

BAYS BAYSHORE DRIVE TRAFFIC LIGHT LAT. 30.505306, LONG. -86.479210

BAYSHORE DRIVE TRAFFIC LIGHT LAT. 30.505323, LONG. -86.479207

영상 인식을 통한 자율주행

박준범 심진섭 최동혁 한대호

TABLE OF CONTENTS

1. 개요

2. 목표

3. 과정

4. 결론

01

자율주행

02

03

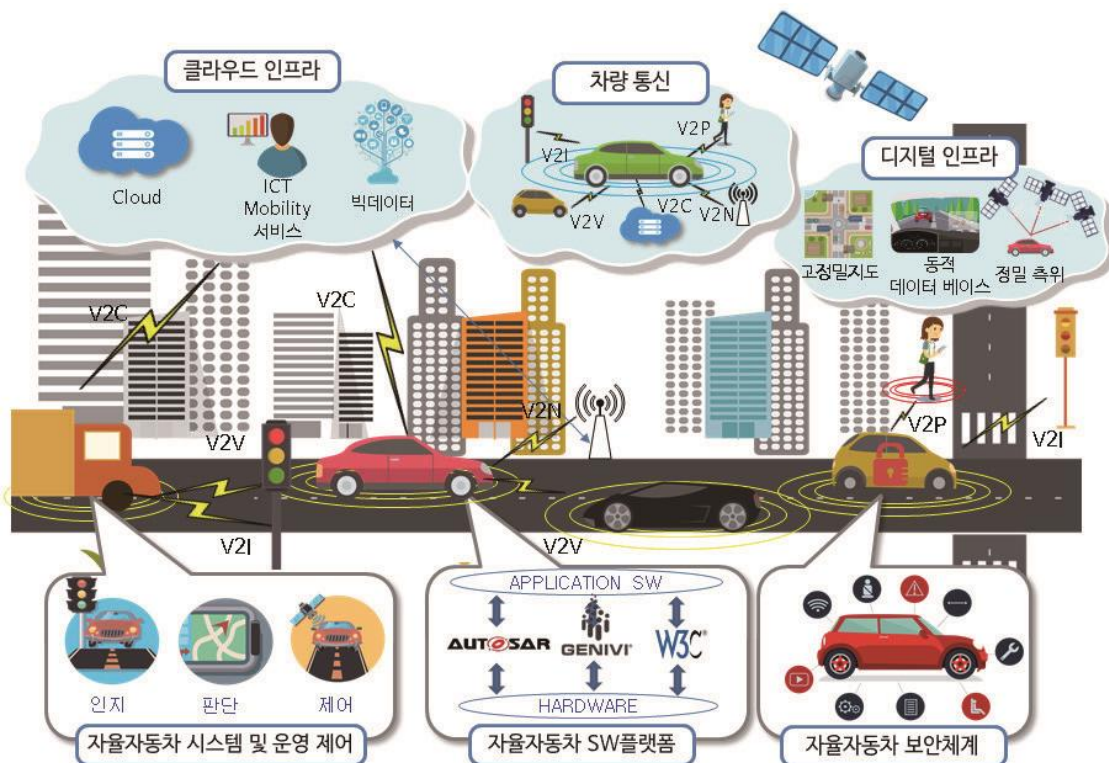
04

05

06

07

08



ICT 표준화전략맵 Ver. 2019 디바이스 자율주행차, TTA, 2018

- ◆ 운전자 또는 승객의 조작 없이 자동차 스스로 운행이 가능한 자동차 (자동차관리법 제2조)
- ◆ 고성능/고신뢰 자동주행 기능이 탑재된 차량이 인프라 및 통신 기술 등과 유기적으로 결합되어 운전자의 개입 없이 스스로 운행하는 개념으로 센서 등으로부터 획득한 다양한 정보를 활용하여 차량의 정밀한 위치와 주변환경을 인식하고 이를 기반으로 충돌없이 안전한 운행 가능한 자동차

01

02

03

04

05

06

07

08

자율주행 자동차의 역사



01

02

03

04

05

06

07

08

자율주행 자동차의 기술

〈표 1〉 자율주행자동차의 주요 구성기술

구성기술	내용
환경인식	<ul style="list-style-type: none"> · 레이더, 카메라 등의 센서 사용 · 정적장애물(가로등, 전봇대 등), 동적장애물(차량, 보행자 등), 도로표식(차선, 정지선, 횡단 보도 등), 신호 등을 인식
위치인식 및 맵핑	<ul style="list-style-type: none"> · GPS/INS/Encoder, 기타 맵핑을 위한 센서 사용 · 자동차의 절대/상대 위치 추정
판단	<ul style="list-style-type: none"> · 목적지까지의 경로 및 장애물 회피 경로 계획 · 차선유지, 차선변경, 좌우회전, 추월, 유턴, 급정지, 주정차 등 주행 상황별 행동 판단
제어	<ul style="list-style-type: none"> · 운전자가 지정한 경로대로 주행하기 위해 조향, 속도변경, 기어 등 액추에이터 제어
인터랙션(HCI)	<ul style="list-style-type: none"> · 인간자동차인터페이스(HVI, Human Vehicle Interface)를 통해 운전자에게 경고 및 정보를 제공, 운전자의 명령을 입력 · V2X(Vehicle To Everything) 통신을 통하여 인프라 및 주변차량과 주행정보 교환

(출처 : 자율주행자동차 기술개발의 특징 및 정책동향, 융합연구정책센터, 2017)

- ◆ HCI(Human Computer Interaction)
사람 - 컴퓨터 간 상화 작용을 돕는 작동 시스템 설계기술 및 학문
- ◆ 자동차가 스스로 주변환경을 인식·판단하여 자동으로 운전하기 위해 인공지능, 빅데이터, 고성능 처리 SW 및 HW 플랫폼, 센서 등 첨단 기술이 필요함

01

02

03

04

05

06

07

08

자율주행 자동차의 기술

단계별 자율주행 자동차 분류



(SAE 기준에 따른 자율주행자동차의 기술 5단계)

- ◆ 자율주행 기술은 아직 국제 표준이 정해져 있지 않지만
미국 도로교통안전국(NHTSA)에서 구분한 자율주행
기술 5단계(0~4단계)와 미국 자동차기술학회(SAE)
구분한 6단계(0~5단계) 사용

자율주행 기술 수준 단계



출처 : NHTSA 참고, KT경제경영연구소 재구성

01

02

03

04

05

06

07

08

자율주행 자동차의 원리

자율주행자동차 원리 3

1단계 : 인식

교통 상황이나 운행 환경 등 파악
(핵심 부품 : GPS, 레이저 스캐너, 라이다, 영상센서)

