

지능형 거울

Smart Mirror

2018265103 이형주

2018210143 박효열

Contents

작품 동기 & 목표

01

지능형 거울 이란?

02

설계 구성도

03

설계 일정표

04

팀원 역할 및 진행 사항

05

01

「작품 동기 & 목표」

지능형 거울은 GUI에 날씨, 기온, 일정(구글 캘린더), 핸드폰 알림 출력, 스크린캐스트(화면 미러링) 등이 함께 출력되어 현대인들이 짧은 시간에 유용한 정보를 얻을 수 있도록 하는 IOT와 거울을 접목시킨 지능형 거울이다. 우리는 기존의 지능형 거울에서 얼굴 인식과 음성인식을 통해 사용자를 인식하는 시스템을 더한 지능형 거울을 설계하였다.

13:29

Wednesday the 22/07

73° (78° | 54°)

1 minute

No events remaining today

02

「지능형거울 이란?」

하프미러와 디스플레이를 결합한
거울이면서도 디스플레이의 기능을 갖고 있는 거울

03

설계 구성도

- 하프미러
- 화면 출력용 모니터
- 라즈베리파이 3B+
- 마이크 & 스피커
- 카메라

하프미러

하프미러는 일반적인 반사만 존재하는 거울과는 달리 반사율과 투과율을 조절한 거울입니다.



하프미러 35 (1524mm X 1M)

「 출력용 모니터 」



모니터(j22wdg)

프로그램을 출력하게될
22인치 크기의 모니터

「라즈베리파이」

라즈베리파이3b+와
기본키트 (케이블, sd카트 등)



라즈베리파이 기본키트

「마이크,스피커」



음성인식 입력을 넣기위한 마이크
& 소리 출력을 위한 스피커

마이크,스피커

카메라

얼굴인식에서 사용하게될 카메라
(raspberry pi camera)



