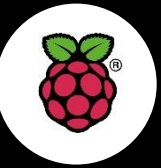


# Facial recognition Temperature sensing

With RASPBERRY PI 4

김규태 강민영 서강원





2020년 10월 30일 기준 코로나 위기 경보 심

스드를 반영한 기준

출처 : <https://www.kdca.go.kr>

## 2. 감염예방을 위한 관리 철저

### □ 감염관리를 위한 전담직원 지정 배치

- 출입 시 사전위생 확인 등 전담직원 배치하여 관리 책임성 부여

### □ 시설출입 시 방역 관리 강화

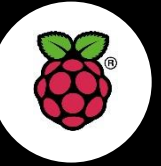
- 시설 종사자 등은 **개인위생 준수**(마스크 착용, 손 씻기) 후 출입 안내
- 시설 관리자의 판단에 따라 시설 종사자 등에 대한 1일 2회 발열 또는 호흡기 증상 확인(체온 확인 등 사전체크) **(붙임 1)**

### □ 다음과 같은 직원 또는 이용자는 출근 또는 이용을 중단할 것

- ① 발열 또는 호흡기 증상이 나타난 직원 또는 이용자는 증상이 나타나지 않을 때까지 출근 또는 이용을 중단할 것

출처 : 중앙 방역 대책 본부

코로나19\_집단시설\_다중이용시설\_대응지침(2판\_\_0226)



라즈베리  
파이 3

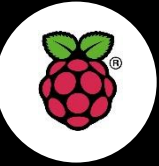
안면인식

열화상  
카메라

PYTHON



체온 측정 및 판독 시스템 구현



## QMainWindow

### 웹캠

얼굴 인식

얼굴 좌표 출력

인식 박스 위에 온도 출력

### 열화상 카메라

온도 측정

얼굴 center 온도 출력

체온 범위 판독 및 출력

◦ 얼굴 단일 인식

◦ 카메라 - 얼굴 사이의 거리 약 60cm

◦ 얼굴의 특정 부위 온도로 출력해서  
체온의 이상 유무를 판독

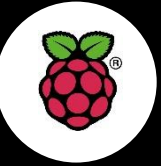
- 저온 : 35°C 이하

- 정상 : 35°C ~ 38°C

- 고온 : 38°C 이상

# 부품 리스트 (1) 라즈베리파이 3

Enjoy your stylish business and campus life with BIZCAM



## 라즈베리파이3 모델 B Raspberry pi3 Model B v 1.2



### 사양

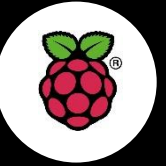
- 1GB LPDDR2 SDRAM 메모리
- 5V/2.5A DC 전원 입력(마이크로 USB) 전원 공급 장치
- 무선 연결:
  - 2.4GHz 및 5GHz IEEE 802.11.b/g/n/ac 무선 LAN 및 Bluetooth 4.2/BLE
  - USB 2.0을 통한 기가비트 이더넷(최대 처리량: 300Mbps)
  - USB 2.0 포트 4개
- 확장 40핀 GPIO(Genere Purpose Input Output) 헤더
- 풀 사이즈 HDMI 비디오 출력

### 설계 환경

- Python 3.7.3
- OpenCV 4.4.0

# 부품 리스트 (2) Webcam

Enjoy your stylish business and campus life with BIZCAM



Eyeview DJ-1000



Logitech C270 HD Webcam 720p



VS

## 사양

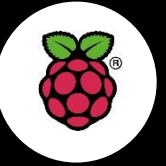
- 300만 화소 지원
- VGA ¼" CMOS 컬러 이미지 센서
- 전송속도 30fps
- 초점거리 5cm~무한대, 초점 수동 조정 가능
- USB 포트 연결

