

CH. 12. 자율주행을 위한 데이터 수집

Section 01 모델 학습을 위한 이미지 보정

- 이미지를 보정하는 이유는 사람이 운전할 때도 햇빛이 강한 날에는 선글라스를 착용하여 잘보이도록 하는 것과 같은 이유로 사진을 보정하여 학습이 편하도록 하기 위함이다.
- 또 하나의 이유는 라즈베리파이에서 사진 사이즈를 줄여 학습 속도를 빠르게 하기 위해서이다.
- nvidia에서는 자율주행 자동차를 위한 알고리즘으로 end-to-end Learning이라는 딥러닝 학습모델을 배포하였다.
- 본 강의에서는 이 모델을 사용하여 자율주행 자동차를 학습시킬 것이다.
- 이 모델을 사용하기 위해서 이미지의 보정을 해보자.

Section 01 모델 학습을 위한 이미지 보정

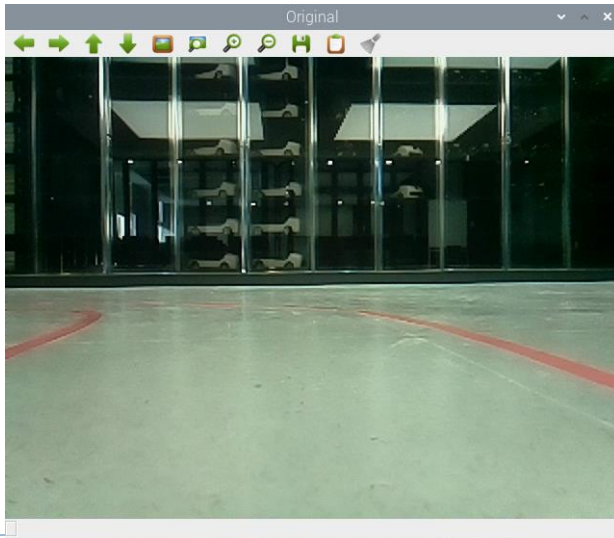
□ ex12_1.py

```
1  import cv2
2
3  cam = cv2.VideoCapture(0)
4  cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
5  cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
6
7  while( cam.isOpened() ):
8      keyVal = cv2.waitKey(0)
9      if keyVal == ord('q'):
10         break
11     elif keyVal == 82:
12         print("go")
13     elif keyVal == 81:
14         print("left")
15     elif keyVal == 83:
16         print("right")
17     elif keyVal == 32:
18         print("stop")
```

Section 01 모델 학습을 위한 이미지 보정

□ ex12_1.py

```
19
20     _, image = cam.read()
21     cv2.imshow('Original', image)
22
23     height, _, _ = image.shape
24     save_image = image[int(height/2):, :, :]
25     cv2.imshow('Save', save_image)
26     cv2.destroyAllWindows()
```



Section 01 모델 학습을 위한 이미지 보정

- 우리가 일반적으로 보는 RGB의 영상을 인공지능이 학습하기 좋은 YUV형식으로 변환해보자.
- YUV형식은 RGB표현을 사용할 때보다 더 효율적으로 특징을 찾아낼 수 있다는 장점이 있다.
- 다음과 같은 코드를 작성해보자.
- ex12_2.py

```
1  import cv2
2
3  cam = cv2.VideoCapture(0)
4  cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
5  cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
6
7  while( cam.isOpened() ):
8      keyValue = cv2.waitKey(0)
9      if keyValue == ord('q'):
10         break
11     elif keyValue == 82:
12         print("go")
```

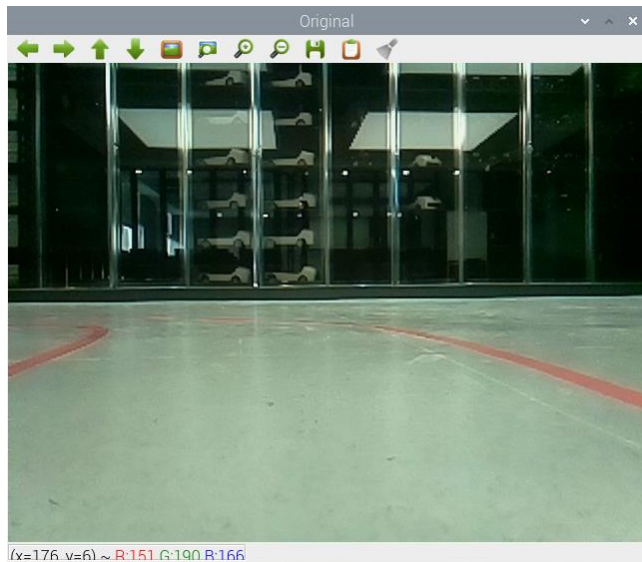
Section 01 모델 학습을 위한 이미지 보정

□ ex12_2.py

```
13     elif keyVal == 81:
14         print("left")
15     elif keyVal == 83:
16         print("right")
17     elif keyVal == 32:
18         print("stop")
19
20     _, image = cam.read()
21     cv2.imshow('Original', image)
22
23     height, _, _ = image.shape
24     save_image = image[int(height/2):, :, :]
25     save_image = cv2.cvtColor(save_image, cv2.COLOR_BGR2YUV)
26     cv2.imshow('Save', save_image)
27     cv2.destroyAllWindows()
```

Section 01 모델 학습을 위한 이미지 보정

- 왼쪽 그림은 원본 이미지이다.
- 오른쪽 그림을 보면 RGB타입의 영상이 YUV로 변환되었다.
- RGB방식보다 선이 또렷하게 표현되었다.



Section 01 모델 학습을 위한 이미지 보정

□ 이제 이미지 처리의 마지막 단계로 블러링 처리와 end-to-end learning 모델의 입력 영상의 크기인 200x66 픽셀로 변경하는 코드를 만들어보자.

□ ex12_3.py

```
1 import cv2
2
3 cam = cv2.VideoCapture(0)
4 cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
5 cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
6
7 while( cam.isOpened() ):
8     keyValue = cv2.waitKey(0)
9     if keyValue == ord('q'):
10         break
11     elif keyValue == 82:
12         print("up")
13     elif keyValue == 81:
14         print("left")
```


Section 01 모델 학습을 위한 이미지 보정

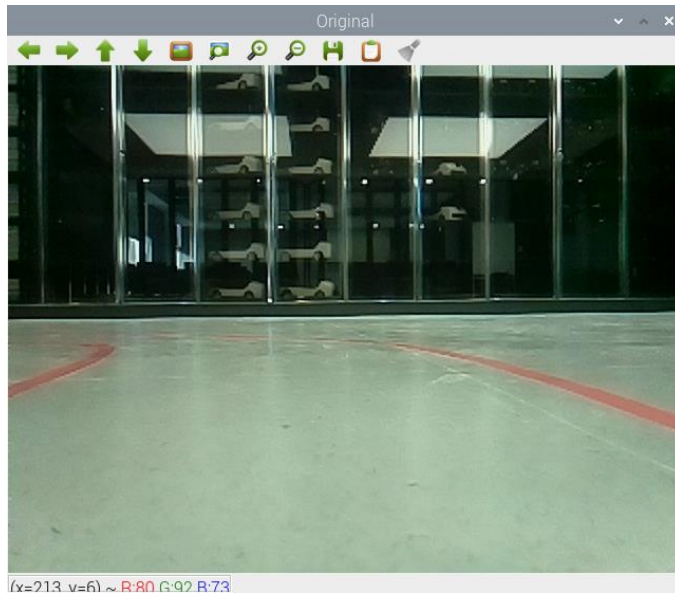
□ ex12_3.py

```
15     elif keyVal == 83:
16         print("right")
17     elif keyVal == 32:
18         print("stop")
19
20     _, image = camera.read()
21     cv2.imshow('Original', image)
22
23     height, _, _ = image.shape
24     save_image = image[int(height/2):, :, :]
25     save_image = cv2.cvtColor(save_image, cv2.COLOR_BGR2YUV)
26     save_image = cv2.GaussianBlur(save_image, (3, 3), 0)
27     save_image = cv2.resize(save_image, (200,66))
28     cv2.imshow('Save', save_image)
29     cv2.destroyAllWindows()
```

Section 01 모델 학습을 위한 이미지 보정

□ ex12_3.py

□ 예제 실행 결과에서 블러링처리와 크기가 200x66으로 조절된 영상을 확인할 수 있다.