라즈베리파이용 카메라

04

설계 일정표

05

「팀원 역할 및 진행 사항」

이형주 - module 수정 & 얼굴인식 python 설계

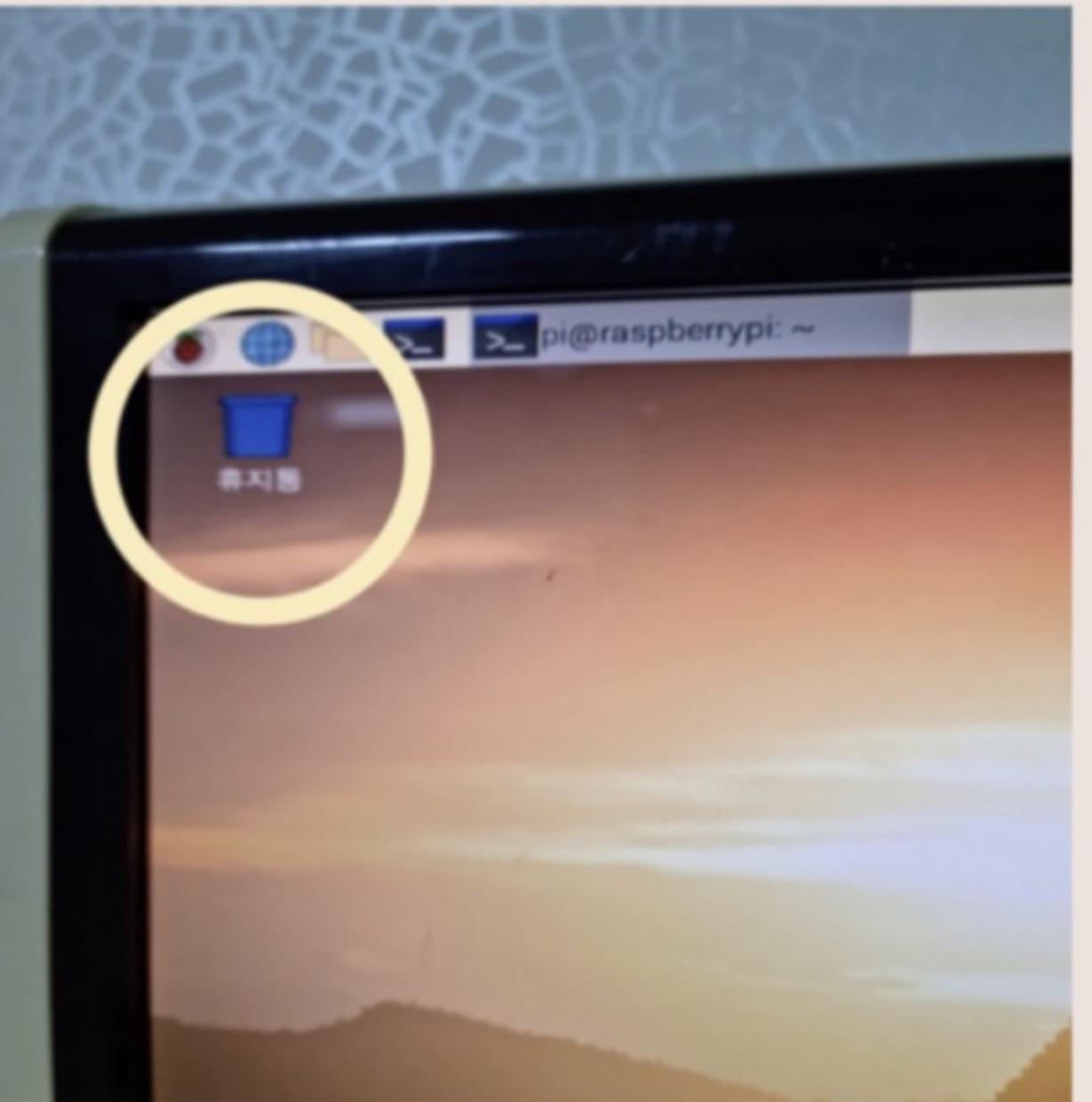
박효열 - API 사용코드 수정& 음성인식 python 설계

기본 설정

라즈베리파이와 지능형 거울의 기본 설정

진행사항

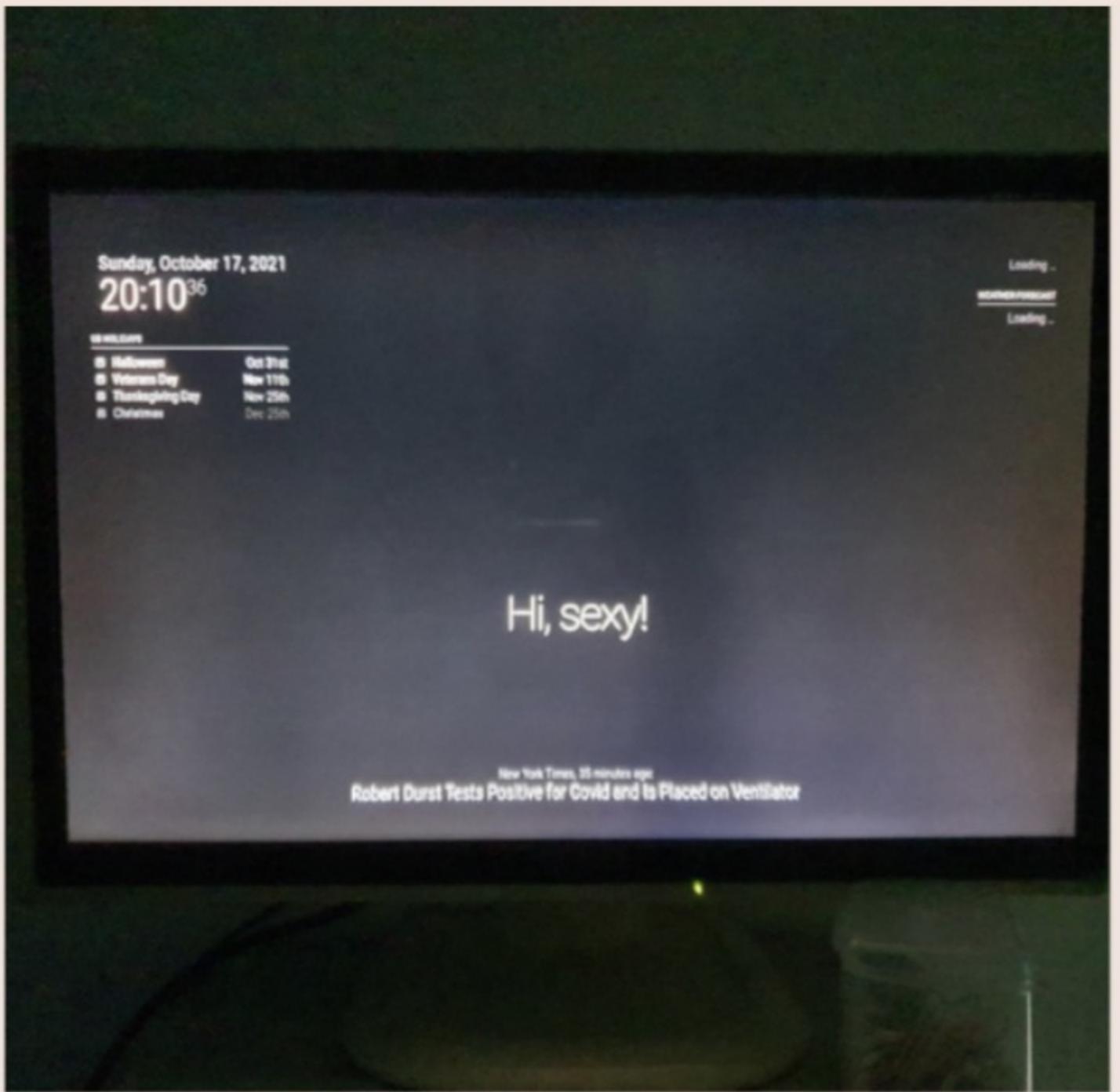
라즈베리파이를 모니터에 연결하여 기본설정을 하여 본격적으로 코딩을 할 수 있는 환경이 조성되었다.



라즈베리파이 출력

진행사항

기존에 있던 오픈소스
MagicMirror2의 코드 중에서
날씨 모듈을 직접 수정하여 한글
이 출력되도록 하였다.



기존의 오픈소스 출력화면

진행사항

기존에 있던 오픈소스
MagicMirror2의 코드 중에서
날씨 모듈을 직접 수정하여 한글
이 출력되도록 하였다.

```
module: "compliments",
position: "lower_third",
config: {
    compliments: {
        anytime: ["Hi, sexy!"]
    }
}
```

기존에 있던 오픈소스 (날씨 모듈)

진행사항

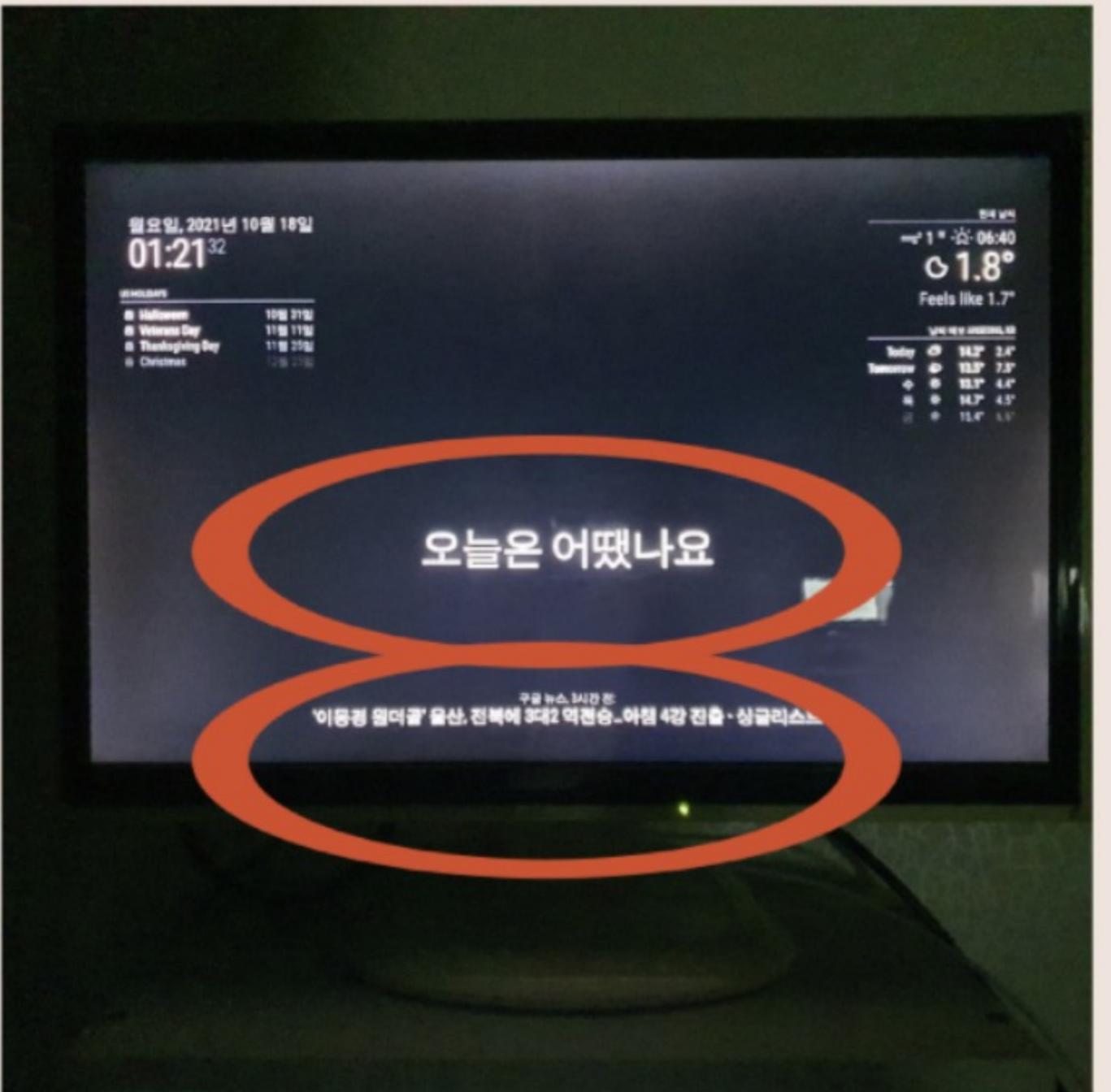
기존에 있던 오픈소스
MagicMirror2의 코드 중에서
날씨 모듈을 직접 수정하여 한글
이 출력되도록 하였다.

```
module: "compliments",
position: "lower_third",
config: {
    compliments: {
        anytime: ["멋져 보입니다",
                  "오늘 하루 화이팅"]
    }
}
```

오픈소스 수정(날씨 모듈)

진행사항

같은원리로 기존에 있던 오픈소스 MagicMirror2의 코드 중에서 뉴스 모듈과 사용자에게 인사말을 건네는 부분 또한 수정해보았다.