## 사양

- 720만 화소 지원
- 전송속도 30fps
- 고정 초점
- USB 포트 연결

# 부품 리스트 (2) Webcam

Enjoy your stylish business and campus life with BIZCAM

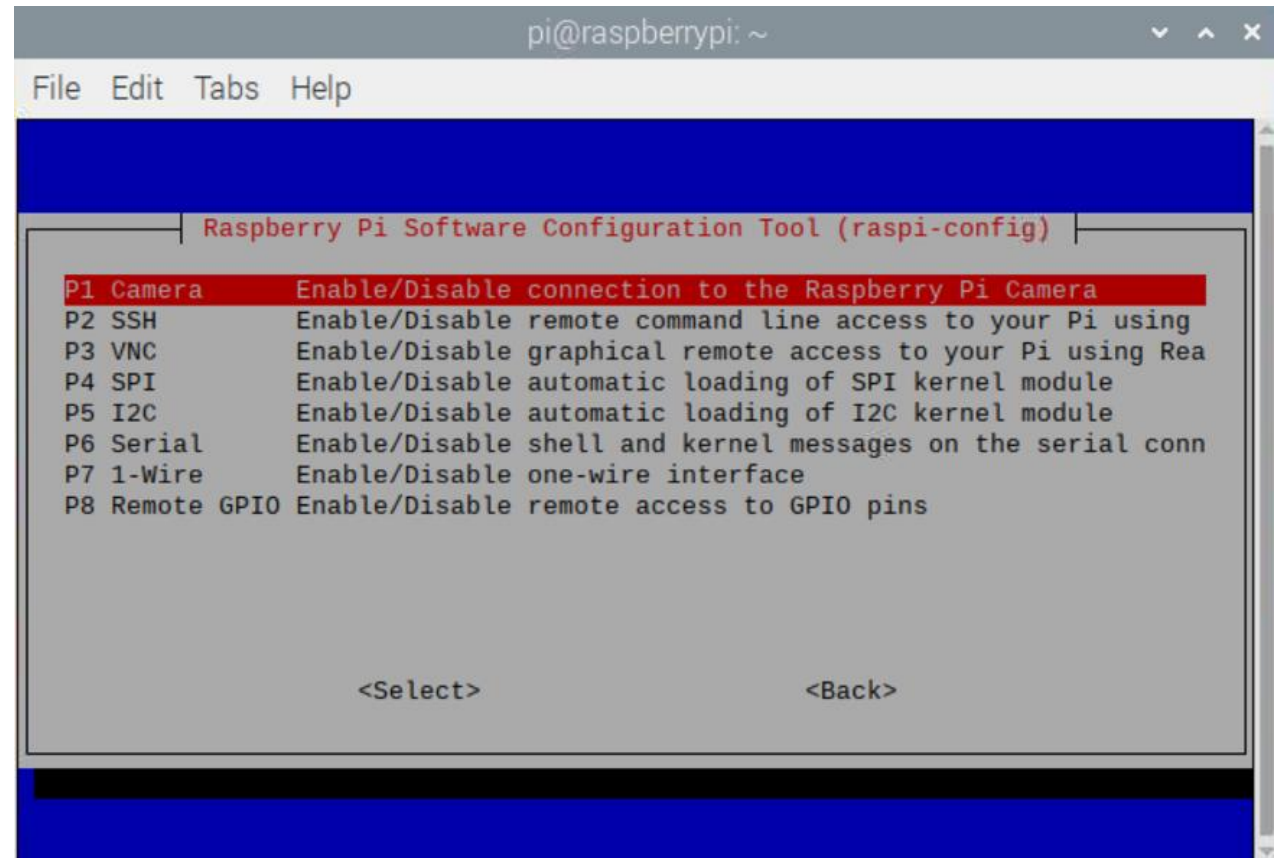


## SW

라즈베리파이에서 camera 설정을 enable로 바꾸기  
sudo raspi-config  
Interfacing Option -> Camera -> Enable

## HW

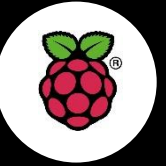
라즈베리파이 USB포트에 연결





# 부품 리스트 (3) 열화상 카메라

Enjoy your stylish business and campus life with BIZCAM

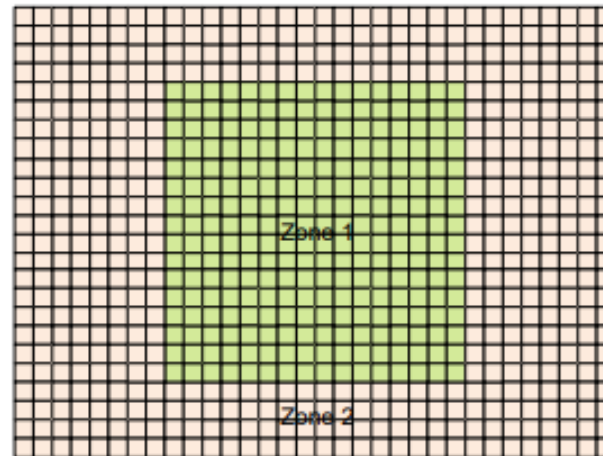


## 열화상 카메라 MLX90640 sensor



### 사양

- Refresh rate : 0.5Hz ~ 64Hz (4Hz : 250ms 마다 RAM에 기록)
- Definition : 32x24 (768 pixels)
- Temperature :  $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$   
( 체온 범위 내 )



