딥러닝 활용

자동차게임 자율주행하기

대전대학교
전자정보통신공학과 4학년 재학중

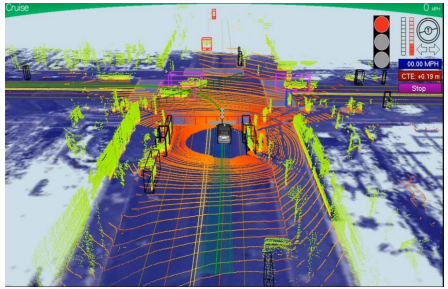
딥러닝 창업동아리(D-Lab) 운영중

BDA[BigData Analysis] Lab

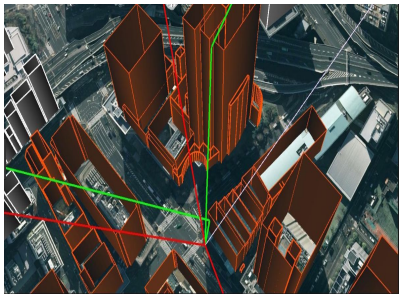
대학원 진학 준비중

자율주행을 하려면?

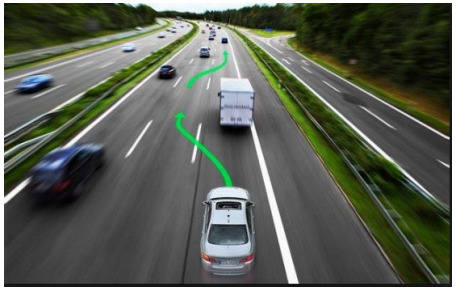
Self-driving technology



mapping

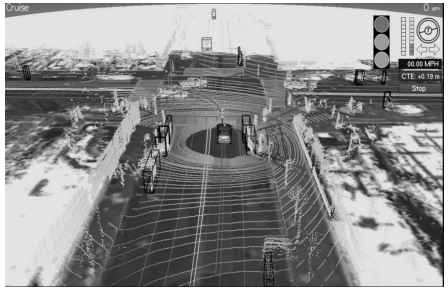


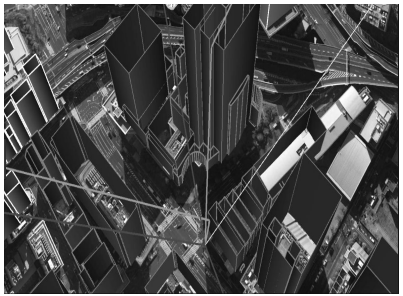
localization

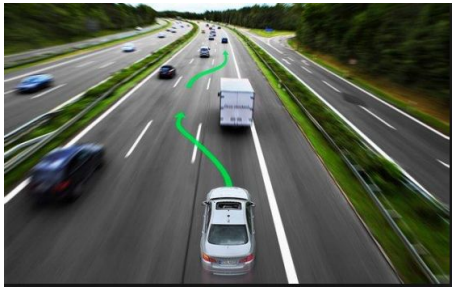


path planning

Self-driving technology







path planning

Why?
Deep Learning?

https://www.youtube.com/watch?v=zxxfvP8-lrU

End to End Learning for Self-Driving Cars

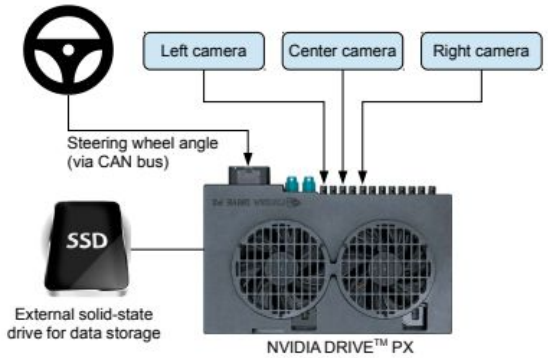


Figure 1: High-level view of the data collection system.