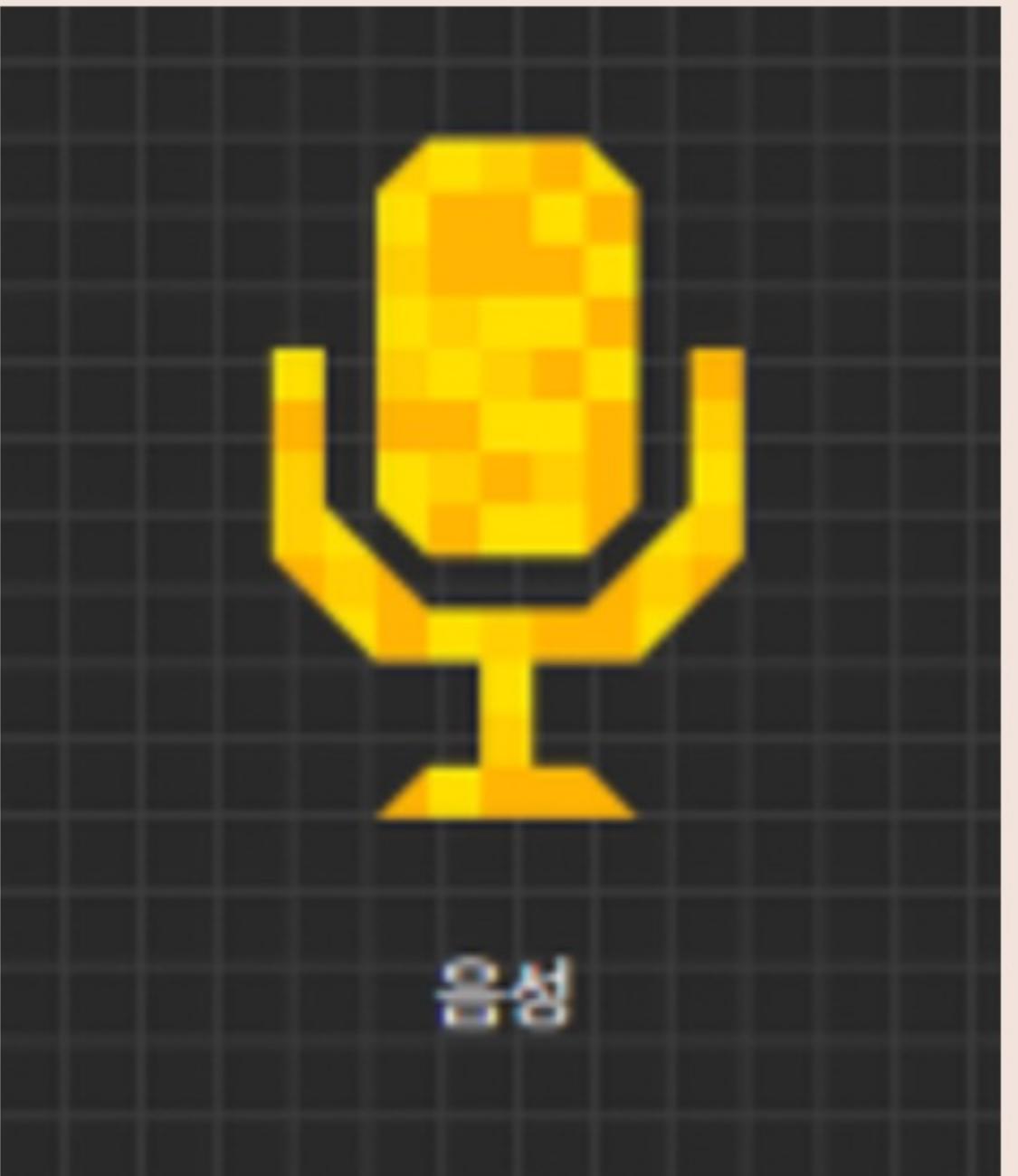
오픈소스 수정 후 출력화면

음성 인식 (Sound To Text)

카카오API를 활용하여 소리를 문서화하고
특정 키워드에 동작하는 조건문을 활용하여 임무 수행

API

API는 컴퓨터와 컴퓨터 프로그램 사이를 연결하는 것 말합니다.
그래서 API를 통해 구글이나 카카오 등의 회사에서 무료로 제공하는
프로그램을 우리가 사용할 수 있습니다.



카카오API

```
import requests, json

url = "https://kakaoi-newtone-openapi.kakao.com/v1/recognize"
key = '793b6c9257acd09a4d5c279601b21f40'
headers = {
    "Content-Type": "application/octet-stream",
    "X-DSS-Service": "DICTATION",
    "Authorization": "KakaoAK " + key,
}
with open('heykakao.wav', 'rb') as fp:
    audio = fp.read()

res = requests.post(url, headers=headers, data=audio)
print(res.text)

result_json_string = res.text[res.text.index('{"type":"finalResult"}'):res.text.rindex('}')+1]
result = json.loads(result_json_string)
print(result)
print(result['value'])
```

카카오API

진행사항

카카오API를 활용하여 저장된 음성파일을 텍스트로 전환하는 코드이다.

```
import speech_recognition as sr
```

- sr : sampling rate (주파수 분석 및 파형의 시간 간격을 결정)

```
def get_speech():
    #마이크에서 음성을 추출하는 객체
    recognizer = sr.Recognizer()
        #마이크 설정
    microphone = sr.Microphone(sample_rate=16000)
        #마이크 소음 수치 반영 _ 보류
    with microphone as source:
        recognizer.adjust_for_ambient_noise(source)
    #    print("소음 수치를 반영하여 음성을 청취합니다.{}".format(recognizer.energy_threshold)) - 빼도 되는듯
        #음성수집
    with microphone as source:
        print("say something!")
        audio_data = recognizer.listen(source)
    audio = audio_data.get_raw_data()
```

```
KEYWORD_NEWS = "뉴스"
KEYWORD_COVID= "코로나"

if text in 웃슨:
    print("무엇을 도와드릴까요?")
    get_speech()
    if KEYWORD_NEWS in text:
        print("뉴스 프로젝트 실행")
        news_kakao.do()

elif KEYWORD_COVID in text:
    print("코로나 프로젝트 실행")
    covid_19.do()
```

news_kakao.py

```
# BeautifulSoup 객체 생성, url을 요청해서 beautifulsoup 객체를 생성하는 함수, url을 input으로 받
def get_soup_obj(url):
    headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
    res = requests.get(url, headers = headers)
    soup = BeautifulSoup(res.text, 'lxml')

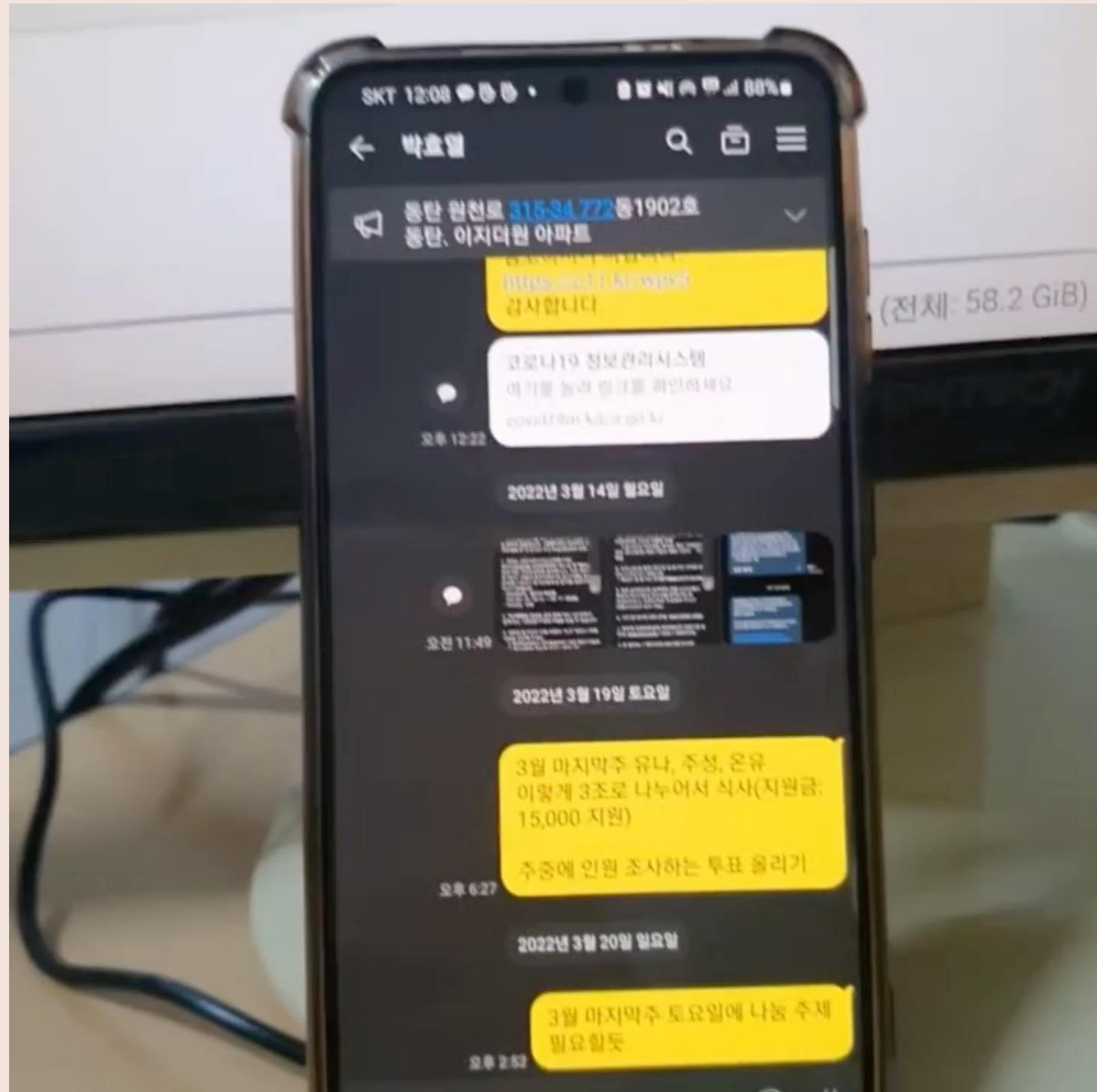
    return soup
```

