Self-driving for GTA5

발표자 소개

대전대학교
전자정보통신공학과 4학년 재학 중

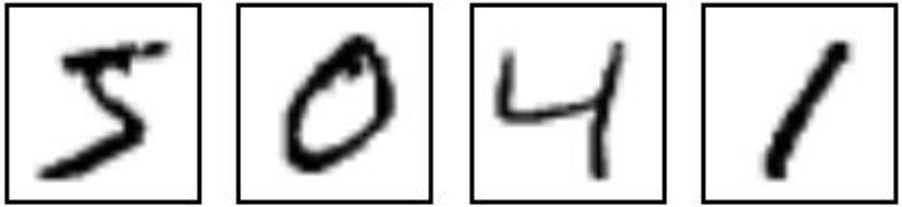
BDA Lab

대학원 진학 준비 중

Kaggle-KR

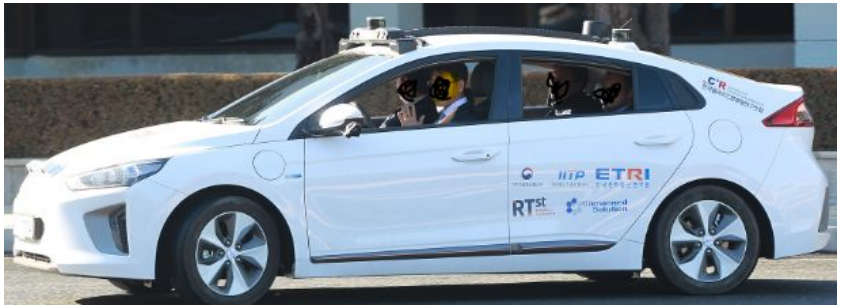
ETRI 인턴 중

이제 뭐하지...?



5 0 4 1 Acc:99.7%

내가 자율주행 자동차를 만들 수 있을까??

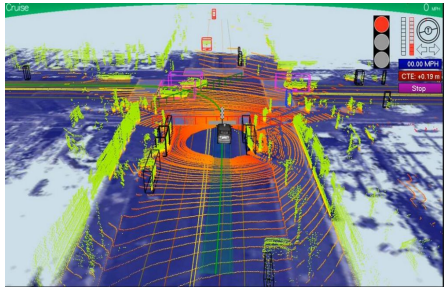


http://m.bsnews.kr/news/articleView.html?idxno=6703

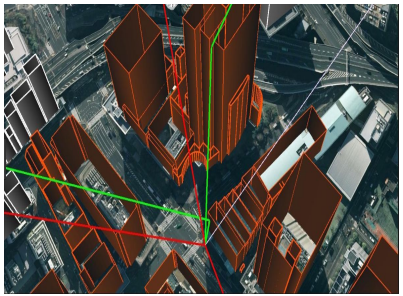
여러 **삽질** 끝에
자율주행 도전‼

자율주행을 위한 **삽질** 이야기
시작됩니다.

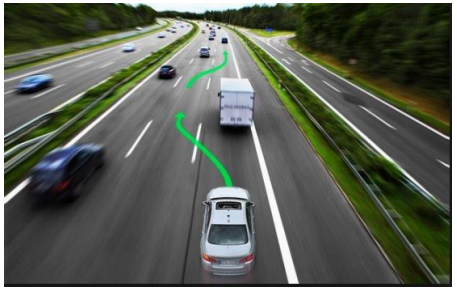
Self-driving technology



mapping

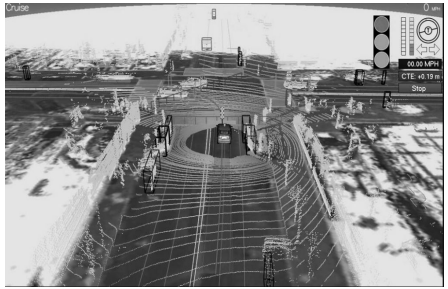


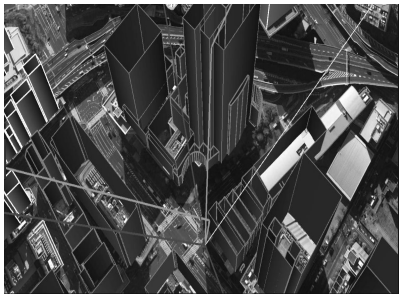
localization

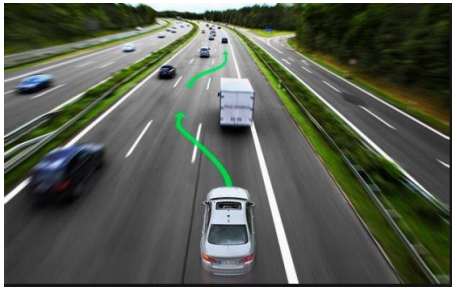


path planning

Self-driving technology







path planning

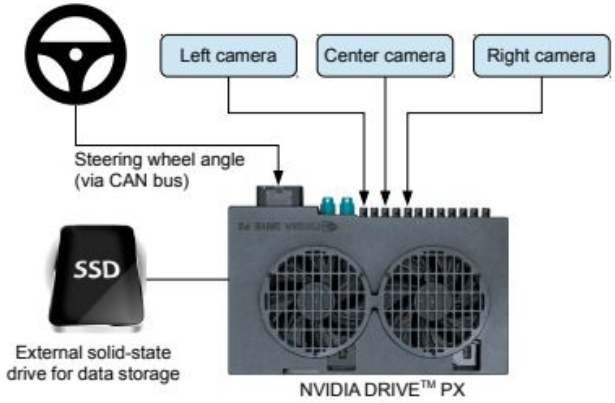


Figure 1: High-level view of the data collection system.