- Time-stamped video from the cameras is captured simultaneously with the steering angle applied by the human driver.
- Training data contains single images sampled from the video, paired with the corresponding steering command (1/r)

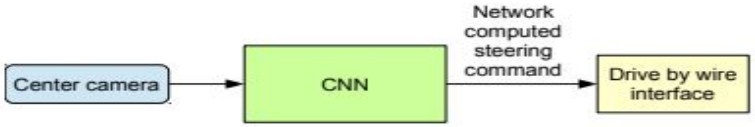


Figure 3: The trained network is used to generate steering commands from a single front-facing center camera.

Karol Zieba. End to End Learning for Self-Driving Cars 25 Apr 2016

Nvidia model

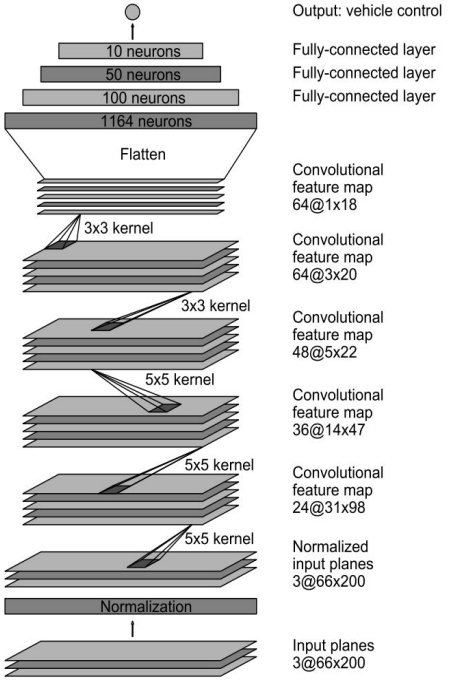
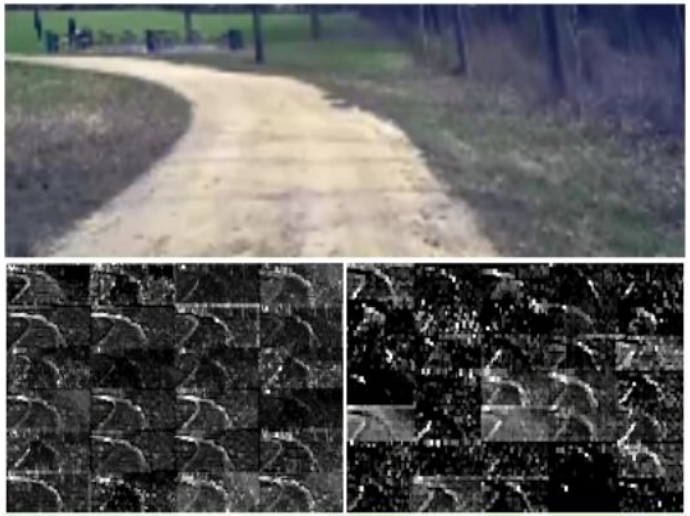
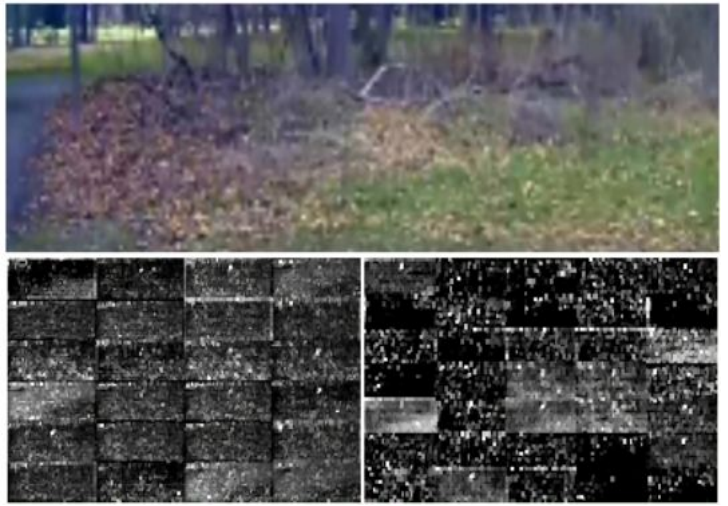


