

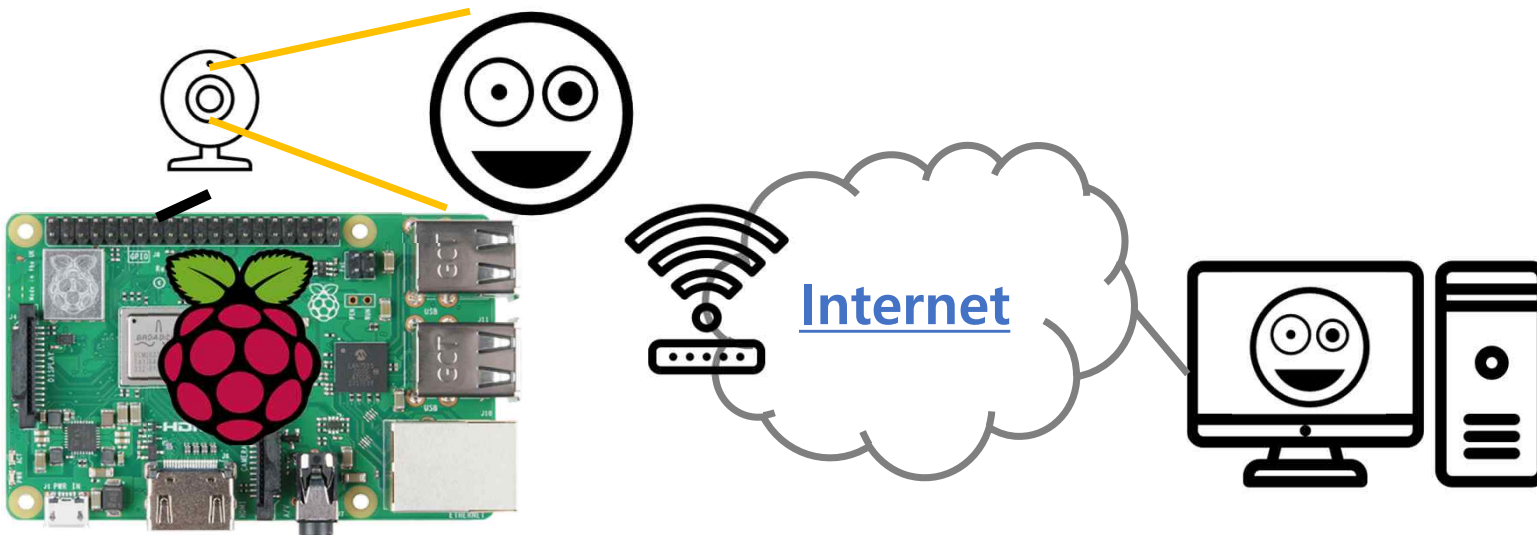
Lecture 10. 라즈베리파이 카메라 & Web CCTV

이재욱 (Jaewook Lee)

jlee0315@pknu.ac.kr

수업 목표

- 라즈베리파이 **Web CCTV** 만들기
 - 라즈베리파이를 통한 카메라 제어
 - Web을 통한 카메라 제어 및 스트리밍 서비스 구현

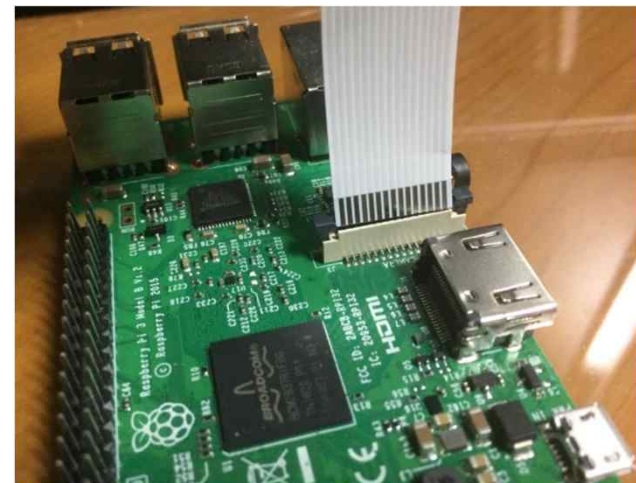
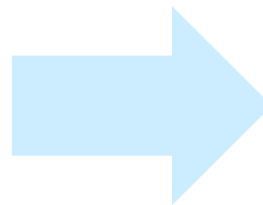


라즈베리파이를 통한 카메라

Pi 카메라

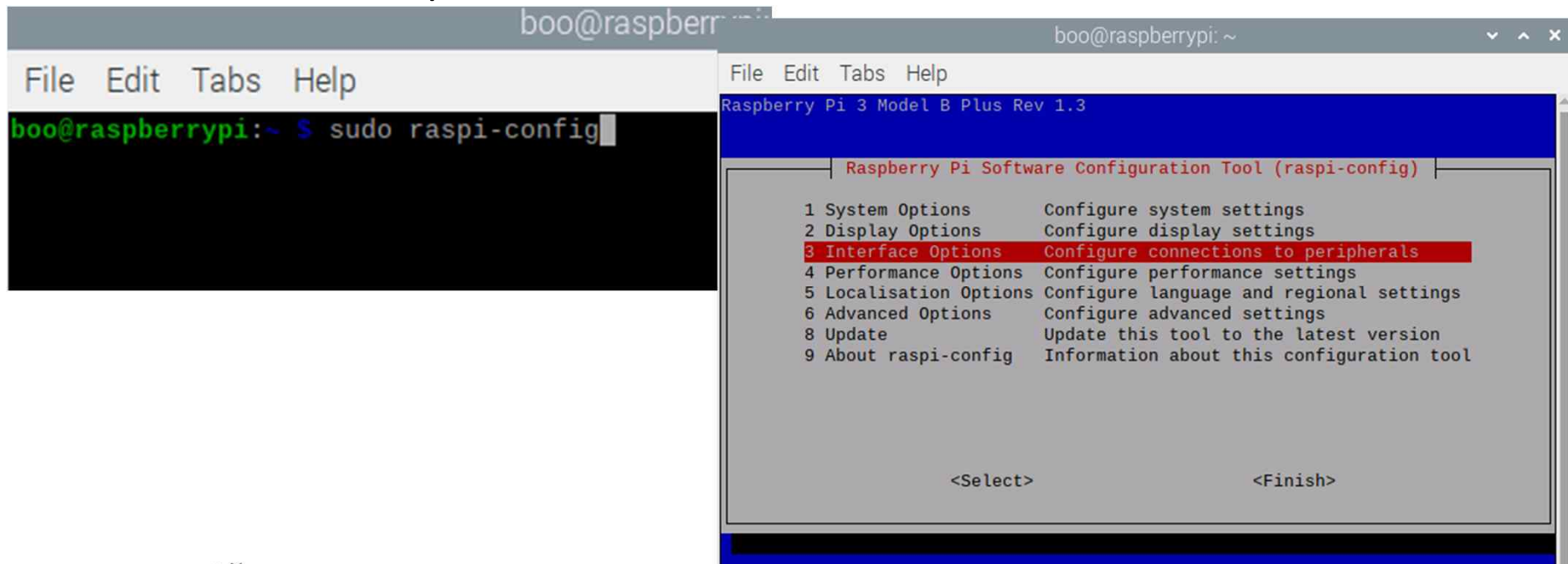
- Pi 카메라

- 라즈베리파이 카메라모듈
- 라즈베리파이 카메라 입력 핀에 카메라 모듈을 아래 그림과 같이 연결



Legacy Camera 설정

- 터미널 창에 아래와 같이 작성
 - sudo raspi-config (터미널)
 - ➔ 3. Interface Options



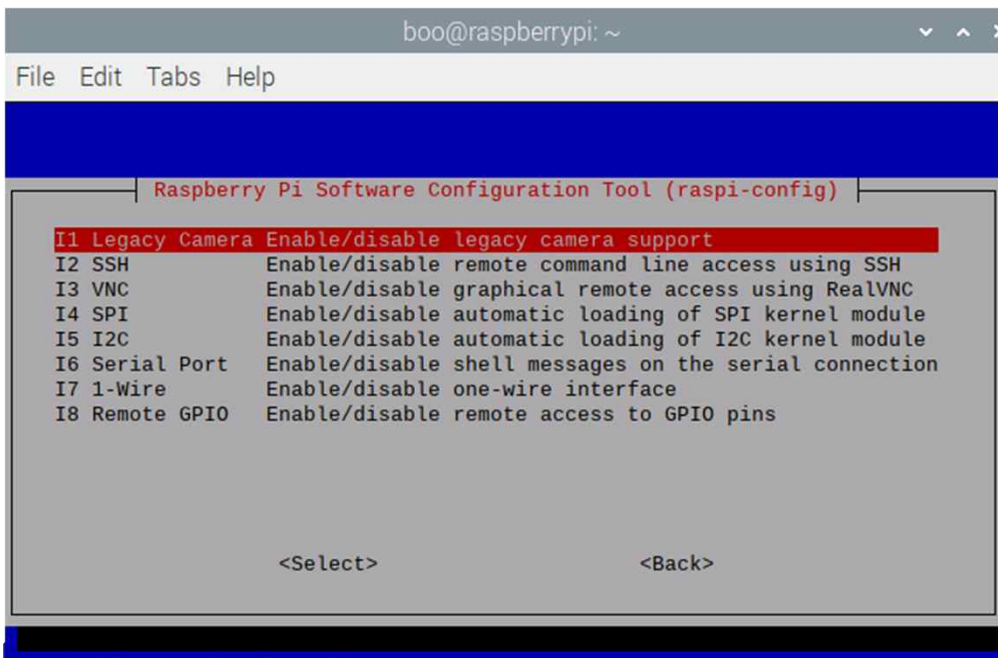
The image shows two terminal windows. The left window shows the command `boo@raspberrypi:~$ sudo raspi-config` being entered. The right window shows the output of the command, which is the Raspberry Pi Software Configuration Tool (raspi-config) menu. The menu lists several options, with '3 Interface Options' highlighted in red. The options are:

Raspberry Pi Software Configuration Tool (raspi-config)	
1 System Options	Configure system settings
2 Display Options	Configure display settings
3 Interface Options	Configure connections to peripherals
4 Performance Options	Configure performance settings
5 Localisation Options	Configure language and regional settings
6 Advanced Options	Configure advanced settings
8 Update	Update this tool to the latest version
9 About raspi-config	Information about this configuration tool

At the bottom of the menu, there are two buttons: '<Select>' and '<Finish>'.