- 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)

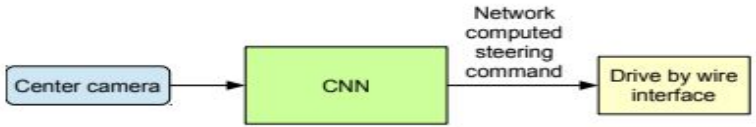
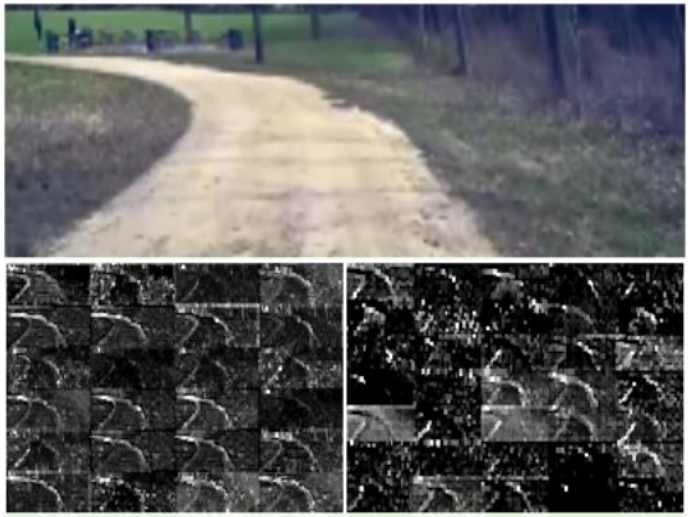


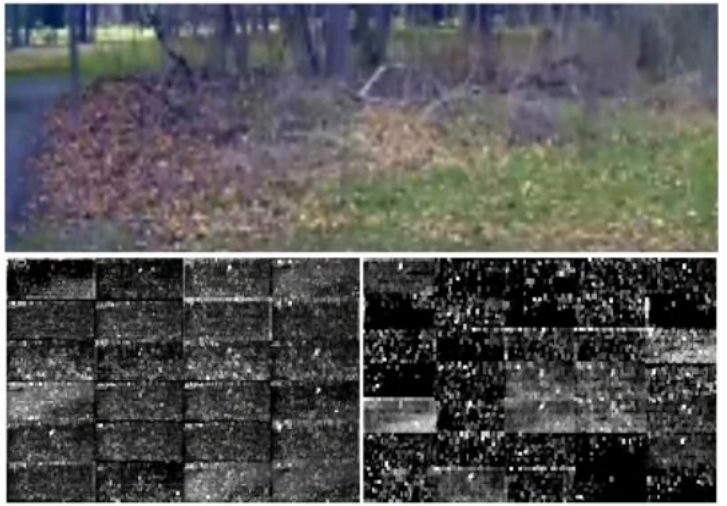
Figure 3: The trained network is used to generate steering commands from a single front-facing center camera.

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

End to End Learning for Self-Driving Cars



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


self-driving car test run
steering wheel
CLEAR
KEEP

주행데이터는 어디에서…

게임속에서 모을 수 있지 않을까?

Supervised Learning

GTA5 이용

Why GTA5?

Real

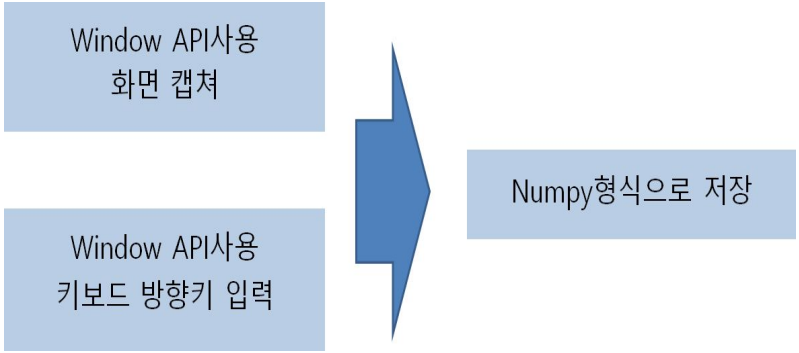
Game

Image

Label

A, W, S, D

방향키



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]

이제 게임을 열심히 하시면 됩니다!!!

언제까지?

질릴때 까지!!!