Figure 4: CNN architecture. The network has about 27 million connections and 250 thousand parameters.

```python
def get_model():
    pool_size = (2, 2)
    stride = (2, 2)
    dropout = 0.2
    model = Sequential()
    model.add(Lambda(lambda x: x/127.5 - 1.,
              input_shape=(66, 200, 3),
              output_shape=(66, 200, 3)))
    model.add(Convolution2D(5, 5, 24, subsample=stride, border_mode="same"))
    model.add(MaxPooling2D(pool_size = pool_size))
    model.add(ELU())
    model.add(Convolution2D(5, 5, 36, subsample=stride, border_mode="same"))
    model.add(ELU())
    model.add(Convolution2D(5, 5, 48, subsample=stride, border_mode="same"))
    model.add(ELU())
    model.add(Convolution2D(3, 3, 64, subsample=stride, border_mode="same"))
    model.add(ELU())
    model.add(Convolution2D(3, 3, 64, subsample=stride, border_mode="same"))
    model.add(Flatten())
    model.add(Dropout(.2))
    model.add(ELU())
    model.add(Dense(1164))
    model.add(Dropout(dropout))
    model.add(ELU())
    model.add(Dense(100))
    model.add(Dropout(dropout))
    model.add(ELU())
    model.add(Dense(50))
    model.add(Dropout(dropout))
    model.add(ELU())
    model.add(Dense(10))
    model.add(Dropout(dropout))
    model.add(ELU())
    model.add(Dense(1))
    adam = Adam(lr = 0.00001)
    model.compile(optimizer=adam, loss="mse", metrics = ['accuracy'])
    return model
```





Bottom left: Activation of the first layer feature maps.
Bottom right: Activation of the second layer feature maps

The activations of the first two feature maps appear to contain mostly noise

Video Test

First
Let's get it using the game

GTA5

Data Collection

Image

Label

A, W, S, D

방향키

Data Collection

[0]	[1]
	[0, 1, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0]

[A, W, D, S, AW, WD, AS, SD, 0]

Make the custom data

Window API사용
화면 캡쳐

Window API사용
키보드 방향키 입력

Numpy형식으로 저장

이미지 데이터	라벨
[200, 200, 200, 199, 199, 199, 199, 198, 198,...]	[0, 1, 0]
[198, 200, 195, 199, 199, 199, 199, 198, 198,...]	[0, 0, 1]
[192, 191, 191, 191, 191, 191, 191, 191, 190,...]	[0, 0, 1]

```python
def grab_screen(region=None, title=None):
    hwin = win32gui.GetDesktopWindow()
    if region:
        left,top,x2,y2 = region
        width = x2 - left + 1
        height = y2 - top + 1
    elif title:
        gtawin = win32gui.FindWindow(None, title)
        if not gtawin:
            raise Exception('window title not found')
        #get the bounding box of the window
        left, top, x2, y2 = win32gui.GetWindowRect(gtawin)
        width = x2 - left +1
        height = y2 - top +1
    else:
        width = win32api.GetSystemMetrics(win32con.SM_CXVIRTUALSCREEN)
        height = win32api.GetSystemMetrics(win32con.SM_CYVIRTUALSCREEN)
        left = win32api.GetSystemMetrics(win32con.SM_XVIRTUALSCREEN)
        top = win32api.GetSystemMetrics(win32con.SM_YVIRTUALSCREEN)

    hwindc = win32gui.GetWindowDC(hwin)
    srcdc = win32ui.CreateDCFromHandle(hwindc)
    memdc = srcdc.CreateCompatibleDC()
    bmp = win32ui.CreateBitmap()
    bmp.CreateCompatibleBitmap(srcdc, width, height)
    memdc.SelectObject(bmp)
    memdc.BitBlt((0, 0), (width, height), srcdc, (left, top), win32con.SRCCOPY)

    signedIntsArray = bmp.GetBitmapBits(True)
    img = np.fromstring(signedIntsArray, dtype='uint8')
    img.shape = (height,width,4)

    srcdc.DeleteDC()
    memdc.DeleteDC()
    win32gui.ReleaseDC(hwin, hwindc)
    win32gui.DeleteObject(bmp.GetHandle())

    return cv2.cvtColor(img, cv2.COLOR_BGRA2RGB)
```

Win32.GetAsyncKeyState()사용

Image

key value

Now you have to drive hard.

