

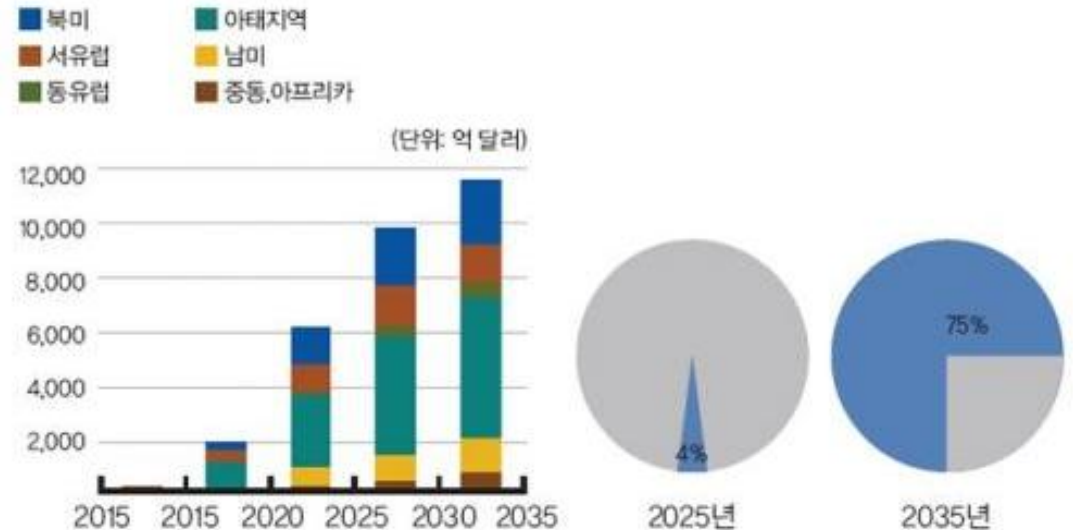
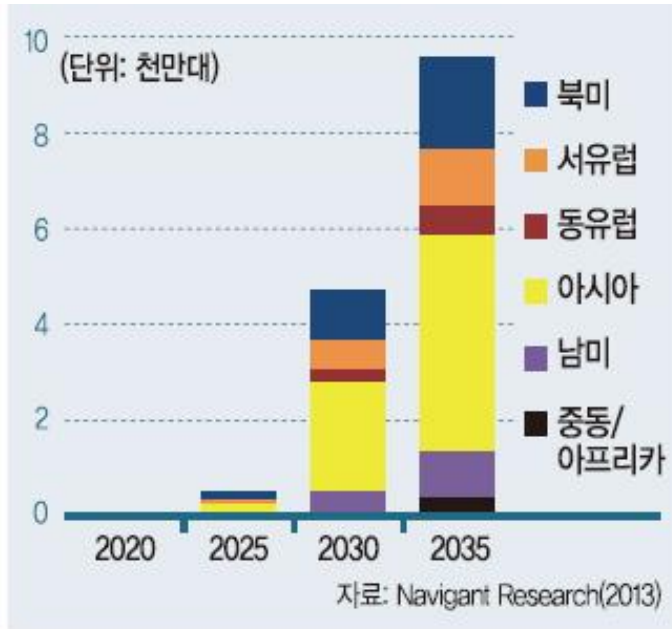
# ***Lane & Vehicle Detection in Self Driving***