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

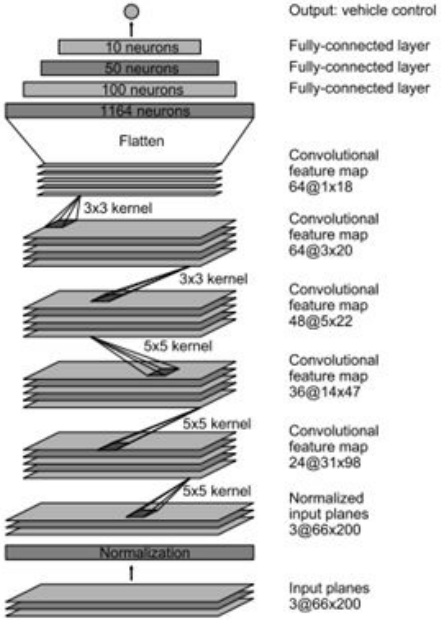
대학생&직장인의 일주일

워어어어어어어어얼 화아아아아아아 수우우우우우 모오오오오오오옥 금으음 퇠

딥러닝은?

데이터수우우우우우우우우집 전처어어어어어어어리 하아아아아아아아아아악습 테스트

Train



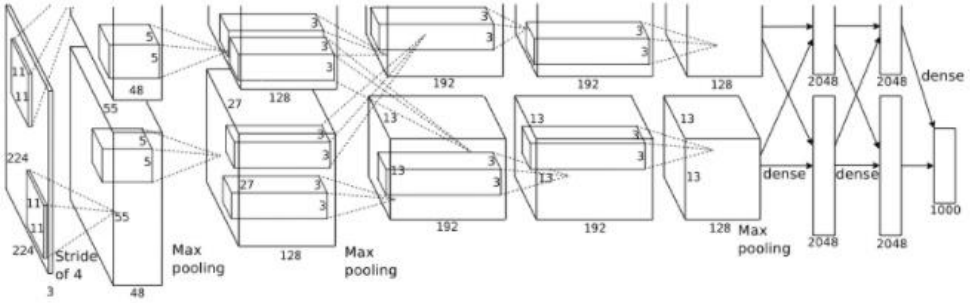
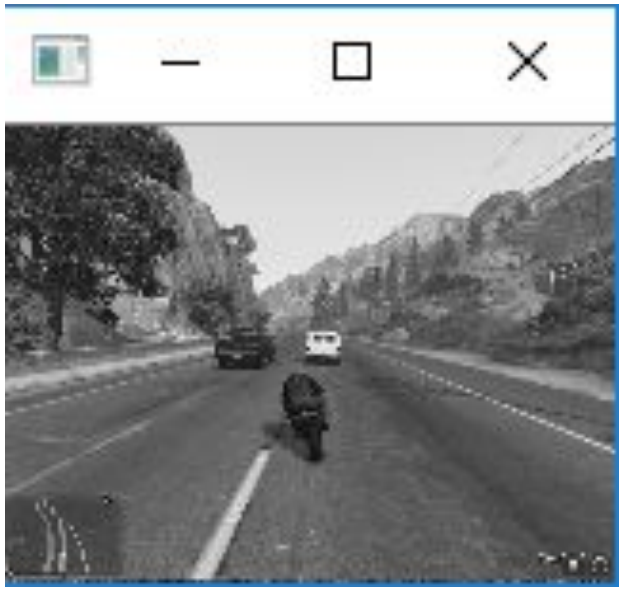


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

AlexNet

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

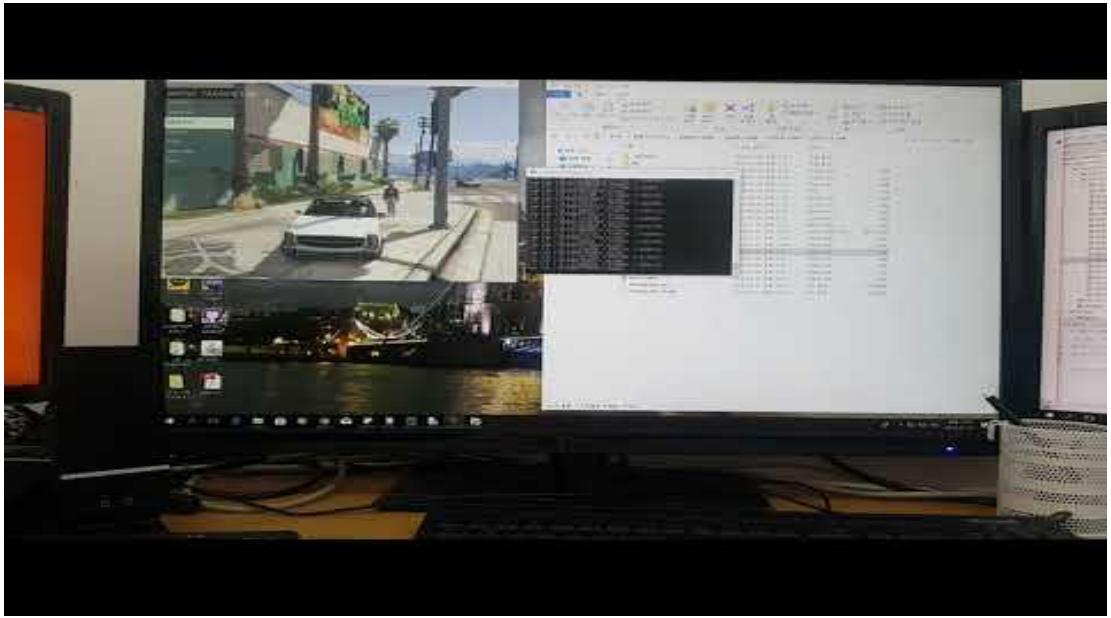


160 x 120

gray image 300K

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

데이터가 부족하면?