Legacy Camera 설정

- 터미널 창에 아래와 같이 작성
 - sudo raspi-config (터미널)
 - sudo raspi-config → 3. Interface Options → I1 Legacy Camera → Yes

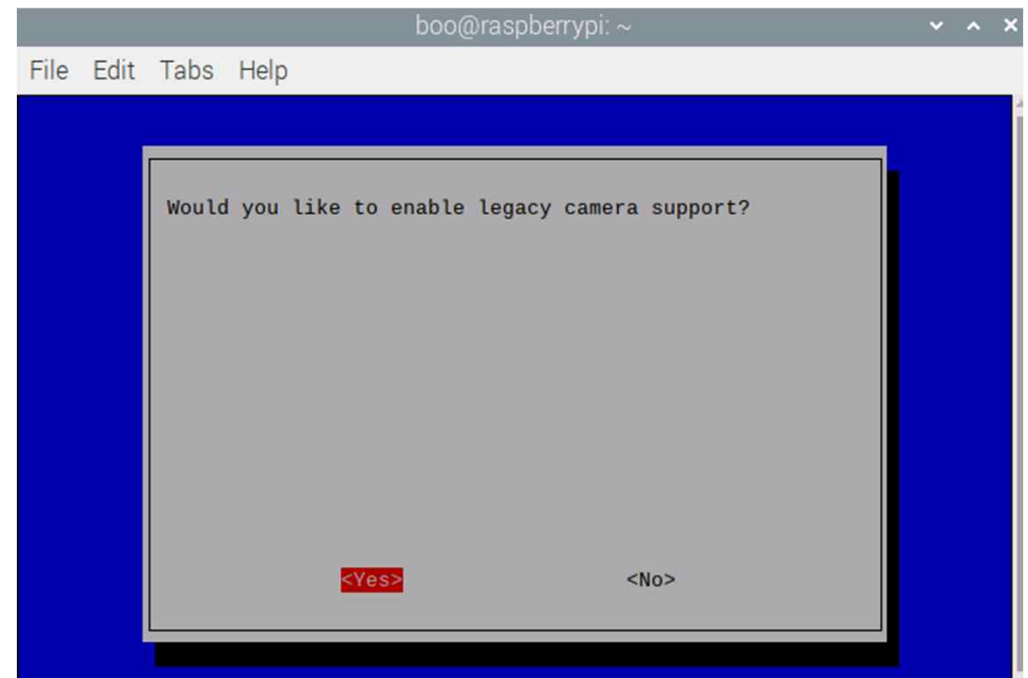


```
boo@raspberrypi: ~
File Edit Tabs Help

Raspberry Pi Software Configuration Tool (raspi-config)

I1 Legacy Camera Enable/disable legacy camera support
I2 SSH Enable/disable remote command line access using SSH
I3 VNC Enable/disable graphical remote access using RealVNC
I4 SPI Enable/disable automatic loading of SPI kernel module
I5 I2C Enable/disable automatic loading of I2C kernel module
I6 Serial Port Enable/disable shell messages on the serial connection
I7 1-Wire Enable/disable one-wire interface
I8 Remote GPIO Enable/disable remote access to GPIO pins

<Select> <Back>
```



```
boo@raspberrypi: ~
File Edit Tabs Help

Would you like to enable legacy camera support?

<Yes> <No>
```



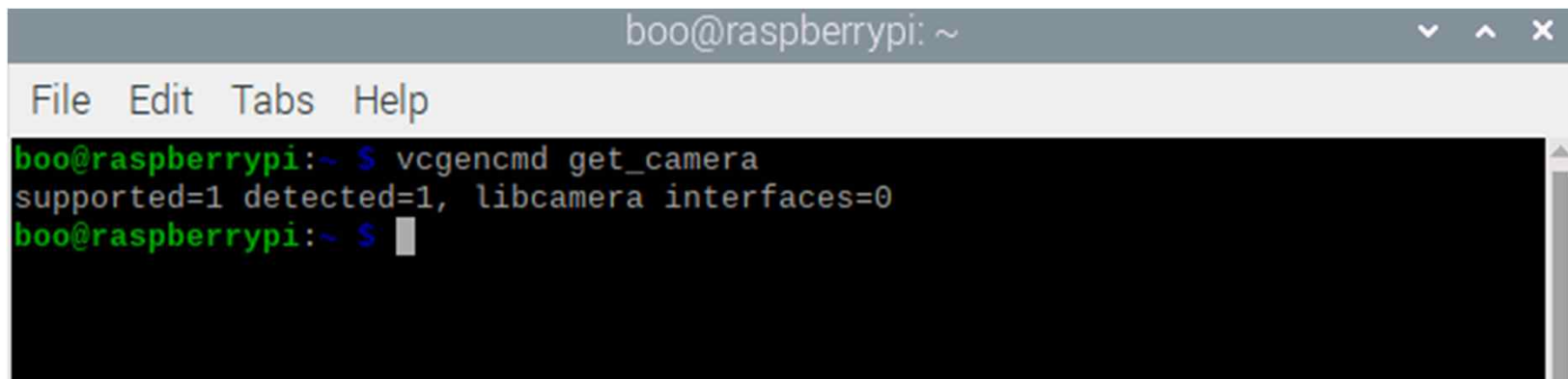
Legacy Camera 설정

- 터미널 창에 아래와 같이 작성
 - sudo raspi-config (터미널)
 - ➔ 3. Interface Options ➔ I1 Legacy Camera ➔ Yes ➔ Reboot



Pi 카메라 설치 확인

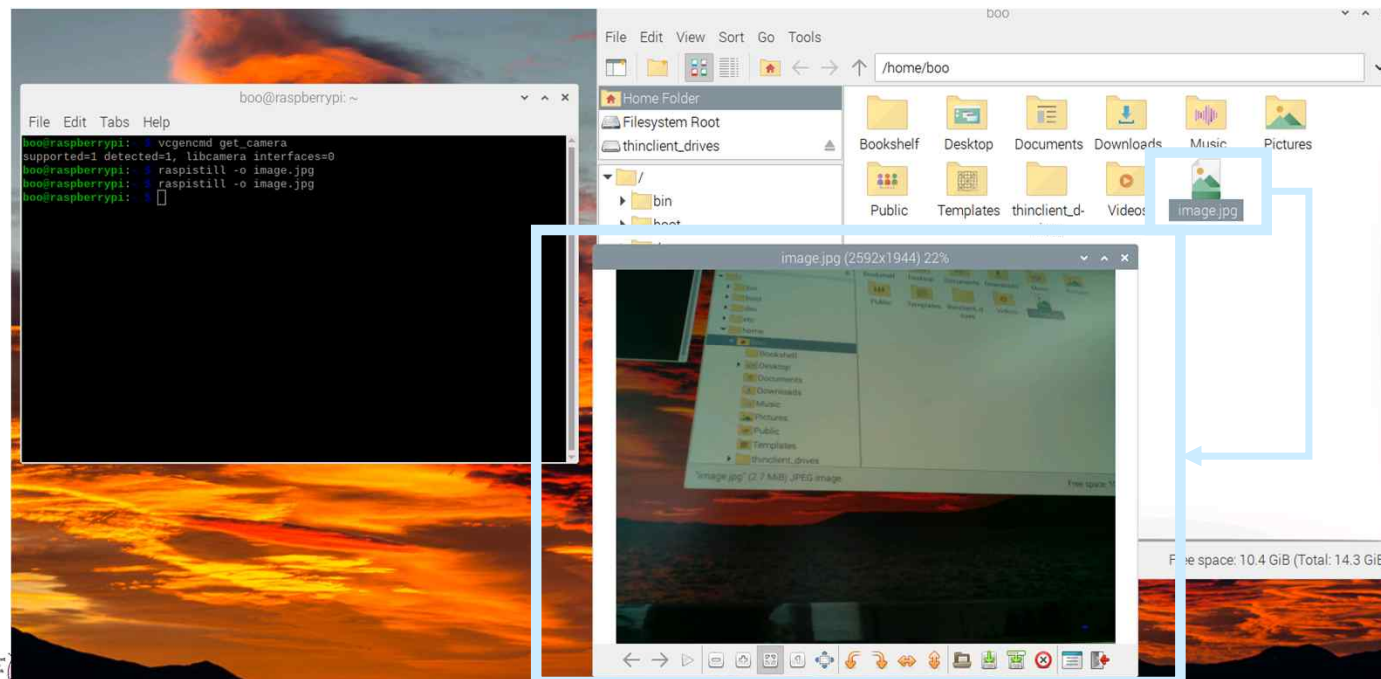
- 터미널 창에 아래와 같이 입력
 - vcgencmd get_camera
 - detected = 1 인 것 확인



```
boo@raspberrypi: ~  
File Edit Tabs Help  
boo@raspberrypi:~ $ vcgencmd get_camera  
supported=1 detected=1, libcamera interfaces=0  
boo@raspberrypi:~ $
```


Pi 카메라 작동확인

- 터미널 창에 아래와 같이 입력
 - raspistill -o image.jpg
 - 바탕화면에 image.jpg 사진 파일 생성 (카메라가 사진 찍고 사진 저장)



picamera 라이브러리

- **picamera 라이브러리**

- Methods

- capture()
 - start_recording(), stop_recording()
 - wait_recording()
 - start_preview(), stop_preview()

picamera 라이브러리

- **picamera 라이브러리**

- Attributes

- rotation, resolution, framerate, brightness, contrast
 - annotate_text, annotate_test_size
 - annotate_background, annotate_foreground
 - image_effect