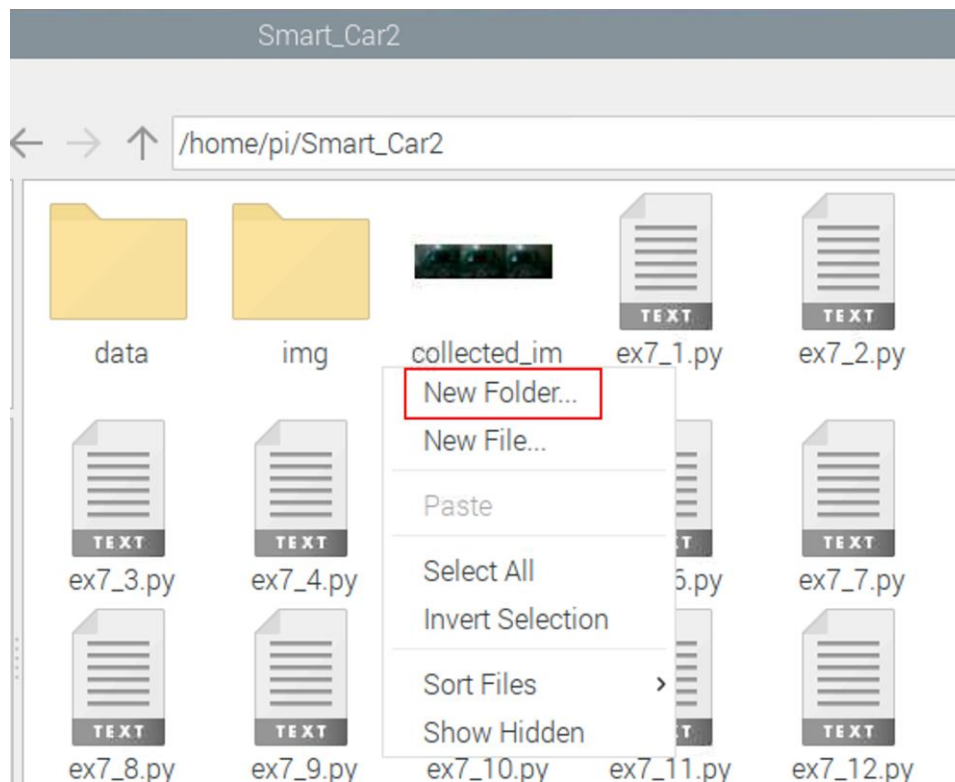
Section 02 이미지 저장

- 이미지를 보정하는 이유는 사람이 운전할 때도 햇빛이 강한 날에는 선글라스를 착용하여 잘보이도록 하는 것과 같은 이유로 사진을 보정하여 학습이 편하도록 하기 위함이다.
- 또 하나의 이유는 라즈베리파이에서 사진 사이즈를 줄여 학습 속도를 빠르게 하기 위해서이다.
- 학습된 데이터가 저장될 폴더를 생성하기 위해 [파일 매니저]를 연다.



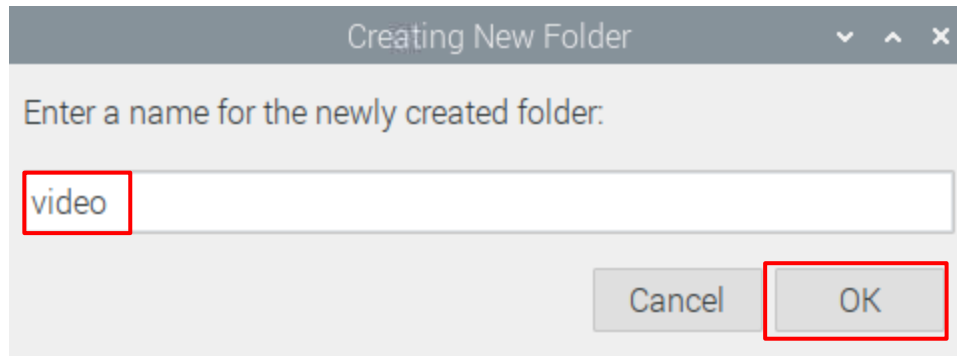
Section 02 이미지 저장

- /home/pi/Smart_Car2 폴더의 위치로 이동한 후, 빈곳에 마우스 오른쪽쪽을 클릭한 다음 [Create New] → [Folder] 버튼을 클릭하여 새로운 폴더를 생성한다.

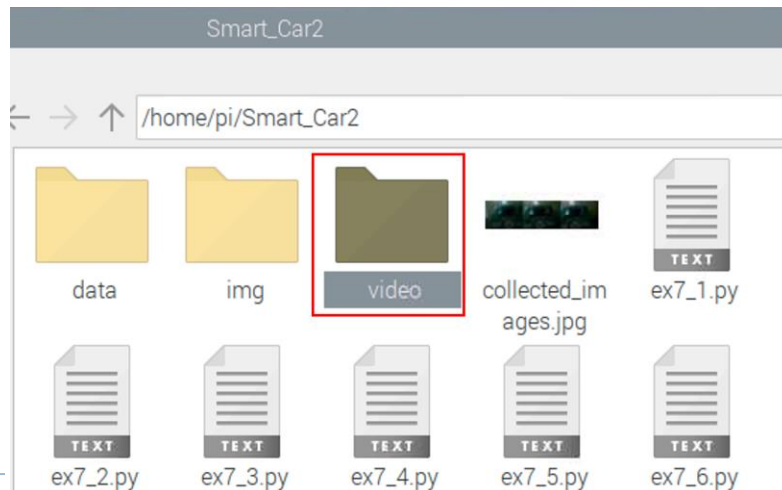


Section 02 이미지 저장

- ❑ “Video” 이름으로 폴더를 생성한다.



- ❑ 데이터가 저장될 video 폴더가 생성되었다.



Section 02 이미지 저장

□ 이미지를 저장하는 코드를 작성해보자.

□ ex12_4.py

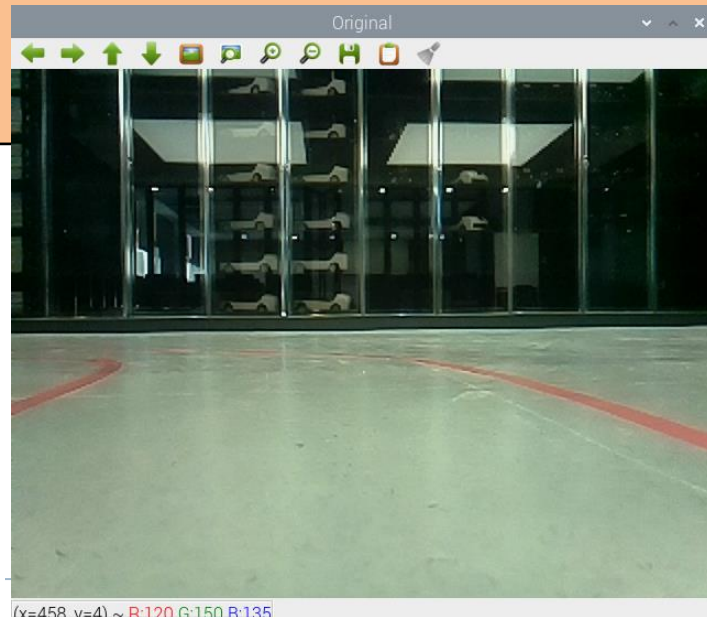
```
1  import cv2
2  import time
3
4  cam = cv2.VideoCapture(0)
5  cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
6  cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
7
8  while( cam.isOpened() ):
9      keyValue = cv2.waitKey(0)
10     if keyValue == ord('q'):
11         break
12
```

Section 02 이미지 저장

□ 이미지를 저장하는 코드를 작성해보자.