https://www.youtube.com/watch?v=7h1Ts784bTg&feature=youtu.be

Test Video

https://youtu.be/B5nhJUo5S6g

Test Video

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

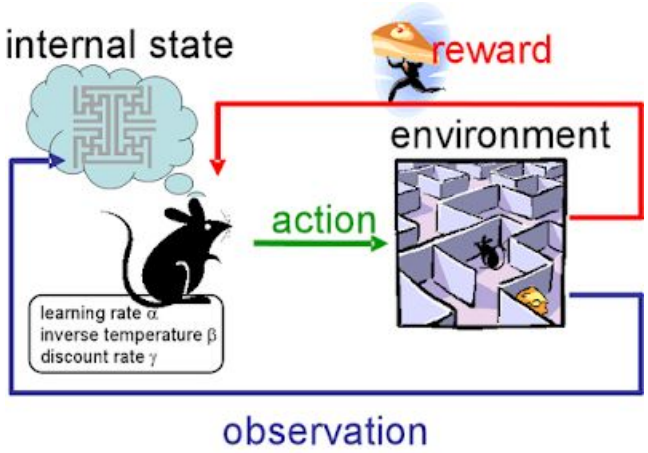
Reinforcement Learning

앞 **발표**를 통해 이제는 너무나 잘 아는

강화학습

Definition

Learn to make good sequences of decisions



agent : 쥐

environment : 미로

reward : 치즈

action : 방향

Environment


```
import gym

env = gym.make('CartPole-v1')

next_state, reward, done, info = env.step(action)
```

env = gym.make(‘GTA5’)???????

GTA5는 **GYM**환경도 없고
API도 없다...!!

API를 어떻게 만들지?

오직 **영상처리**로!!

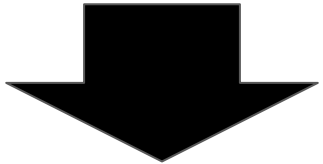
Environment

Agent


습격에 적합한 도주
차량입니다. 로스 산토스
커스텀에서 더욱 빠르고
튼튼하게 개조할 수 있습니다.
그런 다음 "도주 차량 준비"를
선택해서 레스터에게
전화하십시오.

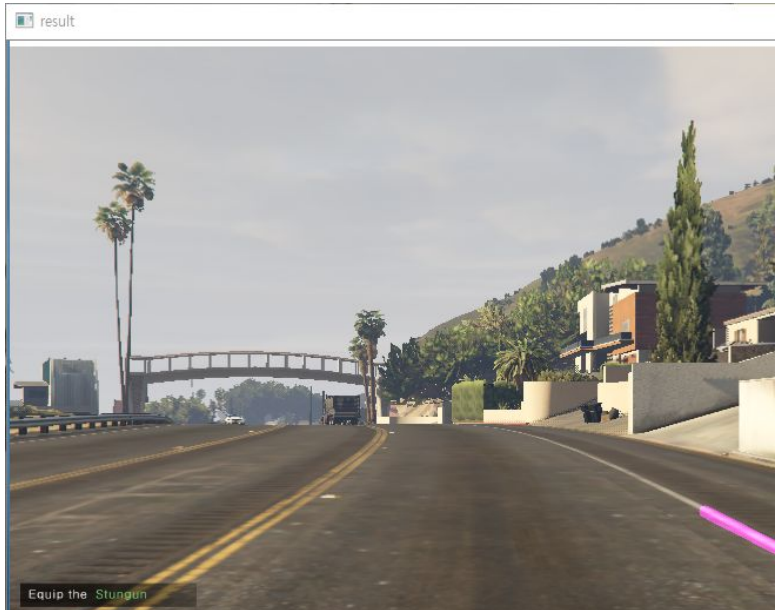
Action

A(좌회전)　　W(직진)　　D(우회전)



W　A　D　S　AW　DW　AS　DS

처음에는…



차선을 검출하면

보상**1**

Reward

앞차와의 거리가 **0.3**

보상 **1**

앞차와의 거리가 **0.1** 이하

보상 **-1**

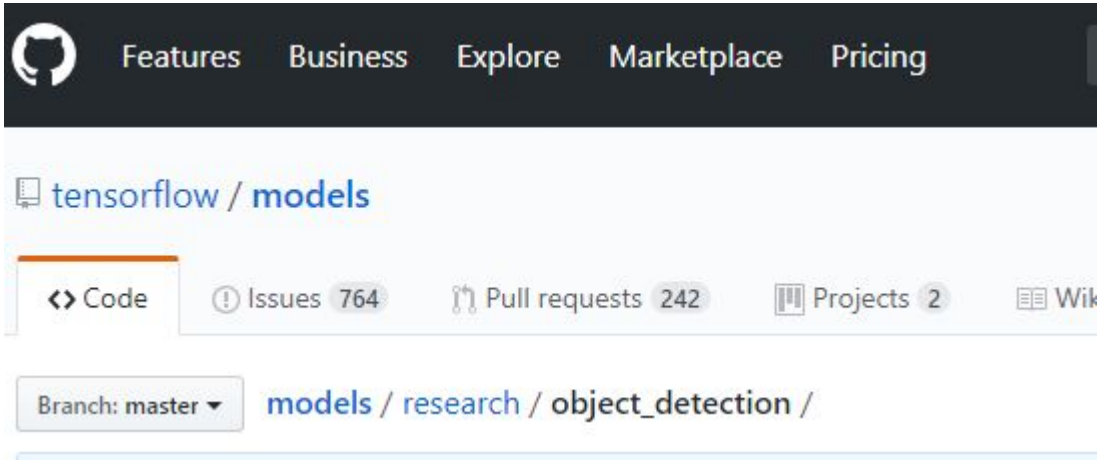
2D 영상으로 어떻게 거리를 측정할까?