news_kakao.py

```
# 각 리스트에 들어갈 내용(content) 만들기
for news_info in news_list3:
    content = {
        "title" : news_info.get('title'),
        "description" : "작성일 : " + news_info.get('date'),
        "image_url" : news_info.get('image_url'),
        "image_width" : 50, "image_height" : 50,
        "link": {
            "web_url": news_info.get('news_url'),
            "mobile_web_url": news_info.get('news_url')
        }
    }

    contents.append(content)

data = {
    'template_object':json.dumps(template)
}
```



covid19_kakao.py

```
#data를 요청
resp = requests.get(service_url, params=params)
#요청된 data를 dictionary 형태로 저장
parsed_content = xmltodict.parse(resp.content)
json_string=json.loads(json.dumps(parsed_content))
    #daily_cnt: 일별 확진자수, stateDt: 기준일, decideCnt: 누적 확진자수 -> api에서 오는 정보는 list형태이기 때문
daily_cnt = []
stateDt = []
decideCnt = []
    #response 안에 body 안에 items 안에 item
items = json_string['response']['body']['items']['item']
    #items를 추출
for i in items:
    stateDt.append(i['stateDt'][4:]) #연도는 제외하고 월,일 만 가져와서 stateDt(리스트)에 추가
    decideCnt.append(int(i['decideCnt']))

#일일 확진자 수 구하기 -> 오늘 확진자 수를 알 수 없어서 1일 빼서 제외시키는 거임, #len 문자열 길이 구하기 함수
for i in range(len(stateDt)-1):
    daily_cnt.append(decideCnt[i]-decideCnt[i+1])
print ("오늘 확진자 수는 %d명 이고 누적 확진자 수는 %d명입니다." %(daily_cnt[0],decideCnt[0]))
```

응답메시지

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <header>
        <resultCode>00</resultCode>
        <resultMsg>NORMAL SERVICE.</resultMsg>
    </header>
    <body>
        <items>
            <item>
                <accDefRate>3.2396602365</accDefRate>
                <accExamCnt>268212</accExamCnt>
                <accExamCompCnt>251940</accExamCompCnt>
                <careCnt>7253</careCnt>
                <clearCnt>834</clearCnt>
                <createDt>2020-03-15 10:01:22.000</createDt>
                <deathCnt>75</deathCnt>
                <decideCnt>8162</decideCnt>
                <examCnt>16272</examCnt>
                <resutlNegCnt>243778</resutlNegCnt>
                <seq>74</seq>
                <stateDt>20200315</stateDt>
                <stateTime>00:00</stateTime>
                <updateDt />
            </item>
            <item>
                <accDefRate>3.3180003364</accDefRate>
                <accExamCnt>261335</accExamCnt>
                <accExamCompCnt>243701</accExamCompCnt>
                <careCnt>7300</careCnt>
                <clearCnt>714</clearCnt>
                <createDt>2020-03-14 10:29:11.000</createDt>
            </item>
        </items>
    </body>
</response>
```

```
print ("오늘 확진자 수는 %d명이고 누적 확진자 수는 %d명입니다." %(daily_cnt[5],decideCnt[5]))
```

```
python-2022.2.1924087327\pythonFiles\lib\python\debugpy\launcher  
Users\Hyo-Yeol\.vscode\extensions\kakao_send_1\covid_19.py "  
오늘 확진자 수는 187213명이고 누적 확진자 수는 12003054명입니다.
```