MLX90640BAB

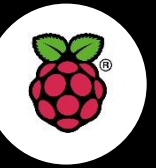
### 측정 위치

- Zone1 온도 오차범위는  $\pm 0.5^{\circ}\text{C}$
  - Zone2는 실제 온도보다  $8^{\circ}\text{C}$  낮게 측정
- > 얼굴 가운데 온도 측정



# 부품 리스트 (3) 열화상 카메라

Enjoy your stylish business and campus life with BIZCAM



## SW

### 1) 이미지를 표현하기 위해 필요한 라이브러리 설치

```
sudo pip3 install matplotlib scipy numpy
```

### 2) I2C tools 설치

```
sudo apt-get install -y python-smbus
```

```
sudo apt-get install -y i2c-tools
```

### 3) I2C 작동하는지 확인

```
sudo nano /boot/config.txt
```

### 4) 재부팅

```
sudo reboot
```

### 5) 라즈베리파이가 MLX90640 센서를 제대로 등록했는지 확인

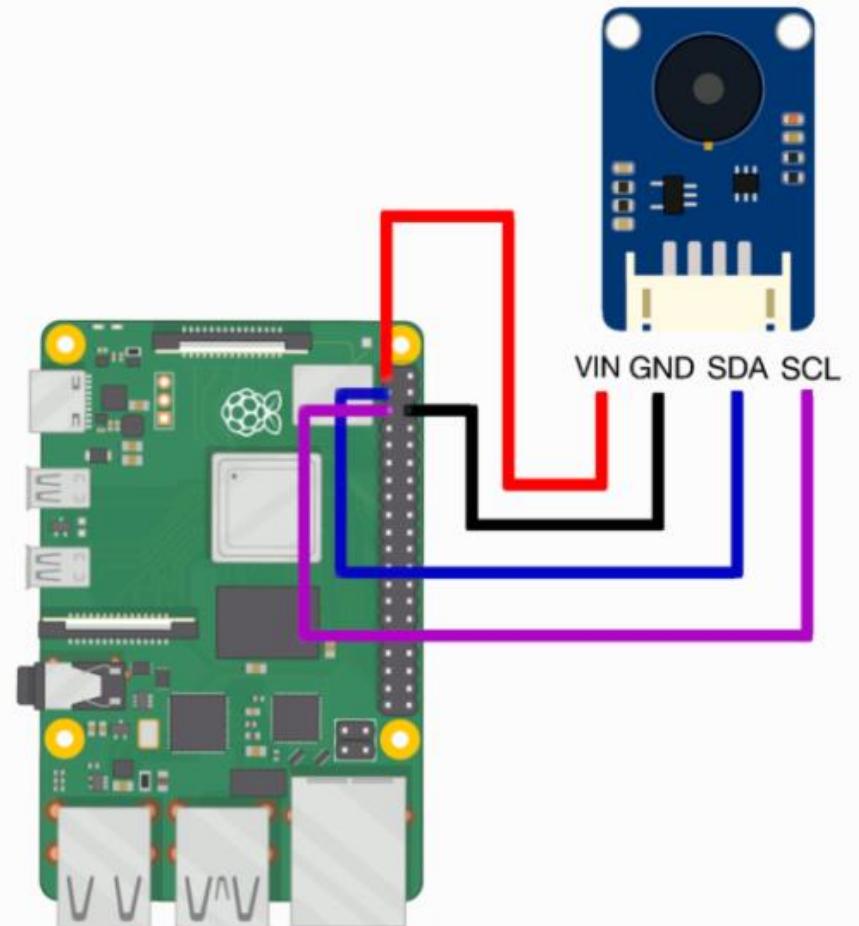
```
sudo i2cdetect -y 1
```

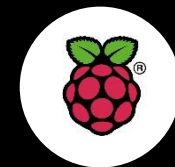
### Adafruit 라이브러리 설치

```
Sudo pip3 install RPI.GPIO adafruit-blinka
```

```
Sudo pip3 install adafruit-circuitpython-mlx90640
```