2단계 : 판단

자동차 스스로 가장 이상적인 결정을 수행

3단계 : 제어

자동차 스스로 속도나 조향장치 등 조절하여
사고를 예방, 안전운전 수행

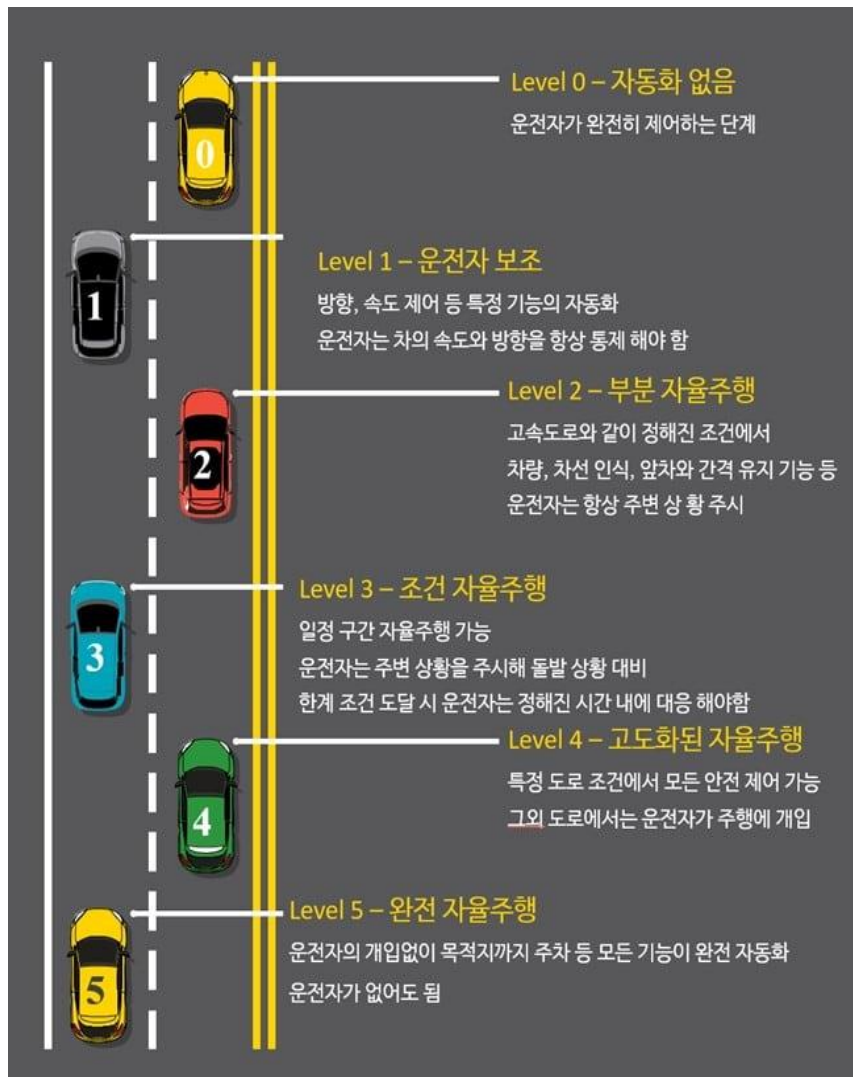
Kb손해보험 인사이트



<http://blog.lginnotek.com/549>

01
02
03
04
05
06
07
08

점진 vs 급진



◆ 점진적 접근

자율주행 1단계부터 4단계까지 점진적 기술 개발을 통해 기존 자동차 산업의 주도권을 놓치지 않은 상태에서 자율주행 시대 실현

VS

◆ 급진적 접근

전통적 자동차 제조 기술은 없지만 인공지능 지능과 S/W 기술을 기반으로 초기 (1~3) 단계를 뛰어넘어 바로 완전 자율주행 (4~5) 단계를 구현함으로써 자율주행 자동차 알고리즘과 소프트웨어 플랫폼 분야에서 주도권을 차지하는 게 목표



Volkswagen



TOYOTA



HYUNDAI



Google

amazon



UBER

01 ■ 세계 최초의 자율주행 자동차

02

03

04

05

06

07

08



(이미지 클릭 후 영상 재생)

01

02

03

04

05

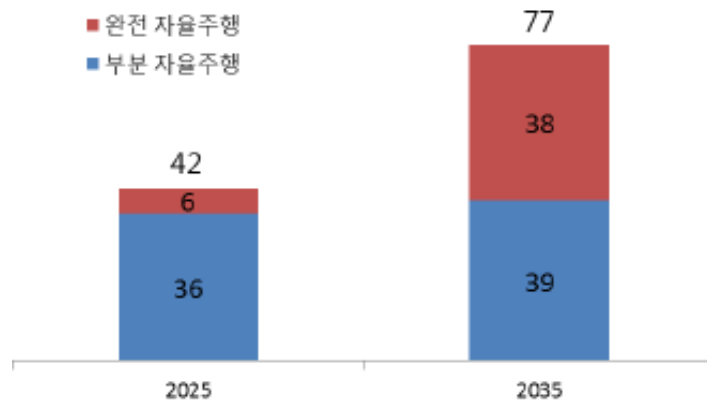
06

07

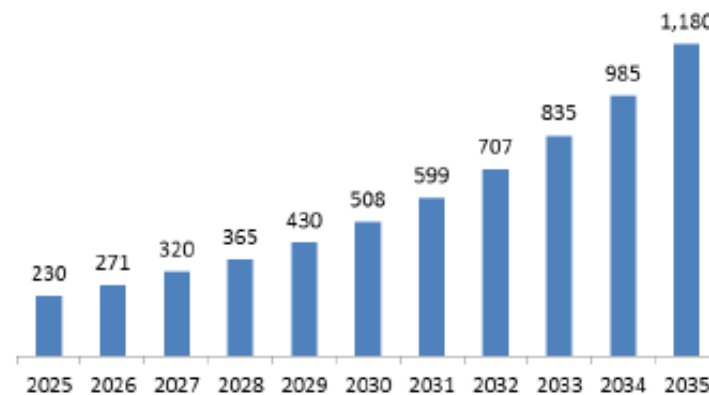
08

자율주행 자동차 시장전망

[자율주행차 시장 규모(10억달러) 및 판매량(만대) 전망]



출처 : BCG(2015.3)



출처 : IHS Automotive(2014.12)

BCG(보스턴 컨설팅 그룹)

- 2025년 약 420억 달러
- 2035년 770억 달러
- 2035년 세계 자동차 판매량의 25% 자율주행

HIS오토모티브

- 2035년 자율주행 자동차 판매량 1,000만대 이상
- 자동차 시장의 약 10%

네비건트 리서치

- 자율 주행자동차의 보급률
- 2025년 4%에서 2030년 41%, 2035년 75% 예상

맥킨지

- 본격적인 상용화 2030년
- 2040년 미국내 차량의 75% 이상 자율주행 자동차

01 자율주행 자동차가 각광받는 이유

02

03

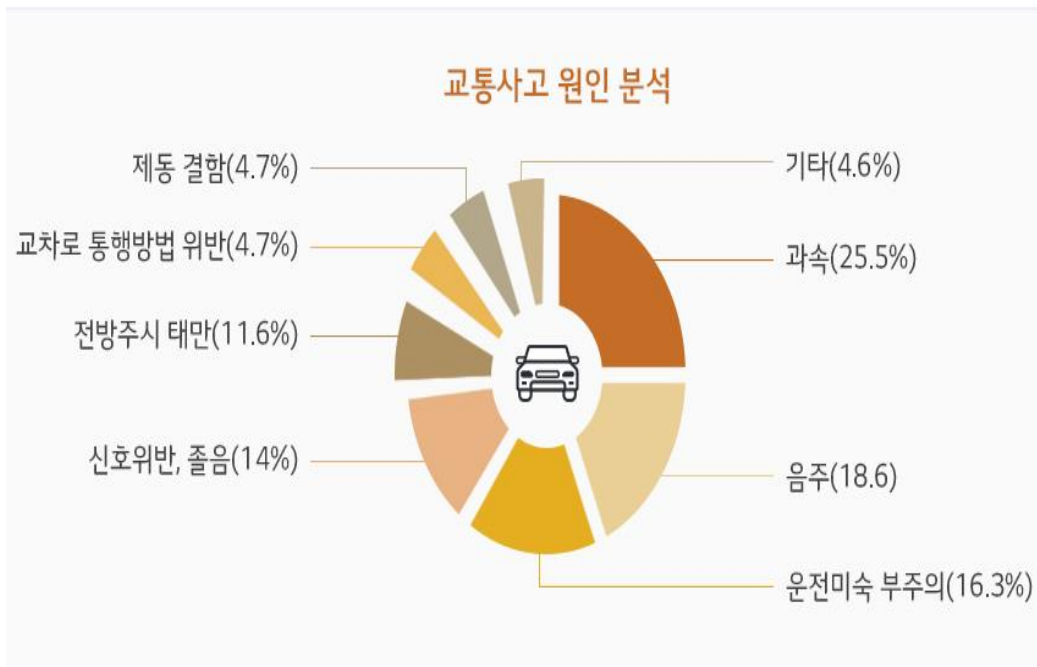
04

05

06

07

08



도로교통공단

- ◆ 국내에서 음주, 운전미숙, 졸음, 전방주시 태만 등의 이유로 발생한 사고 비율 60% 이상
- ◆ 차량 흐름 제어로 교통체증감소 효과
- ◆ 운전자가 목적지까지 직접 자동차 통제 없이 이동 가능
- ◆ 나이, 장애 여부 등과 관계없이 차량 이용 가능
- ◆ 과속/신호위반의 감소, 교통경찰의 효율적 배치 운전대의 제거를 통한 차량 실내 공간 활용 등 가능