```
28     "service": "DICTATION",
29     "Authorization": "KakaoAK " + "793b6c9257acd09a4d5c27366
30
31     res = requests.post(kakao_speech_url, headers=headers, data=
32     #print(res.text) -> 없어도 된다
33     result = res.text[res.text.index('{"type": "finalResult"}'):res
34     global text
35     text = json.loads(result).get('value')
36     print(text)
37
38     #추가 프로젝트 -> 날씨?
39     KEYWORD_NEWS = "뉴스"
40     KEYWORD_COVID= "코로나"
41
문제    출력    디버그 콘솔    터미널
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hyo-Yeol.vscode\extensions\kakao_send_1> cmd /C ""C:\Program Files\Python\Files\lib\python\debugpy\launcher 68425 -- c:\Users\Hyo-Yeol.vscode\extensions\kakao_send_1> c: && cd c:\Users\Hyo-Yeol.vscode\extensions\ms-python.python-2022.2.1924887327\pythonFiles\lib\python\idle
C:\Users\Hyo-Yeol.vscode\extensions\kakao_send_1> c: && cd c:\Users\Hyo-Yeol.vscode\extensions\ms-python.python-2022.2.1924887327\pythonFiles\lib\python\idle
```

얼굴 인식 (Face recognition)

Haar cascade 기법과 LBPH 알고리즘을 활용한 얼굴 인식

목표

카메라와 openCV를 활용하여
스마트미러를 활용하여 사람을 인
식하는 것을 넘어 사용자를 인식
하게 할 것이다.



라즈베리파이용 카메라

진행사항

카메라와 opencv를 활용하여 스마트미러를 활용하여 사람을 인식하는 것을 넘어 사용자를 인식하게 할 것이고, 이후 카메라를 활용한 추가적인 장점을 기대하고 있다.

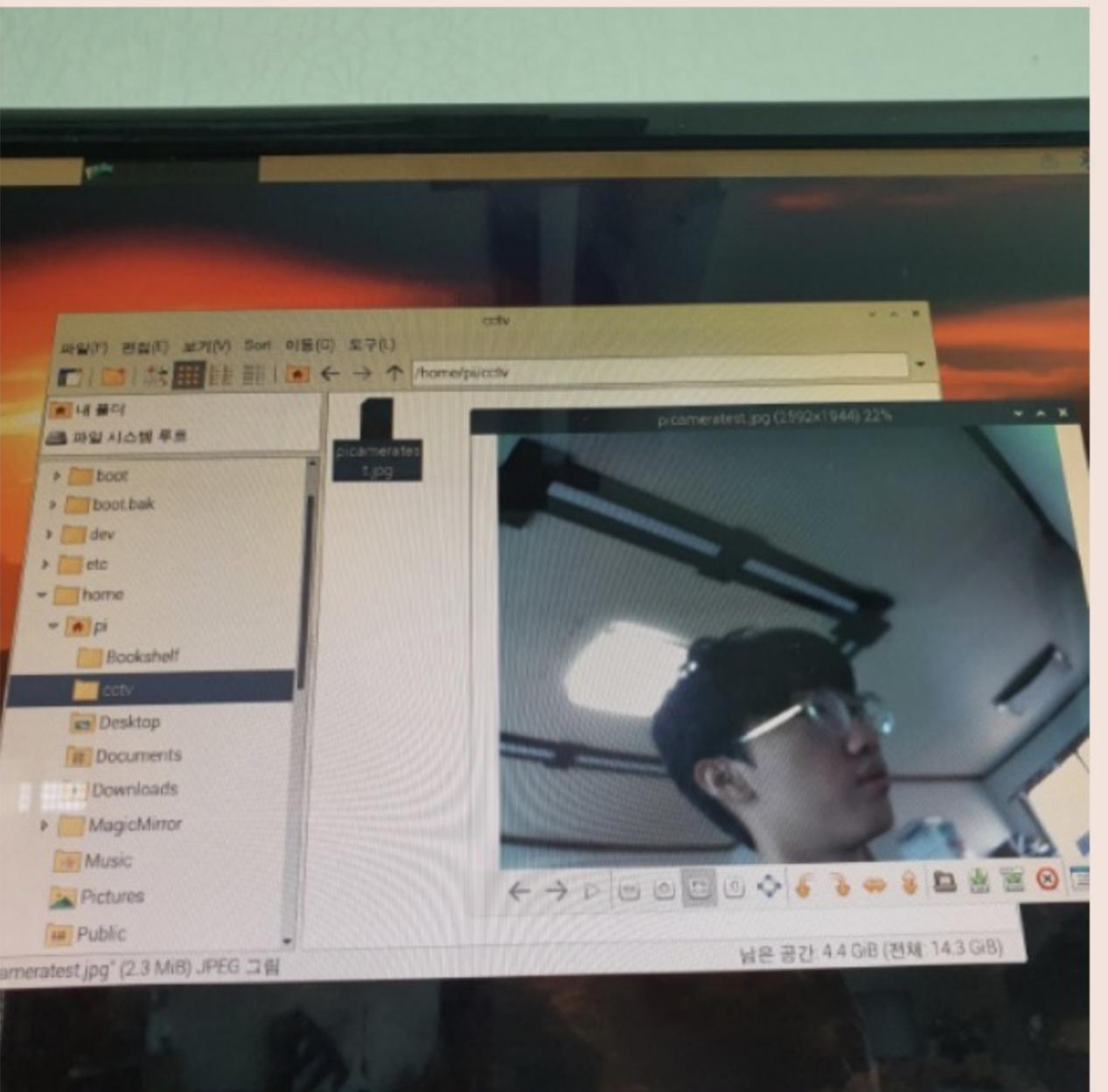


카메라

」

진행사항

리눅스 명령어를 통해 라즈베리파이에 카메라를 설치하였고 출력은 다음과 같았다.



라즈베리파이 카메라로 찍은 출력화면

진행사항

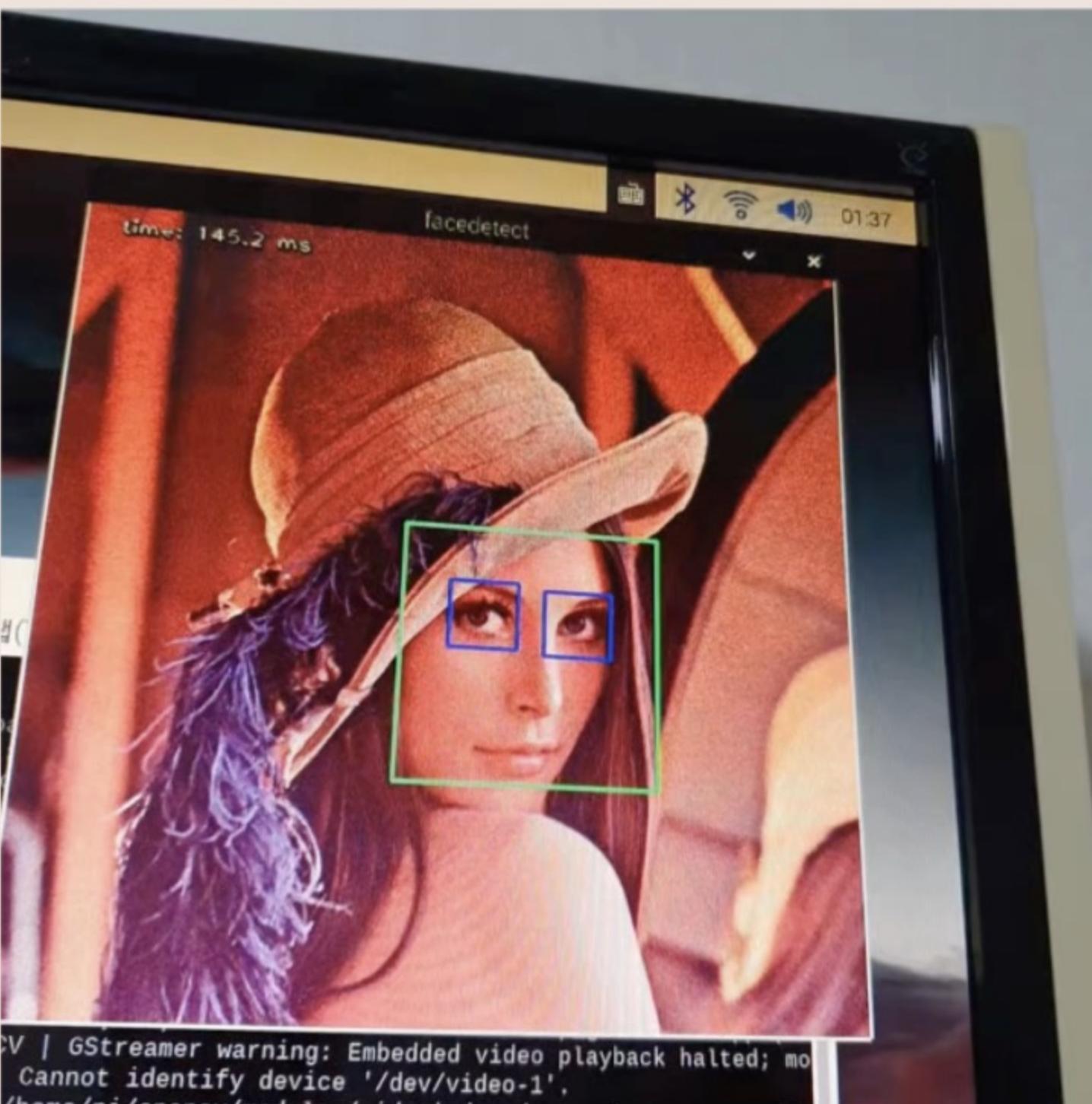
```
pi@raspberrypi:~ $ cd opencv  
pi@raspberrypi:~/opencv $ cd build  
pi@raspberrypi:~/opencv/build $ python3  
Python 3.9.2 (default, Mar 12 2021, 04:06:34)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import cv2  
>>> cv2.__version__  
'4.5.5'
```

openCV 4.5.4 / python3.9.2
를 성공적으로 설치하였다

opencv가 정상적으로 설치된 모습

진행사항

sample파일로 확인해보았을 때
얼굴을 인식하고 눈과같은 특징점
을 잡아냄



opencv가 정상적으로 설치된 모습

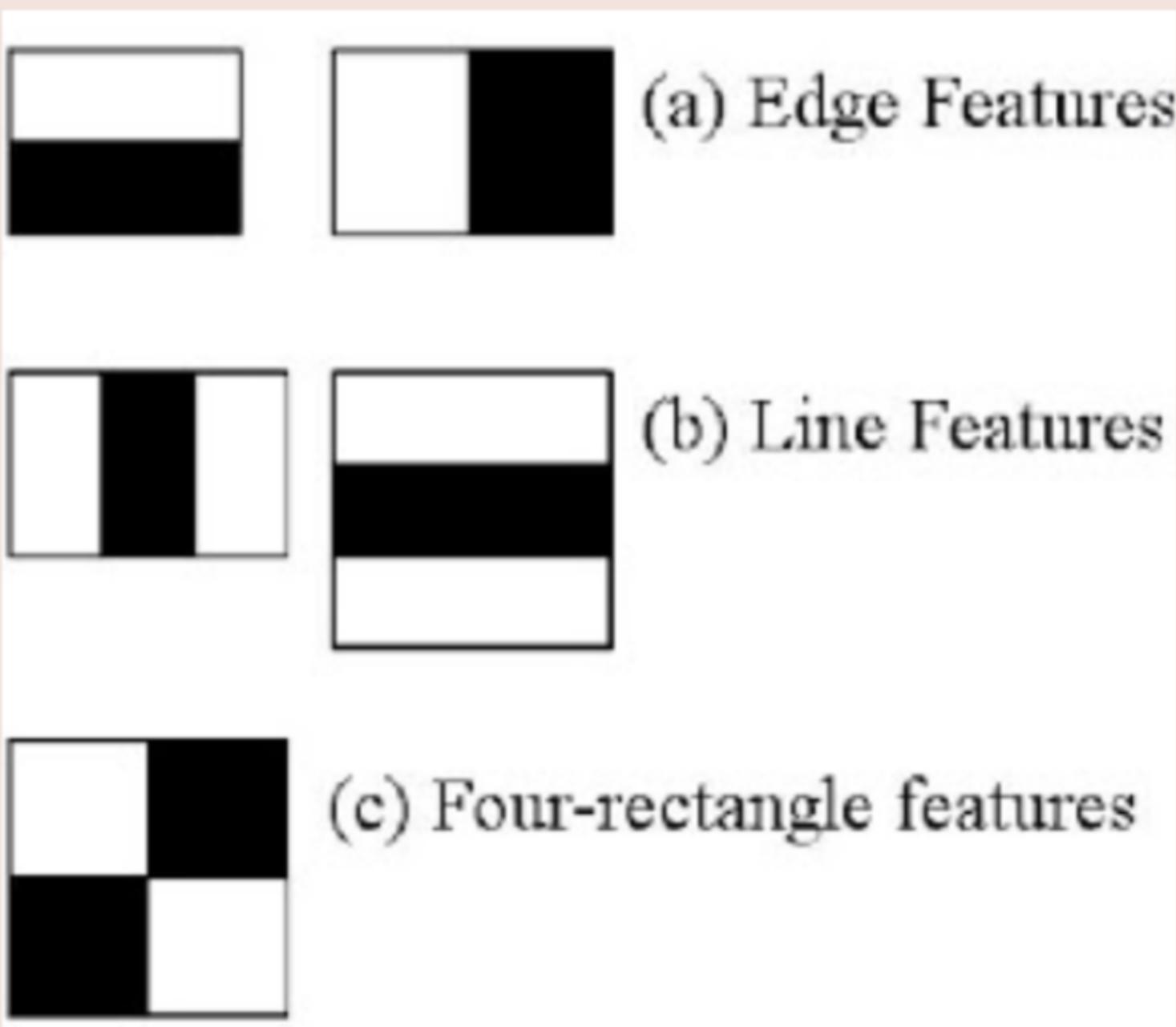
```
import cv2
import numpy as np
# 얼굴 인식용 xml 파일
face_classifier = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# 전체 사진에서 얼굴 부위만 잘라 리턴
def face_extractor(img):
    # 흑백처리
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # 얼굴 찾기
    faces = face_classifier.detectMultiScale(gray, 1.3, 5)
    # 찾은 얼굴이 없으면 None으로 리턴
    if faces is ():
        return None
    # 얼굴들이 있으면
    for(x, y, w, h) in faces:
        # 해당 얼굴 크기만큼 cropped_face에 잘라 넣기
        cropped_face = img[y:y+h, x:x+w]
    # cropped_face 리턴
    return cropped_face
```