쉽게 쉽게 해봐요~

Reward

Tensorflow
Object Detection API 사용

Reward



Tensorflow Github -> models->research -> object-detection

Reward

```
with detection_graph.as_default():
    with tf.Session(graph=detection_graph) as sess:
        while True:

            #ret, image_np = cap.read()
            image_np = grab_screen(region=(0, 40, 800, 640))
            image_np = cv2.cvtColor(image_np, cv2.COLOR_BGR2RGB)
            # Expand dimensions since the model expects images to have shape: [1, None, None, 3]
            image_np_expanded = np.expand_dims(image_np, axis=0)
            image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
            # Each box represents a part of the image where a particular object was detected.
            boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

            # Each score represent how level of confidence for each of the objects.
            # Score is shown on the result image, together with the class label.
            scores = detection_graph.get_tensor_by_name('detection_scores:0')
            classes = detection_graph.get_tensor_by_name('detection_classes:0')
            num_detections = detection_graph.get_tensor_by_name('num_detections:0')
            # Actual detection.
            (boxes, scores, classes, num_detections) = sess.run(
                [boxes, scores, classes, num_detections],
                feed_dict={image_tensor: image_np_expanded})
            # Visualization of the results of a detection.
            vis_util.visualize_boxes_and_labels_on_image_array(
                image_np,
                np.squeeze(boxes),
                np.squeeze(classes).astype(np.int32),
                np.squeeze(scores),
                category_index,
                use_normalized_coordinates=True,
                line_thickness=8)
```

Reward

```
for i, b in enumerate(boxes[0]):
```

i 를 사용

```
classes[0][i] == ‘number’
```

Branch: master ▾ **models** / research / object_detection / data / **mscoco_label_map.pbtxt**

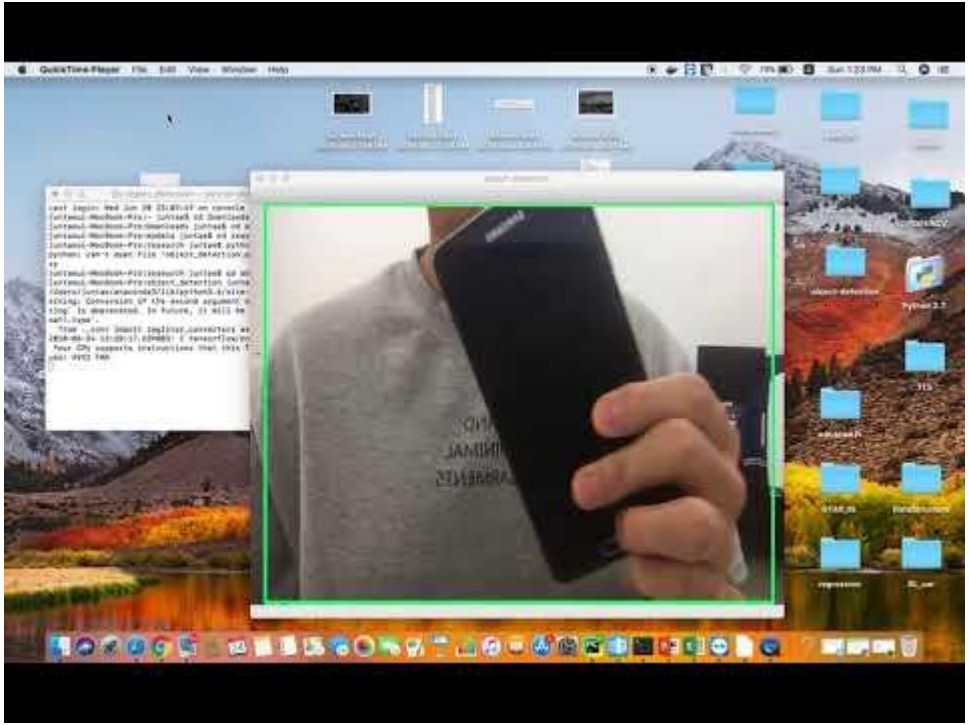
nealwu Move the research models into a research subfolder (#2430)

1 contributor

401 lines (400 sloc) | 4.94 KB

```
item {
  name: "/m/01g317"
  id: 1
  display_name: "person"
}
item {
  name: "/m/0199g"
  id: 2
  display_name: "bicycle"
}
item {
  name: "/m/0k4j"
  id: 3
  display_name: "car"
}
item {
  name: "/m/04_sv"
  id: 4
  display_name: "motorcycle"
}
item {
  name: "/m/05czz6l"
  id: 5
  display_name: "airplane"
}
```