Pi 카메라 실습

- Pi 카메라 실습

- 사진촬영
- 연속 사진 촬영
- 화질 및 상하좌우 반전
- 밝기조절
- 녹화
- OPENCV를 이용한 사진촬영 및 편집

실습1. 사진 촬영

1_capture.py

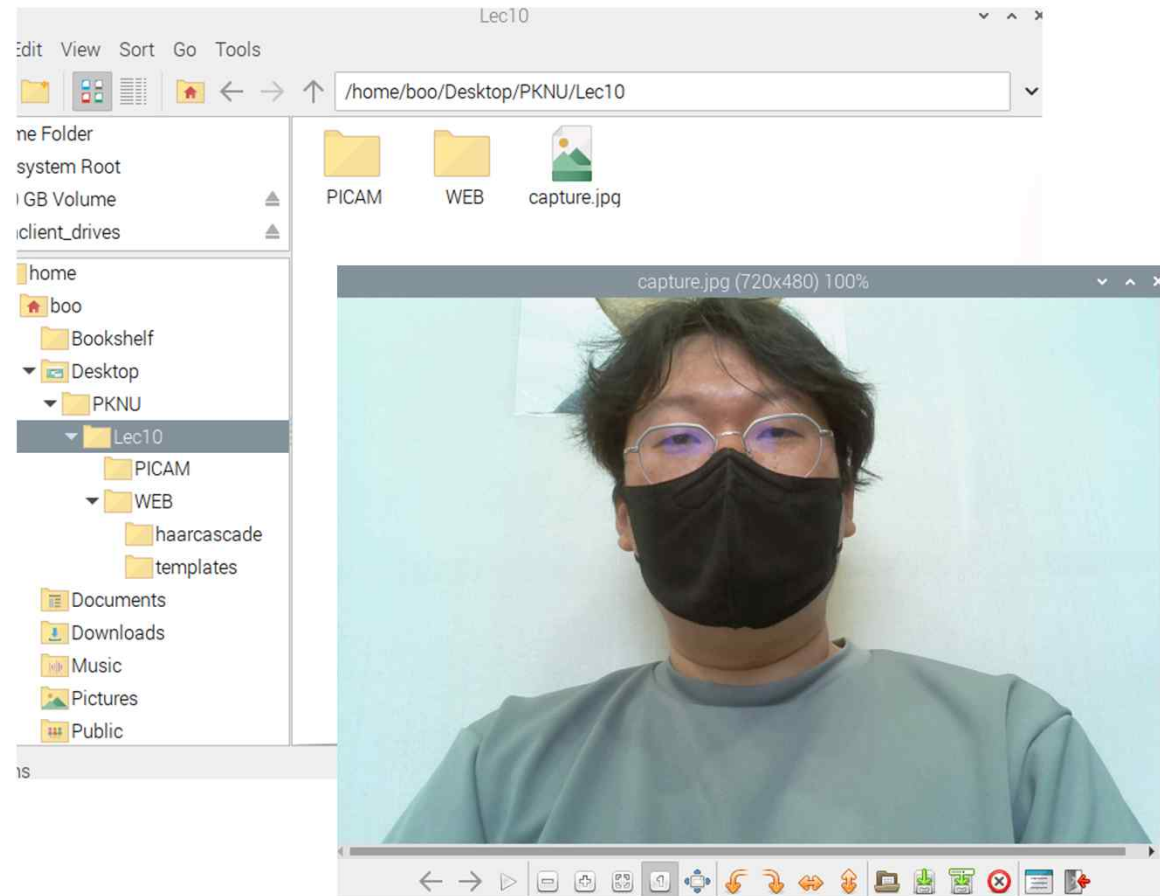
```
import picamera  
import time
```

```
camera = picamera.PiCamera()
```

```
camera.start_preview()  
time.sleep(5)  
camera.stop_preview()
```

```
camera.capture("/home/boo/Desktop/P  
KNU/Lec10/1_capture.jpg")
```

저장할 폴더 명/ 파일 명



실습2. 연속 사진 촬영

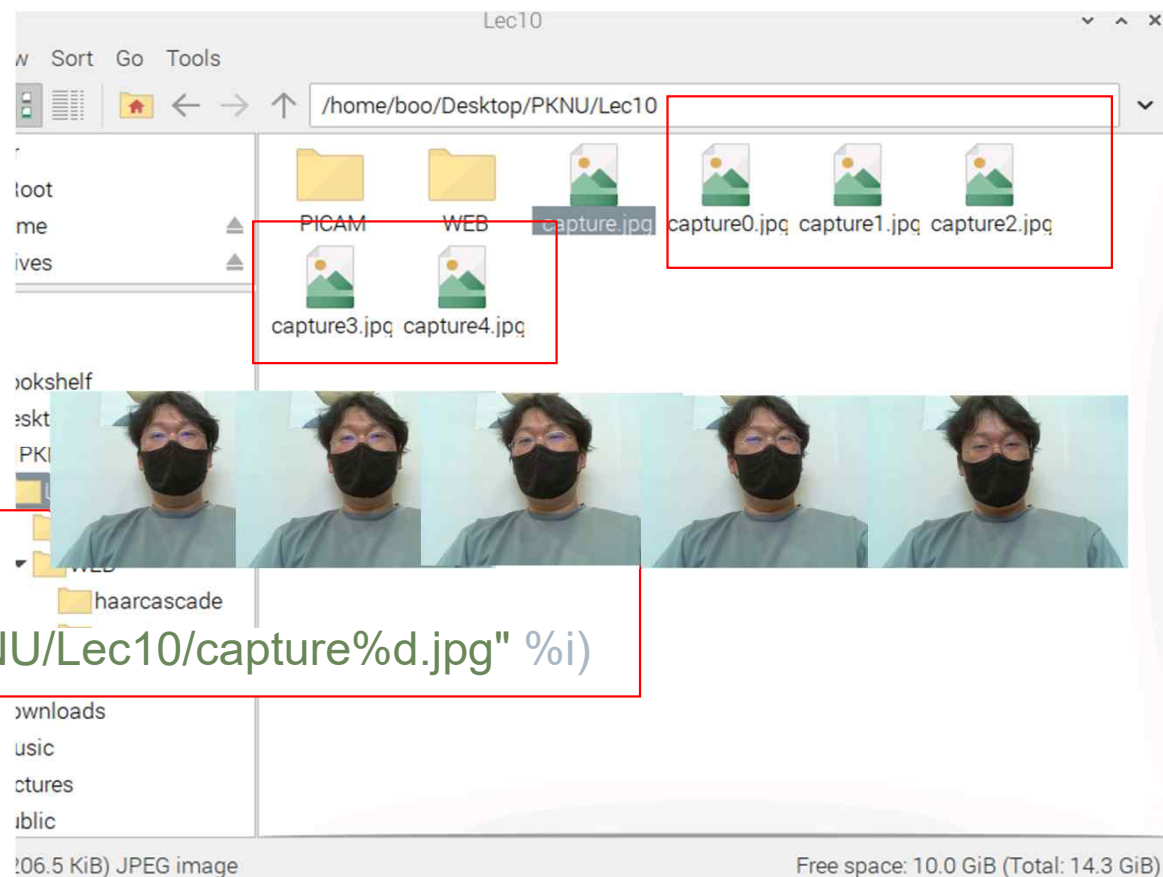
2_snapshot.py

```
import picamera  
import time
```

```
camera = picamera.PiCamera()  
camera.start_preview()
```

```
for i in range(5):  
    time.sleep(5)  
    camera.capture("/home/boo/Desktop/PKNU/Lec10/capture%d.jpg" %i)
```

```
camera.stop_preview()
```



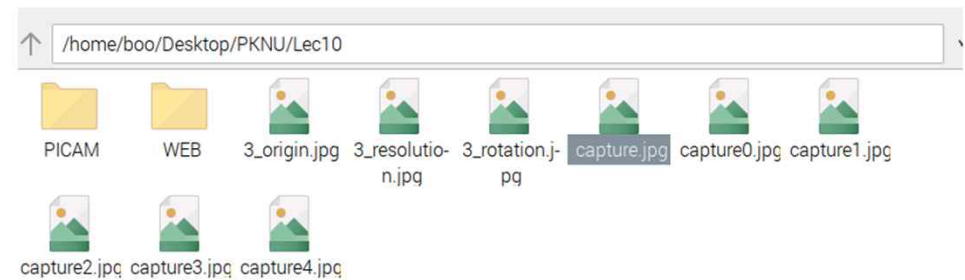
실습 3. Resolution & rotation

3_resolandrot.py

```
import picamera  
import time
```

```
camera = picamera.PiCamera()  
camera.start_preview()  
time.sleep(5)  
camera.stop_preview()  
camera.capture("/home/boo/Desktop/PKNU/Lec1  
0/3_origin.jpg")
```

```
camera.resolution = (920, 480)  
camera.start_preview()  
time.sleep(5)  
camera.stop_preview()  
camera.capture("/home/boo/Desktop/PKNU/Lec1  
0/3_resolution.jpg")
```

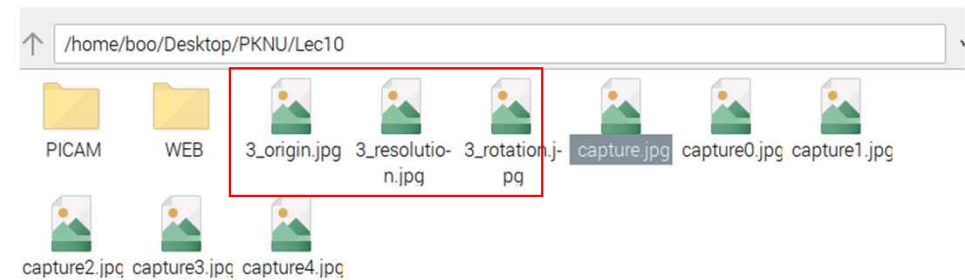


실습 3. Resolution & rotation

3_resolandrot.py

```
camera.rotation = 180  
camera.start_preview()  
time.sleep(5)  
camera.stop_preview()  
camera.capture("/home/boo/Desktop/PKNU/Lec  
10/3_rotation.jpg")
```

파이 카메라로 찍는 사진의 최대 해상도는 **2592 x 1944**, **framerate**를 15로 설정
- camera.framerate = 15