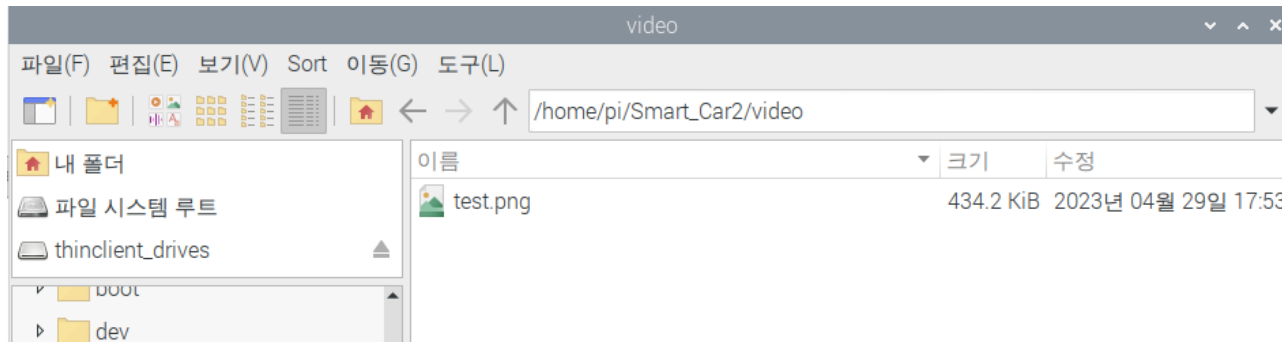
□ ex12_4.py

```
13  _, image = cam.read()
14  cv2.imshow('Original', image)
15
16  cv2.imwrite("/home/pi/Smart_Car2/video/test.png" , image)
17
18  time.sleep(1.0)
19
20  cv2.destroyAllWindows()
```



Section 02 이미지 저장

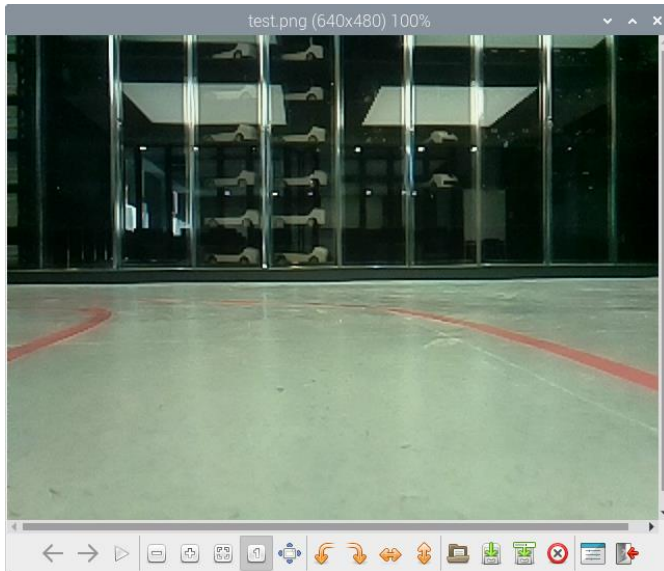
- ex12_4.py
- 예제가 실행되면 OpenCV의 이미지가 출력된다.
- `time.sleep(1.0)`으로 되어 있기 때문에 1초마다 사진이 갱신된다.
- [파일 매니저]를 열어 `/home/pi/Smart_Car2/video` 폴더에 접속하면 `test.png` 파일이 생성되었음을 확인할 수 있다.



Section 02 이미지 저장

□ ex12_4.py

□ test.png 파일을 더블클릭하여 알아보면 잘 저장되었음을 확인할 수 있다.



Section 02 이미지 저장

- 사진을 저장하는 이름이 같아 항상 같은 이름으로 덮어쓰기 되어 저장된다.
- 새로운 사진은 새로운 이름을 설정하여 저장해보자.
- ex12_5.py

```
1  import cv2
2  import time
3
4  cam = cv2.VideoCapture(0)
5  cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
6  cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
7
8  filepath = "/home/pi/Smart_Car2/video/test"
9  i = 0
10
11  while( cam.isOpened() ):
12      keyValue = cv2.waitKey(0)
13      if keyValue == ord('q'):
14          break
```

Section 02 이미지 저장

- 사진을 저장하는 이름이 같아 항상 같은 이름으로 덮어쓰기 되어 저장된다.
- 새로운 사진은 새로운 이름을 설정하여 저장해보자.
- ex12_5.py

```
15
16     _, image = cam.read()
17     cv2.imshow('Original', image)
18
19     cv2.imwrite(f'{filepath}_{i:05d}.png', image)
20     i += 1
21
22     time.sleep(1.0)
23
24     cv2.destroyAllWindows()
```

Section 02 이미지 저장

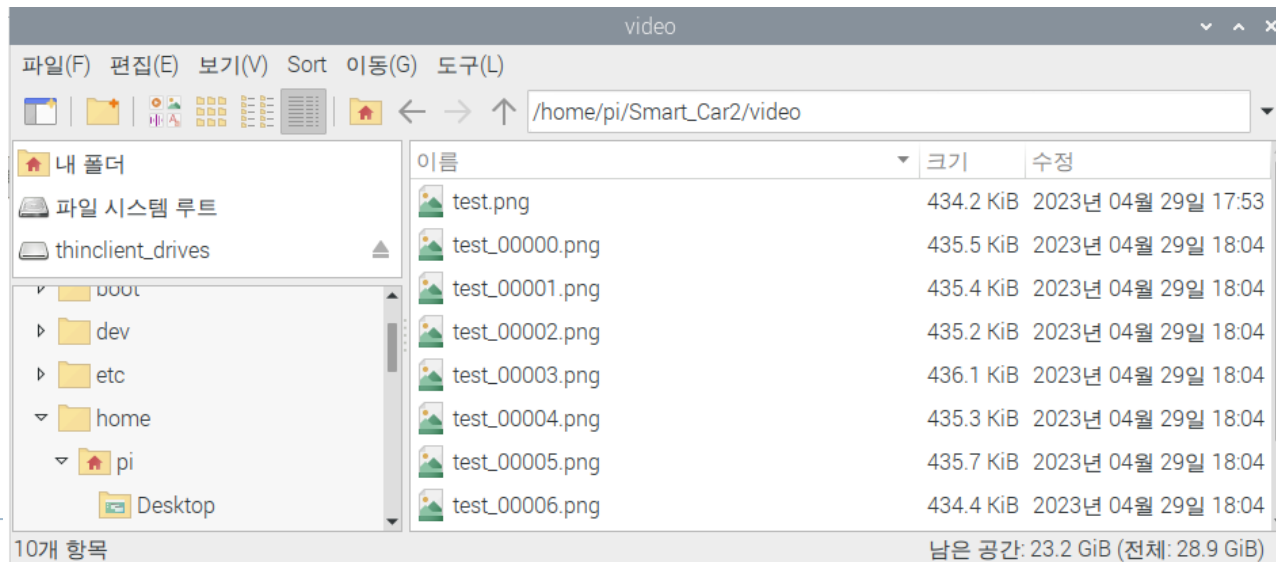
□ ex12_5.py

□ 코드가 실행되면 test 00000부터 1씩 증가하여 사진이 저장되어
진다.

□ 실제 자동차 주행 데이터를 얻기 위해 test 데이터는 삭제한다.

□ video 폴더에서 [Ctrl + A] 를 눌러 전체 선택 후 [delete] 를 눌러
휴지통으로 이동한다.

□ 또는 [Shift + delete] 를 눌러 완전 삭제 할 수 있다.



Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ 키보드의 입력에 따라 조건문을 실행하는 코드를 만들어보자.

□ ex12_6.py

```
1 import cv2
2 import time
3
4 cam = cv2.VideoCapture(0)
5 cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
6 cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
7
8 filepath = "/home/pi/Smart_Car2/video/train"
9 i = 0
10 carState = "stop"
11 while( cam.isOpened() ):
12     _, image = cam.read()
13     cv2.imshow('Original', image)
14
15     keyValue = cv2.waitKey(10)
```

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

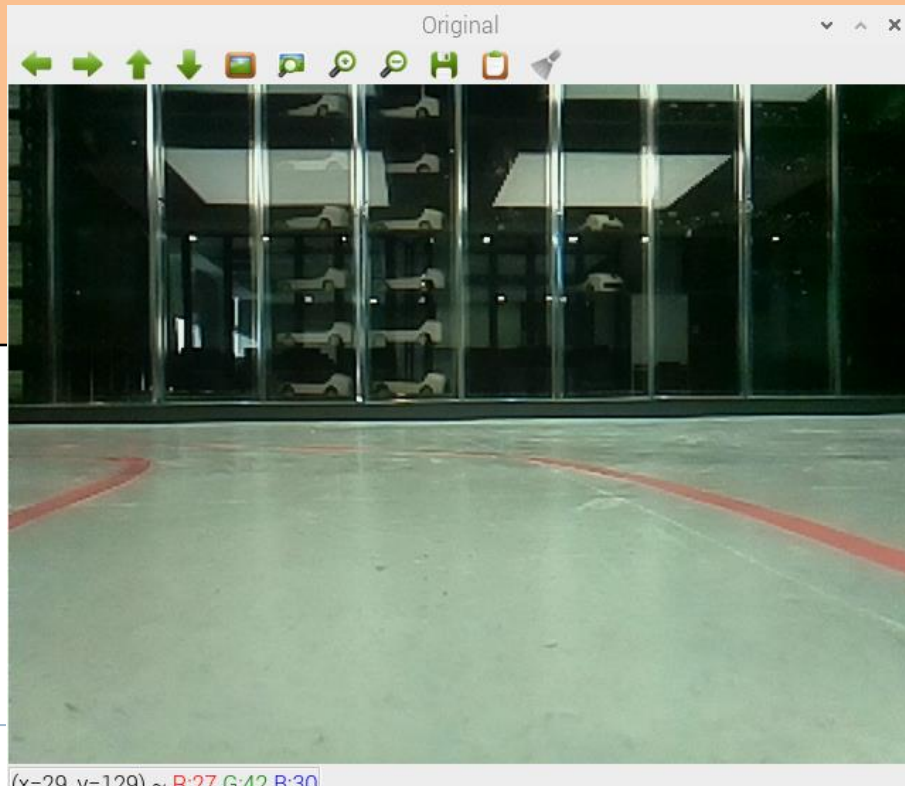
□ ex12_6.py

```
16     if keyValue == ord('q'):
17         break
18     elif keyValue == 82:
19         print("go")
20         carState = "go"
21     elif keyValue == 81:
22         print("left")
23         carState = "left"
24     elif keyValue == 83:
25         print("right")
26         carState = "right"
27     elif keyValue == 32:
28         print("stop")
29         carState = "stop"
30
31     height, _, _ = image.shape
32     save_image = image[int(height/2) : , :, :]
33     save_image = cv2.cvtColor(save_image, cv2.COLOR_BGR2YUV)
34     save_image = cv2.GaussianBlur(save_image, (3, 3), 0)
35     save_image = cv2.resize(save_image, (200, 66))
```