Reward



Reward

```
        tts.play('이 것은 포크입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 49:
        tts.play('이 것은 나이프입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 50:
        tts.play('이 것은 숟가락입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 51:
        tts.play('이 것은 그릇입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 52:
        tts.play('이 것은 바나나입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 53:
        tts.play('이 것은 사과입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 54:
        tts.play('이 것은 샌드위치입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 55:
        tts.play('이 것은 오렌지입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 56:
        tts.play('이 것은 브로콜리입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 57:
        tts.play('이 것은 당근입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 58:
        tts.play('이 것은 핫도그입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 59:
        tts.play('이 것은 피자입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 60:
        tts.play('이 것은 도넛입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 61:
        tts.play('이 것은 케이크입니다')
        playsound('tmp.mp3')
    elif classes[0][i] == 62:
        tts.play('이 것은 의자입니다')
        playsound('tmp.mp3')
```

```
        tts.play('이것은 코끼리입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 23:
        tts.play('이것은 곰입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 24:
        tts.play('이것은 얼룩말입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 25:
        tts.play('이것은 기린입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 27:
        tts.play('이것은 배낭입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 28:
        tts.play('이것은 우산입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 31:
        tts.play('이것은 핸드백입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 32:
        tts.play('이것은 타이입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 33:
        tts.play('이것은 캐리어입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 35:
        tts.play('이것은 스키입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 36:
        tts.play('이것은 스노우보드 입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 37:
        tts.play('이것은 공입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 38:
        tts.play('이것은 연입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 39:
        tts.play('이것은 야구배트입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 40:
        tts.play('이것은 야구글러브 입니다.')
        playsound('tmp.mp3')
```

```
        tts.play('이것은 차입니다.')
        playsound('tmp.mp3')
    elif classes[0][i]==4:
        tts.play('이것은 오토바이입니다.')
        playsound('tmp.mp3')
    elif classes[0][i]==5:
        tts.play('이것은 비행기입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 6:
        tts.play('이것은 버스입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 7:
        tts.play('이것은 기차입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 8:
        tts.play('이것은 트럭입니다.')
        playsound('tmp.mp3')
    elif classes[0][i]==9:
        tts.play('이것은 보트입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 10:
        tts.play('이것은 신호등입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 15:
        tts.play('이것은 벤치입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 16:
        tts.play('이것은 새입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 17:
        tts.play('이것은 고양이입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 18:
        tts.play('이것은 개입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 19:
        tts.play('이것은 말입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 20:
        tts.play('이것은 양입니다.')
        playsound('tmp.mp3')
    elif classes[0][i] == 21:
        tts.play('이것은 소입니다.')
        playsound('tmp.mp3')
```

Reward

```
scores[0][i]
```

```
if scores[0][i] >= 0.7:
```

Accuracy

임계값 70%

Reward

이제 거리를 측정 합시다‼

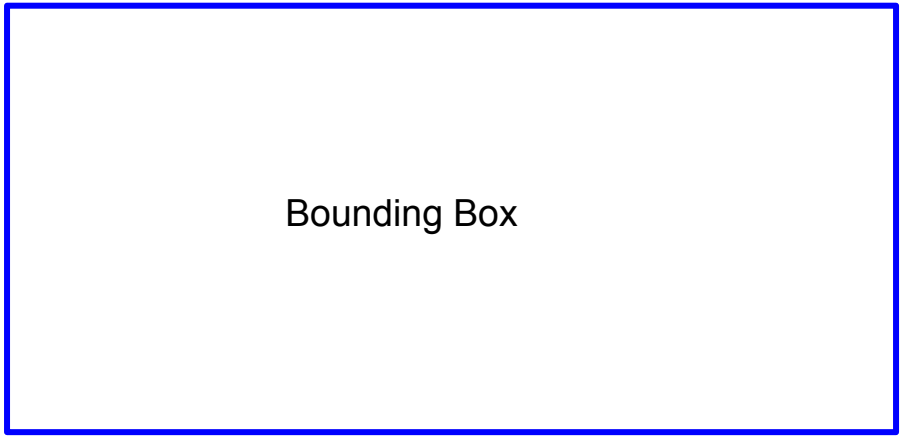
Reward

`boxes[0][i][3] - boxes[0][i][1]`

x좌표의 차이

`boxes[0][i][2] - boxes[0][i][0]`

y좌표의 차이



Reward

왼쪽 박스가 더 크다 = 가깝다

Reward

Done

Breakout은 공을 놓치면 게임 다시시작

CartPole은 쓰러지면 게임 다시시작

슈퍼마리오도 죽으면 게임 다시시작

GTA5는 죽으면….?

Done

병원에서 시작...

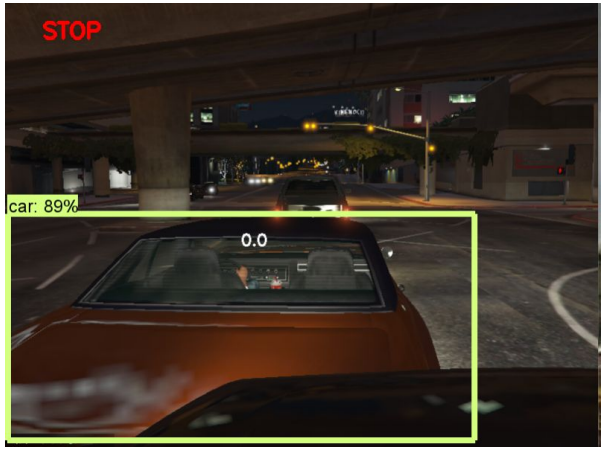
Done

운전에서의 에피소드가 끝난다는것은?