## HW





Python 명령어 학습  
데이터 분석 학습  
데이터 전처리  
Heatmap 그래프

PYTHON

라즈베리  
파이

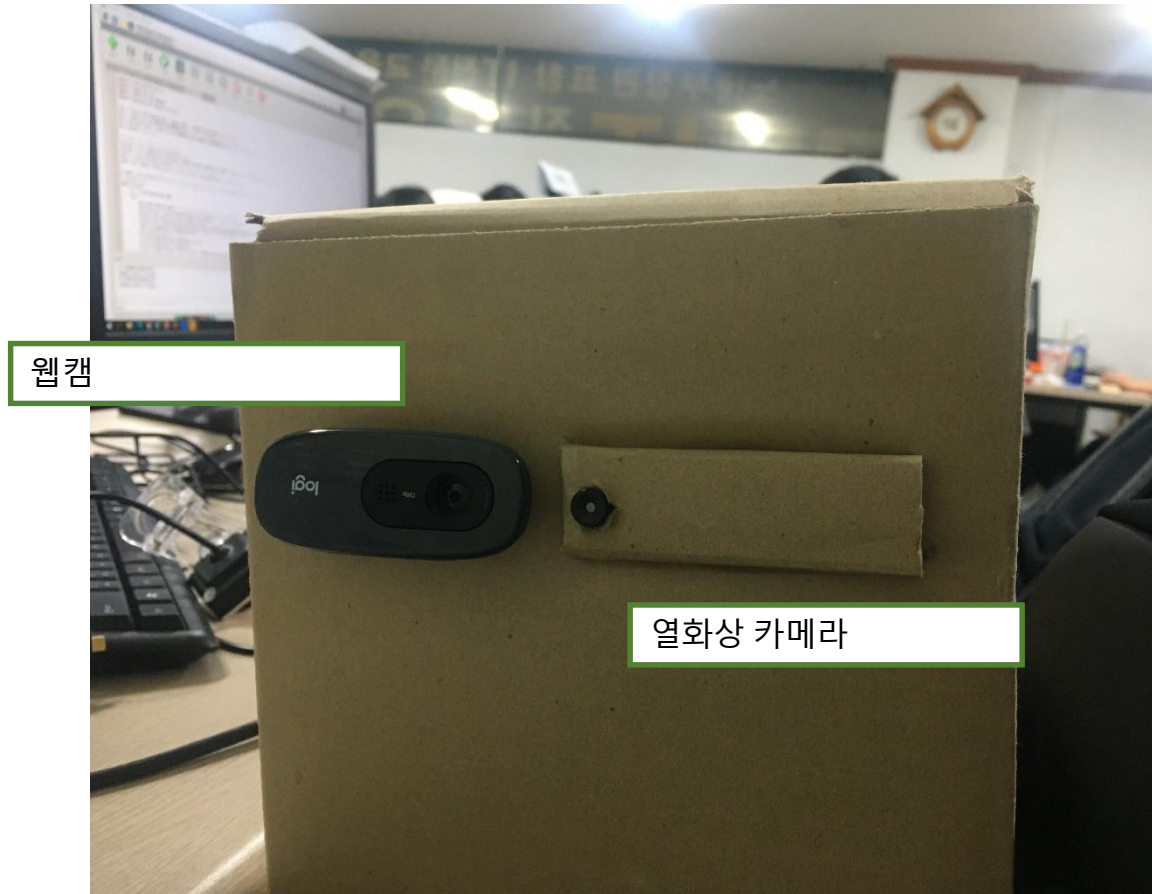
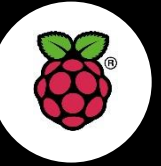
개발 환경 구현  
웹캠 및 열화상 카메라 연결