Human Error 개선

상당수의 사고 및 인명손실 예방

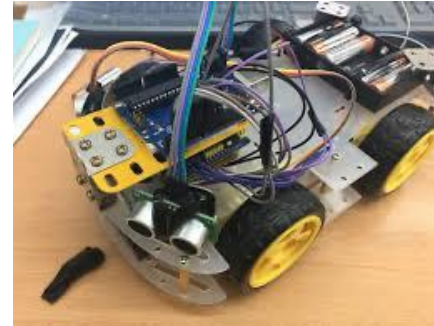
01

02



GTA5를 활용한
자율 주행

유로트럭을 활용한
자율 주행



라즈베리파이
자율주행 RC카

아두이노 자율주행
드론



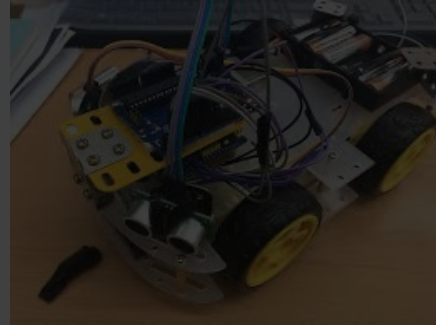
01

02



GTA5를 활용한
자율 주행

유로트럭을 활용한
자율 주행



아두이노 자율주행
드론



라즈베리파이
자율주행 RC카

01

02

자율주행

- ◆ 사람의 조작없이 스스로 주행하는 차량 제작
당초 기획은 하드웨어까지 직접 제작하는 것이었으나,
하드웨어 관련 문제와 투자하는 자원 대비 이익률이 매우 저조할 것으로 판단 되어,
시중 자동차 시뮬레이터를 사용해 개발
- ◆ 여러 제약 조건과 환경 등을 미루어 보아 차량과 여러 시설물, 보행자 등의 장애물이 기본적으로 구현이 되어 있는
GTA5(by 락스타 게임즈) 사용
- ◆ 차량 시뮬레이터를 사용한 자율주행 프로그램 연구 및 개발을 통해 OPENCV의 **Object Detection Algorism**
Hough transform을 적용한 차선 인식, User의 Key Interrupt을 기반으로한 **Supervised Learning**
학습된 데이터를 분류 및 예측하는 Inception과 같은 **Deep-Learning Neural Network Algorism**
YOLO V3와 같은 진화된 **Object Detection Algorism** 적용, object가 무엇인지 파악하고
그 상황에 맞는 행동을 스스로 학습하고 동작하는 **Unsupervised Learning**을 연구하고자 한다.

01

02

03

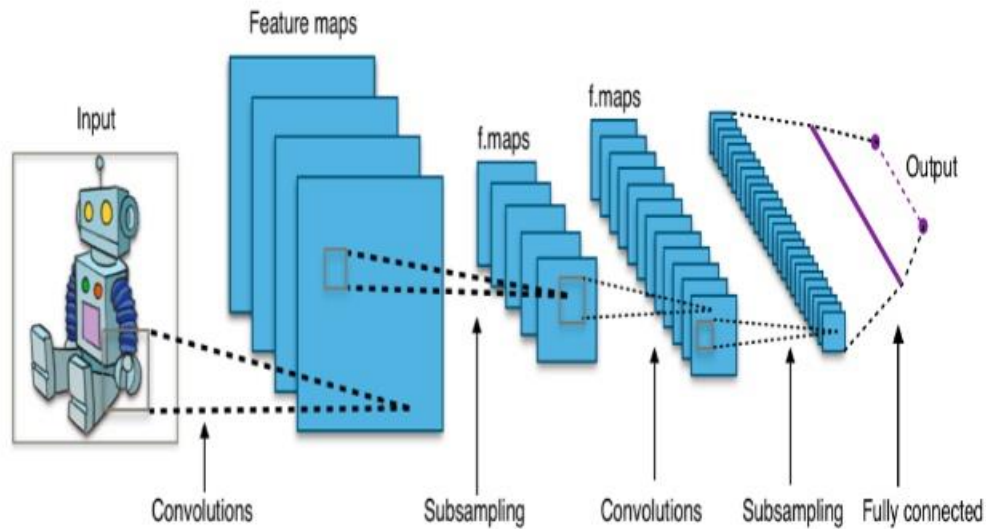
04

05

06

합성곱 신경망(CNN)

Deep learning - CNN



- ◆ Convolutional Neural Network
시각적 이미지를 분석하는 데 사용되는 인공신경망
- ◆ 이미지 및 비디오 인식, 추천 시스템, 이미지 분류 등
이미지 분석 및 자연어 처리에 응용
- ◆ 차선 인식, 차량, 보행자 및 객체 인식
자율주행을 위한 end to end 학습, 운전자 상태 인식
탑승자 인식 등의 분야에 사용

01

순환 인공 신경망(RNN)

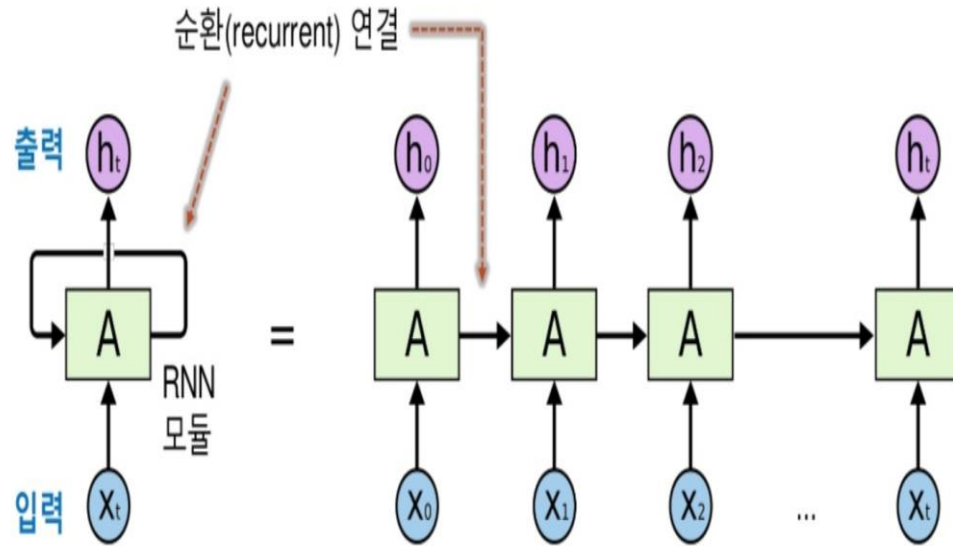
02

03

04

05

06



- ◆ Recurrent Neural Network
유닛간의 연결이 순환적 구조를 갖는 인공신경망
- ◆ 필기체 인식이나 음성 인식과 같이 시변적 특징을 가지는 데이터 처리
- ◆ 음성 인식, 교통 상황 예측, 운전자 상태 인식 등

01

02

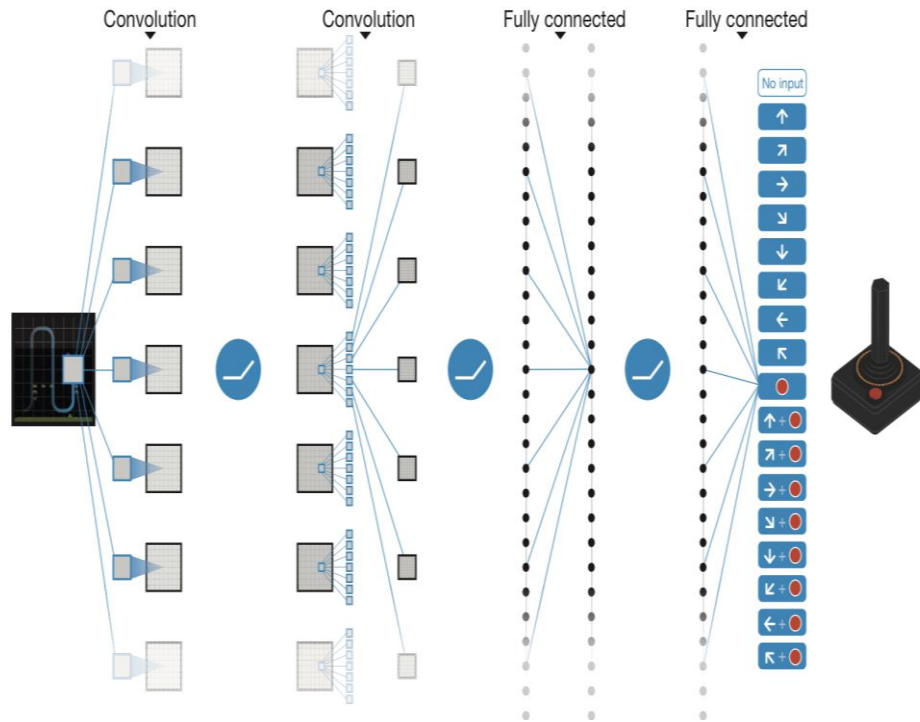
03

04

05

06

심층 Q-네트워크(DQN)



