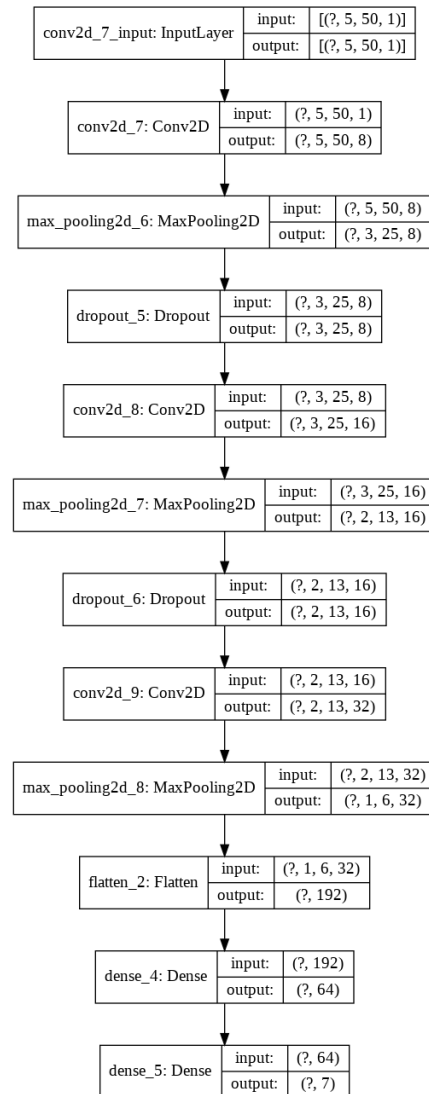
- Tensorflow 기반의 Machine Learning
- pandas 모듈로 읽은 csv 데이터를 Numpy 모듈을 사용하여 numpy array형태로 변환
- Deep Neural Network기반 학습 모델
- 학습된 모델에 테스트용 데이터를 넣어서 정확도 측정
- 구동 환경은 Google Colab TensorFlow 2.2



(그림 11. Deep Neural Network 모델 모형)



(그림 12. Deep Neural Network 모델)



(그림 13. Convolution Neural Network 모델)

TEST	3694	670	732	217	42	175	149	5679
	65%	12%	13%	4%	1%	3%	3%	
REAL	40067	4243	3109	1520	1583	1571	1168	53261
	75%	8%	6%	3%	3%	3%	2%	
TRAIN	39021	6343	6290	5929	5925	2810	2351	68669
	57%	9%	9%	9%	9%	4%	3%	

(그림 14. 테스트, 실제, 학습 데이터 Label별 개수)

- 현재 시뮬레이터로부터 나오는 데이터는 약 30Hz로 초당 30~32개의 데이터가 나온다.
- 실제 차량 데이터는 10Hz로 초당 10개의 데이터로 추출
- 시뮬레이터 데이터를 10Hz로 조절 후 학습 및 평가 진행

예측 결과)

(0: 직진, 1: 좌측 커브, 2: 우측 커브, 3. 좌회전, 4. 우회전, 5. 좌 차선변경, 6. 우 차선변경)

DNN TEST Prediction							
PRED ↓	0	1	2	3	4	5	6 ←TEST
0	61.759117	9.8	12.31	3.927	6.751	2.751	2.706
1	66.584	10.22444	9.776	0	10.87	1.247	1.297
2	51.595	12.22	15.36007	3.008	10.89	4.923	2.005
3	68.664	15.51	8.141	1.996928	1.536	1.69	2.458
4	52.404	12.1	13.78	8.494	8.974359	2.965	1.282
5	57.955	15.72	9.659	14.02	0	0	2.652
6	51.794	22.87	0	8.52	15.7	1.121	0

(그림 15. Deep Neural Network 테스트 데이터 예측 정확도)

DNN REAL Prediction							
PRED ↓	0	1	2	3	4	5	6 ←TEST
0	88.32705219	1.897	1.188	1.318	5.264	1.255	0.751
1	69.314	18.80745	1.579	5.892	1.084	2.805	0.519
2	62.85	3.538	19.90994	1.19	8.62	1.866	2.026
3	66.689	2.719	2.039	19.23861	7.954	1.36	0
4	35.502	5.938	4.485	6.317	42.26153	0.695	4.801
5	86.633	3.883	1.591	1.146	0.827	4.20115	1.719
6	96.318	1.113	0.856	0	1.541	0.171	0