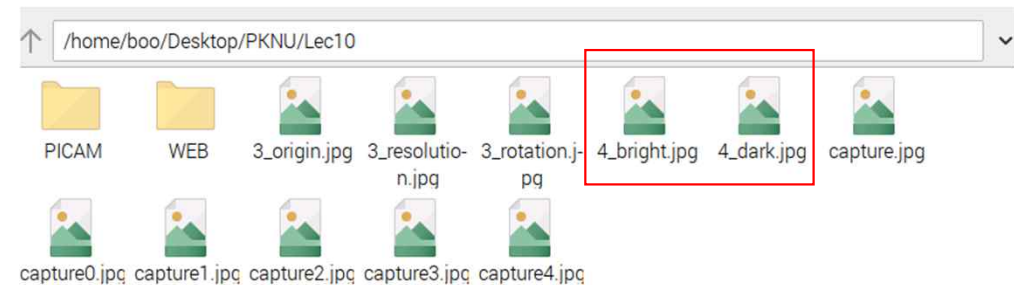
실습4. 밝기 조절

4_brightness.py

```
import picamera  
import time
```

```
camera = picamera.PiCamera()  
camera.start_preview()  
time.sleep(5)  
camera.stop_preview()  
camera.capture("/home/boo/Desktop/PKNU  
/Lec10/4_bright.jpg")
```

```
camera.brightness = 20  
camera.start_preview()  
time.sleep(5)  
camera.stop_preview()  
camera.capture("/home/boo/Desktop/PKNU  
/Lec10/4_dark.jpg")
```



실습 5. 주석 달기 (annotation)

5_annotation.py

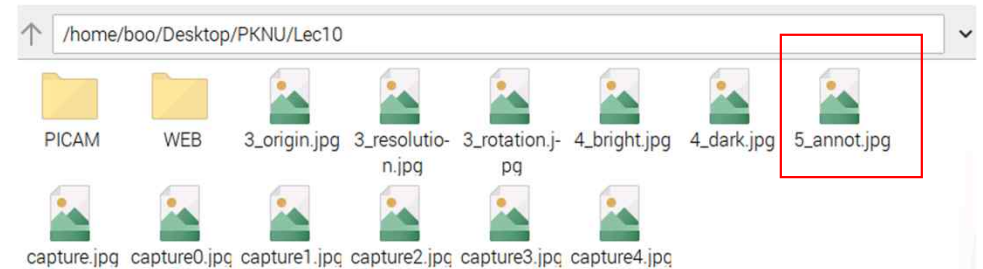
```
import picamera  
import time
```

```
camera = picamera.PiCamera()
```

```
camera.annotate_text_size = 30  
camera.annotate_background = picamera.Color('blue')  
camera.annotate_foreground = picamera.Color('yellow')  
camera.annotate_text = "Embedded System Design"
```

```
camera.start_preview()  
time.sleep(5)  
camera.stop_preview()
```

```
camera.capture("/home/boo/Desktop/PKNU/Lec10/5_  
annot.jpg")
```



실습 6. 녹화

6_recording.py

```
import picamera  
import time
```

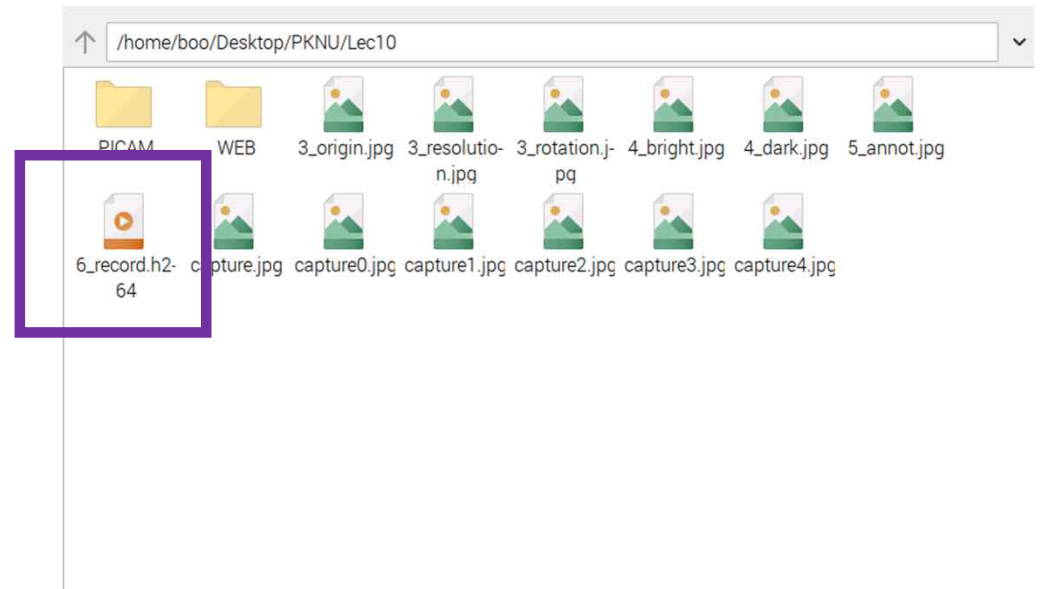
```
camera = picamera.PiCamera()
```

```
camera.start_preview()
```

```
camera.start_recording("/home/boo/Desktop/PKNU/Lec10/6_record.h264")  
camera.wait_recording(10)
```

```
camera.stop_preview()
```

```
camera.stop_recording()
```



OpenCV



- **Open Source Computer Vision (OpenCV)**

- 영상 처리에 사용할 수 있는 오픈 소스 라이브러리
- 컴퓨터가 사람의 눈처럼 인식할 수 있게 처리해주는 역할을 수행 하게 해줌
 - 공장에서 제품 검사할 때
 - 의료 영상 처리 및 보정 그리고 판단
 - CCTV영상
 - 로봇틱스
- 카메라로 찍어서 할 수 있는 모든 일은 OpenCV로 처리 가능

OpenCV 설치

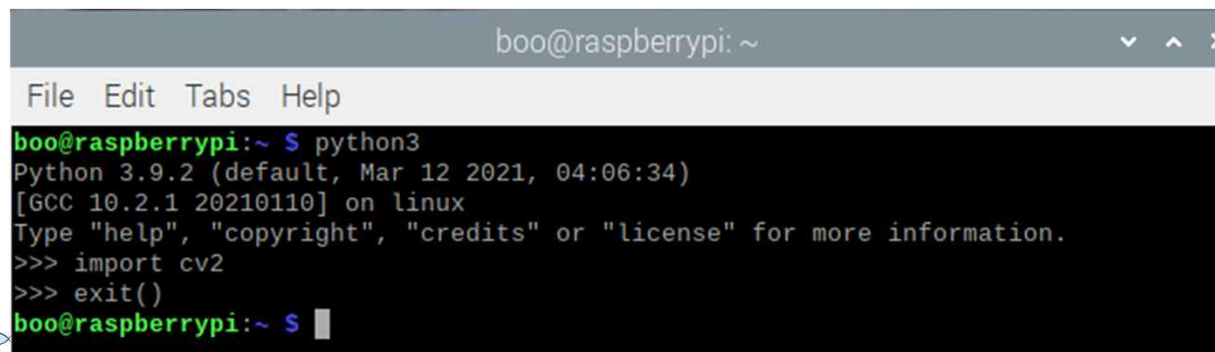
- 터미널 창에 아래와 같이 명령어 기입

```
sudo apt-get update
sudo apt-get upgrade

sudo apt-get install python3-opencv
```

- 설치확인

- 터미널 창에 python3 친 후 import cv2
- 설치에 이상 없으면 exit() 친 후 빠져 나올 것



```
boo@raspberrypi: ~
File Edit Tabs Help
boo@raspberrypi:~ $ python3
Python 3.9.2 (default, Mar 12 2021, 04:06:34)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> exit()
boo@raspberrypi:~ $
```



OpenCV 라이브러리

- **Capture & read**

- `cap = cv2.VideoCapture(file_path or index)`

- `file_path`: 동영상 파일 경로
 - `index`: 카메라 장치 번호 (0, 1, 2, ...)
 - `cap`: VideoCapture 객체

- `ret, img = cap.read()`

- `ret`: 프레임 읽기 성공 여부 (True/False)
 - `img`: 프레임 이미지, Numpy array

OpenCV 라이브러리

- **Print**

- cv2.imshow(title, img)

- title: 윈도우 창의 제목
 - img: 출력할 이미지 객체, Numpy array

- cv2.waitKey(time)

- time: 입력 대기 시간 (0: 무한대기)

OpenCV 라이브러리

- **Save**

- `cv2.imwrite(file_name, img)`

- `file_name`: 저장할 영상 파일 이름
 - `img`: 저장할 이미지 객체, Numpy array

- **Color conversion**

- `cv2.cvtColor(img, flag)`

- `img`: 색상을 변경할 이미지 객체, Numpy array
 - `flag`: `cv2.COLOR_BGR2RGB`, `cv2.COLOR_BGR2GRAY`, `cv2.COLOR_GRAY2RGB`