```
# 얼굴 인식용 xml 파일  
face_classifier = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

캐스케이드 분류기

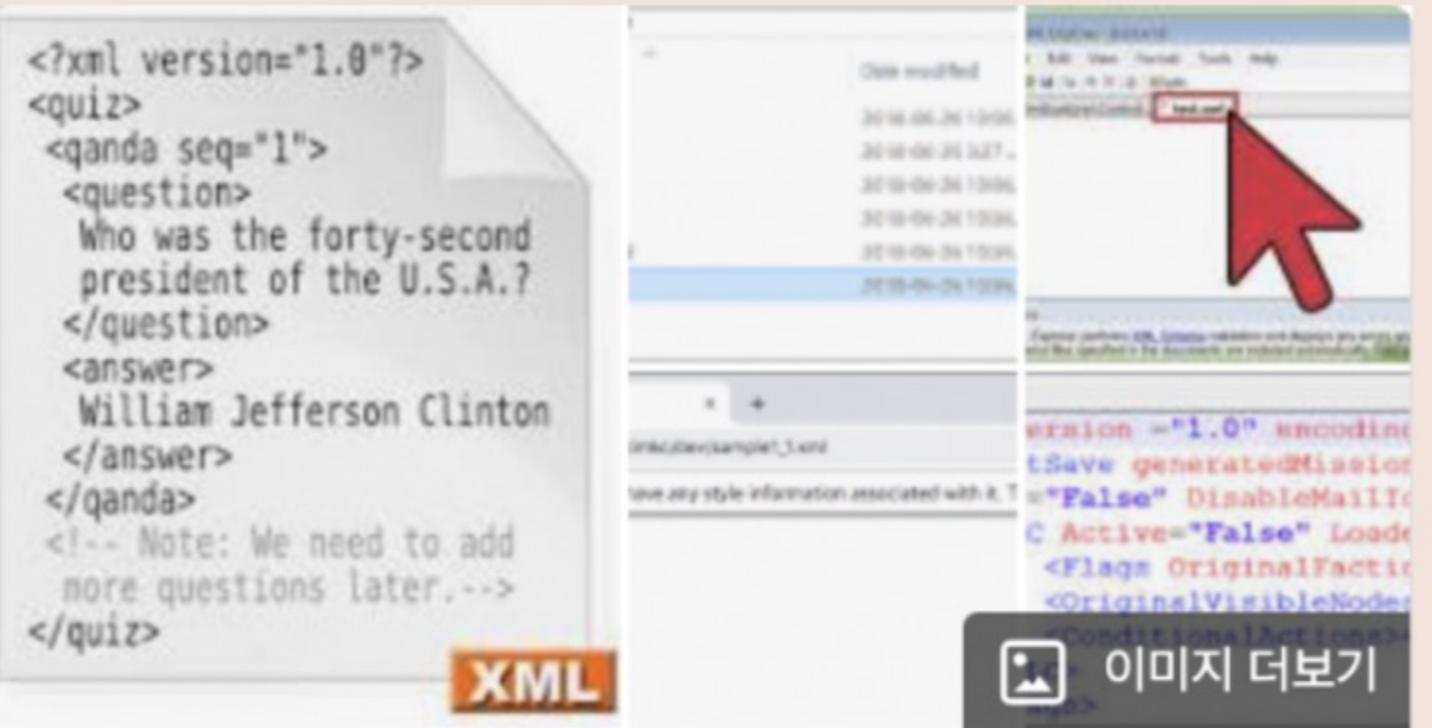
Haar-features



헝가리 수학자 Alfred Haar에
의해 1909년에 발표된 개념
얼굴인식 특징점을 빠르게 추출할
수 있는 특징추출 기법

XML

HTML과 같은 마크업 언어로서
정보를 담고 있는 오브젝트의 개념
으로 이해하면 좋다.