- ◆ Deep Q Networks
구글 딥마인드가 개발한 기계학습 알고리즘
Deep Neural Network + Q-Learning
- ◆ 다양한 상황에서의 주행 대응 방법 학습
인공지능 시뮬레이터에서의 주행 학습 알고리즘

01

자율주행 인식 과정

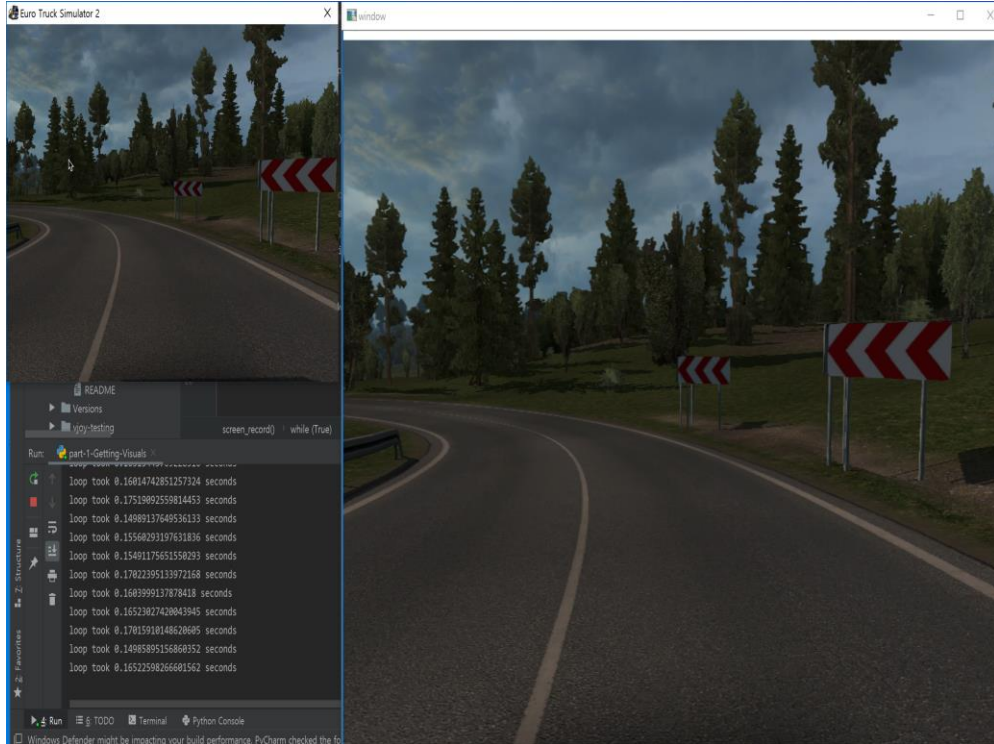
02

03

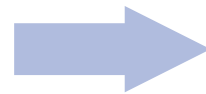
04

05

06



1. Screen Grab



2. Line Finding

01

02

03

04

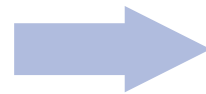
05

06

자율주행 인식 과정



3. Lane Finder



4. Self Driving

01

02

03

04

05

06

자율주행 인식과정



5. Object Detection

01

02

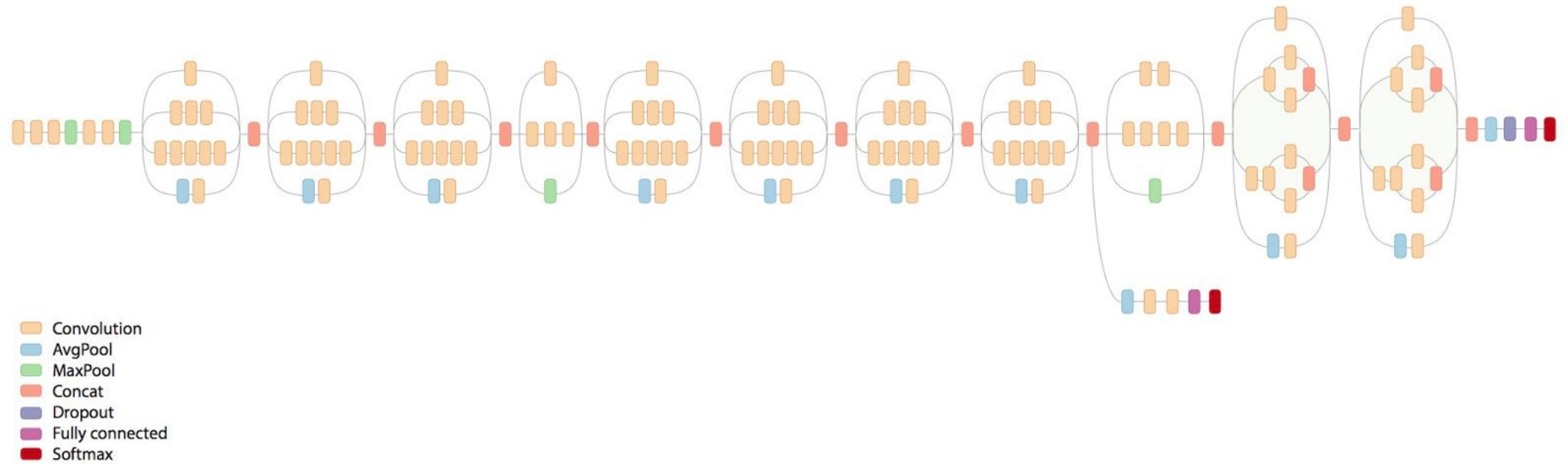
03

04

05

06

자율주행(GTA5)