**김광록**

**김원섭**

# 자율주행자동차 시장 전망 및 점유율 전망



국내 자율주행차, 선택이 아닌 **필수**

## 다양한 환경의 도로 주행 영상



태양 고도, 진행 방향에 따라 불필요한 그림자나 유리창에 비치는 반사광에 영향



## “테슬라 사고, 태양 역광 탓” ... 자율주행차 또 날씨 오작동

[중앙일보] 입력 2018.04.10 01:07 | 종합 8면 지면보기 ▶



가

가



오전 시간, 밝은 태양, **차선** 잘못 인식  
작년 중앙분리대 충돌사고와 비슷  
2년 전에 흰색 트레일러 **차량** 을 하늘로 오인

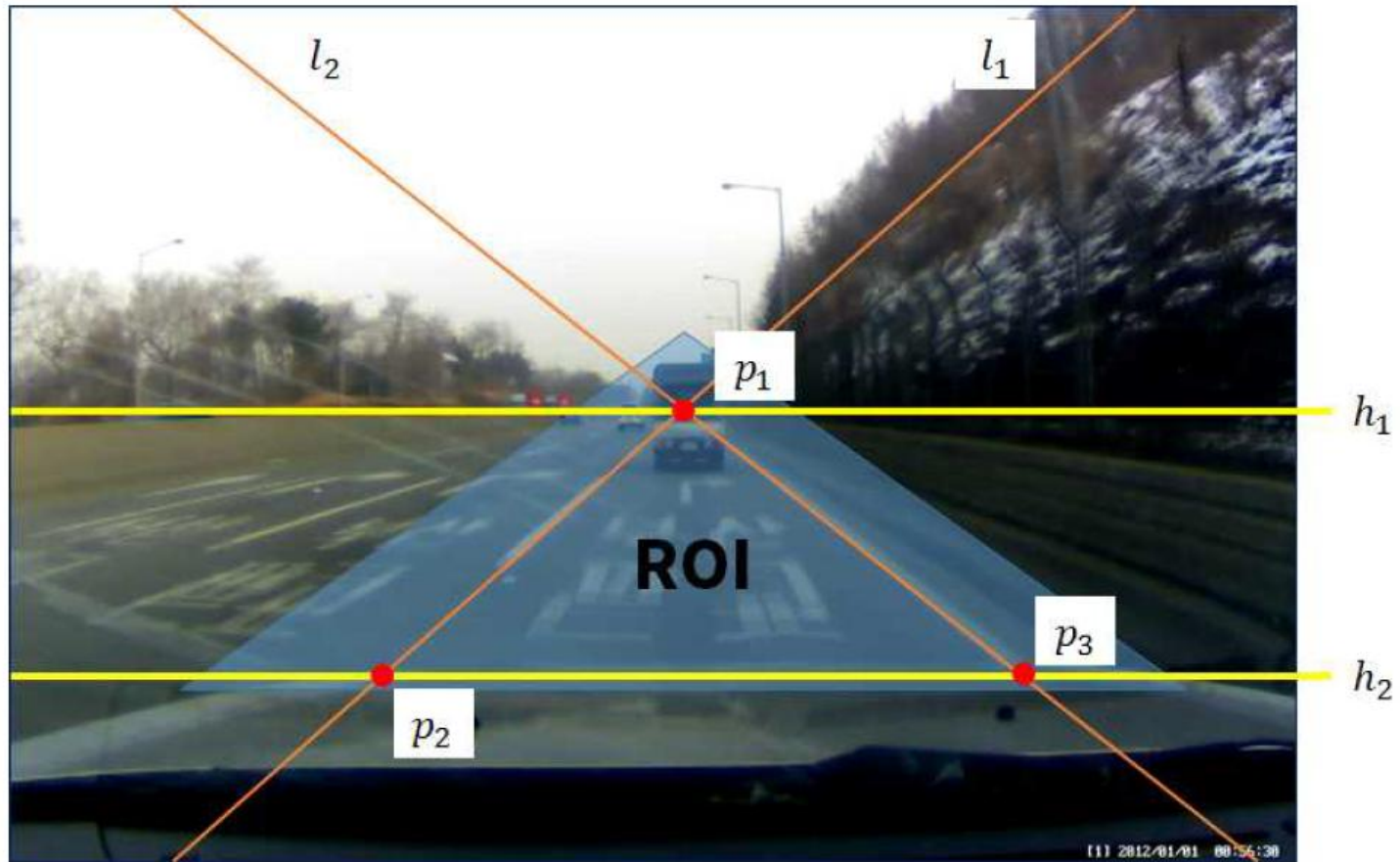
## ***원래 구현하고자 했던 영상 예시***



<https://www.youtube.com/watch?v=6hUbKyD-2X8>

# ***Lane Detection***

# ***ROI 영역, Region of Interest***



정확도 ↑ 처리하는 양 ↓ 처리 속도 ↑

단계1: 경계선 검출 (Canny Edge)

단계2: 허프변환(Hough Transform)을 통한 직선 검출

단계3: 직선들로부터 소실점 위치 판정

단계4: 수평 경계선 누적을 이용한 보닛위치 검출

단계5: 검출된 직선들을 통해 차선 폭 추정



## 단계1: 경계선 검출 (Canny Edge)



단계1: 경계선 검출 (Canny Edge)

단계2: 허프변환(Hough Transform)을 통한 직선 검출

단계3: 직선들로부터 소실점 위치 판정

단계4: 수평 경계선 누적을 이용한 보닛위치 검출

단계5: 검출된 직선들을 통해 차선 폭 추정

## **단계2: 허프변환(Hough Transform)을 통한 직선 검출**

특징	차선의 각도	차선의 시작점과 끝점 위치
왼쪽 차선	$1^{\circ} \sim 89^{\circ}$	화면의 좌하단, 우상단
오른쪽 차선	$91^{\circ} \sim 179^{\circ}$	화면의 좌상단, 우하단

단계1: 경계선 검출 (Canny Edge)