이미지 프로세싱  
딥러닝 객체 인식 기술  
안면 인식 기술

OpenCV

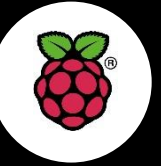
QT design

출력 화면 구현

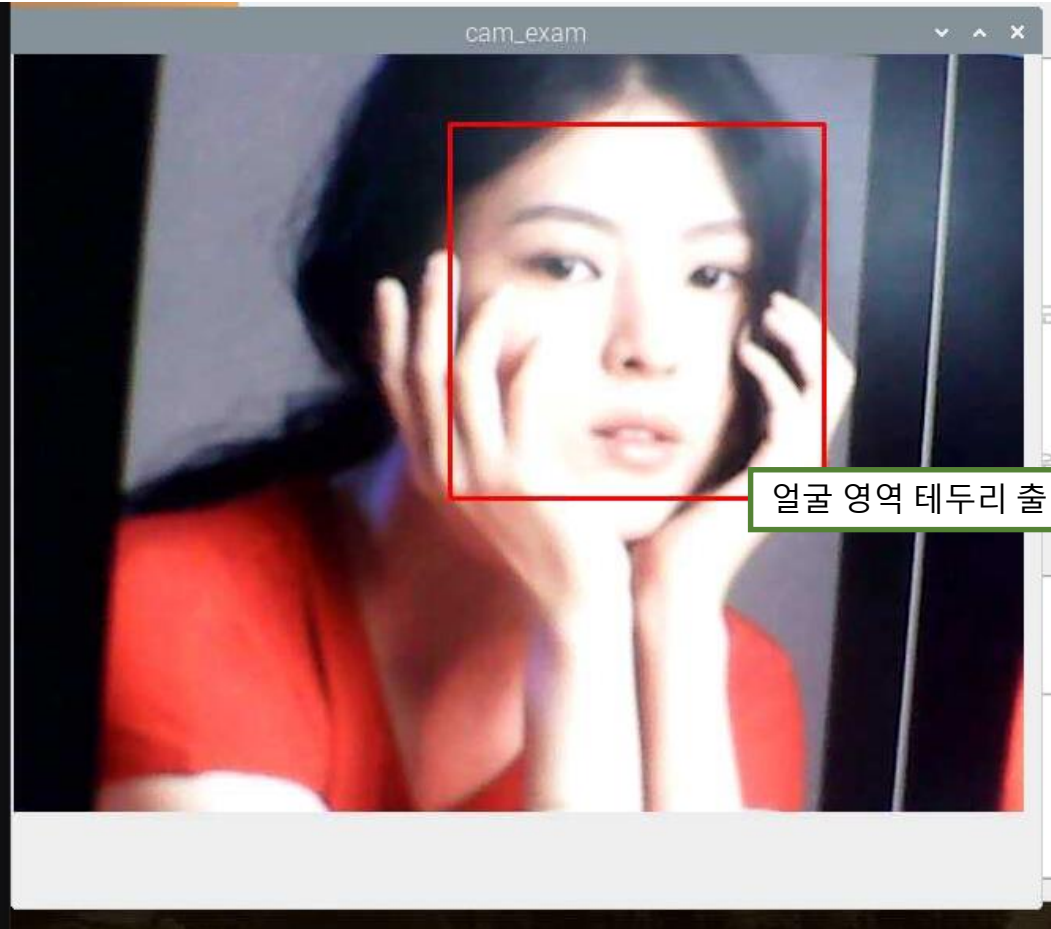


# 얼굴인식 코드 예시

Enjoy your stylish business and campus life with BIZCAM



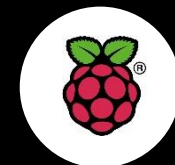
준비된 이미지



얼굴 영역 테두리 출력

# 얼굴인식 코드 설명

Enjoy your stylish business and campus life with BIZCAM



```
from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import sys, cv2, numpy

class Example(QWidget):
    # 생성자
    def __init__(self):
        super().__init__()
        self.setWindowTitle("cam_exam")
        self.setGeometry(150,150,650,540)
        self.iniUI()
        self.nextFrameSlot()

    # UI 정보
    def iniUI(self):
        self.cpt = cv2.VideoCapture(0)
        self.frame = QLabel(self)
        self.frame.resize(640,480)
        # 레이블에 맞도록 레이블의 이미지를 조정
        self.frame.setScaledContents(True)
        self.show()

        self.timer = QTimer()
        self.timer.timeout.connect(self.nextFrameSlot)
        self.timer.start(1000/24)
```

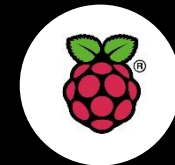
```
# 출력 관련 코드
def nextFrameSlot(self):
    success, cam = self.cpt.read()
    if success:
        cam = cv2.cvtColor(cam, cv2.COLOR_BGR2RGB)
        self.img_p = cv2.cvtColor(cam, cv2.COLOR_RGB2GRAY)

    # =====얼굴인식 코드 =====
    faceCascade = cv2.CascadeClassifier('gui/haarcascade_frontface.xml')
    faces = faceCascade.detectMultiScale(
        self.img_p, scaleFactor=1.2, minNeighbors=5, minSize=(20, 20)
    )
    for (x,y,w,h) in faces:
        cv2.rectangle(cam,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = self.img_p[y:y+h, x:x+w]
        roi_color = cam[y:y+h, x:x+w]
    # =====
    img = QImage(cam, cam.shape[1], cam.shape[0], QImage.Format_RGB888)
    pix = QPixmap.fromImage(img)
    self.frame.setPixmap(pix)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    ex = Example()
    sys.exit(app.exec_())
```