구글이 만든 Inception V3 Neural Network Architecture 모델 사용

01

02

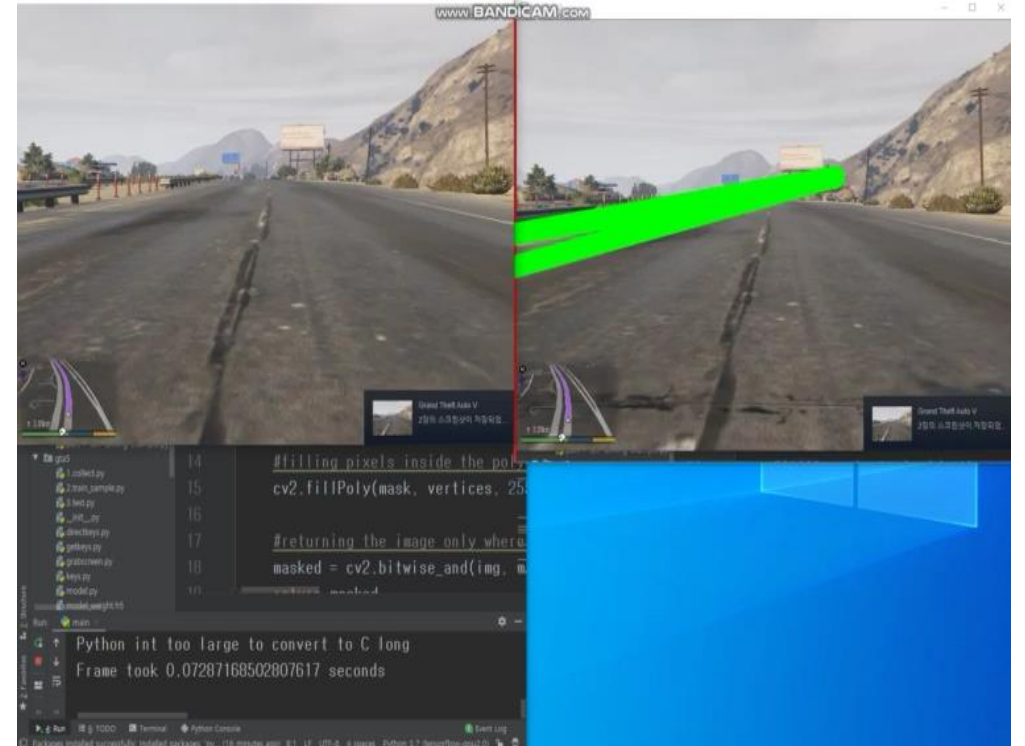
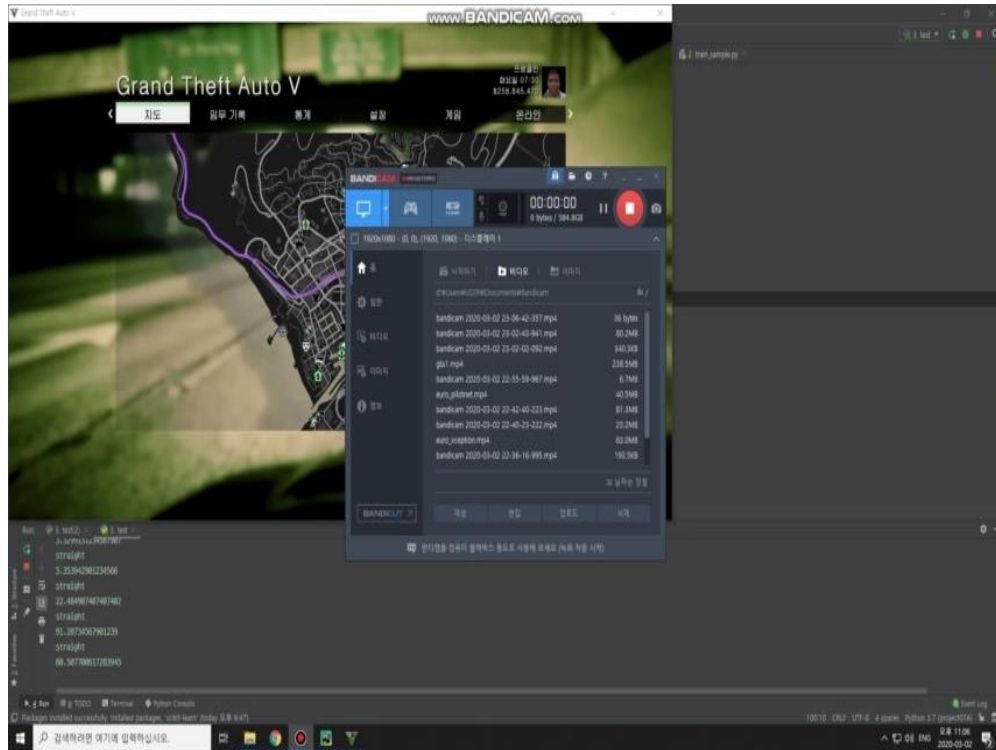
03

04

05

06

자율주행(GTA5)



Inception V3 모델을 사용한 학습 트레이닝 결과

자율주행(GTA5)

Figure 1. A canonical Inception module (Inception V3).

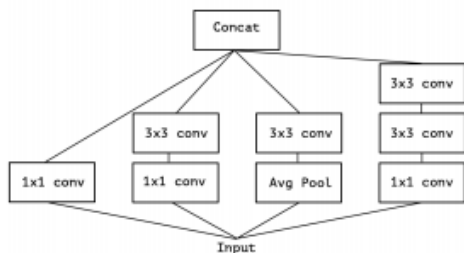


Figure 2. A simplified Inception module.

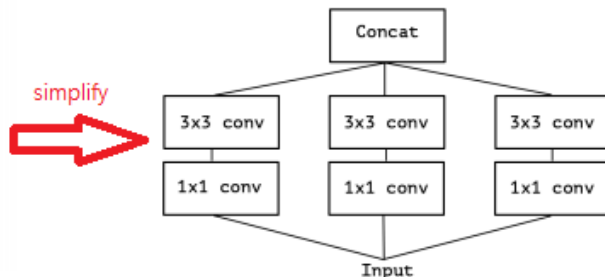
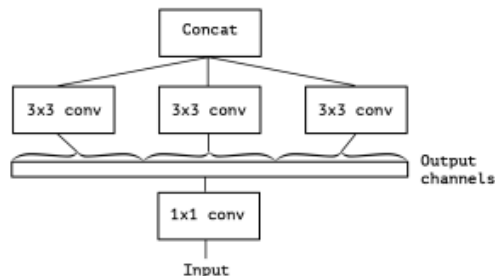


Figure 3. A strictly equivalent reformulation of the simplified Inception module.

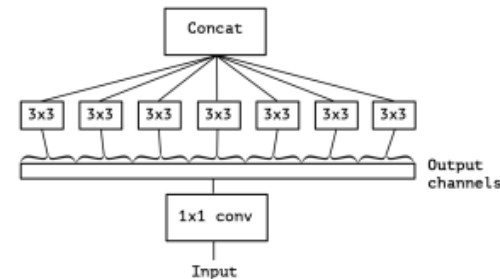
1x1 convolution을 통합하여,
깊이에 따라 다른 convolution 적용



각 convolution이 적용되는 깊이를 줄이고,
더 많은 convolution을 사용



Figure 4. An "extreme" version of our Inception module, with one spatial convolution per output channel of the 1x1 convolution.



구글이 만든 Xception Neural Network Architecture 모델 사용
Inception보다 cross-channel correlation과 spatial correlation을 더 독립적으로 연산