단계2: 허프변환(Hough Transform)을 통한 직선 검출

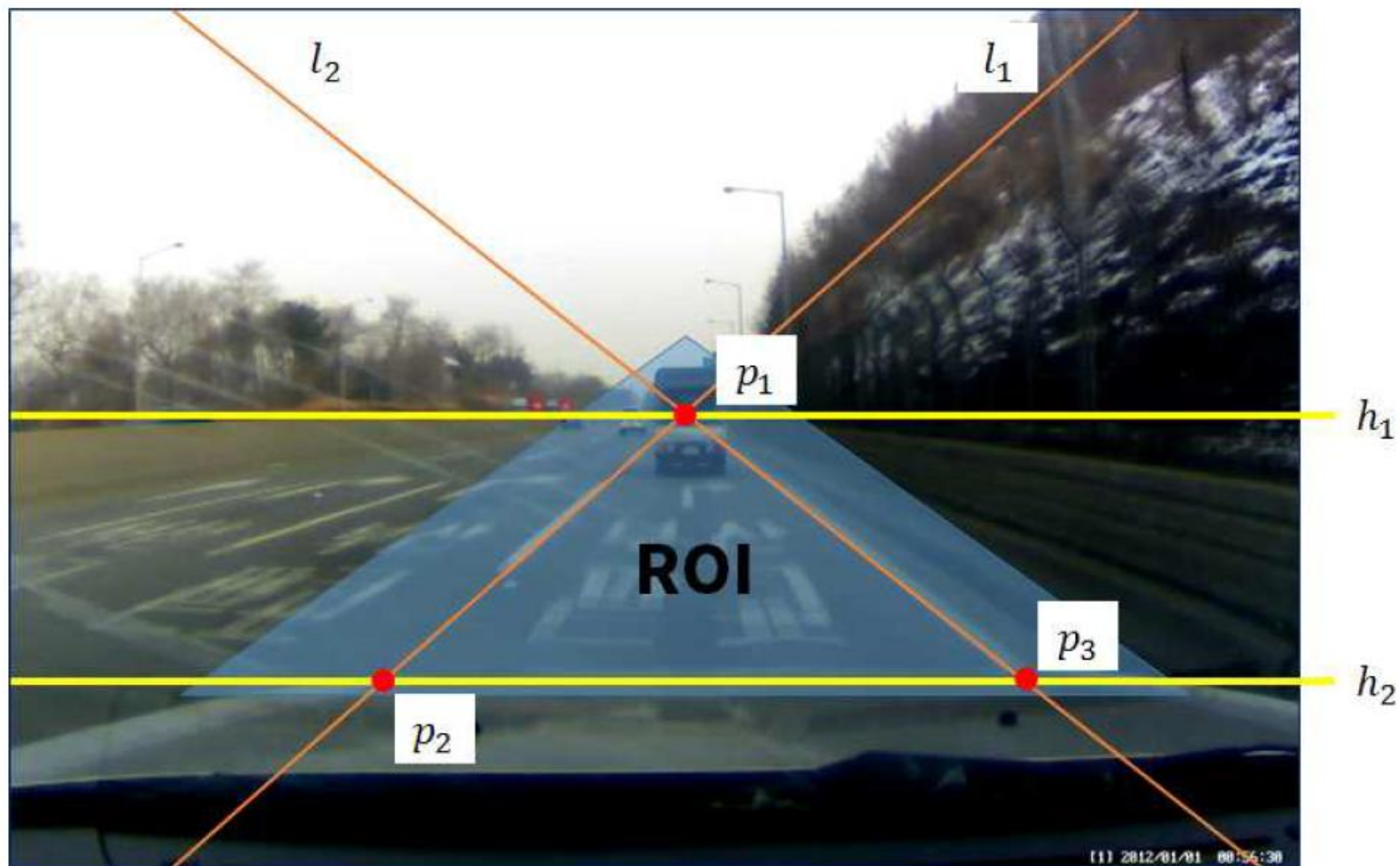
단계3: 직선들로부터 소실점 위치 판정

단계4: 수평 경계선 누적을 이용한 보닛위치 검출

단계5: 검출된 직선들을 통해 차선 폭 추정

# ***Lane Detection***

# ***캘리브레이션(Calibration) 과정***





# Computer Vision vs Deep Learning

결제카드 등록

KG Inicis

KG 이니시스에서 제공하는 결제 서비스

카드종류

개인

법인

카드번호

스캔

유효기간

01월

▼

2014년

▼

생년월일

(6자리)

비밀번호

••

아래 약관 내용에 모두 동의합니다.