(그림 16. Deep Neural Network 실제 데이터 예측 정확도)

CNN TEST Prediction							
PRED ↓	0	1	2	3	4	5	6 ←TEST
0	53.017826	11.27	14.58	5.248	8.054	4.054	3.782
1	54.414	11.82045	13.62	1.646	9.975	3.491	5.037
2	50.091	13.26	13.99271	2.461	14.4	3.555	2.233
3	49.462	28.57	1.69	8.294931	5.837	3.687	2.458
4	44.631	10.58	16.51	8.574	12.58013	3.045	4.087
5	40.341	26.52	12.31	13.07	6.061	1.13636	0.568
6	41.928	31.17	0	8.52	9.193	8.072	1.12108

(그림 17. Convolution Neural Network 테스트 데이터 예측 정확도)

PRED ↓	0	1	2	3	4	5	6 ←TEST
0	82.61412135	0.809	1.887	6.731	4.156	2.633	1.171
1	53.241	23.144	2.263	17.53	1.202	1.744	0.872
2	53.651	0.901	26.47153	8.106	6.433	1.962	2.477
3	53.569	3.331	1.428	25.83277	3.875	9.993	1.971
4	44.093	0	1.642	3.411	40.36639	8.97	1.516
5	67.091	1.273	8.02	20.05	1.018	0.50923	2.037
6	89.726	0.257	0.171	8.818	0.428	0	0.59932

(그림 18. Convolution Neural Network 실제 데이터 예측 정확도)

5. 진행 예정 사항

(1) 이석준:

- 차량 상태 분석 프로그램의 인터페이스 작성 시작

(2) 방형진:

- 시뮬레이터 데이터와 실제 차량 데이터의 차이 분석 및 정규화 개량 중

(3) 정희석:

- 차량 상태 분석용 CNN, DS-CNN 모델 분석 및 구현 및 DNN 모델 개량 중

6. 개발 일정

일자(혹은 주차)	진행할 일
5/11	팀 결성
5/13	신청서 제출
5/18	시뮬레이터 사용법 및 데이터 사용법 교육
6/5	졸업과제 1주차 발표
6/12	졸업과제 보고서 작성 및 제출
6월 3주차	1차 멘토링 진행
7/31	졸업과제 중간 보고서 작성 및 제출
8월 1주차	2차 멘토링 진행
9월 1주차	포스터 작성
9월 3주차	최종 보고서 작성
9월 4주차	9/25 졸업과제 발표
10월 말	결과물 제출 & SW등록

	6				7				8				9			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
시뮬레이터 교육	■															
센서 데이터 수집	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
센서 데이터 처리 관련 자료 수집		■	■	■	■	■										
SIM2REAL Transfer 학습				■	■	■										
Data Adaptation 학습				■	■											
센서 데이터 처리 및 정규화				■	■	■	■	■	■	■	■	■	■	■	■	■
데이터 학습					■	■	■	■								
분류 학습 모델 작성 및 수정					■	■	■	■	■	■	■	■	■	■	■	■
학습 모델 성능 테스트 및 수정						■	■	■	■	■	■	■	■	■	■	■
실제 데이터 모델 테스트								■	■	■	■	■	■	■	■	■
프로그램 디버깅 및 테스트									■	■	■	■	■	■	■	■
발표준비 및 시연 준비															■	■

- 완료 - 진행 중 - 진행 예정

7. 구성원 별 역할

팀원	역할
정희석	시뮬레이터를 이용하여 센서 데이터를 수집하고 Google Colab with TensorFlow 2.0을 사용하여 Sim2Real 모델을 구현, 이후 자동차 상태 분석 프로그램의 분석 모델을 구현을 담당
방형진	시뮬레이터를 이용하여 센서 데이터를 수집 및 분석하고, 데이터를 처리하며 Sim2Real 모델에 데이터 학습을 진행, 이후 자동차 상태 분석 프로그램의 데이터 처리를 담당
이석준	시뮬레이터를 이용하여 센서 데이터를 수집하고 데이터 처리 프로그램 작성 및 Sim2Real 모델에 실제 자동차 데이터를 사용하여 모델 성능 테스트 실시, 이후 자동차 상태 분석 프로그램 인터페이스 작성을 담당