실습 7. OpenCV 실습

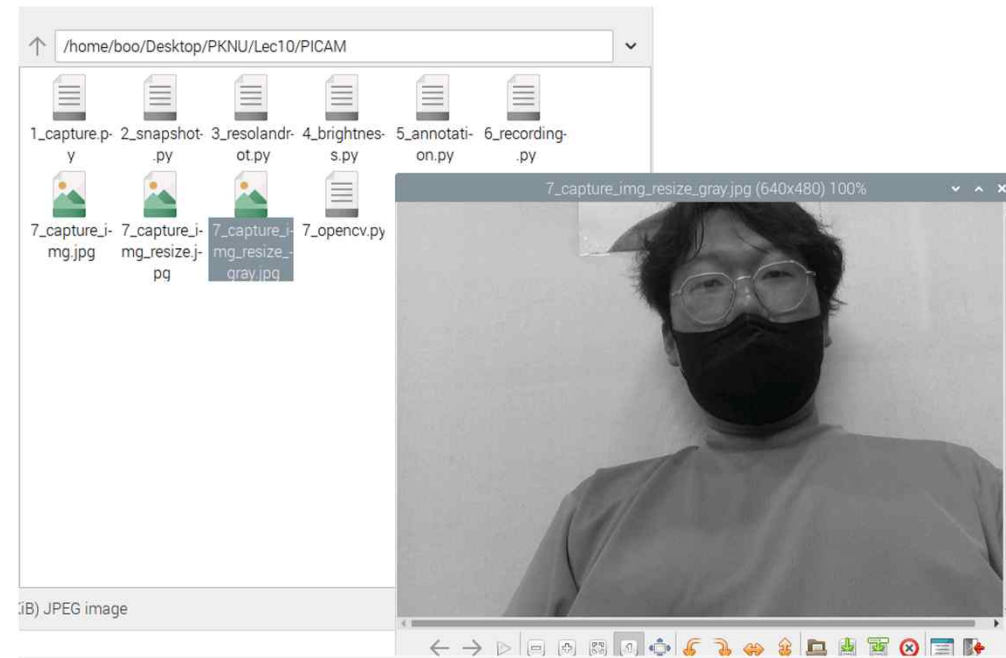
```
import cv2
import numpy as np
```

```
cap = cv2.VideoCapture(0)
ret, img = cap.read()
cap.release()
```

```
cv2.imshow('capture_img', img)
cv2.waitKey(0)
cv2.imwrite('7_capture_img.jpg', img)
```

```
img_resize = cv2.resize(img, dsize=(640, 480))
cv2.imshow('capture_img_resize', img_resize)
cv2.waitKey(0)
cv2.imwrite('7_capture_img_resize.jpg', img_resize)
```

```
img_gray = cv2.cvtColor(img_resize, cv2.COLOR_BGR2GRAY)
cv2.imshow('capture_img_resize_gray', img_gray)
cv2.waitKey(0)
cv2.imwrite('7_capture_img_resize_gray.jpg', img_gray)
```



PI 카메라 과제

• PI 카메라 과제

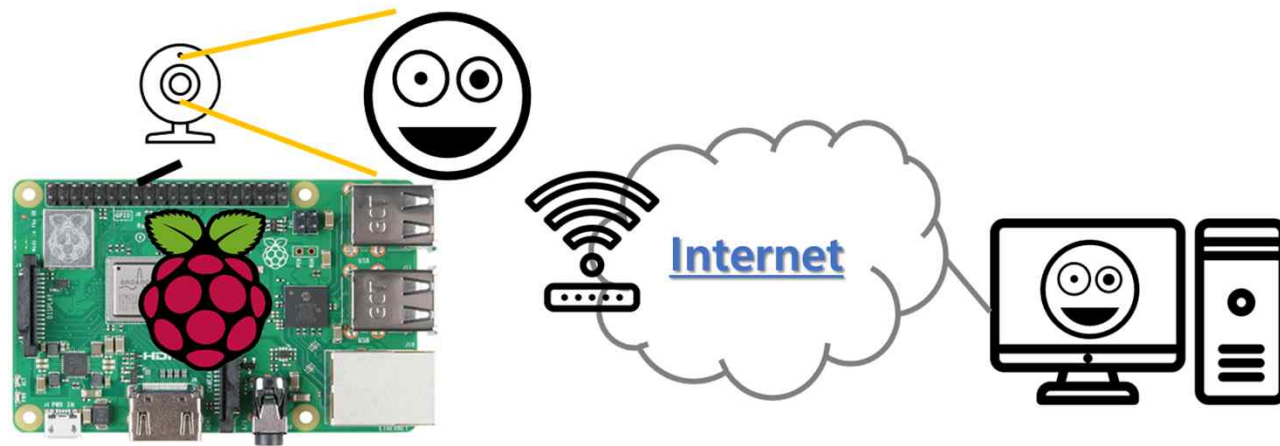
– PI 카메라 실습 1~*7까지 결과를 보고서에 적어 제출하시오

- 1. 사진촬영
- 2. 연속 사진 촬영
- 3. 화질 및 상하좌우 반전
- 4. 밝기조절
- 5. 주석 달기 < 조원 이름들을 영어로>
- 6. 녹화
- 7. OPENCV를 이용한 사진촬영 및 편집

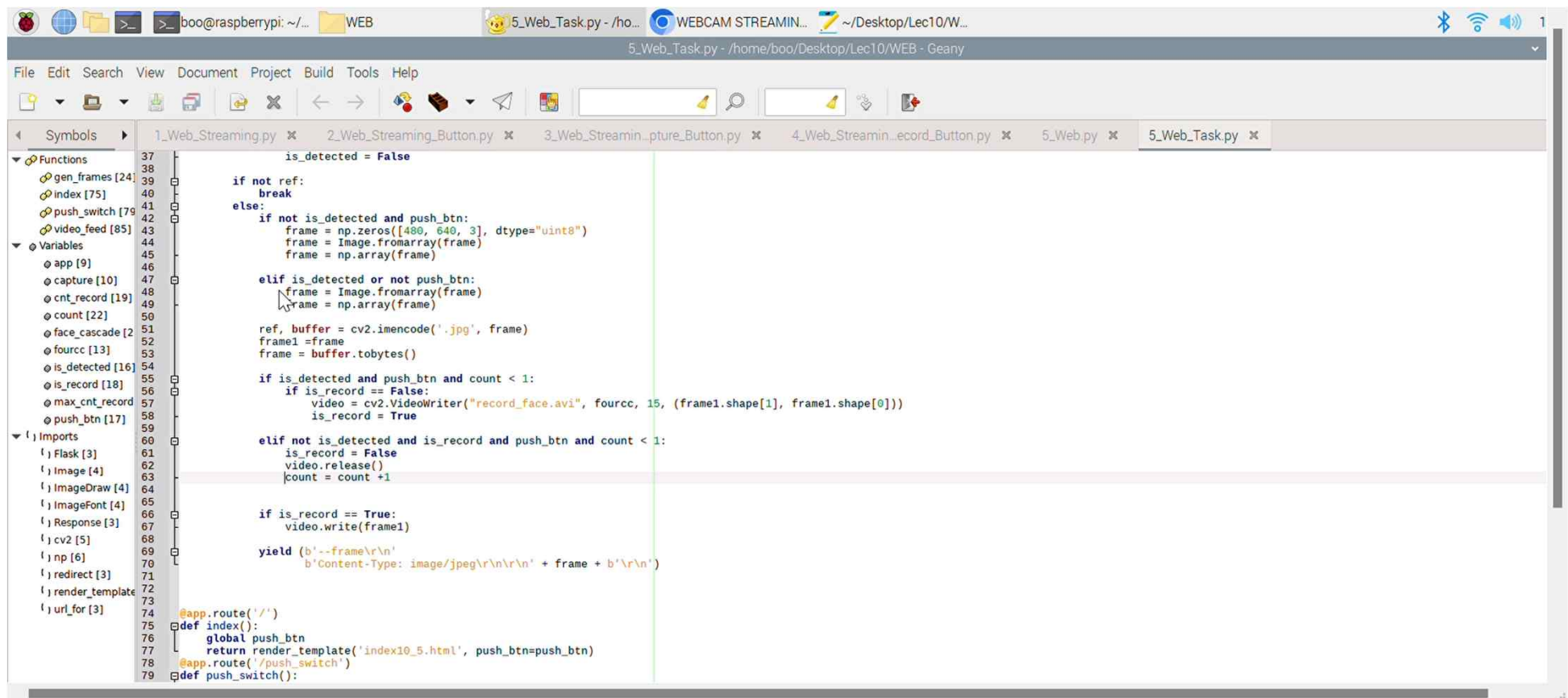
– 코드 X, 사진 및 영상 파일 업로드

웹 스트리밍 실습

- 웹 스트리밍 실습
 - 1) 웹 스트리밍 기본
 - 2) ON/OFF 웹 스트리밍
 - 3) 녹화 및 캡처 기능이 있는 웹 스트리밍
 - 4) 얼굴 인식 웹 스트리밍



목표: 얼굴인식 웹 스트리밍



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders for 'Functions', 'Variables', and 'Imports'. The code editor displays a Python script for a web application that streams video frames from a camera. The script includes logic for detecting faces, recording them, and serving them over a web browser. The code is written in Python 3 and uses libraries like Flask, cv2, and numpy.

```
37 is_detected = False
38
39 if not ref:
40     break
41 else:
42     if not is_detected and push_btn:
43         frame = np.zeros([480, 640, 3], dtype="uint8")
44         frame = Image.fromarray(frame)
45         frame = np.array(frame)
46
47     elif is_detected or not push_btn:
48         frame = Image.fromarray(frame)
49         frame = np.array(frame)
50
51     ref, buffer = cv2.imencode('.jpg', frame)
52     frame1 = frame
53     frame = buffer.tobytes()
54
55     if is_detected and push_btn and count < 1:
56         if is_record == False:
57             video = cv2.VideoWriter("record_face.avi", fourcc, 15, (frame1.shape[1], frame1.shape[0]))
58             is_record = True
59
60     elif not is_detected and is_record and push_btn and count < 1:
61         is_record = False
62         video.release()
63         count = count + 1
64
65     if is_record == True:
66         video.write(frame1)
67
68     yield (b'--frame\r\n'
69           b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
70
71 @app.route('/')
72 def index():
73     global push_btn
74     return render_template('index10_5.html', push_btn=push_btn)
75
76 @app.route('/push_switch')
77 def push_switch():
```



기본 HTML 배대

html 파일은 templates 폴더를
만들어서 옮기세요

```
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
      integrity="sha384-MCw98/SFnGE8fJT36XE0ngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPM0" crossorigin="anonymous">
    <title>WEBCAM STREAMING</title>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-lg-8 offset-lg-2">
          <h3 class="mt-5">Live Streaming</h3>
          
        </div>
        <div class="col-lg-8 offset-lg-2">
          <a href="{{ url_for('push_capture') }}"><input type="button" value="Capture" ></a>
          <a href="{{ url_for('push_record') }}"><input type="button" value="Recording/Ready" ></a>
          <b> {% if is_record!=0 %} Recording {% elif is_record==0 %} Ready {%
            endif %}
          </b>
        </div>
      </div>
    </div>
  </body>
</html>
```