[정기과금 이용약관](#)

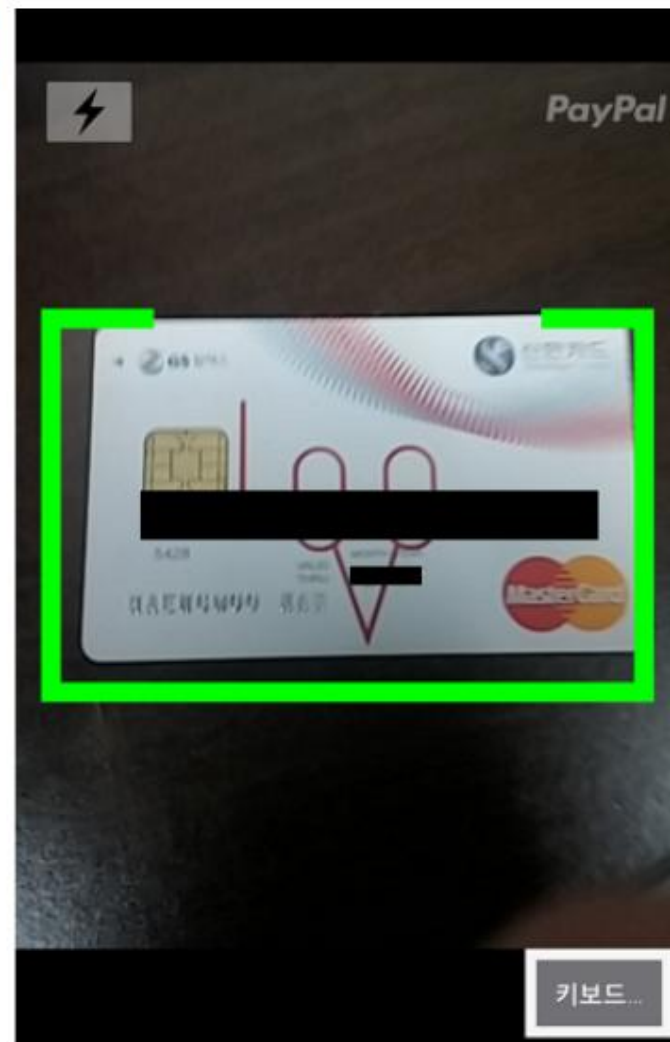
[전자금융거래 이용약관](#)

[개인정보 수집 및 이용안내](#)

[개인정보제공 및 위탁안내](#)

[금융정보 수집 및 이용안내](#)