01

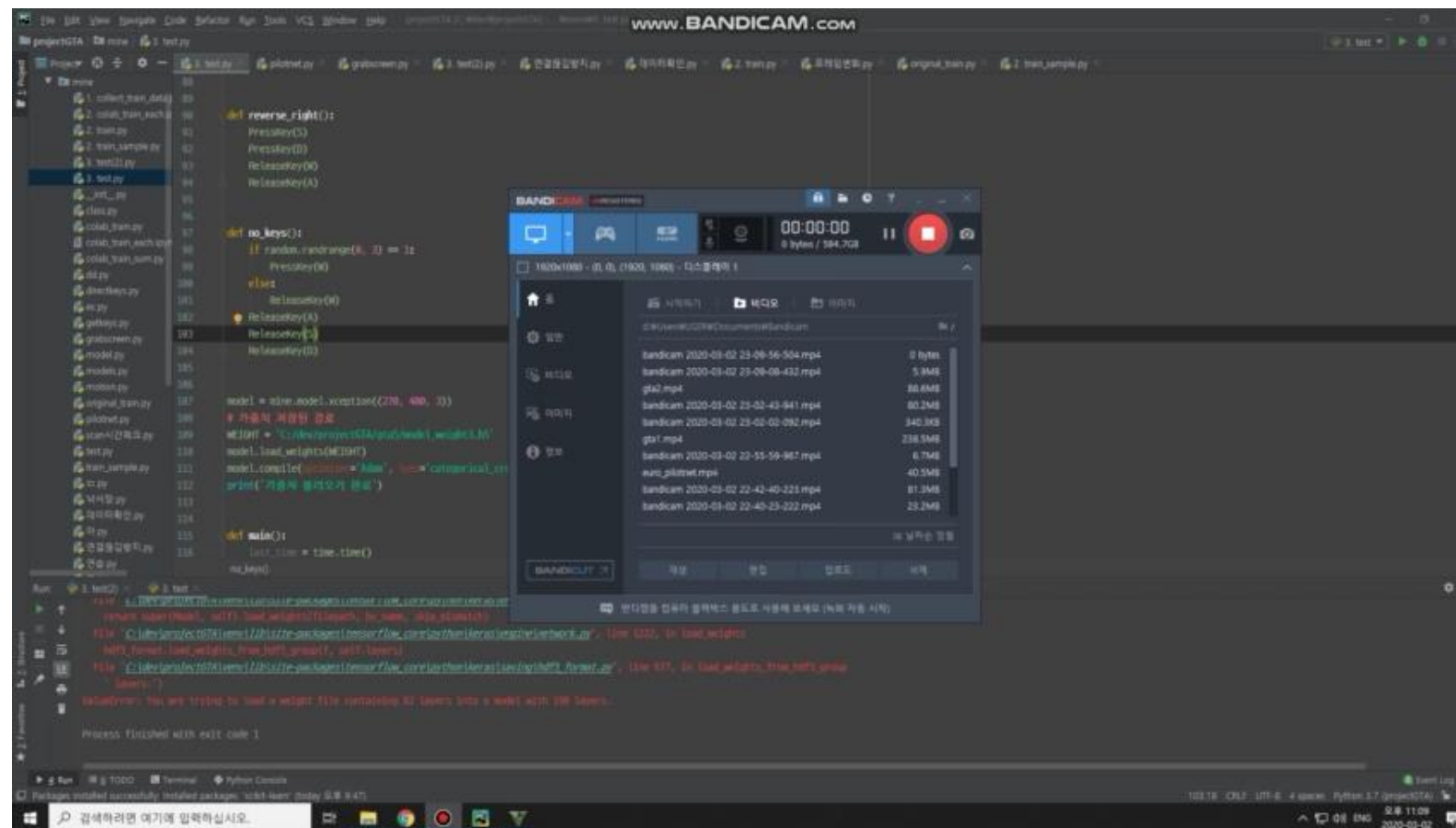
02

03

04

05

06



Xception 모델을 사용한 학습 트레이닝 결과

01 자율주행 (Euro Truck Simulator 2)

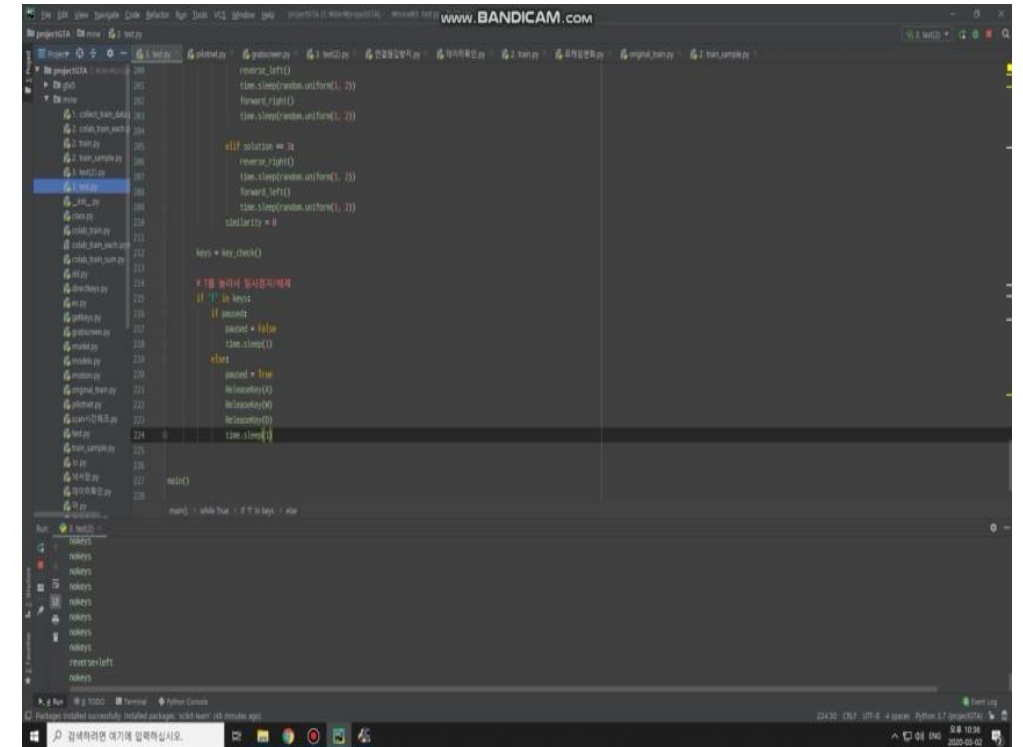
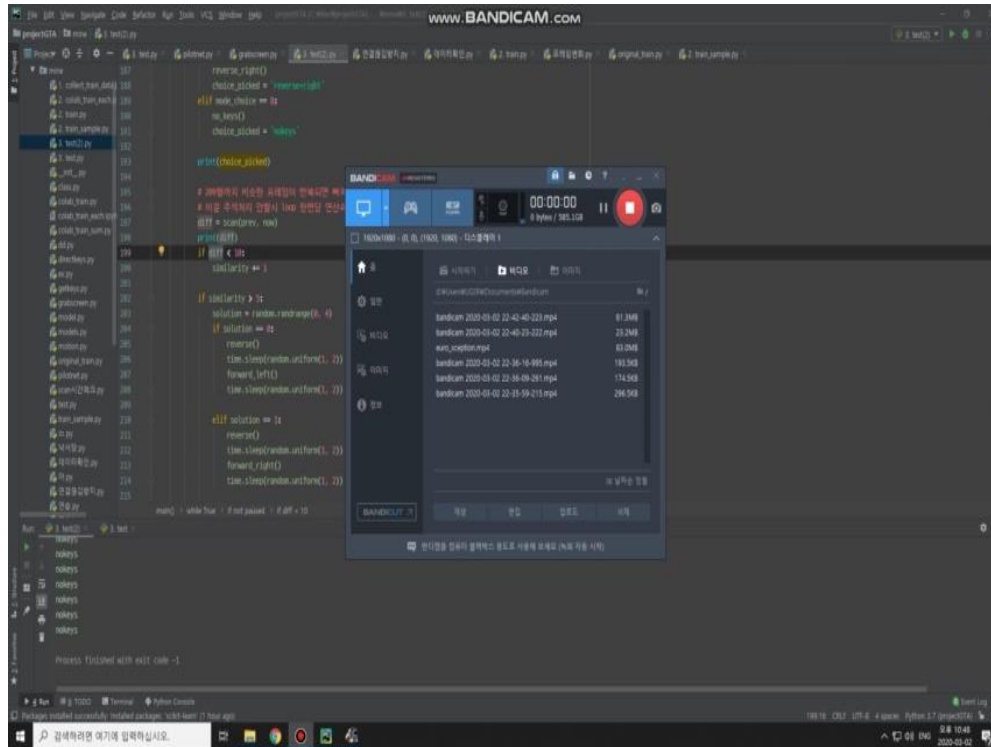
02

03

04

05

06



Xception 모델을 사용한 학습 트레이닝 결과

자율주행 모드 문제점 및 대응 방안

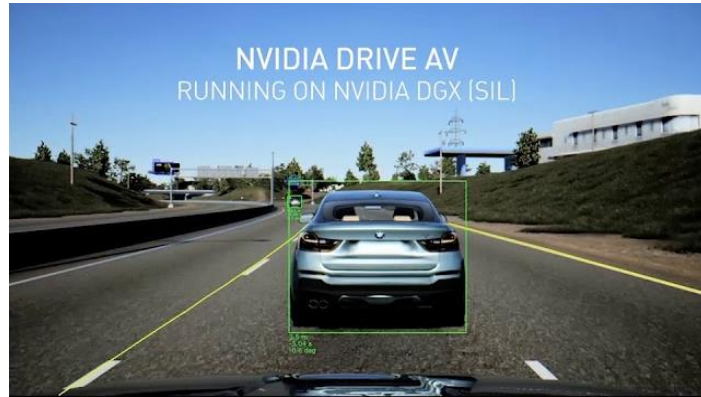
- ◆ End to End 학습을 시도했지만 데이터 질 or 모델 부적합의 원인으로 학습이 제대로 이루어지지 않음
- ◆ 키 입력 데이터가 W(가속)에 너무 편향 되어 있어서 예측 값이 W(가속)가 많이 나오는 경향이 있음
- ◆ 키 입력이 레이블이기 때문에 파일럿(trainer)에 따라 레이블 편차가 큼
- ◆ 하드웨어적인 문제로 각각의 파일을 불러와 학습하기 때문에 학습이 제대로 이루어지지 않을 가능성?
(파일 병합시 메모리에러 발생)