XML

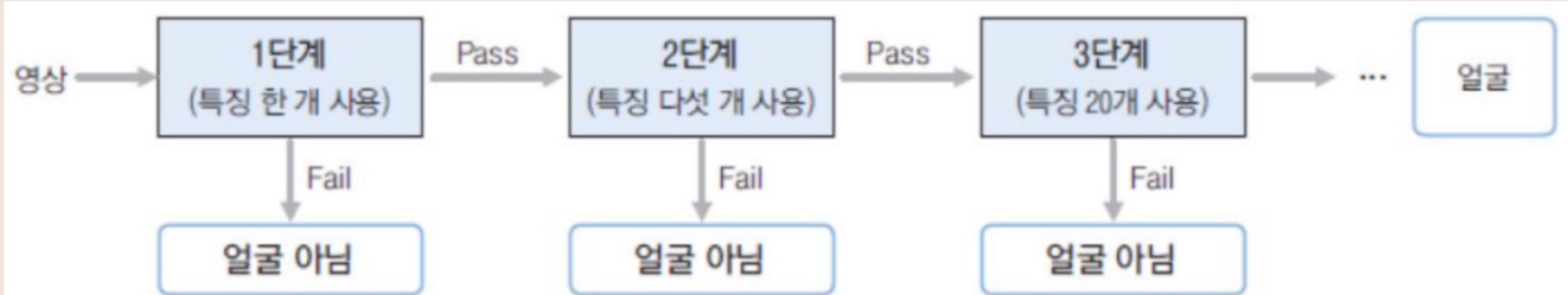
프로그램 언어

XML은 W3C에서 개발된, 다른 특수한 목적을 갖는 마크업 언어를 만드는데 사용하도록 권장하는 다목적 마크업 언어이다.

XML은 SGML의 단순화된 부분집합으로, 다른 많은 종류의 데이터를 기술하는 데 사용할 수 있다. [위키팩](#)과

```
# 전체 사진에서 얼굴 부위만 잘라 리턴
def face_extractor(img):
    # 흑백처리
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # 얼굴 찾기
    faces = face_classifier.detectMultiScale(gray, 1.3, 5)
    # 찾은 얼굴이 없으면 None으로 리턴
    if faces is ():
        return None
    # 얼굴들이 있으면
    for(x, y, w, h) in faces:
        # 해당 얼굴 크기만큼 cropped_face에 잘라 넣기
        cropped_face = img[y:y+h, x:x+w]
    # cropped_face 리턴
    return cropped_face
```

detectMultiScale



캐스케이드 분류기



Feature 1/16
Stage 2/4 (18%)

「학습데이터 쌓기」

```
# 카메라 실행
cap = cv2.VideoCapture(0)
# 저장할 이미지 카운트 변수
count = 0
while True:
    # 카메라로 부터 사진 1장 얻기
    ret, frame = cap.read()
    # 얼굴 감지 하여 얼굴만 가져오기
    if face_extractor(frame) is not None:
        count += 1
        # 얼굴 이미지 크기를 200x200으로 조정
        face = cv2.resize(face_extractor(frame), (200, 200))
        # 조정된 이미지를 흑백으로 변환
        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
        # faces폴더에 jpg파일로 저장
        # ex > faces/user0.jpg   faces/user1.jpg ....
        file_name_path = 'faces/user'+str(count)+'.jpg'
        cv2.imwrite(file_name_path, face)

        # 화면에 얼굴과 현재 저장 개수 표시
        cv2.putText(face, str(count), (50, 50),
                    cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow('Face Cropper', face)
    else:
        print("Face not Found")
        pass

    if cv2.waitKey(1) == 13 or count == 100:
        break

cap.release()
cv2.destroyAllWindows()
print('Collecting Samples Complete!!!!')
```

「학습데이터 학습」

```
import cv2
import numpy as np
from os import listdir
from os.path import isfile, join
data_path = 'faces/'

# faces 폴더에 있는 파일 리스트 얻기
onlyfiles = [f for f in listdir(data_path) if isfile(join(data_path, f))]
# 데이터와 매칭될 라벨 변수
Training_Data, Labels = [], []
# 파일 개수 만큼 루프
for i, files in enumerate(onlyfiles):
    image_path = data_path + onlyfiles[i]
    # 이미지 불러오기
    images = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    # 이미지 파일이 아니거나 못 읽어 왔다면 무시
    if images is None:
        continue
    # Training_Data 리스트에 이미지를 바이트 배열로 추가
    Training_Data.append(np.asarray(images, dtype=np.uint8))
    # Labels 리스트엔 카운트 번호 추가
    Labels.append(i)

# 훈련할 데이터가 없다면 종료.
if len(Labels) == 0:
    print("There is no data to train.")
    exit()

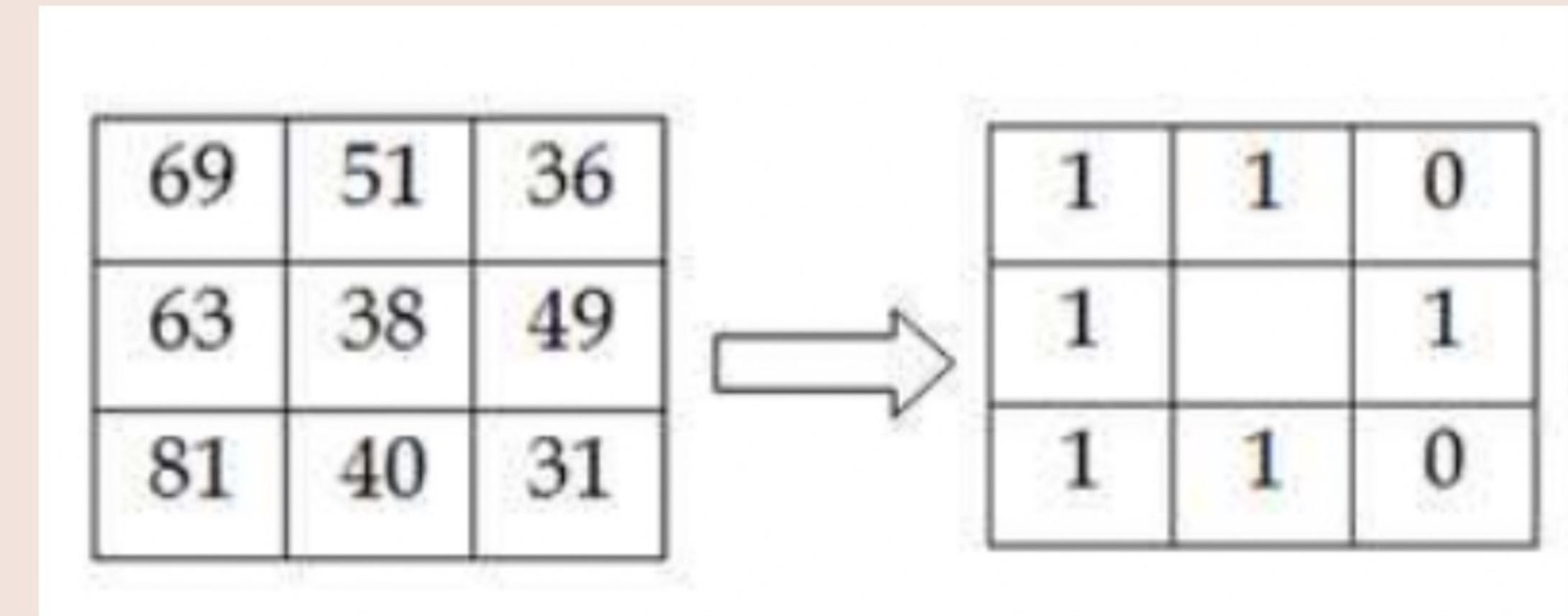
# Labels를 32비트 정수로 변환
Labels = np.asarray(Labels, dtype=np.int32)
# 모델 생성
model = cv2.face.LBPHFaceRecognizer_create()
# 학습 시작
model.train(np.asarray(Training_Data), np.asarray(Labels))
print("Model Training Complete!!!!")
```