# 얼굴인식 코드 설명 - Haar Cascade

Enjoy your stylish business and campus life with BIZCAM



## Haar Cascade

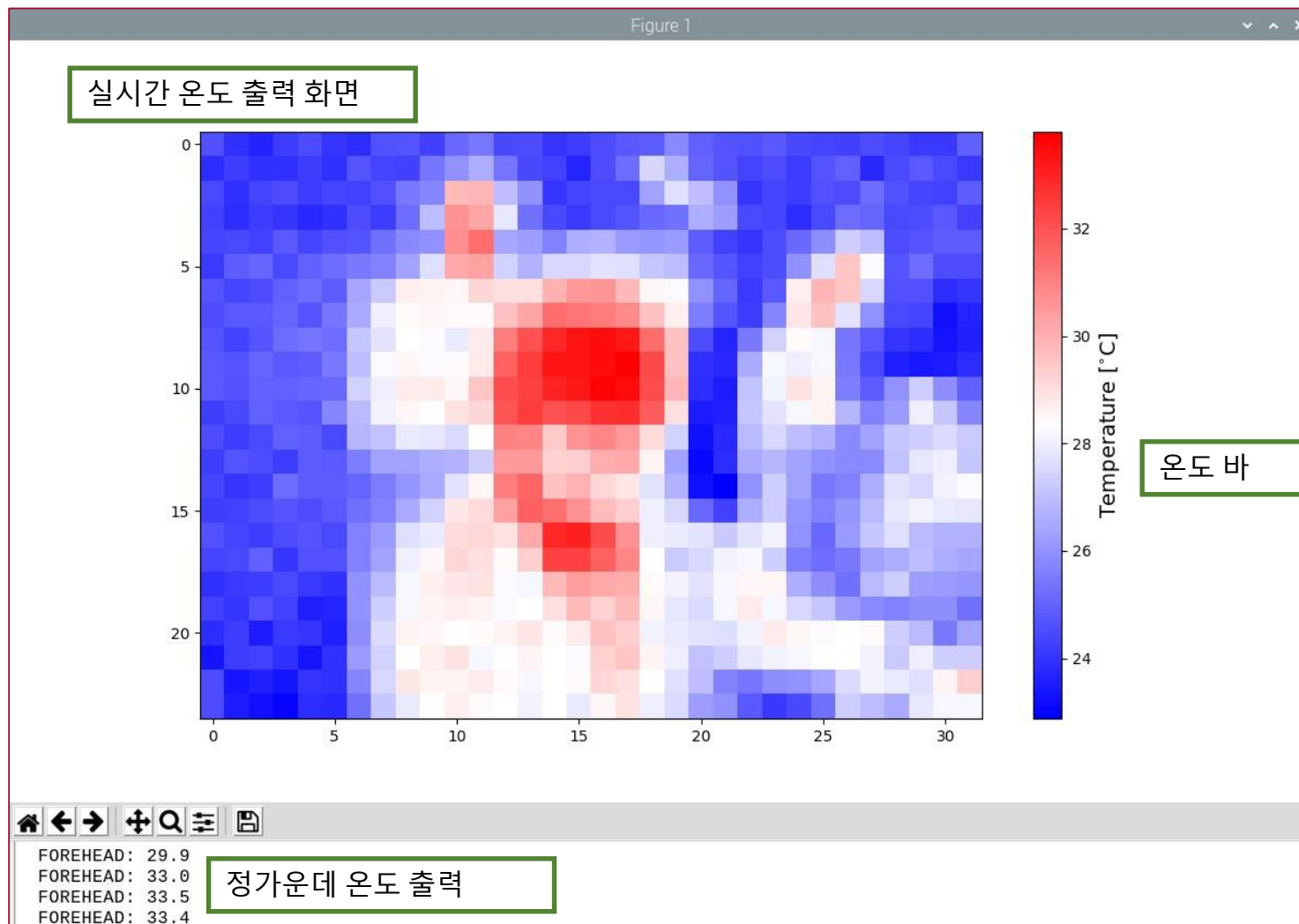
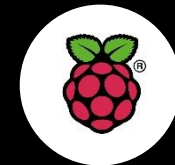
머신 러닝기반의 오브젝트 검출 알고리즘  
특징을 기반으로 비디오 또는 이미지에서 오브젝트를 검출하기 위해 사용  
직사각형 영역으로 구성되는 특징을 사용하기 때문에 픽셀을 직접 사용할 때 보다 동작 속도가 빠름  
얼굴을 검출하기 위해 미리 학습시켜 놓은 XML 포맷으로 저장된 분류기인  
'haarcascade\_frontalface.xml' 를 다운로드 해서 사용

```
import numpy as np
import cv2 as cv

# 얼굴과 눈을 검출하기 위해 미리 학습시켜 놓은 XML 포맷으로 저장된 분류기를 로드.
face_cascade = cv.CascadeClassifier('haarcascade_frontalface_default.xml')
# 얼굴과 눈을 검출할 그레이스케일 이미지를 준비.
img = cv.imread('sachin.jpg')
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
# 이미지에서 얼굴을 검출합니다.
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
# 얼굴이 검출되었다면 얼굴 위치에 대한 좌표 정보를 리턴 받음
for (x,y,w,h) in faces:
    # 원본 이미지에 얼굴의 위치를 표시
    cv.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
# 얼굴과 눈 검출 결과를 화면에 보여줍니다.
cv.imshow('img',img)
cv.waitKey(0)
cv.destroyAllWindows()
```