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ ex12_6.py

```
36 cv2.imshow('Save', save_image)
37
38 if carState == "left":
39     print("L")
40 elif carState == "right":
41     print("R")
42 elif carState == "go":
43     print("G")
44 elif carState == "stop":
45     print("S")
46
47 cv2.destroyAllWindows()
```



Shell

L
left
L
left
L
left
L
stop
S
right
R

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ ex12_7.py

```
1 import cv2
2 import time
3
4 cam = cv2.VideoCapture(0)
5 cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
6 cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
7
8 filepath = "/home/pi/Smart_Car2/video/train"
9 i = 0
10 carState = "stop"
11 while( cam.isOpened() ):
12     _, image = cam.read()
13     cv2.imshow('Original', image)
14
15     keyValue = cv2.waitKey(0)
16     if keyValue == ord('q'):
17         break
18     elif keyValue == 82:
19         print("go")
20         carState = "go"
```


Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ ex12_7.py

```
21     elif keyValue == 84:
22         print("back")
23         carState = "back"
24     elif keyValue == 81:
25         print("left")
26         carState = "left"
27     elif keyValue == 83:
28         print("right")
29         carState = "right"
30     elif keyValue == 32:
31         print("stop")
32         carState = "stop"
33
34     height, _, _ = image.shape
35     save_image = image[int(height/2) : , :, :]
36     save_image = cv2.cvtColor(save_image, cv2.COLOR_BGR2YUV)
37     save_image = cv2.GaussianBlur(save_image, (3, 3), 0)
38     save_image = cv2.resize(save_image, (200, 66))
39     cv2.imshow('Save', save_image)
40
```

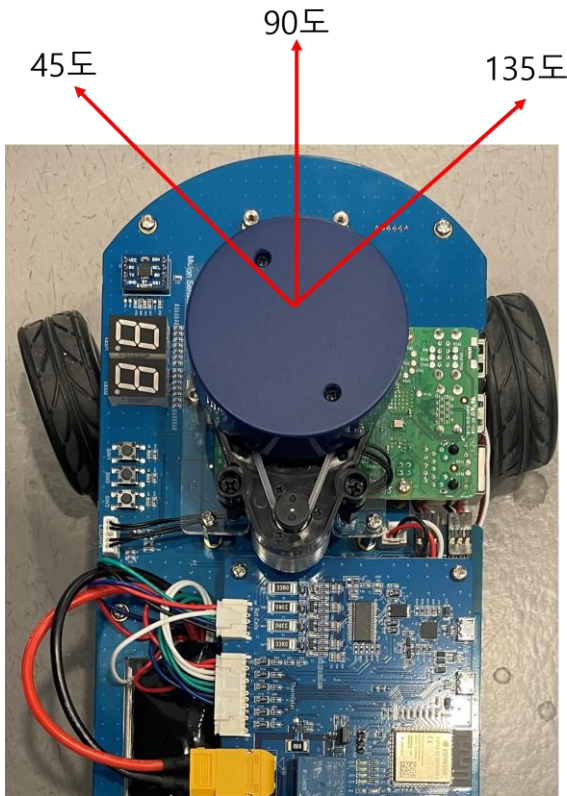
Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ ex12_7.py

```
41     if carState == "left":
42         cv2.imwrite(f"{filepath}_{i:05d}_{carState}.png", save_image)
43         i += 1
44     elif carState == "right":
45         cv2.imwrite(f"{filepath}_{i:05d}_{carState}.png", save_image)
46         i += 1
47     elif carState == "go":
48         cv2.imwrite(f"{filepath}_{i:05d}_{carState}.png", save_image)
49         i += 1
50     elif carState == "stop":
51         print("S")
52
53 cv2.destroyAllWindows()
```

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

- 학습모델을 만들기 위해서 자동차가 직진할 때는 0도, 왼쪽으로 이동할 때는 -45도, 오른쪽으로 이동할 때는 45도로 정하였다.

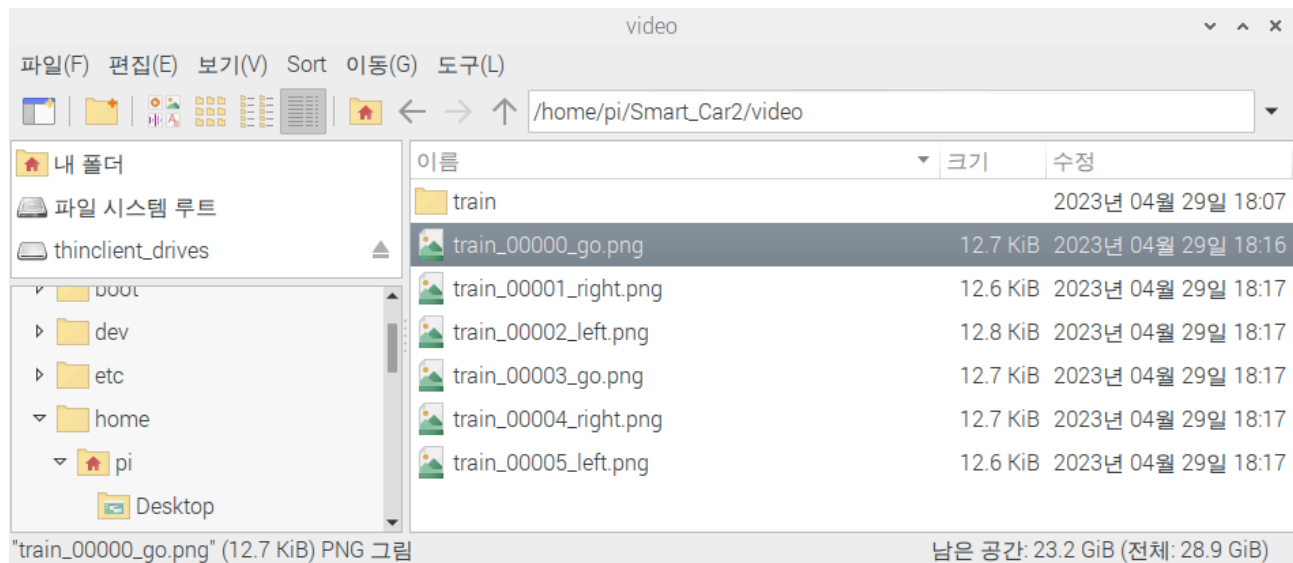


Shell

```
qt5ct: u  
go  
stop  
S  
right  
left  
go  
right  
left  
stop  
S
```