- ◆ 온라인에서 배포하는 자율주행 시뮬레이터 프로그램을 활용하여 코드 구현

01
02
03
04
05
06

자율주행 시뮬레이터



NVIDIA
DRIVE
Constellation

LGSVL
Simulator

AirSim
on
Unity

Udacity's
Self-Driving
Car Simulator



01

02

03

04

05

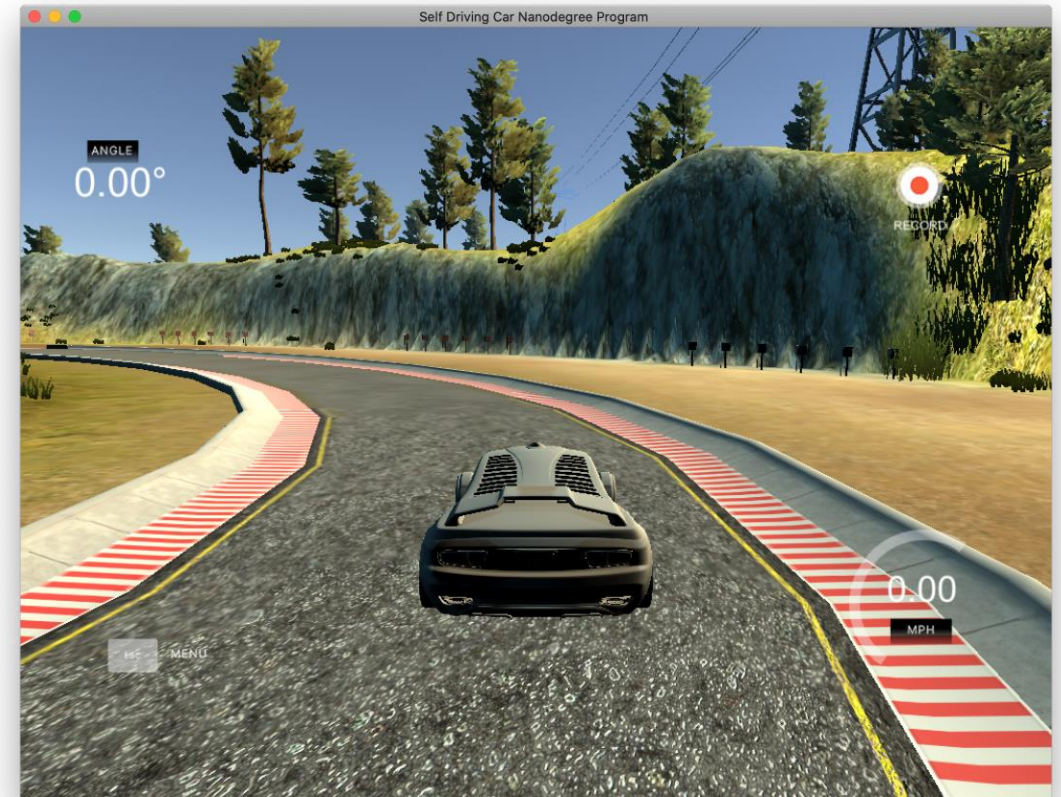
06

Udacity Self-Driving Car Nanodegree Program



01
02
03
04
05
06

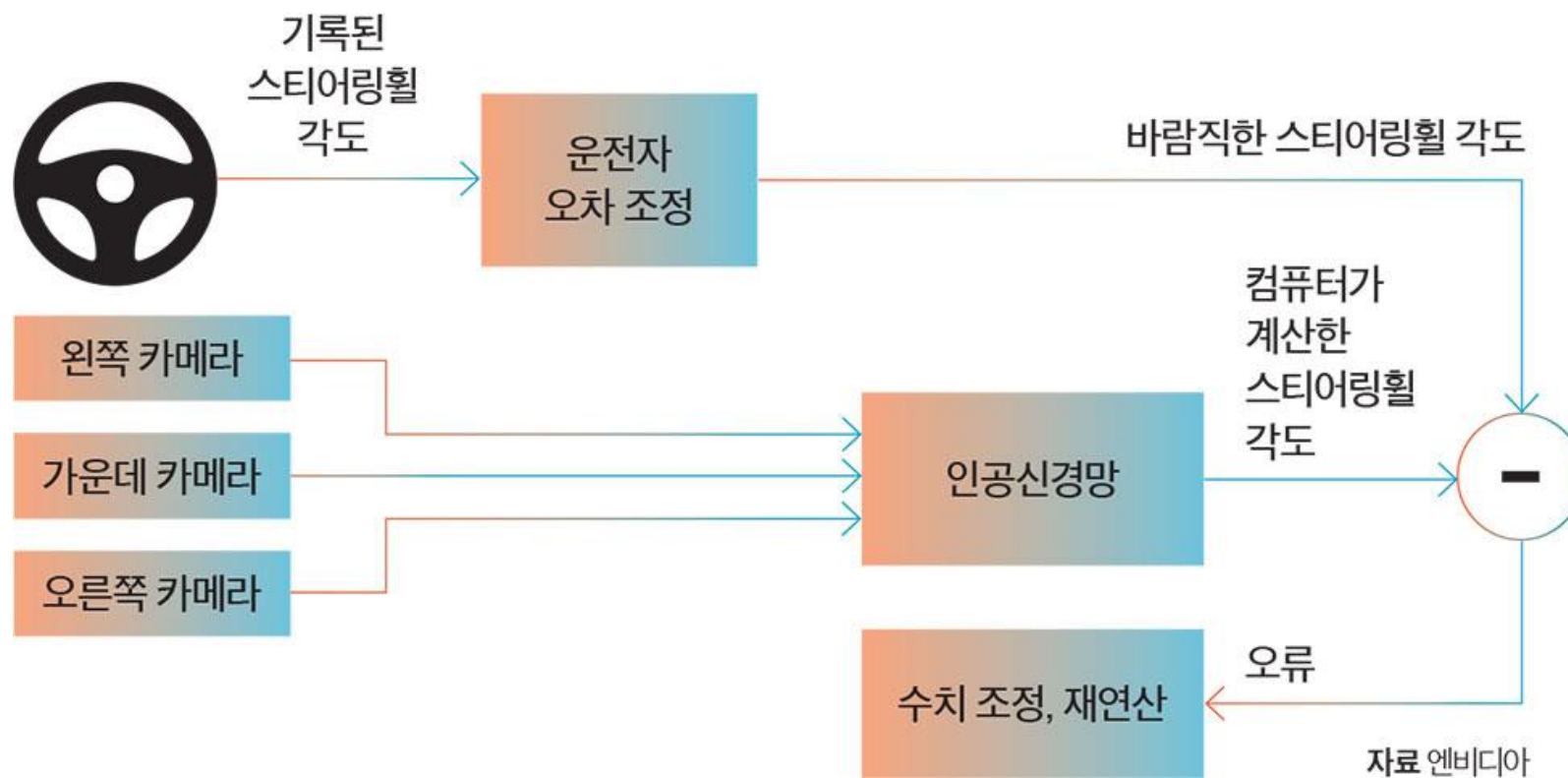
Udacity Self-Driving Car Nanodegree Program



Track 1에서 데이터 수집하여 Track 2에서 Test

Udacity Self-Driving Car Nanodegree Program

자율주행차 학습 과정



Udacity Self-Driving Car Nanodegree Program



left

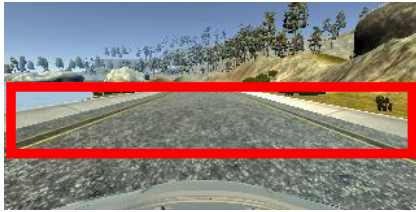


center



right

왼쪽, 중앙, 오른쪽 카메라에서 찍은 세 개 이미지 저장



불필요한 이미지 제거 및
이미지 사이즈 조절 (64 X 64)

flip Image (ang : 0.0)



shift Image (ang : 0.0)



brightness Image (ang : 0.0) shadow Image (ang : 0.0)