# 열화상 카메라 실습 capture

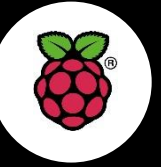
Enjoy your stylish business and campus life with BIZCAM





# 열화상 카메라 세부 코드

Enjoy your stylish business and campus life with BIZCAM



```
import time, board, busio
import numpy as np
import pandas as pd
import adafruit_mlx90640
import matplotlib.pyplot as plt
```

# 열화상카메라에서 데이터를 읽어옴

```
i2c = busio.I2C(board.SCL, board.SDA, frequency=400000)
mlx = adafruit_mlx90640.MLX90640(i2c)
mlx.refresh_rate = adafruit_mlx90640.RefreshRate.REFRESH_4_HZ
mlx_shape = (24,32)
```

# figure 생성 및 설정

```
fig, ax = plt.subplots(figsize=(12,7))
therm1 = ax.imshow(np.zeros(mlx_shape), vmin=0, vmax=60, cmap = 'bwr')
```

# Colorbar 생성

```
cbar = fig.colorbar(therm1)
cbar.set_label('Temperature [°C]', fontsize=14)
```

# frame 크기 설정

```
frame = np.zeros((24*32,))
t_array = []
```

# heatmap 반복 그리기

```
while True:
```

```
    t1 = time.monotonic()
```

```
    try:
```

```
        mlx.getFrame(frame) # read MLX temperatures into frame var
        data_array = (np.reshape(frame, mlx_shape)) # reshape to 24x32
```

```
        therm1.set_data(data_array)
```

```
        therm1.set_clim(vmin=np.min(data_array), vmax=np.max(data_array))
```

```
        cbar.on_mappable_changed(therm1) # colorbar 업데이트
```

```
        plt.pause(0.001) # 화면 출력을 위한 시간지연
```

```
        t_array.append(time.monotonic()-t1)
```

```
        print("FOREHEAD: {}".format(frame[272].round(1)))
```

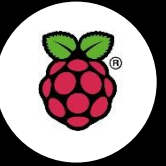
```
        # 이마부분 온도 출력
```

```
    except ValueError:
```

```
        continue # if error, just read again
```

# 중간 평가

Enjoy your stylish business and campus life with BIZCAM



## □ QMainWindow

웹캠

열화상 카메라

☒ 얼굴 인식

☒ 온도 측정

☒ 얼굴 좌표 출력

☒ 얼굴 center 온도 출력

☐ 인식 박스 위에 온도 출력

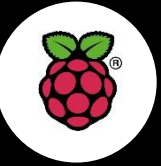
☐ 체온 범위 판독 및 출력

56%

- 웹캠과 열화상 카메라 정보 동기화
- 체온 정보 영상 화면에 출력
- 체온 판독 추가

# 추가 학습 사항

Enjoy your stylish business and campus life with BIZCAM



QT  
design

웹캠과 열화상 카메라의  
수평 레이아웃 설정,  
하단부 정보 출력

초음파  
센서  
(HC\_SR04)