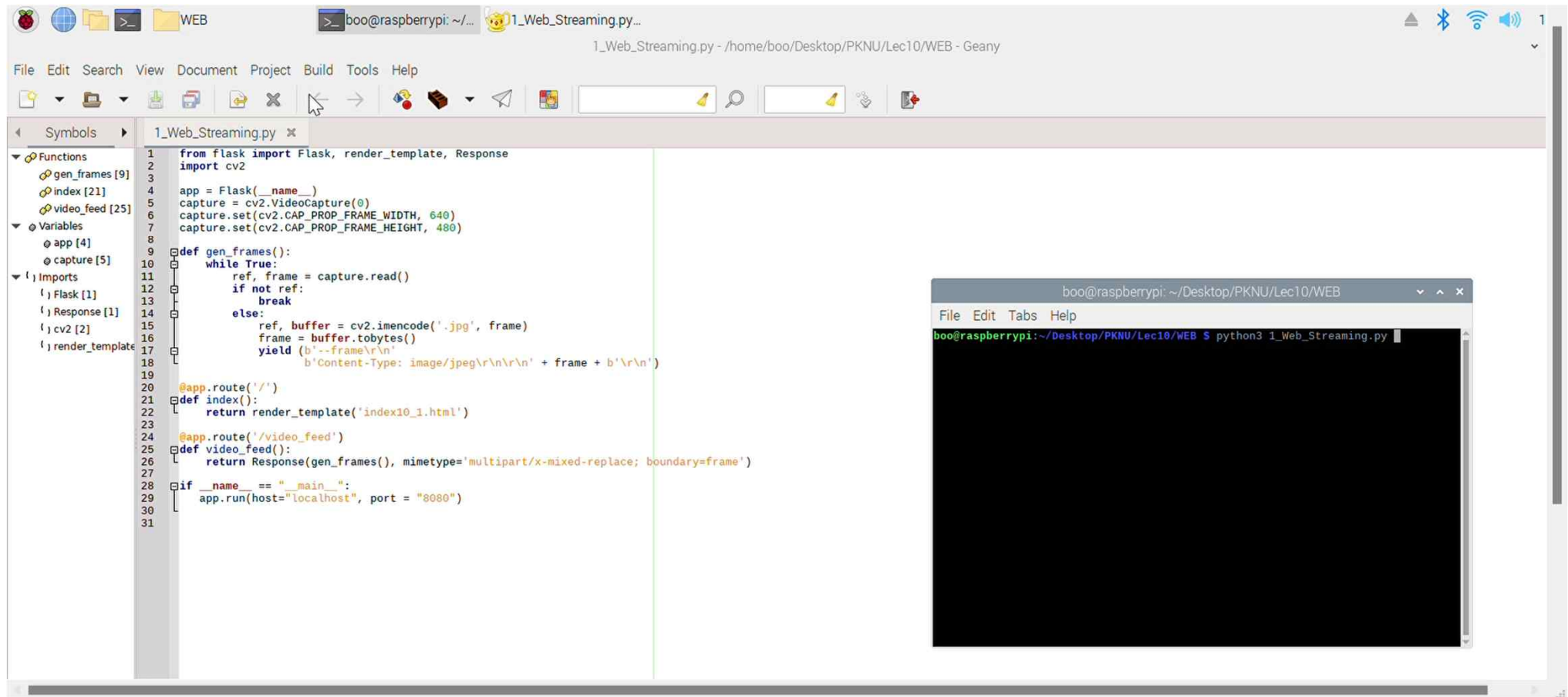
Live Streaming



Capture Recording/Ready {% if is_record!=0 %} Recording {% elif is_record==0 %} Ready {%
endif %}



웹 스트리밍 기초



The screenshot displays a Raspberry Pi desktop environment. The primary window is a code editor titled '1_Web_Streaming.py...'. The code is written in Python and implements a web streaming application using Flask and OpenCV. The code includes imports for Flask, render_template, Response, and cv2. It initializes a Flask app and a video capture object. A generator function 'gen_frames()' reads video frames and yields them as JPEG images with appropriate headers. The app has two routes: a root route 'index()' that renders a template, and a '/video_feed' route that returns the generated frames as a streaming response. The main block runs the app on localhost:8080.

```
1 from flask import Flask, render_template, Response
2 import cv2
3
4 app = Flask(__name__)
5 capture = cv2.VideoCapture(0)
6 capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
7 capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
8
9 def gen_frames():
10     while True:
11         ref, frame = capture.read()
12         if not ref:
13             break
14         else:
15             ref, buffer = cv2.imencode('.jpg', frame)
16             frame = buffer.tobytes()
17             yield (b'--frame\r\n'
18                  b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
19
20 @app.route('/')
21 def index():
22     return render_template('index10_1.html')
23
24 @app.route('/video_feed')
25 def video_feed():
26     return Response(gen_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')
27
28 if __name__ == "__main__":
29     app.run(host="localhost", port = "8080")
30
31
```

In the bottom right corner, a terminal window is open, showing the command 'python3 1_Web_Streaming.py' being executed in the directory '~/Desktop/PKNU/Lec10/WEB'.



웹 스트리밍 기초

```
from flask import Flask, render_template, Response
import cv2
```

```
app = Flask(__name__)
capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
```

```
def gen_frames():
    while True:
        ref, frame = capture.read()
        if not ref:
            break
        else:
            ref, buffer = cv2.imencode('.jpg', frame)
            frame = buffer.tobytes()
            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```



웹 스트리밍 기초

```
@app.route('/')  
def index():  
    return render_template('index10_1.html')
```

```
@app.route('/video_feed')  
def video_feed():  
    return Response(gen_frames(),  
                    mimetype='multipart/x-mixed-replace;  
                    boundary=frame')
```

```
if __name__ == "__main__":  
    app.run(host="localhost", port = "8080")
```

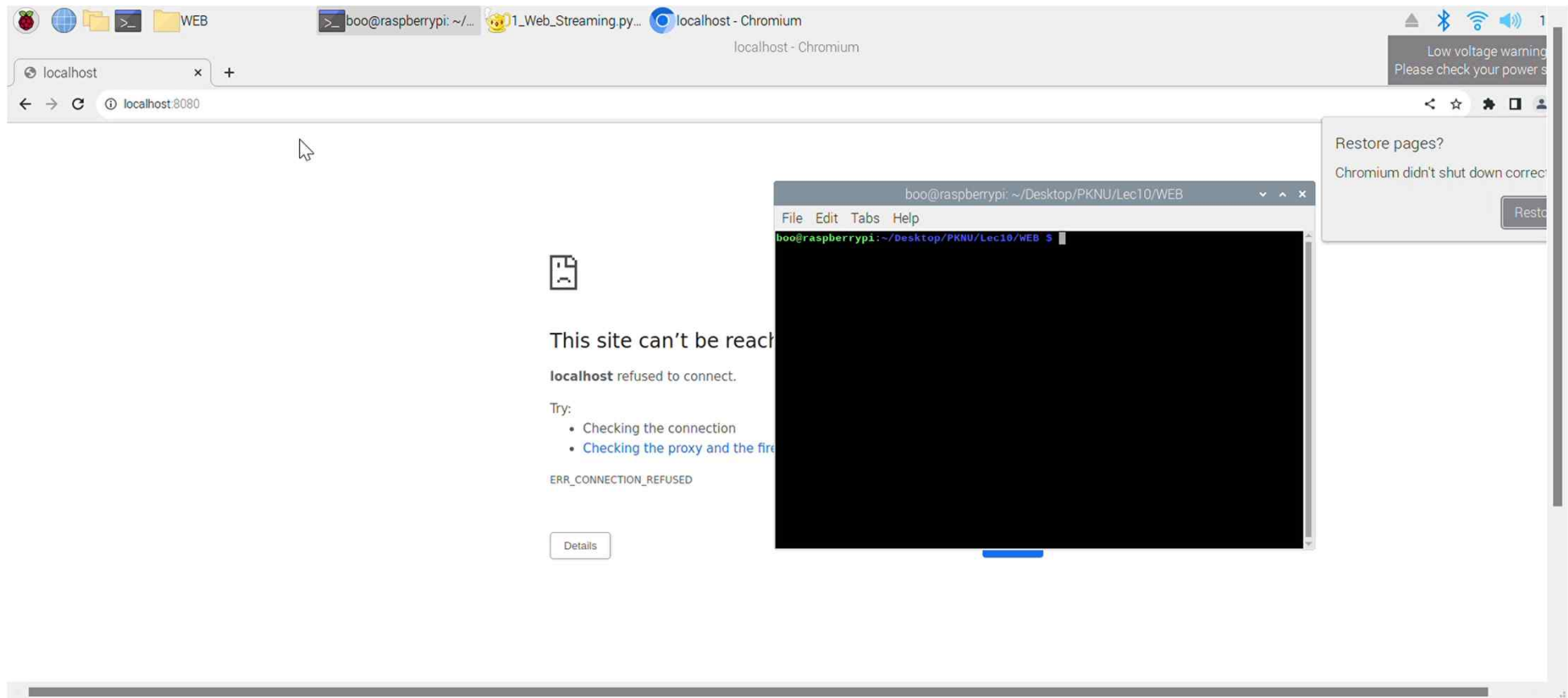
Live Streaming



localhost:8080



ON/OFF 웹 스트리밍



ON/OFF 웹 스트리밍

```
from flask import Flask, render_template, Response, url_for, redirect
from PIL import ImageFont, ImageDraw, Image
import cv2
import numpy as np
```

```
app = Flask(__name__)
capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
```

```
global push_btn
push_btn = True
```

ON/OFF 웹 스트리밍

```
def gen_frames():  
    global push_btn  
    while True:  
        ref, frame = capture.read()  
        if not ref:  
            break  
        else:  
            if push_btn:  
                frame = np.zeros([480, 640, 3], dtype="uint8")  
                frame = Image.fromarray(frame)  
                frame = np.array(frame)  
            ref, buffer = cv2.imencode('.jpg', frame)  
            frame = buffer.tobytes()  
            yield (b'--frame\r\n'  
                  b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```