바로 **충돌**

충돌을 어떻게 인식하지?

Done



자동차와 충돌

건물과의 충돌은?

Done

영상간 얼마나 차이가 있는지를 검출

cv2.absdiff

http://www.noah.org/wiki/movement.py

Done

주행중

Done

충돌 -> Game Done

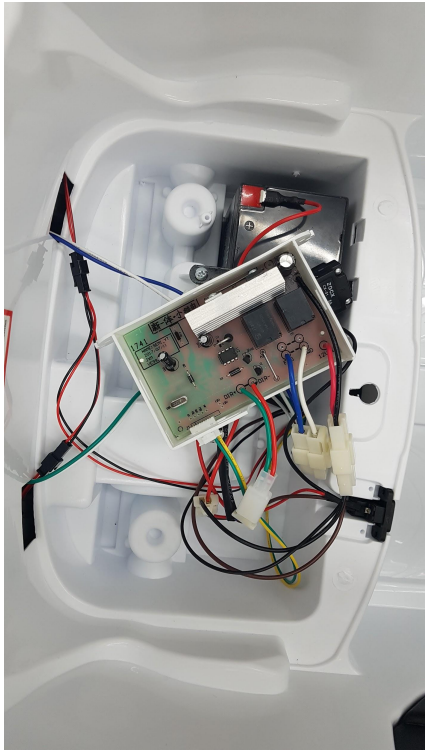
설정이 다 끝난 후 실행을 했지만...