사람과 체온 측정 단말기  
사이의 거리 값 측정

OpenCV  
Python  
추가 학습

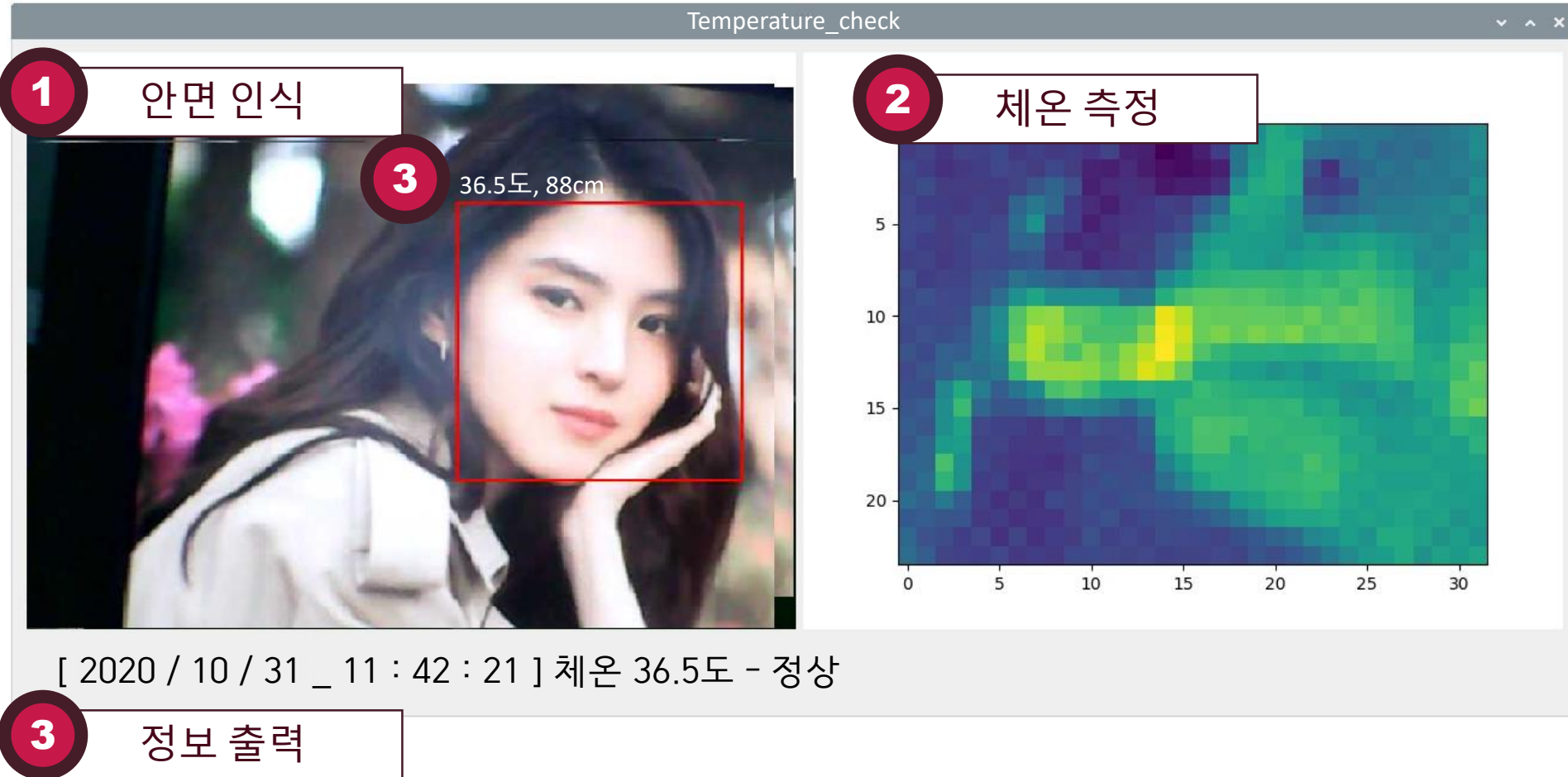
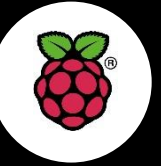
열화상 카메라와 웹캠 사이의  
정보 동기화  
웹캠 화면 내 체온, 거리 정보 출력

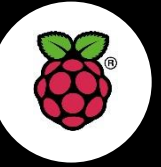
안드로  
이드

화면 및 온도 정보를  
앱을 통해 출력

# 최종 완성본 구상도

Enjoy your stylish business and campus life with BIZCAM





감사합니다