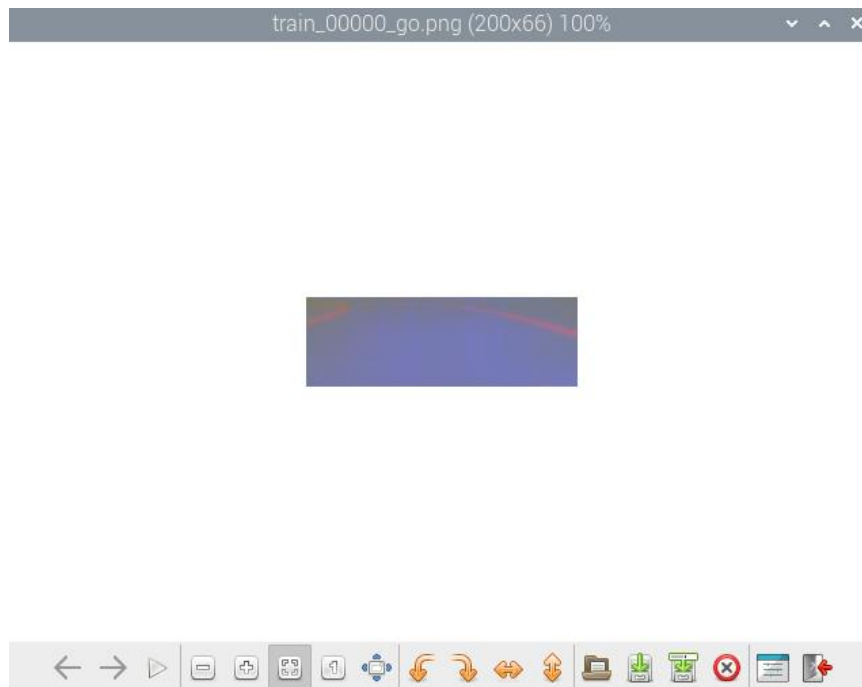
Section 03 조종기능을 추가하여 실제 주행 데이터 수집

- 직진할 때는 go, 왼쪽으로 이동할 때는 left, 오른쪽으로 이동할때는 right가 파일 이름에 함께 저장되었다.
- train_번호_방향의 형태로 저장되었다. 사진은 매우 빠르게 많이 저장된다.
- 스페이스 바를 눌러 자동차가 멈춰있을 때는 사진이 지정되지 않는다.
- 사진을 저장할 때 동일한 이름이 있다면 덮어쓰기가 된다.



Section 03 조종기능을 추가하여 실제 주행 데이터 수집

- 저장된 파일 중 하나를 열어 확인해보자.
- 사진 데이터가 잘 저장되었음을 확인할 수 있다.



Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ 이제 자동차를 움직이는 기능을 추가하여 완성하자.

□ ex12_8.py

```
1 import cv2
2 import time
3 from ctypes import *
4 import os
5
6 WiringPi = CDLL("/home/pi/WiringPi/wiringPi/libwiringPi.so.2.70", mode=RTLD_GLOBAL)
7 swcar = cdll.LoadLibrary('/home/pi/swcar/libswcar.so')
8
9 motor_status = "STOP"
10 servo_status = "CENTER"
11
12 def motor_forward():
13     global motor_status
14     if (motor_status == "REVERSE") :
15         swcar.SIO_ForwardMotor(0)
16         time.sleep(0.1)
17         swcar.SIO_ForwardMotor(20)
18         motor_status = "FORWARD"
19
```

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ 이제 자동차를 움직이는 기능을 추가하여 완성하자.

□ ex12_8.py

```
20 def motor_reverse():
21     global motor_status
22     if (motor_status == "FORWARD") :
23         swcar.SIO_ReverseMotor(0)
24         time.sleep(0.1)
25         swcar.SIO_ReverseMotor(20)
26         motor_status = "REVERSE"
27
28 def motor_stop():
29     global motor_status
30     swcar.SIO_ForwardMotor(0)
31     motor_status = "STOP"
32
33 def servo_left():
34     global servo_status
35     if (motor_status == "RIGHT") :
36         swcar.SIO_WriteServo(100, 50)
37         time.sleep(0.1)
38         swcar.SIO_WriteServo(100, 90)
39         servo_status = "LEFT"
40
```

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ 이제 자동차를 움직이는 기능을 추가하여 완성하자.

□ ex12_8.py

```
41 def servo_right():
42     global servo_status
43     if (motor_status == "LEFT") :
44         swcar.SIO_WriteServo(100, 50)
45         time.sleep(0.1)
46         swcar.SIO_WriteServo(100, 10)
47         servo_status = "RIGHT"
48
49 def servo_center():
50     global servo_status
51     swcar.SIO_WriteServo(100, 50)
52     servo_status = "CENTER"
53
54 swcar.SIO_Init(0)
55
56 motor_stop()
57 servo_center()
58
59 cam = cv2.VideoCapture(0)
60 cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
32 61 cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
62
```


Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ 이제 자동차를 움직이는 기능을 추가하여 완성하자.

□ ex12_8.py

```
63 filepath = "/home/pi/Smart_Car2/video/train"
64 i = 0
65
66 while( cam.isOpened() ):
67     _, image = cam.read()
68     cv2.imshow('Original', img)
69
70     keyValue = cv2.waitKey(0)
71     if key == ord('q'):
72         motor_stop()
73         time.sleep(0.02)
74         servo_center()
75         time.sleep(0.02)
76         break
77     elif key == 82:
78         carState = "go"
79         motor_forward()
80         time.sleep(0.02)
81         motor_stop()
```

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

□ 이제 자동차를 움직이는 기능을 추가하여 완성하자.

□ ex12_8.py

```
82     elif key == 81:
83         carState = "left"
84         servo_left()
85         motor_forward()
86         time.sleep(0.02)
87         motor_stop()
88     elif key == 83:
89         carState = "right"
90         servo_right()
91         motor_forward()
92         time.sleep(0.02)
93         motor_stop()
94     elif key == 32:
95         #carState = "stop"
96         motor_stop()
97         time.sleep(0.02)
98
99     height, _, _ = image.shape
100    save_image = image[int(height/2) : , :, :]
```

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

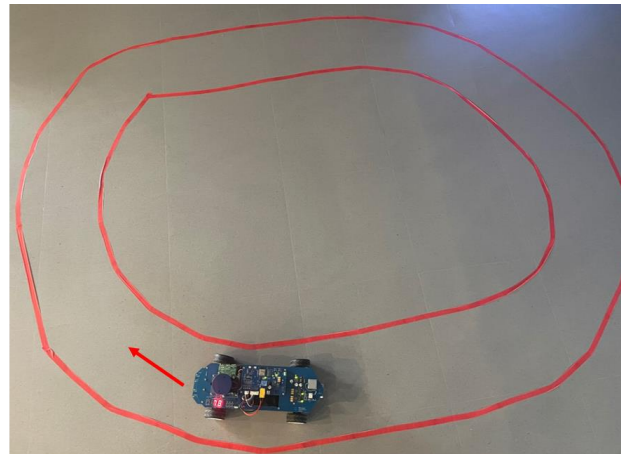
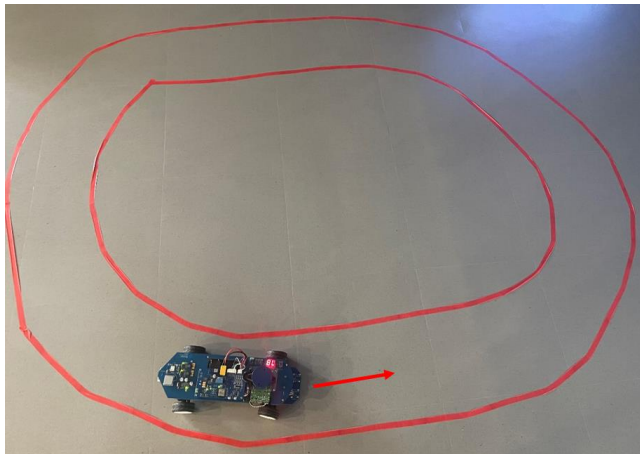
□ 이제 자동차를 움직이는 기능을 추가하여 완성하자.

□ ex12_8.py

```
101 save_image = cv2.cvtColor(save_image, cv2.COLOR_BGR2YUV)
102 save_image = cv2.GaussianBlur(save_image, (3, 3), 0)
103 save_image = cv2.resize(save_image, (200, 66))
104 cv2.imshow('Save', save_image)
105
106 if carState == "left":
107     cv2.imwrite(f"{filepath}_{i:05d}_{carState}.png", save_image)
108     i += 1
109 elif carState == "right":
110     cv2.imwrite(f"{filepath}_{i:05d}_{carState}.png", save_image)
111     i += 1
112 elif carState == "go":
113     cv2.imwrite(f"{filepath}_{i:05d}_{carState}.png", save_image)
114     i += 1
115
116 cam.release()
117 cv2.destroyAllWindows()
```

Section 03 조종기능을 추가하여 실제 주행 데이터 수집

- 실제 트랙을 그려 데이터를 모은다.
- 왼쪽 그림처럼 자동차를 시계 반대 방향으로 1~2 바퀴를 이동하면서 데이터를 학습한다.
- 이 후 오른쪽 그림처럼 자동차를 시계방향으로 1~2바퀴를 이동하면서 데이터를 학습한다.
- 데이터를 학습시킬 때는 PC의 화면을 보고 조종하여 학습한다.
- 자동차를 직접 보면서 조종을 하면 3인칭으로 보여지기 때문에 실제 카메라 영상을 보면서 학습을 시킨다.