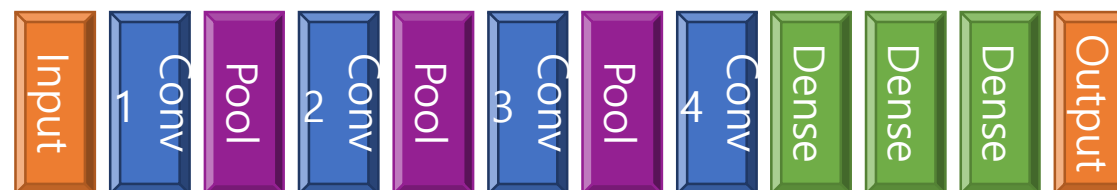
데이터 추가를 위해
뒤집기, 이동, 밝기, 그림자
이미지 생성

Udacity Self-Driving Car Nanodegree Program

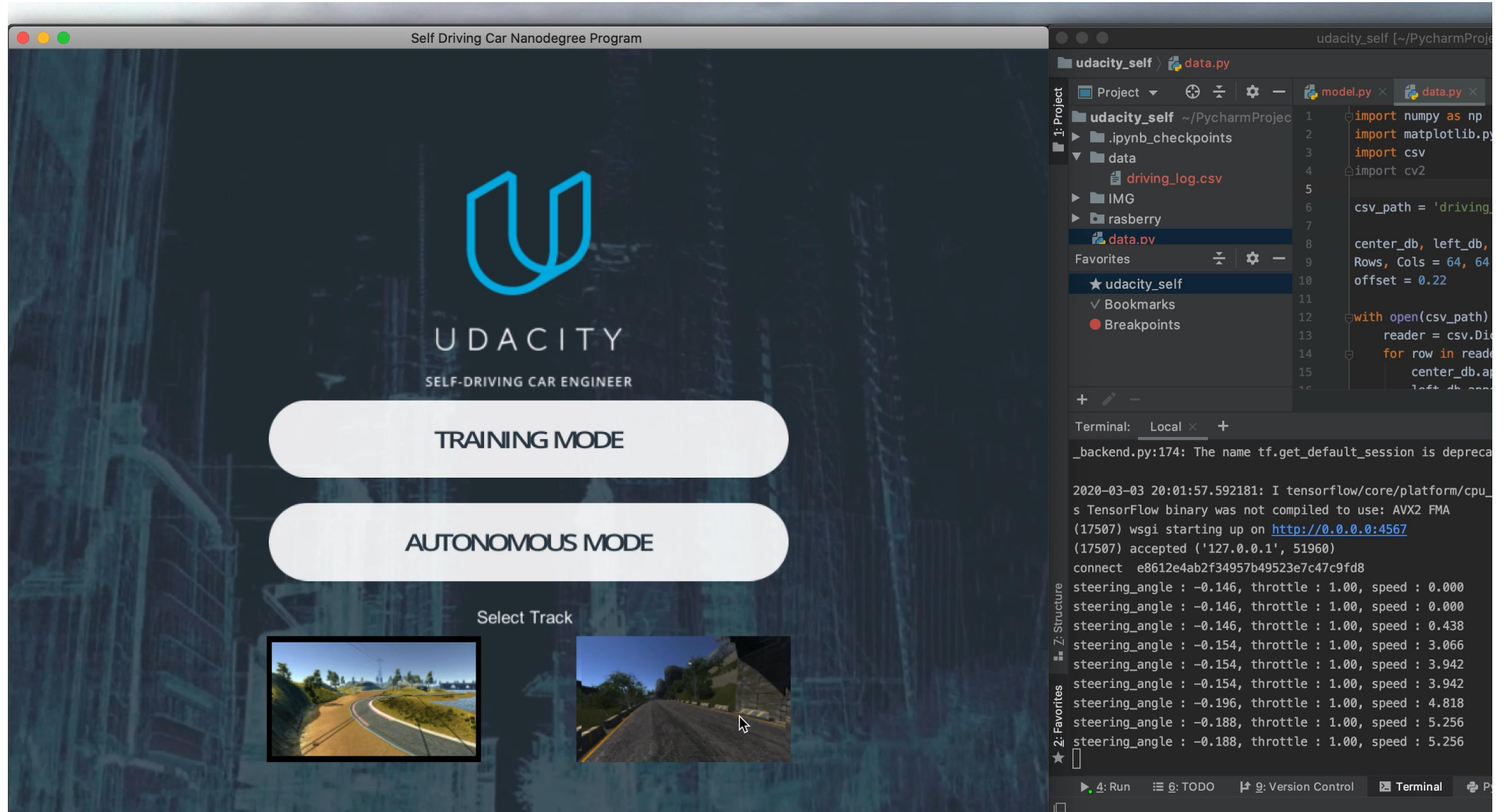
Layer (type)	Output Shape	Param #
lambda_1 (Lambda)	(None, 64, 64, 3)	0
Conv1 (Conv2D)	(None, 32, 32, 32)	896
max_pooling2d_1 (MaxPooling2)	(None, 16, 16, 32)	0
Conv2 (Conv2D)	(None, 8, 8, 64)	18496
max_pooling2d_2 (MaxPooling2)	(None, 4, 4, 64)	0
Conv3 (Conv2D)	(None, 4, 4, 128)	73856
max_pooling2d_3 (MaxPooling2)	(None, 2, 2, 128)	0
Conv4 (Conv2D)	(None, 2, 2, 128)	65664
flatten_1 (Flatten)	(None, 512)	0
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 128)	65664
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 1)	65
Total params: 249,409		
Trainable params: 249,409		
Non-trainable params: 0		



Angle : 0.1670138



Udacity Self-Driving Car Nanodegree Program(Fastest Mode)



Udacity Self-Driving Car Nanodegree Program(Fantastic Mode)



01

02

문제점 및 한계점

- ◆ 실제 상황과 비슷한 GTA5를 사용하여 학습을 하려 했으나 End to End 학습을 할 때 상대적으로 고려해야 할 변수가 많아 학습에 어려움을 겪음
- ◆ 추출한 이미지를 사용하지 않고 전체 이미지를 신경망에 전달하였으나 각 상황 별 특성을 추출하여 학습을 시켰으면 더 좋은 결과를 얻었을 것으로 예상
- ◆ 모델 학습시 W(가속) 키에 대한 입력 비중이 높아 이에 적절한 가중치 설정이 필요
- ◆ 자율 주행 구현 알고리즘이 고사양을 요구하는데 반해 준비된 하드웨어 환경이 높지 않아 테스트에 어려움
- ◆ Objective Detection을 통해 보행자, 신호, 제한 속도 등 변수까지 적용하려 했으나 이를 구현하기에 어려움을 겪음

01

02

결론

- ◆ 시뮬레이터내에서 자율 주행 자동차를 직접 구현해 보며 자율주행 원리 이해
- ◆ 이미지 처리과정에 대한 이론 학습
- ◆ CNN을 이용하여 주행 영상과 핸들 정보만으로 자율 주행을 구현 해보았는데
추후에는 Reinforcement Learning과 Lane Detection을 이용한 자율주행 차량 구현을 목표로 함
- ◆ 이를 기반으로 인공지능 드론, 항공기, 선박 등 다른 수단까지 확장 가능할 것으로 예상



- 감사합니다 -

The image is a simulated street scene, likely from a video game or a simulation software. It depicts a multi-lane road with several cars. Overlaid on the scene are various sensor and navigation elements: red dashed lines radiating from cars, representing their field of vision or sensor range; blue circular outlines around cars, possibly indicating their current position or a specific sensor's footprint; and blue arrows on the road surface, likely representing lane markings or navigation paths. In the background, there are palm trees, modern buildings, and a clear sky. Two large, semi-transparent rectangular boxes are positioned across the road, each containing text. The box on the left reads 'BAYSHORE DRIVE TRAFFIC LIGHT LAT. 30.505306, LONG. -86.479210'. The box on the right reads 'BAYSHORE DRIVE TRAFFIC LIGHT LAT. 30.505323, LONG. -86.479207'. The text is white and appears to be part of the simulation's data display.