ON/OFF 웹 스트리밍

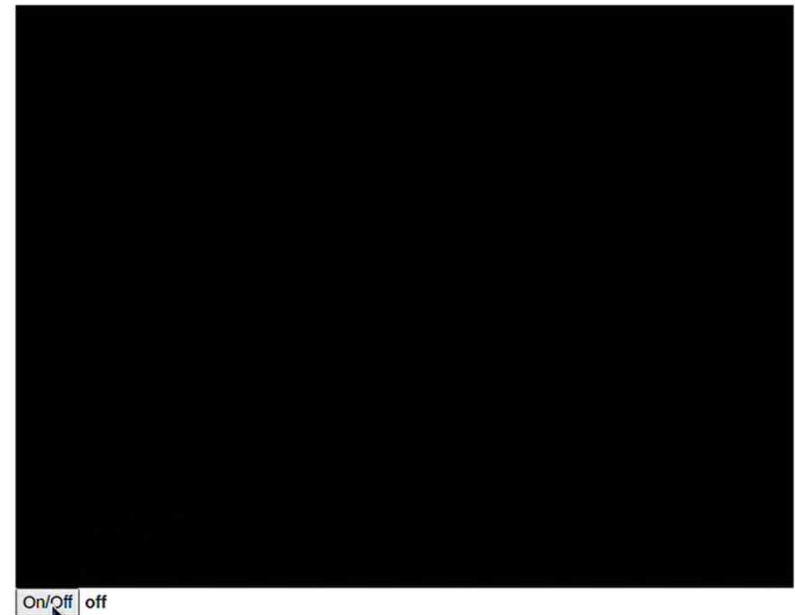
```
@app.route('/')
def index():
    global push_btn
    return render_template('index10_2.html',
push_btn=push_btn)
```

```
@app.route('/video_feed')
def video_feed():
    return Response(gen_frames(),
mimetype='multipart/x-mixed-replace;
boundary=frame')
```

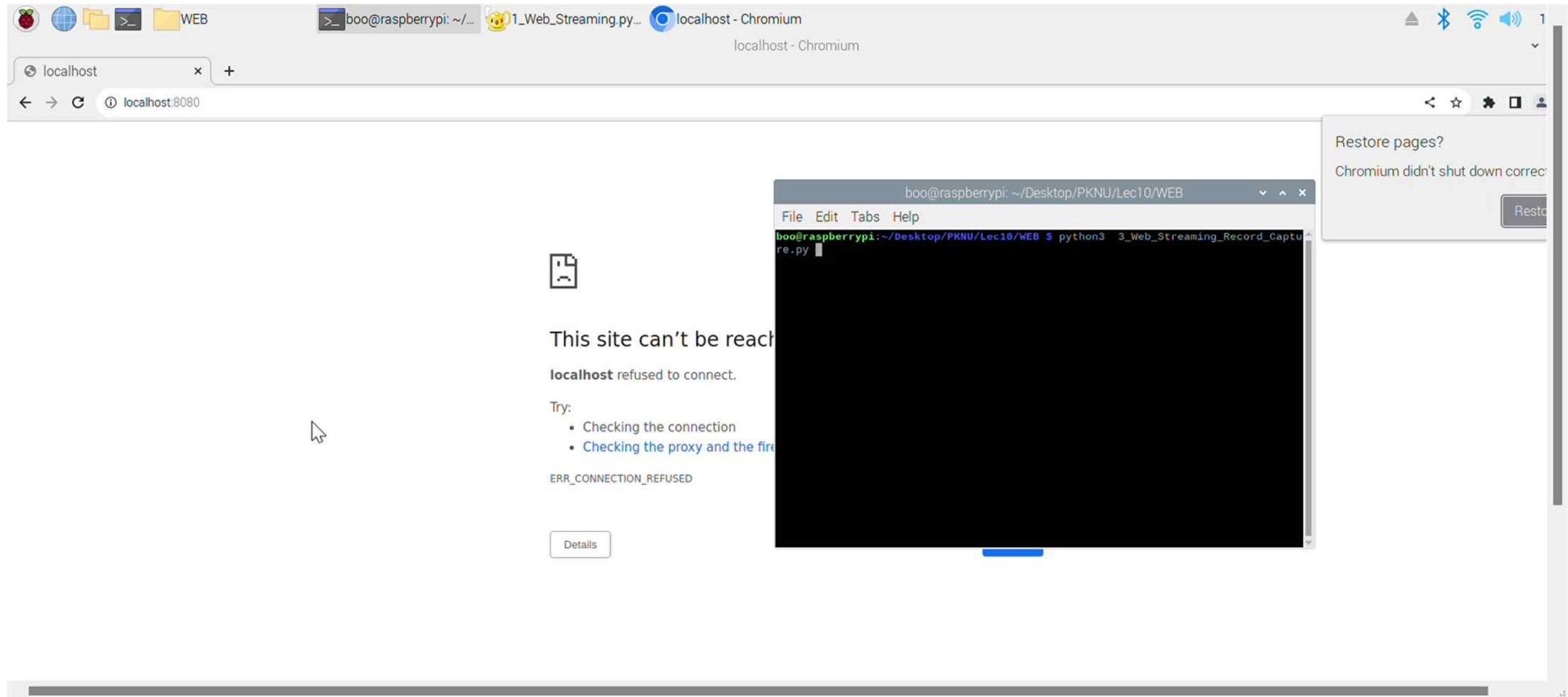
```
@app.route('/push_switch')
def push_switch():
    global push_btn
    push_btn = not push_btn
    return redirect(url_for('index'))
```

```
if __name__ == "__main__":
    app.run(host="localhost", port = "8080")
```

Live Streaming



웹 스트리밍 and 캡처 & 레코딩



웹 스트리밍 and 캡처 & 레코딩

```
from flask import Flask, render_template, Response, url_for, redirect
from PIL import ImageFont, ImageDraw, Image
import cv2
import numpy as np
```

```
app = Flask(__name__)
global is_capture, is_record, start_record
capture = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
is_record = False
is_capture = False
start_record = False
```



웹 스트리밍 and 캡처 & 레코딩

```
def gen_frames():  
    global is_record, start_record, is_capture, video  
    while True:  
        ref, frame = capture.read()  
        if not ref:  
            break  
        else:  
            frame = Image.fromarray(frame)  
            frame = np.array(frame)  
            ref, buffer = cv2.imencode('.jpg', frame)  
            frame1 = frame  
            frame = buffer.tobytes()  
            if start_record == True and is_record == False:  
                is_record = True  
                start_record = False  
                video = cv2.VideoWriter("record.avi", fourcc, 15,  
(frame1.shape[1], frame1.shape[0]))  
            elif start_record == True and is_record == True:  
                is_record = False  
                start_record = False  
                video.release()
```

```
elif is_capture:  
    is_capture = False  
    cv2.imwrite("record_capture.png", frame1)  
  
if is_record == True:  
    video.write(frame1)  
  
yield (b'--frame\r\n'  
        b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```



웹 스트리밍 and 캡처 & 레코딩

```
@app.route('/')
def index():
    global is_record
    return render_template('index10_4.html', is_record=is_record)
```

```
@app.route('/video_feed')
def video_feed():
    return Response(gen_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')
```

```
@app.route('/push_record')
def push_record():
    global start_record
    start_record = not start_record
    return redirect(url_for('index'))
```

```
@app.route('/push_capture')
def push_capture():
    global is_capture
    is_capture = True
    return redirect(url_for('index'))
```