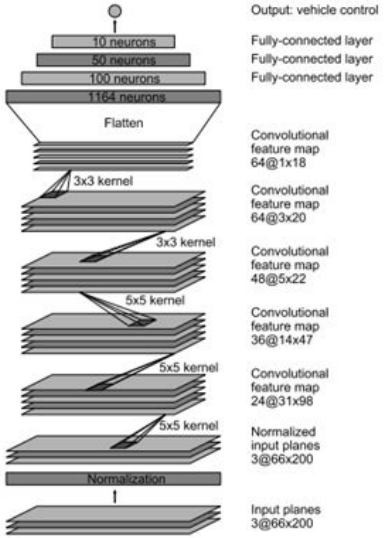
아주 많이....

https://www.youtube.com
/watch?v=xJ7PWTNdcnk

Train

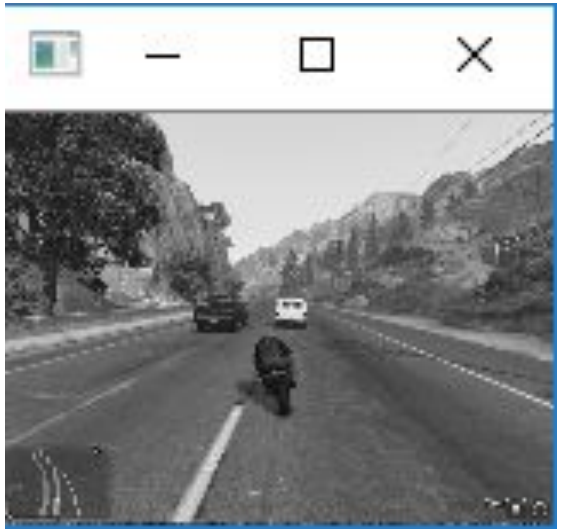
AlexNet, Inception, VGG, Resnet 등등 ⇨ from keras.applications.'모델명' import '모델명

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201



Karol Zieba. End to End Learning for Self-Driving Cars 25 Apr 2016

Train Data



gray image

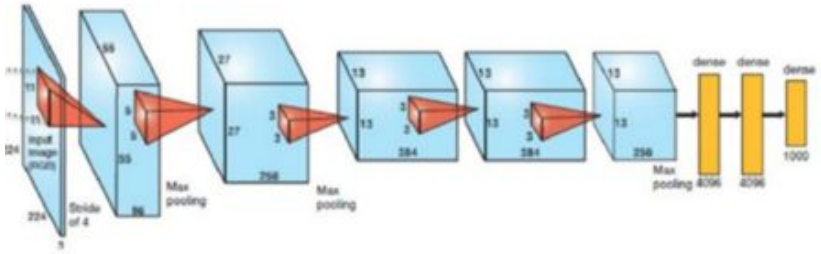
Size: 120x80

300K

Train time : 60H
(CPU : i5
GPU: X)

Test

Game
screen



Standard AlexNet architecture adapted to this dataset

방향키(확률 값)