결제카드 등록



# ***Develop environment & Tools***

Made by 건태형

운영체제

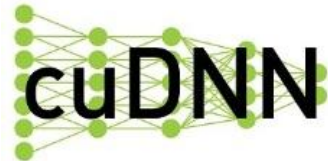
GPU 환경

언어 환경

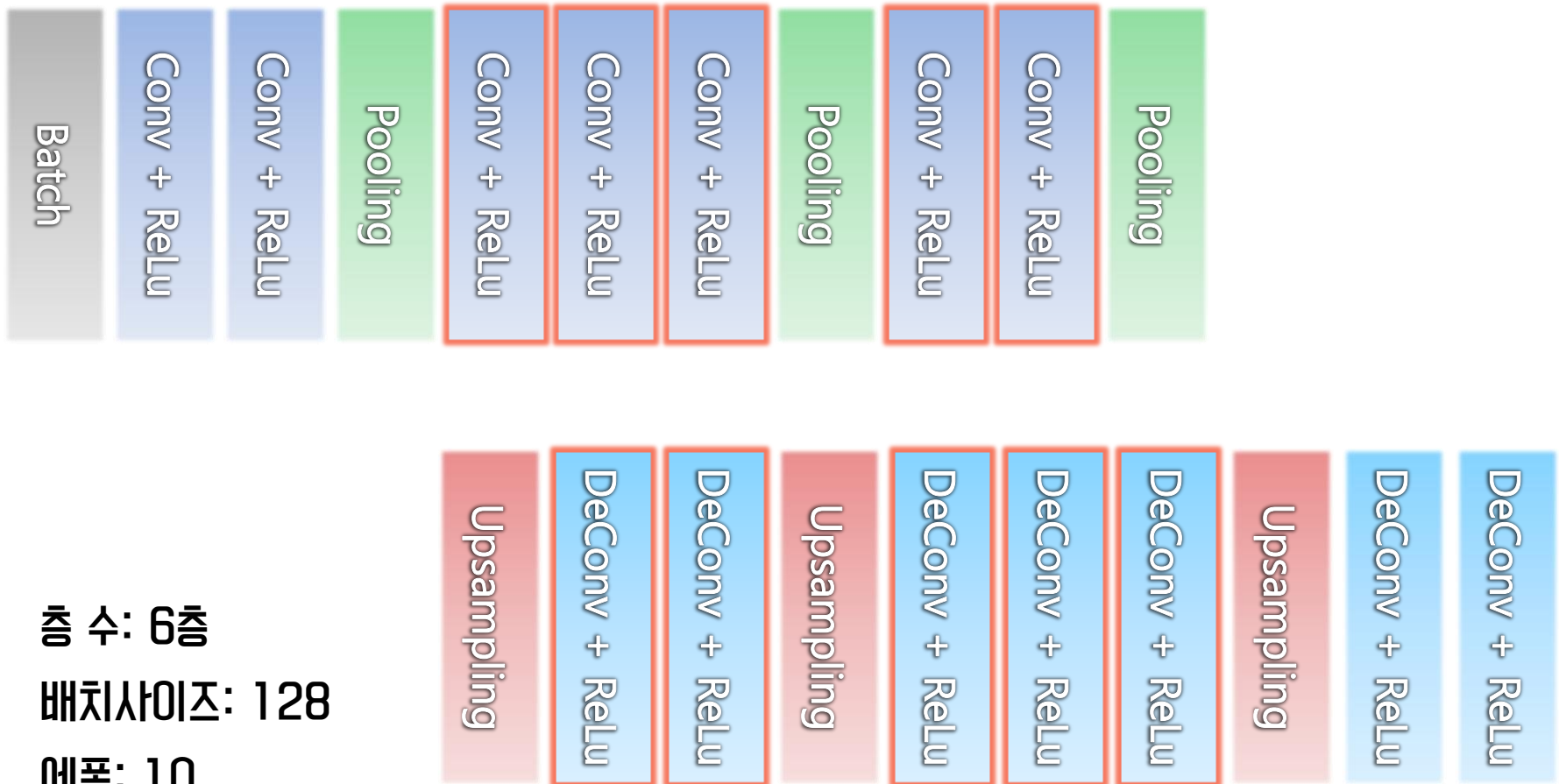
플랫폼 / 소프트웨어



**GEFORCE®  
GTX 1080**



# Network Architecture



층 수: 6층

배치사이즈: 128

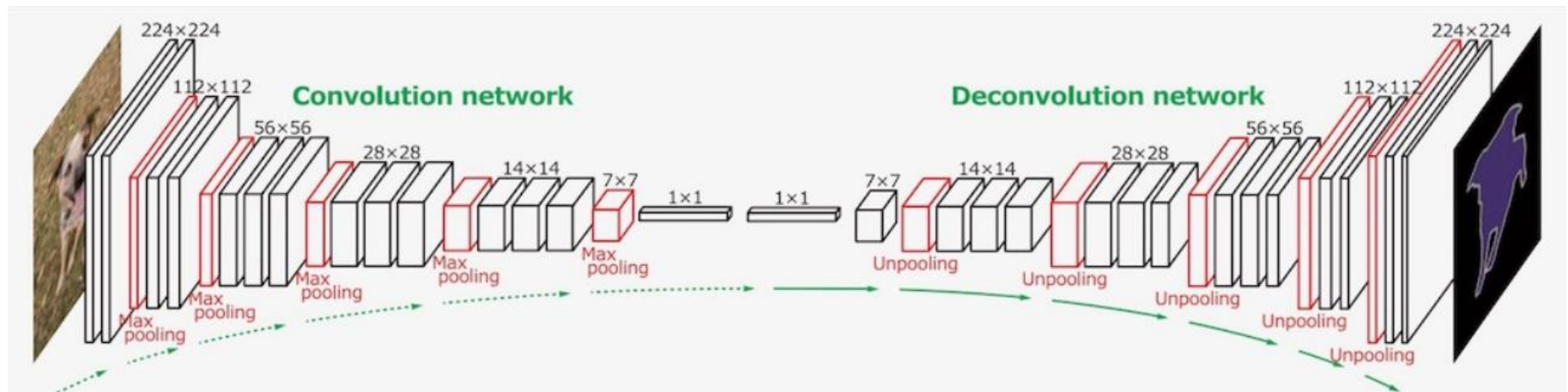
에폭: 10

경사하강법 종류: Adam

드롭아웃: 0.2

: Includes Dropout

# ***Convolution vs Deconvolution***



## 모델 돌린 결과

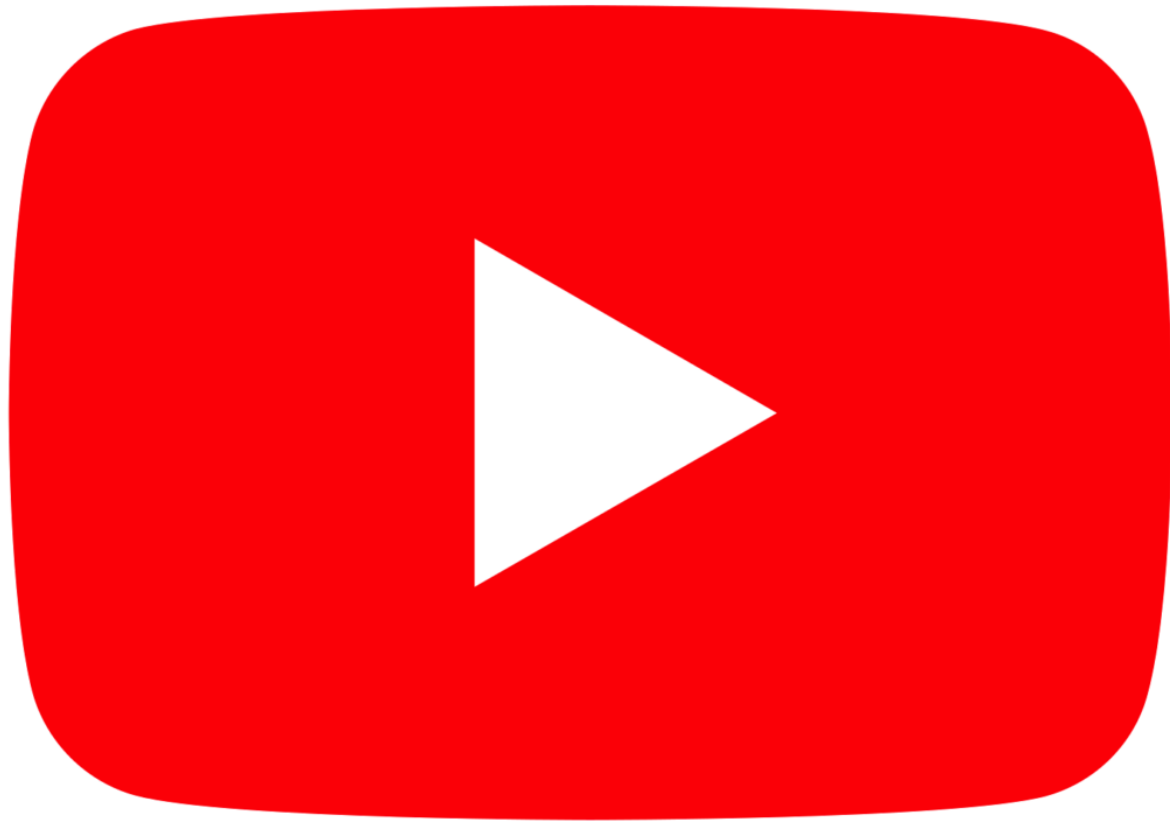
### *SGD*

```
Epoch 1/10
90/89 [=====] - 15s 163ms/step - loss: 0.1318 - val_loss: 0.1283
Epoch 2/10
90/89 [=====] - 11s 121ms/step - loss: 0.1246 - val_loss: 0.1261
Epoch 3/10
90/89 [=====] - 11s 120ms/step - loss: 0.1199 - val_loss: 0.1216
Epoch 4/10
90/89 [=====] - 11s 120ms/step - loss: 0.1130 - val_loss: 0.1136
Epoch 5/10
90/89 [=====] - 11s 119ms/step - loss: 0.1045 - val_loss: 0.1026
Epoch 6/10
90/89 [=====] - 12s 128ms/step - loss: 0.0960 - val_loss: 0.0922
Epoch 7/10
90/89 [=====] - 12s 135ms/step - loss: 0.0890 - val_loss: 0.0863
Epoch 8/10
90/89 [=====] - 11s 126ms/step - loss: 0.0843 - val_loss: 0.0829
Epoch 9/10
90/89 [=====] - 11s 120ms/step - loss: 0.0807 - val_loss: 0.0798
Epoch 10/10
90/89 [=====] - 11s 120ms/step - loss: 0.0775 - val_loss: 0.0768
<keras.engine.sequential.Sequential object at 0x00000216FF782B70>
```