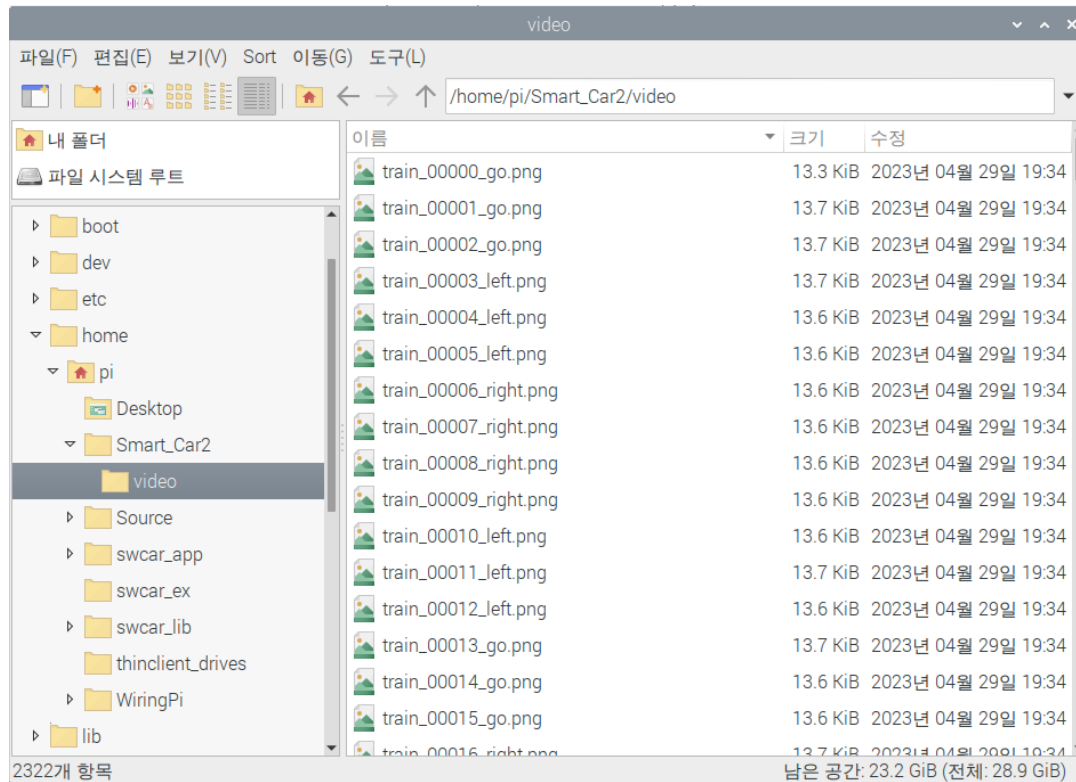


Section 03 조종기능을 추가하여 실제 주행 데이터 수집

- 데이터를 저장하는 시점은 전진, 왼쪽, 오른쪽이다.
- 멈춰있을 때는 저장하지 않으므로 반대방향의 데이터를 모을 때는 자동차를 멈춘 후 손으로 자동차를 들어 반대방향으로 이동시킨 후 주행하여 저장한다.
- 조종할 때는 키보드의 버튼을 계속 누르지 않고 방향을 바꿀 때만 한번씩 눌러준다.
- 좋은 데이터의 학습이 중요하다.
- 운전연습을 통해 가운데로 주행하는 데이터를 모으도록 한다.

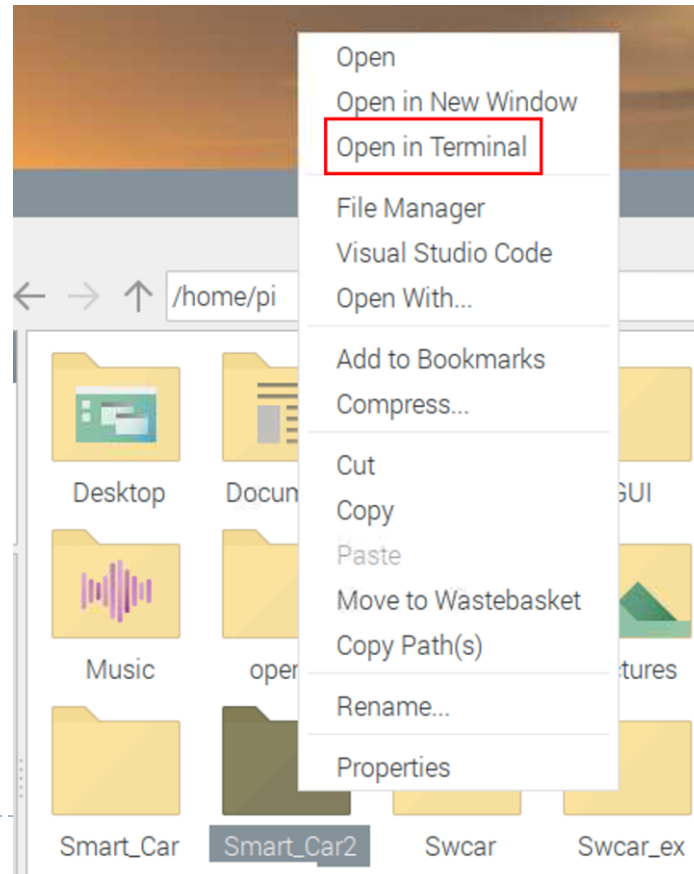
Section 03 조종기능을 추가하여 실제 주행 데이터 수집

- 2~3바퀴의 데이터를 모으면 대략 2천장 이상의 데이터가 모인다.
- 2바퀴 주행하였을 때 약 2천장의 데이터가 모였다.



Section 04 학습된 데이터를 PC로 전달

- 키보드의 입력에 따라 조건문을 실행하는 코드를 만들어보자.
- [파일 매니저]를 열어 /home/pi의 위치로 이동한다.
- [Smart_Car2] 폴더에 마우스 오른쪽 버튼을 클릭 후 [Open in Terminal]을 클릭한다.



Section 04 학습된 데이터를 PC로 전달

- /home/pi/Smart_Car2 경로로 터미널 창이 열렸다.

```
pi@raspberrypi: ~/Smart_Car2
File Edit Tabs Help
pi@raspberrypi:~/Smart_Car2 $
```

- 다음의 명령어를 입력하여 video 폴더를 압축한다.
- zip 명령어는 zip으로 압축하는 명령어이다.
- -r은 폴더내의 모든 파일을 압축하라는 옵션이다.
- video.zip은 압축될 파일명이다.
- video는 압축할 폴더이다.
- video 폴더안의 모든 파일을 video.zip 의 이름으로 압축한다.

```
zip -r video.zip video
```

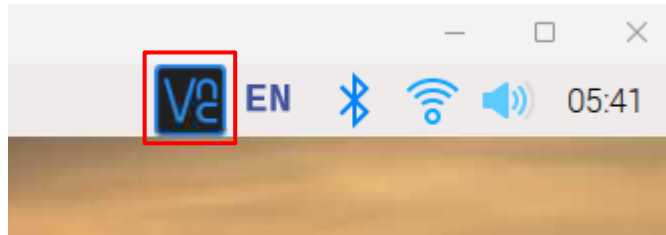
```
pi@raspberrypi: ~/Smart_Car
File Edit Tabs Help
pi@raspberrypi:~/Smart_Car $ zip -r video.zip video
```