Test Video

https://www.youtube.com/watch?v=32ywp_y1QEc

Future Work

Driving Data

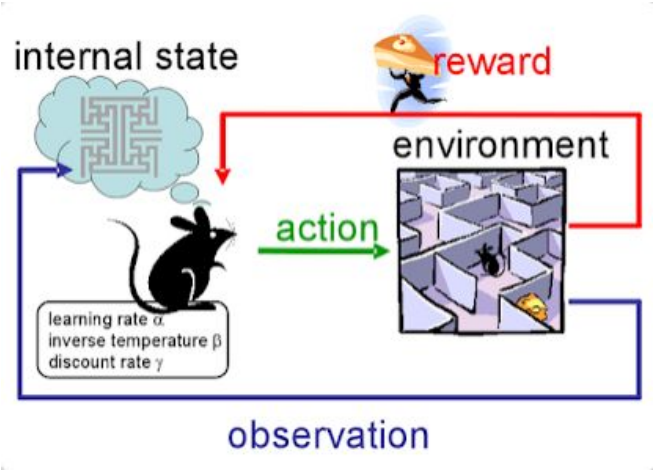
480 x 270 RGB Image

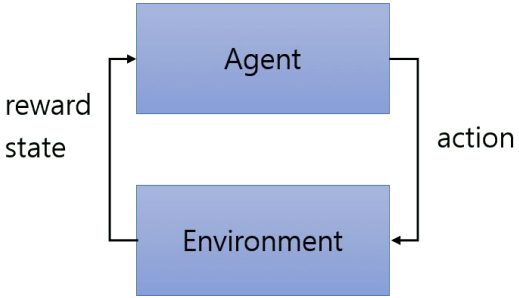
600,000 Image

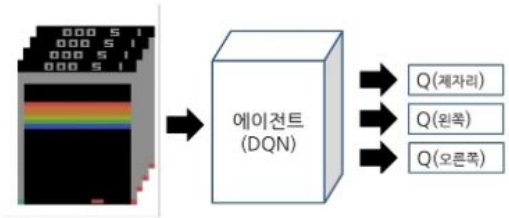
Data collection after navigation setup

Object Detection

Reinforcement Learning

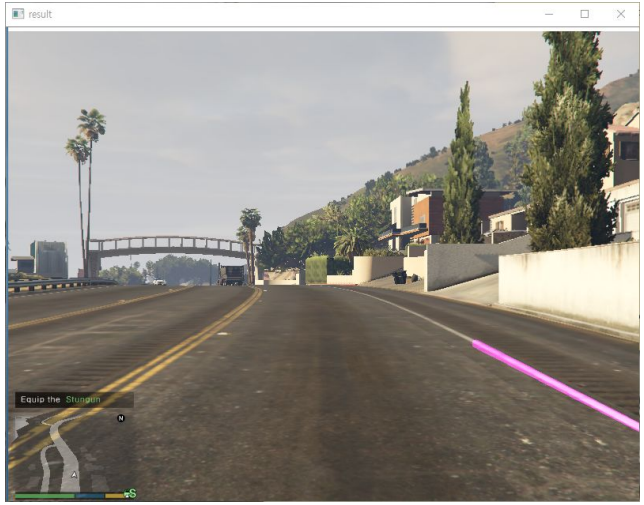


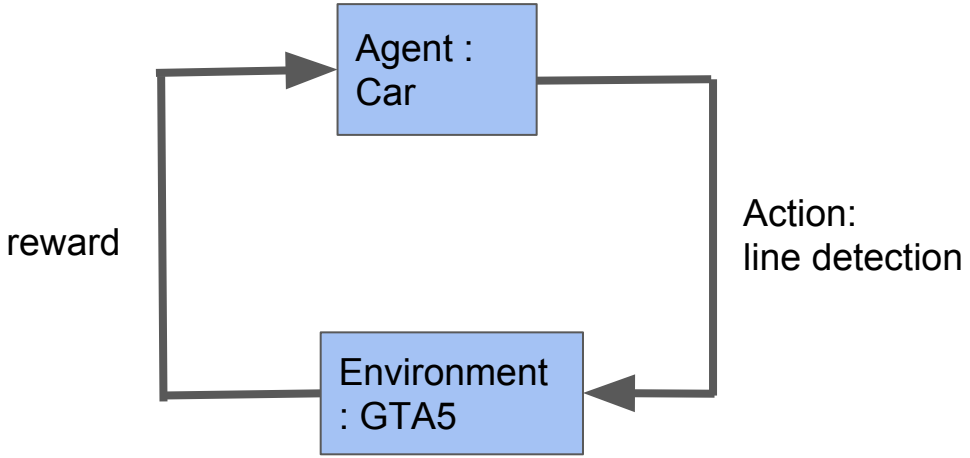




https://www.slideshare.net/WoongwonLee/ss-78783597?from_m_app=android

Reinforcement Learning





Q & A

감사합니다.