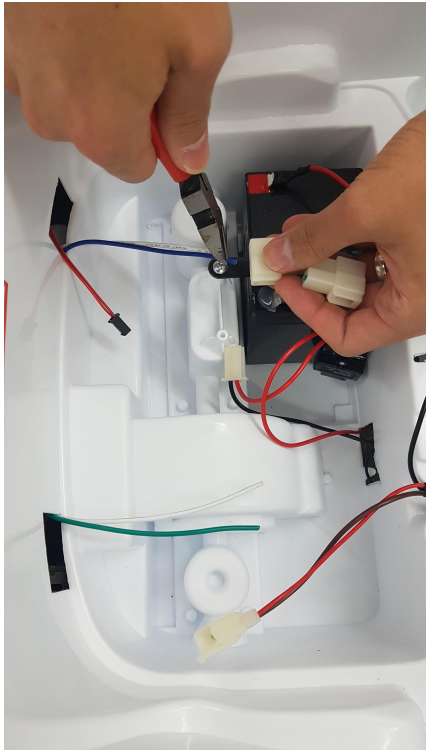
결국 구매

최종 목표

Real World에 적용

딥러닝

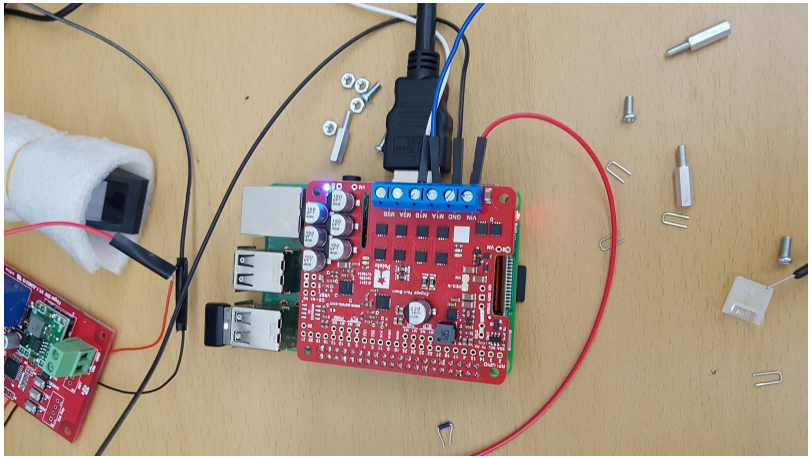




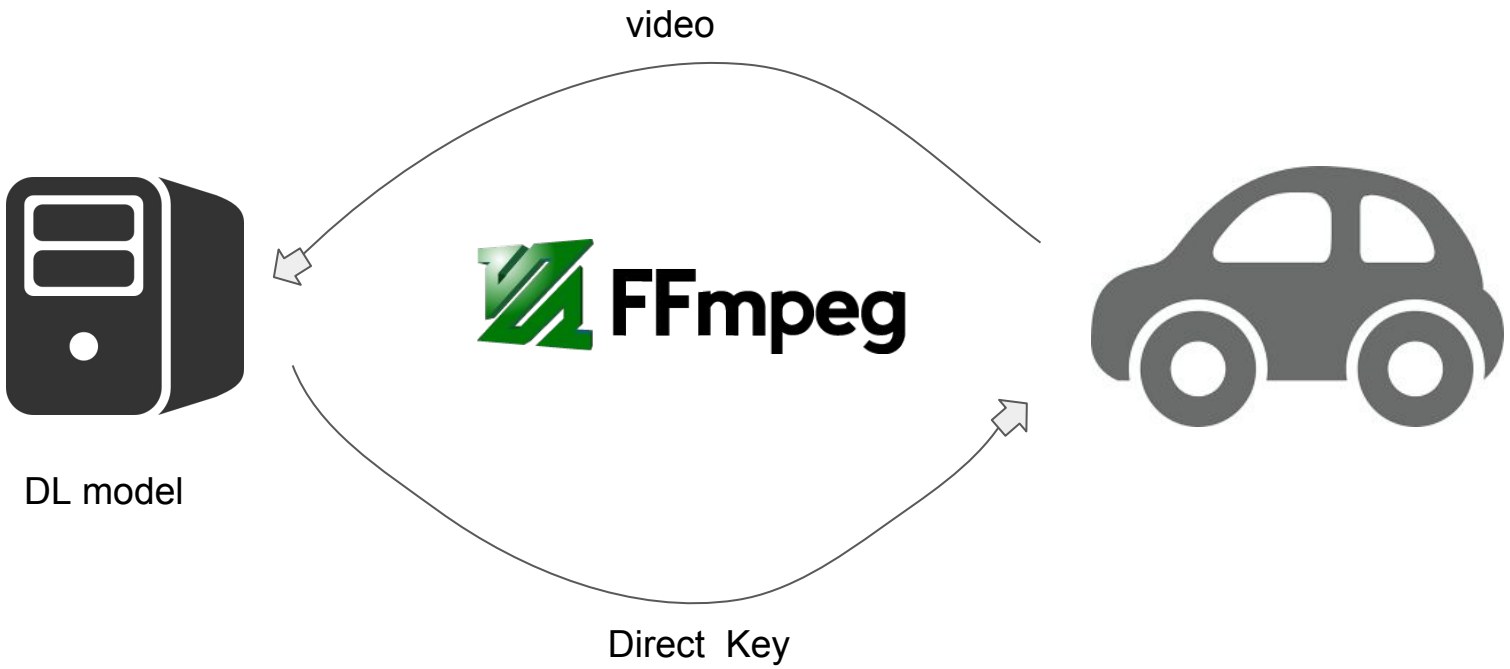
Control Box 연결선 제거

DC모터 분해

DC모터와 점퍼선 납땜



모터드라이버와 라즈베리파이에 연결


video
FFmpeg
DL model
Direct Key

FFmpeg ?

Michael Niedermayer의 주도하에 개발되고 있는 모든 동영상, 음악, 사진 포맷들의 디코딩과 인코딩을 목표로 만들어지고 있는 LGPL와 GPL 이중 라이센스[1]를 따르는 오픈소스 프로젝트.

관련업계에서는 마이크로소프트+애플급의 영향력을 가지고 있는 멀티미디어계의 최종보스. 가능성 없는 이야기긴 하지만 이 프로그램이 유료화를 시전한다면 우리는 내일부터 동영상을 시청 못할수도 있다.[2] 아래의 FFmpeg을 기반[3]으로 하는 프로그램 리스트를 보면 알겠지만 이건 농담이 아니다.

한 일화로 NVIDIA와 AMD는 자사의 그래픽카드의 하드웨어 가속을 위해 FFmpeg에 **'직접'** 소스를 건내주고 FFmpeg이 시키는대로 수정하였다. 대부분의 멀티미디어 재생기가 FFmpeg을 기반으로 하기 때문에, FFmpeg에서 지원만 한다면 그 효과를 당장 볼 수 있기 때문.

https://namu.wiki/w/FFmpeg

```
paused = False
def processor(server, stream_frame, recv_data):
    if not paused:

        print('shape'+str(stream_frame.shape))

        screen = cv2.cvtColor(stream_frame, cv2.COLOR_RGB2GRAY)
        screen = cv2.resize(screen, (160,120))

        prediction = model.predict([screen.reshape(160,120,1)])[0]
        print(prediction)

        turn_thresh = .75
        fwd_thresh = 0.70

        if prediction[1] > fwd_thresh:
            return bytearray([0, 1, 0])
        elif prediction[0] > turn_thresh:
            return bytearray([1, 0, 0])
        elif prediction[2] > turn_thresh:
            return bytearray([0, 0, 1])
        else:
            return bytearray([0, 1, 0])


s1 = remote_model(addr='0.0.0.0', stream_port = 8888, command_port=7777)
s1.set_loop_processor(processor)
s1.open()
```

server

```
if recv_data == bytearray([0,0,1]):
        print('right')
        right = 1
        duty2 = 10
        p2.ChangeDutyCycle(duty2)
        GPIO.output(MOTOR2A,1)
        GPIO.output(MOTOR2B,1)


s = streamer(remote_model_url = '192.168.0.2')
s.set_ffmpeg_flag(streamer.MACOS_FACETIME_PRESET)
s.set_model_respones_callback(response)
s.start()
```

client

같이 할 여러분들의 연락을 기다립니다!!

연락처

- 이메일 : kjt7889@naver.com
- 페이스북 : https://www.facebook.com/kjt7889

감사합니다