Section 04 학습된 데이터를 PC로 전달

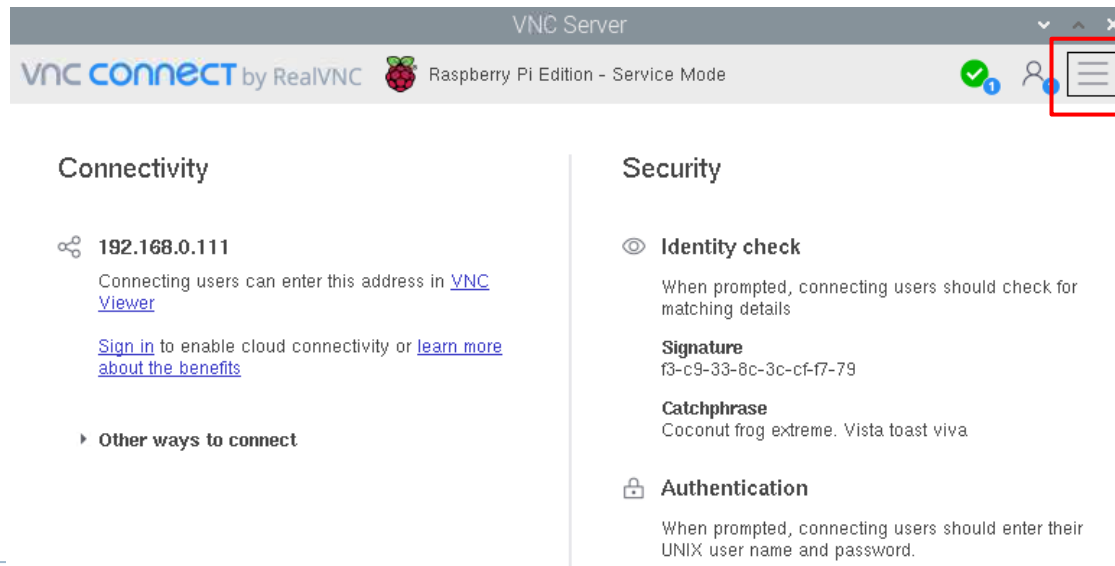
- /home/pi/Smart_Car2 폴더에 접속하면 방금 압축한 video.zip 파일을 확인할 수 있다.
- 데이터를 압축하는 이유는 데이터의 용량은 크지 않으나 데이터의 개수가 많아 데이터를 옮기는데 시간이 오래 걸리기 때문이다.
- 압축을 하여 하나의 단일 파일로 데이터를 옮기면 빠르게 옮길 수 있다. 이제 PC로 데이터를 옮겨보자.
- 획득한 데이터는 라즈베리파이에서 학습하지 않는다.
- 라즈베리파이에서 학습 시 GPU가 없기 때문에 너무 오래 걸린다.
- 데이터를 PC로 가져와서 학습한다.

Section 04 학습된 데이터를 PC로 전달

- 아래 VNC 아이콘을 클릭한다.
- 라즈베리파이 오른쪽 위에 위치한다.

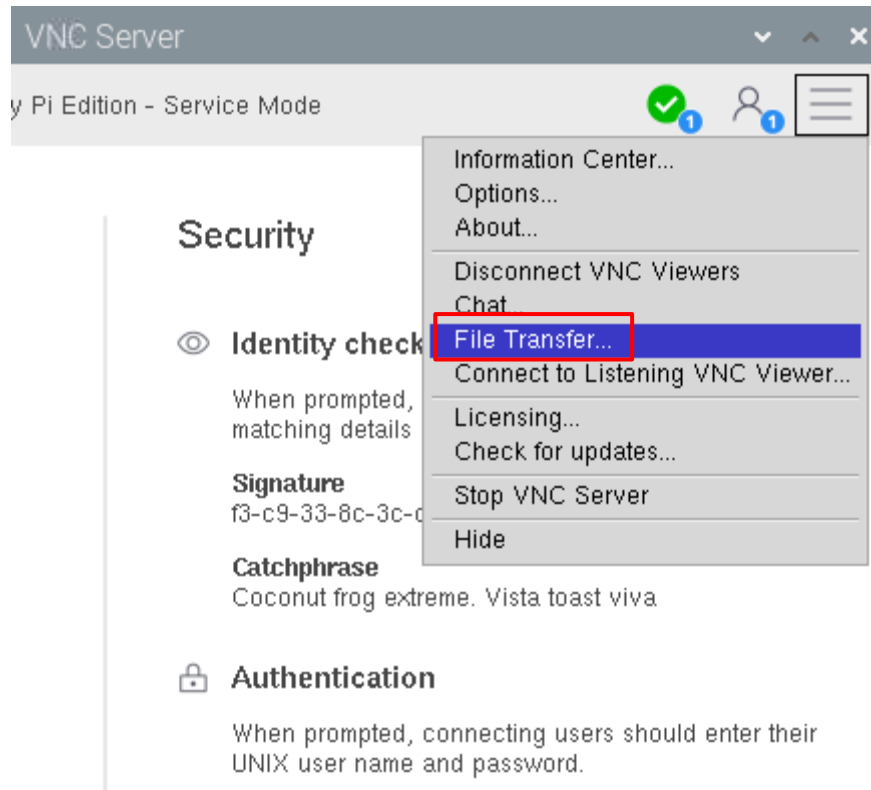


- [더보기] 버튼을 클릭한다.



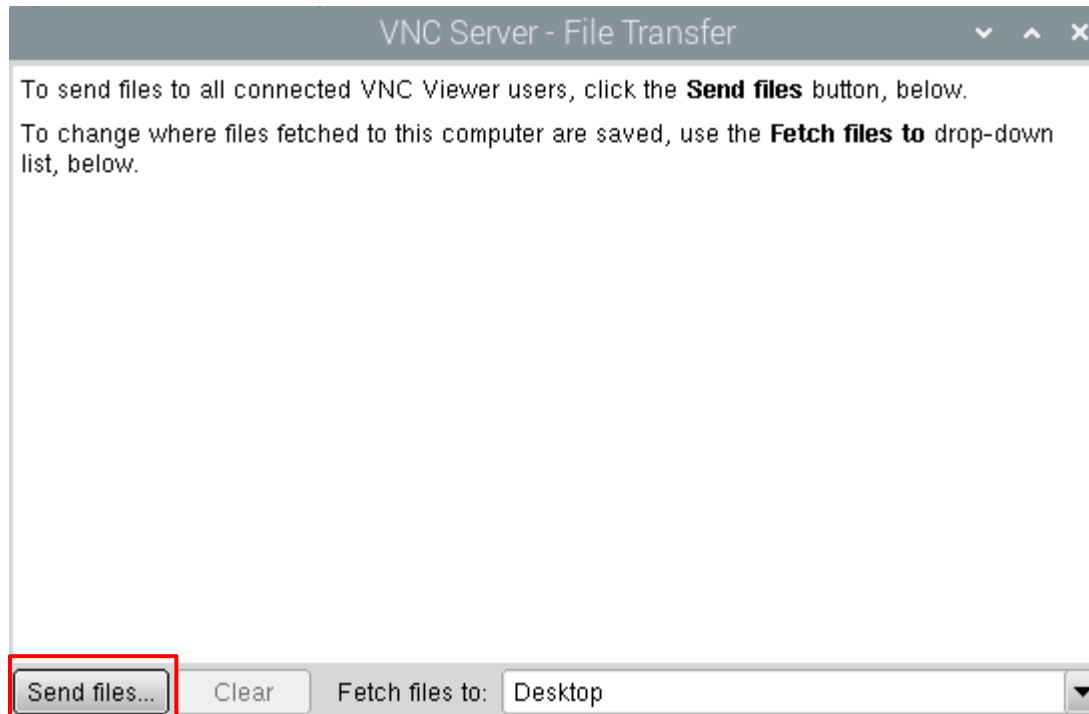
Section 04 학습된 데이터를 PC로 전달

- [File Transfer ...] 버튼을 클릭한다.



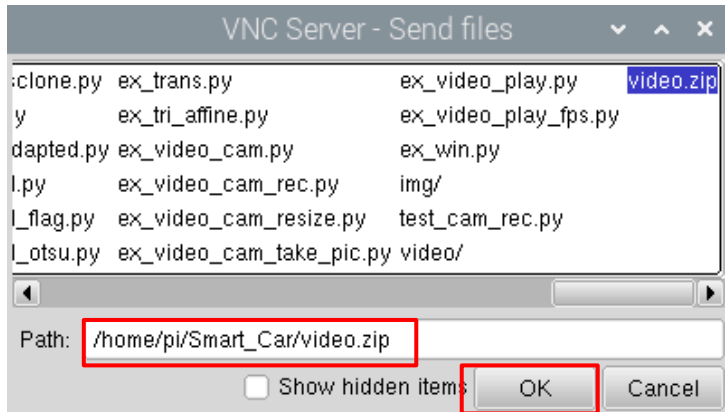
Section 04 학습된 데이터를 PC로 전달

- [Send files ...] 버튼을 클릭한다.

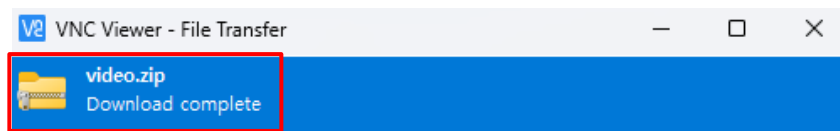


Section 04 학습된 데이터를 PC로 전달

- /home/pi/Smart_Car2 폴더에서 video.zip 파일을 선택한 후 [OK] 버튼을 눌러 PC로 전송한다.