```
if __name__ == "__main__":
    app.run(host="localhost", port = "8080")
```

Live Streaming

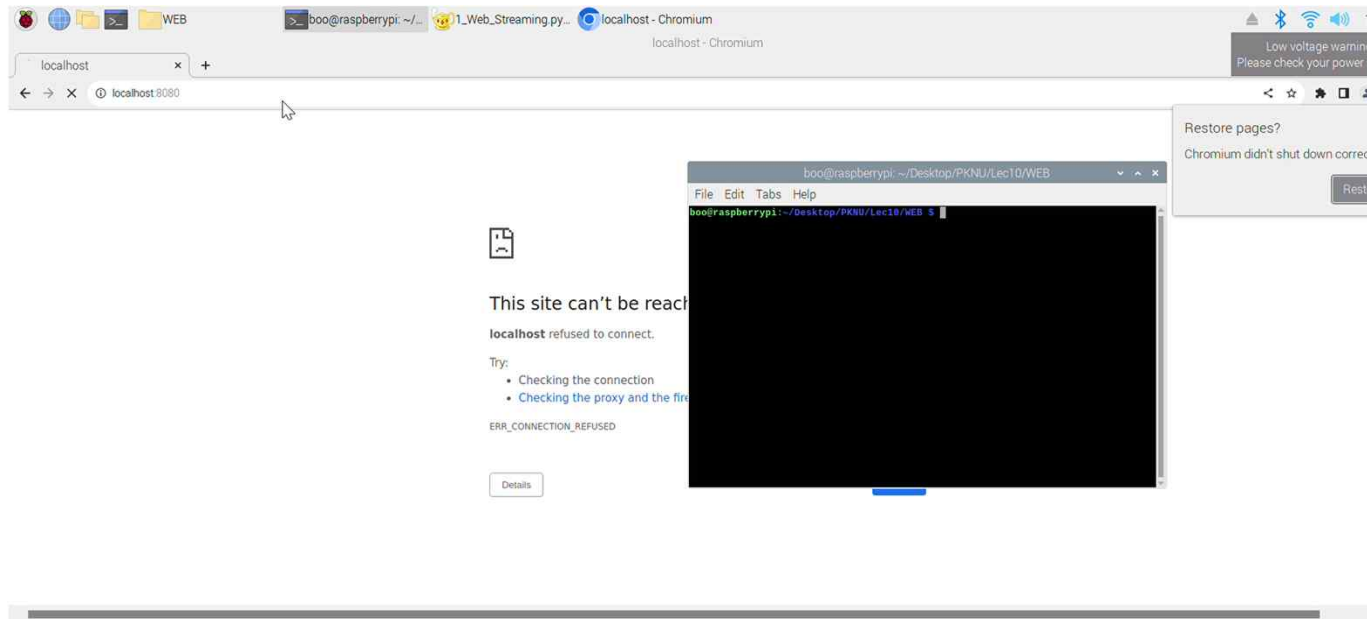


Capture Recording/Ready Ready



Advanced 웹 스트리밍

- Haar Cascade 검출기를 이용한 웹 스트리밍
 - Normal Mode: 보통 스트리밍
 - Face Detection Mode: 얼굴이 인식 된 경우에만 스트리밍 ON

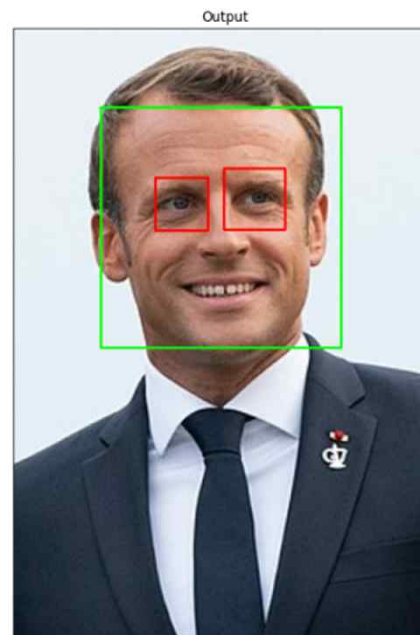


Haar Cascade 검출기

• Haar Cascade 검출기

- 'Rapid Object Detection using a Boosted Cascade of Simple Features' 논문 (2001년 발표)에서 제안한 객체 검출기

파일명	검출 대상
haarcascade_frontalface_default.xml haarcascade_frontalface_alt.xml haarcascade_frontalface_alt2.xml haarcascade_frontalface_alt_tree.xml	정면 얼굴 검출
haarcascade_profileface.xml	측면 얼굴 검출
haarcascade_smile.xml	웃음 검출
haarcascade_eye.xml haarcascade_eye_tree_eyeglasses.xml haarcascade_lefteye_2splits.xml haarcascade_righteye_2splits.xml	눈 검출
haarcascade_frontalcatface.xml haarcascade_frontalcatface_extended.xml	고양이 얼굴 검출
haarcascade_fullbody.xml	사람의 전신 검출
haarcascade_upperbody.xml	사람의 상반신 검출
haarcascade_lowerbody.xml	사람의 하반신 검출
haarcascade_russian_plate_number.xml haarcascade_licence_plate_rus_16stages.xml	러시아 자동차 번호판 검출



```
detectorPaths = {  
    "face": "haarcascade_frontalface_default.xml",  
    "eyes": "haarcascade_eye.xml",  
}
```



Advanced 웹 스트리밍

```
from flask import Flask, render_template, Response, url_for, redirect
from PIL import ImageFont, ImageDraw, Image
import cv2
import numpy as np

app = Flask(__name__)
capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
global is_detected, push_btn, cnt_record, max_cnt_record
is_detected = False
push_btn = False
cnt_record = 0
max_cnt_record = 30
face_cascade = cv2.CascadeClassifier('haarcascade/haarcascade_frontalface_default.xml')
```



Advanced 웹 스트리밍

```
def gen_frames():
    global is_detected, push_btn, cnt_record, max_cnt_record
    while True:
        ref, frame = capture.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor= 1.5, minNeighbors=3, minSize=(20,20))

        if len(faces) :
            is_detected = True
            cnt_record = max_cnt_record
        else:
            cnt_record -= 1
            if cnt_record == 0:
                is_detected = False
        if not ref:
            break
```



Advanced 웹 스트리밍

```
if not ref:
    break
else:
    if not is_detected and push_btn:
        frame = np.zeros([480, 640, 3], dtype="uint8")
        frame = Image.fromarray(frame)
        frame = np.array(frame)
    elif is_detected or not push_btn:
        frame = Image.fromarray(frame)
        frame = np.array(frame)
    ref, buffer = cv2.imencode('.jpg', frame)
    frame = buffer.tobytes()
    yield (b'--frame\r\n'
           b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```



Advanced 웹 스트리밍

```
@app.route('/')
def index():
    global push_btn
    return render_template('index10_5.html', push_btn=push_btn)
@app.route('/push_switch')
def push_switch():
    global push_btn
    push_btn = not push_btn
    return redirect(url_for('index'))

@app.route('/video_feed')
def video_feed():
    return Response(gen_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == "__main__":
    app.run(host="localhost", port = "8080")
```

Live Streaming

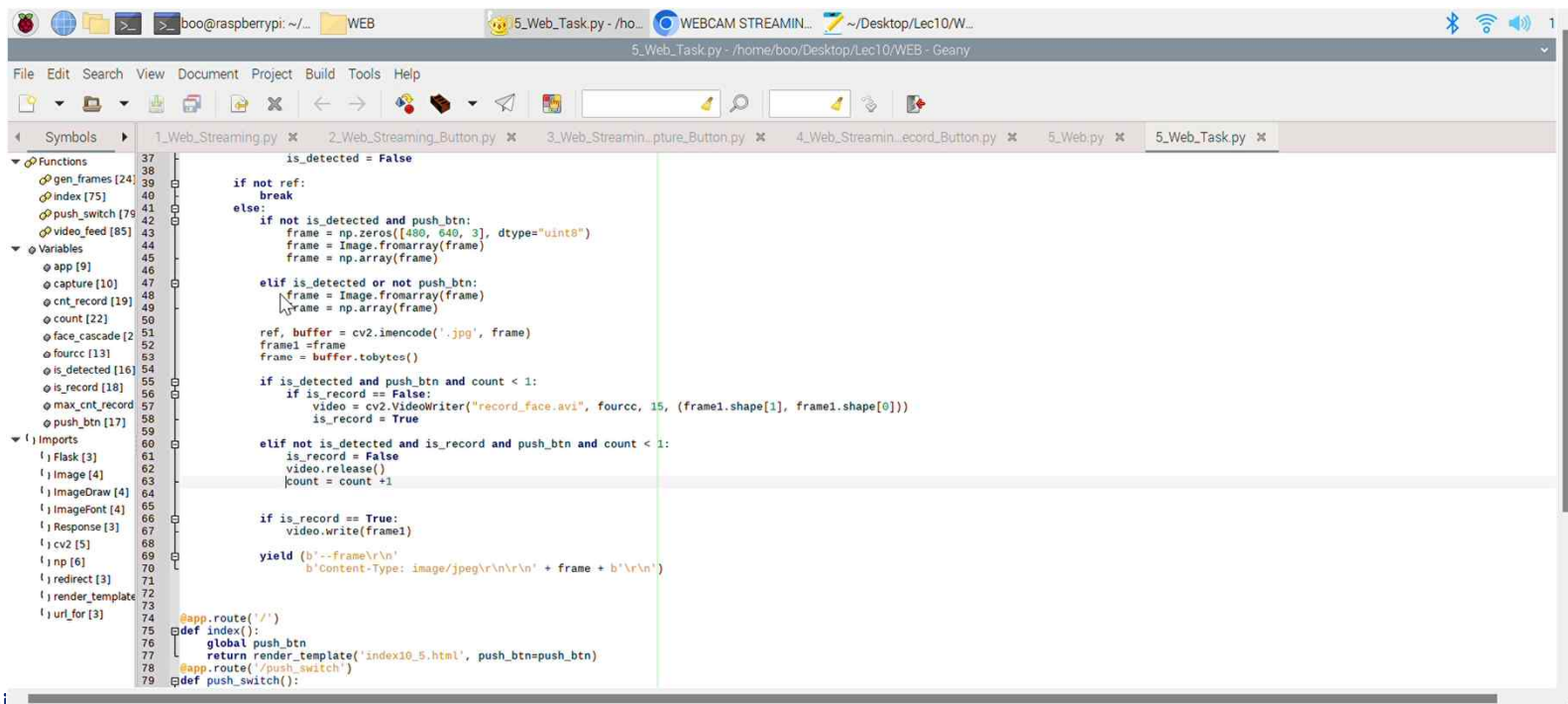


Mode Face Detection Mode



Web 카메라 제어

- Web 카메라 제어 (index10_5.html 사용)
 - 얼굴이 인식모드에서 처음 얼굴이 인식되는 경우에만 레코딩
 - 코드와 결과물 모두 제출



```
37 is_detected = False
38
39 if not ref:
40     break
41 else:
42     if not is_detected and push_btn:
43         frame = np.zeros([480, 640, 3], dtype="uint8")
44         frame = Image.fromarray(frame)
45         frame = np.array(frame)
46
47     elif is_detected or not push_btn:
48         frame = Image.fromarray(frame)
49         frame = np.array(frame)
50
51     ref, buffer = cv2.imencode('.jpg', frame)
52     frame1 = frame
53     frame = buffer.tobytes()
54
55     if is_detected and push_btn and count < 1:
56         if is_record == False:
57             video = cv2.VideoWriter("record_face.avi", fourcc, 15, (frame1.shape[1], frame1.shape[0]))
58             is_record = True
59
60     elif not is_detected and is_record and push_btn and count < 1:
61         is_record = False
62         video.release()
63         count = count + 1
64
65     if is_record == True:
66         video.write(frame1)
67
68     yield (b'--frame\r\n'
69           b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
70
71 @app.route('/')
72 def index():
73     global push_btn
74     return render_template("index10_5.html", push_btn=push_btn)
75 @app.route('/push_switch')
76 def push_switch():
```



과제 팁

- ***을 채우시오
 - Global 변수 선언: is_record, count, video

```
if is_detected and push_btn and count < 1:
    if ***:
        video = cv2.VideoWriter("record_face.avi", fourcc, 15, (frame1.shape[1], frame1.shape[0]))
        is_record = ***

elif not is_detected and is_record and push_btn and count < 1:
    is_record = ***
    video.release()
    count = ***

if ***:
    video.write(frame1)
```