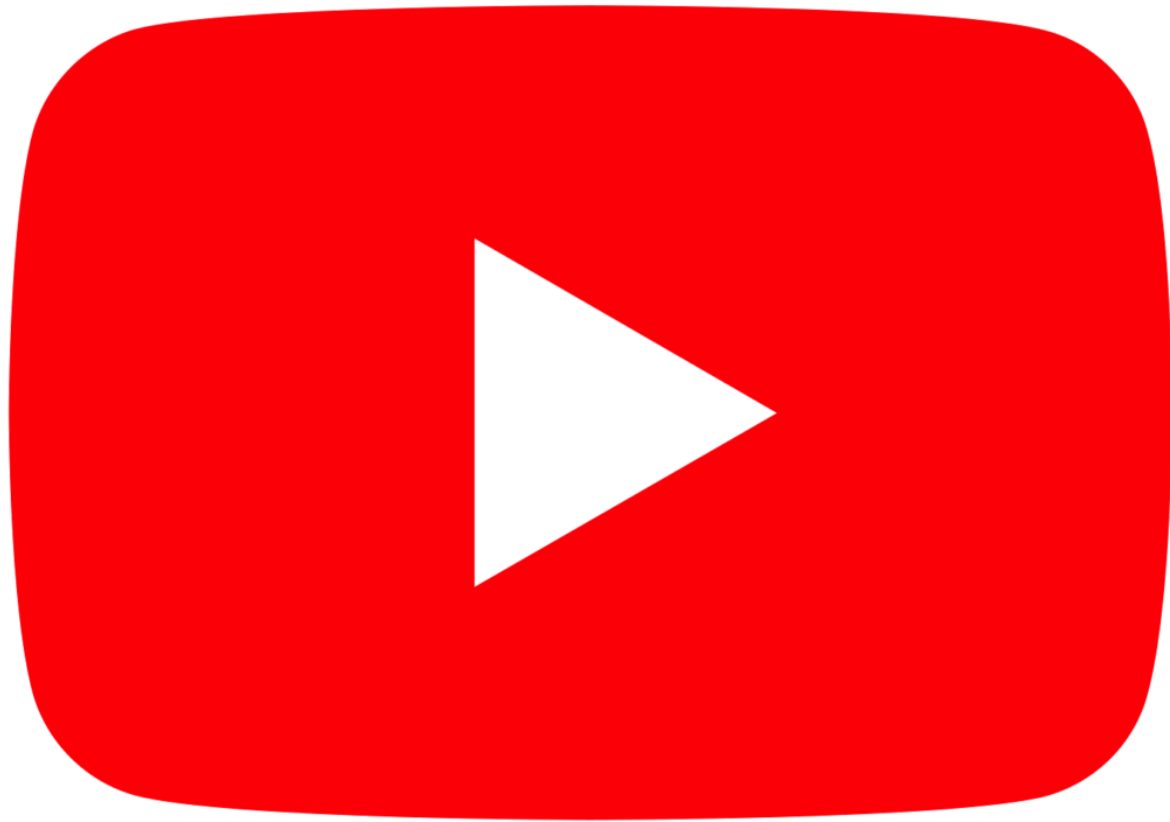
## Adam

```
Epoch 1/10
90/89 [=====] - 1113s 12s/step - loss: 0.0491 - val_loss: 0.0198
Epoch 2/10
90/89 [=====] - 1089s 12s/step - loss: 0.0146 - val_loss: 0.0113
Epoch 3/10
90/89 [=====] - 1092s 12s/step - loss: 0.0116 - val_loss: 0.0098
Epoch 4/10
90/89 [=====] - 1090s 12s/step - loss: 0.0101 - val_loss: 0.0094
Epoch 5/10
90/89 [=====] - 1088s 12s/step - loss: 0.0092 - val_loss: 0.0106
Epoch 6/10
90/89 [=====] - 1093s 12s/step - loss: 0.0087 - val_loss: 0.0088
Epoch 7/10
90/89 [=====] - 1096s 12s/step - loss: 0.0084 - val_loss: 0.0088
Epoch 8/10
90/89 [=====] - 1090s 12s/step - loss: 0.0081 - val_loss: 0.0079
Epoch 9/10
90/89 [=====] - 1095s 12s/step - loss: 0.0078 - val_loss: 0.0075
Epoch 10/10
90/89 [=====] - 1093s 12s/step - loss: 0.0073 - val_loss: 0.0064
<keras.engine.sequential.Sequential object at 0x000001CC2802ADA0>
```



[https://youtu.be/IU6R8s\\_fIZo](https://youtu.be/IU6R8s_fIZo)

***Adam***



<https://youtu.be/zRUFG4mXGOM>

# Vehicle Detection

## Vehicle Image Database



What use will you give to the database? —

to make a model

What is your major field of work? —

i'm a student

Contact

Name

Affiliation

E-mail

Comments:

Please provide additional comments

Thank you for data

Download the database

# ***Vehicle Detection***

## Vehicles



3,425 장

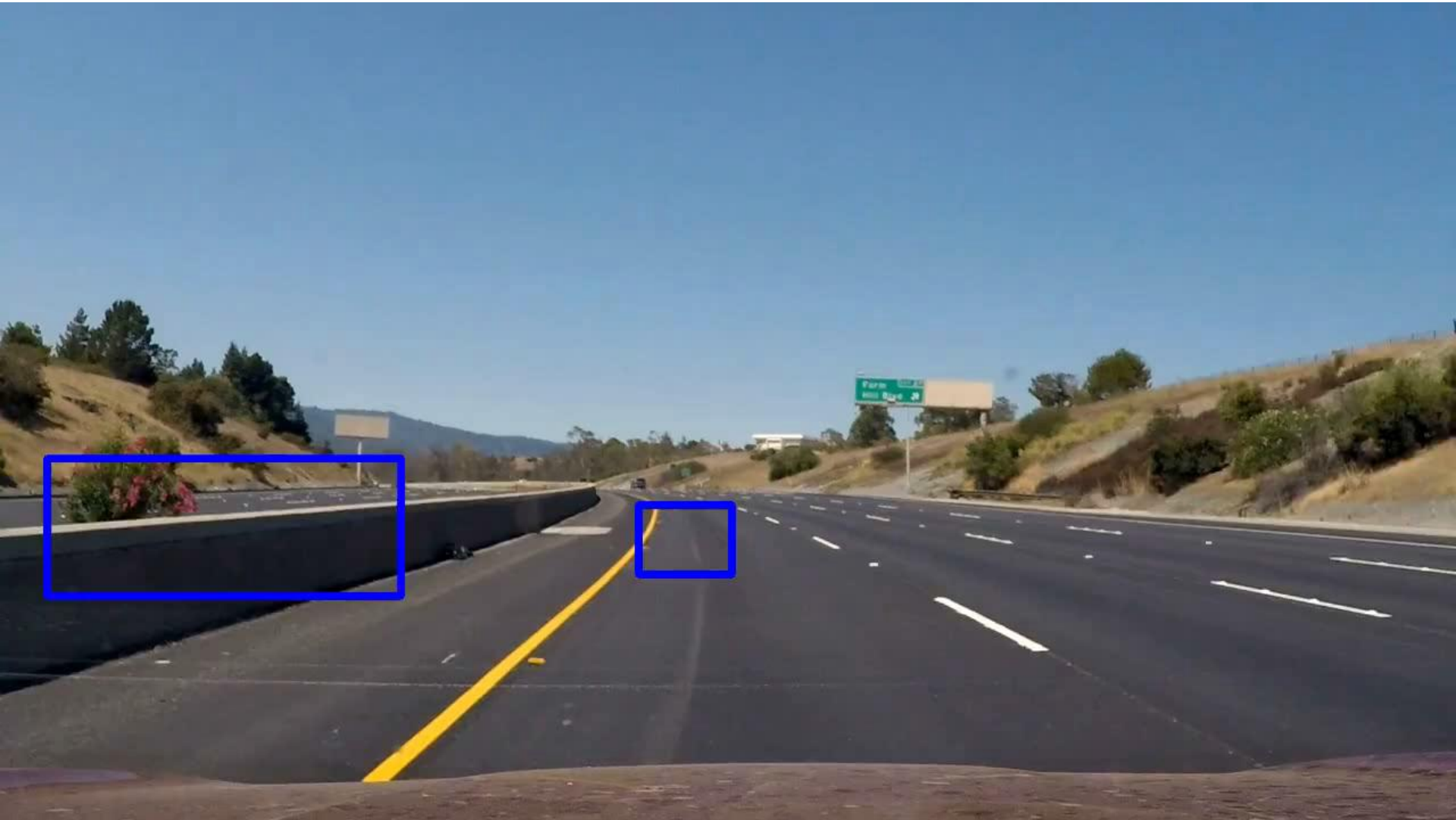
## non-Vehicles



3,900 장



# ***Vehicle Detection***



<https://www.youtube.com/watch?v=2NO0Tw6AUyl>

## ***Reference***

스마트 기기에서의 차선 및 차량 검출에 관한 연구, 2016년 2월, 한국기술대학원, 박종섭

카셰어링 쏘카 카드 스캔 사진: [blog.socar.kr/1670](http://blog.socar.kr/1670)

개발 환경 및 개발 도구 양식: 김건태

자율자동차 시장 전망: Navigant Research (2013)

테슬라 사고 기사: 중앙일보 (2018, 4)

Machine Learning Engineer Nanodegree, Capstone project,

Micheal Virgo, (2017, 5)

Vehicle image Database, GTI: [www.gti.ssr.upm.es/data/](http://www.gti.ssr.upm.es/data/)

Canny Edge 사진: [www.safaribooksonline.com](http://www.safaribooksonline.com)

Vehicle Detection: [github.com/tatsuyah/vehicle-detection](https://github.com/tatsuyah/vehicle-detection)

원래 구현하고자 했던 영상: <https://www.youtube.com/watch?v=P754S8F6Ujo>