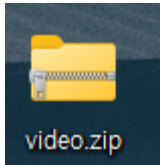


- video.zip 파일이 윈도우의 바탕화면으로 전송되었다.

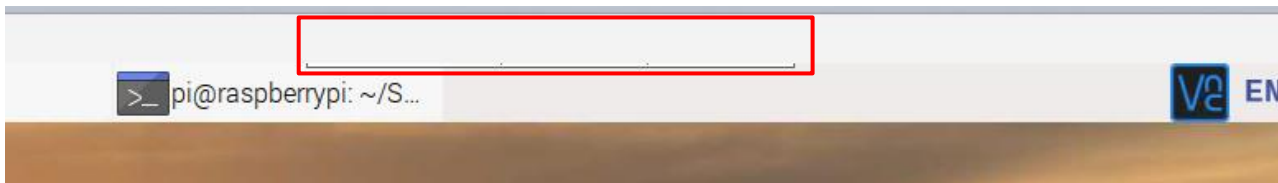


Section 04 학습된 데이터를 PC로 전달

- 윈도우의 바탕화면에서 video.zip 파일을 찾아보자.

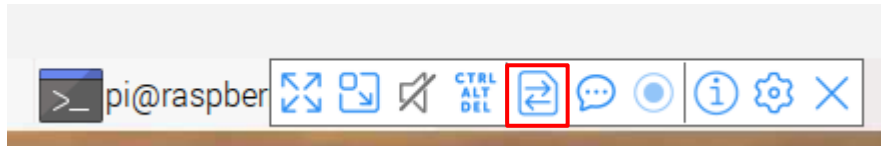


- 이처럼 VNC의 파일 전송 기능을 이용하면 라즈베리파이와 윈도우 PC간의 데이터 이동이 자유롭다.
- PC에서 라즈베리파이로의 데이터 전송 방법은 VNC뷰어에서 가운데 위쪽으로 마우스를 이동한다.

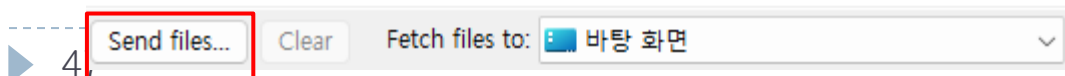
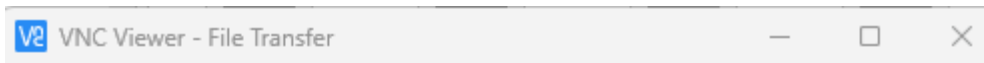


Section 04 학습된 데이터를 PC로 전달

- 다음과 같은 아이콘이 나온다.
- 파일 이동 아이콘을 클릭한다.

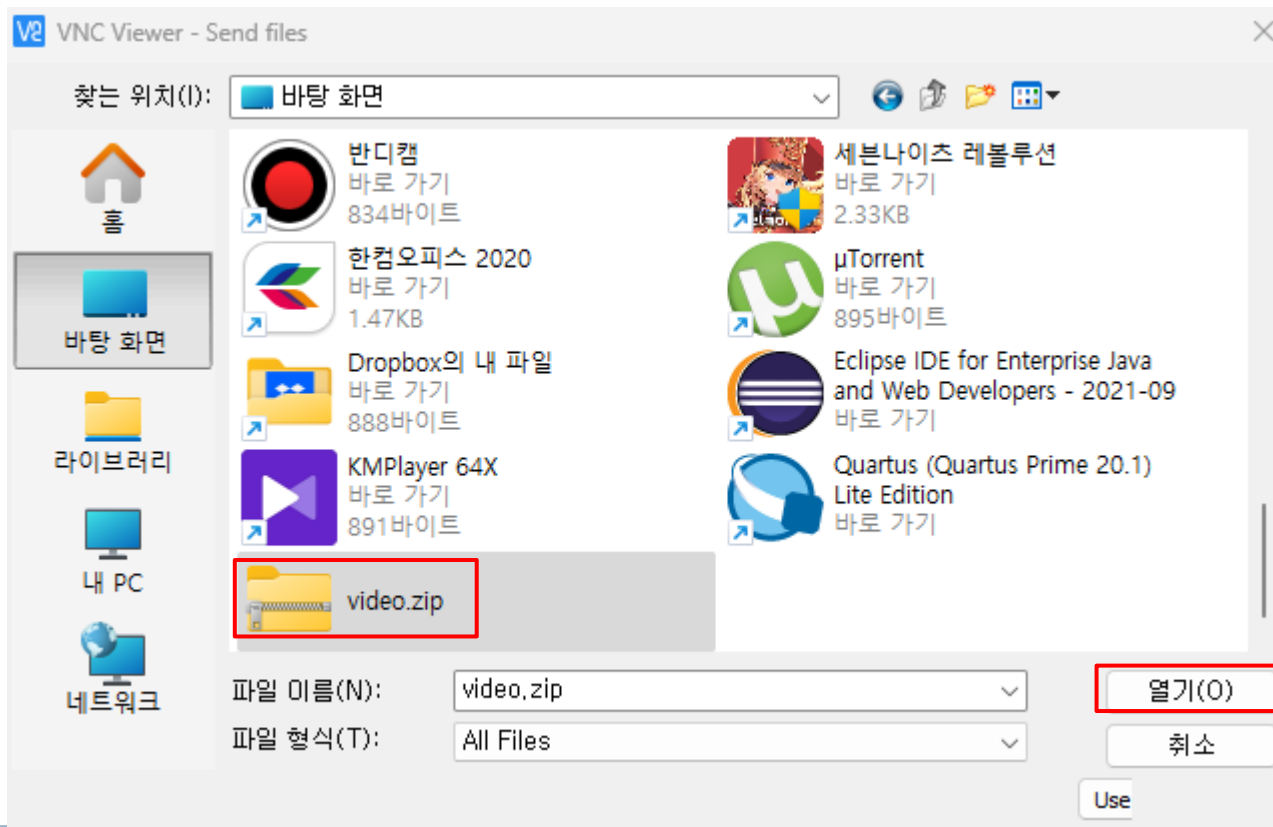


- [Send files ...] 버튼을 클릭한다.



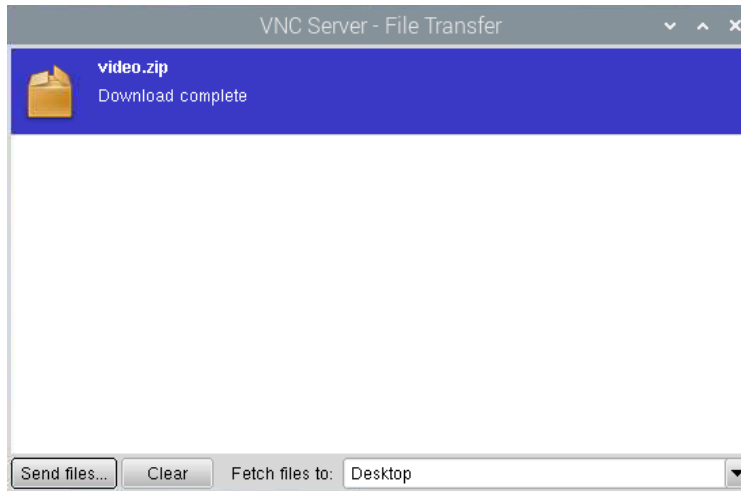
Section 04 학습된 데이터를 PC로 전달

- 바탕화면으로 이동하여 방금전에 받았던 video.zip 파일을 선택한다.
- 보내고 싶은 파일을 선택하면 된다.
- [열기]를 눌러 보낸다.

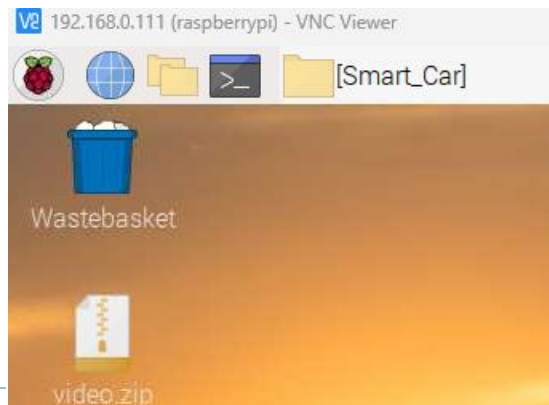


Section 04 학습된 데이터를 PC로 전달

- 이제 라즈베리파이에서 File Trasfer가 열린다.



- video.zip 파일이 PC에서 라즈베리파이로 이동하였다.



Q&A