LBPH

Local Binary Pattern (LBP)
은 각 픽셀의 이웃을 임계 값으로
지정하여 이미지의 픽셀에 레이블
을 지정하고 결과를 이진수로 간주
하는 텍스처 연산자이다.

```
# 모델 생성
model = cv2.face.LBPHFaceRecognizer_create()
# 학습 시작
model.train(np.asarray(Training_Data), np.asarray(Labels))
```

LBPH



지역이진패턴

「얼굴인식 시스템」

```
#카메라 열기
cap = cv2.VideoCapture(0)

while True:
    #카메라로 부터 사진 한장 읽기
    ret, frame = cap.read()
    # 얼굴 검출 시도
    image, face = face_detector(frame)
    try:
        #검출된 사진을 흑백으로 변환
        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
        #위에서 학습한 모델로 예측시도
        result = model.predict(face)
        #result[1]은 신뢰도이고 0에 가까울수록 자신과 같다는 뜻이다.
        if result[1] < 500:
            ##### 어랫든 0~100 표시하려고 한듯
            confidence = int(100*(1-(result[1])/300))
            # 유사도 화면에 표시
            display_string = str(confidence) + '% Confidence it is user'
            cv2.putText(image, display_string, (100,120), cv2.FONT_HERSHEY_COMPLEX, 1, (250,120,255), 2)
            #75 보다 크면 동일 인물로 간주해 UnLocked!
            if confidence > 75:
                cv2.putText(image, "Unlocked", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
                cv2.imshow('Face Cropper', image)
            else:
                #75 이하면 타인.. Locked!!!
                cv2.putText(image, "Locked", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 2)
                cv2.imshow('Face Cropper', image)
        except:
            #얼굴 검출 안됨
            cv2.putText(image, "Face Not Found", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0), 2)
            cv2.imshow('Face Cropper', image)
            pass
        if cv2.waitKey(1)==13:
            break
    cap.release()
    cv2.destroyAllWindows()
```

```
43  
44  
45  
46  
47  
48  
49  
50  
51
```

```
#검출된 사진을 흑백으로 변환  
face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)  
#이미지에서 학습한 모델로 예측시도  
result = model.predict(face)
```



```
C:\Facial-Recognition> c: && cd c:\Facial-Recognition && /C && C:\>  
python3\lib\python\debug\launcher SPSI -- c:\Facial-Recognition\face_3.py  
c:\Facial-Recognition\face_3.py:31: SyntaxWarning: "is" with a literal. Did you  
mean "=="?  
if faces is():  
    if faces is():  
        Model Training complete!!!!
```

써드파티모듈

3rd Party Module

구글 캘린더

스크린 캐스트

핸드폰 알림 시스템

구글 캘린더

구글 API

```
module: "calendar",
header: "일정",
position: "top_left",
config: {
  calendars: [
    {
      symbol: "calendar-check",
      url: "https://calendar.google.com/calendar/ical/"
    }
  ]
}
```

4월

임인년3월



월

화

수

목

금

28

(26)

29

(27)

30

(28)

31

(29)

1

(1)

졸작발표

기상

저녁 약속

점심약속

4

(4)

5

(5)

6

(6)

7

(7)

8

(8)

C++프로그

청명

한식

전자회로 수

식목일

점심약속

점심 약속

운동

월요일, 2022년 4월 4일

10:54⁴⁴

일정

- ▣ C++프로그래밍 수업
- ▣ 점심 약속
- ▣ 점심약속
- ▣ 운동
- ▣ 전자회로 수업

4시간 후
내일 12:00
수요일 12:30
수요일 18:00
목요일 09:00

핸드폰 스크린캐스트 (Screencast API)

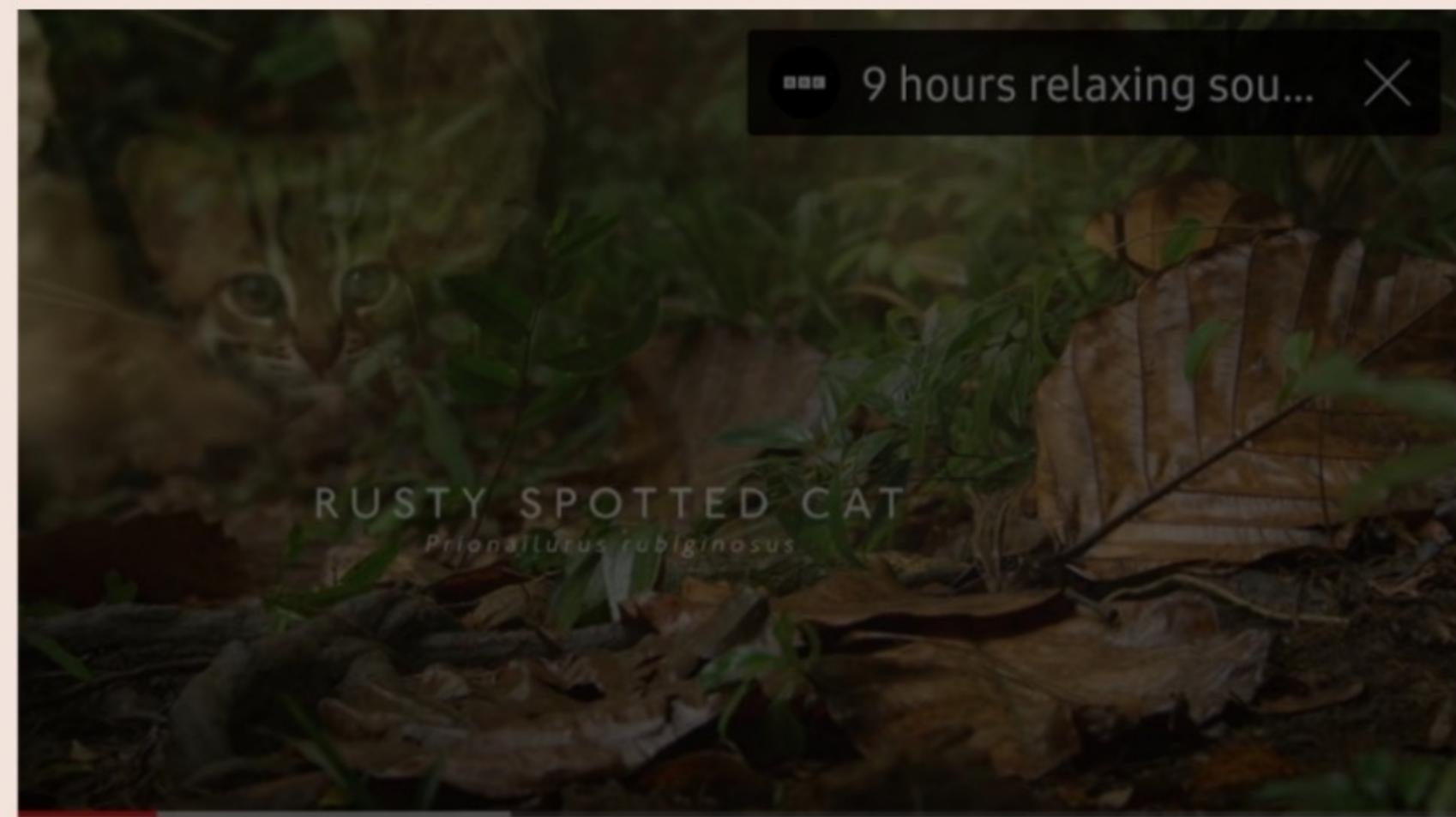
MMM-Screencast

🔗 **MMM-Screencast**

A module to cast to the [MagicMirror²](#). Currently, only YouTube casting is supported. Hopefully, I will have time to add more casting options.

NOTE: MMM-Screencast has been updated to deal with new support, please pull from master to ensure that you have the most up-to-date version. Thanks!

```
module: 'MMM-Screencast',
position: 'bottom_left',
config:{
    position:'bottomLeft',
    height: 270,
    width: 480,
    y: 500,
}
```



9 hours relaxing sou... X

World's smallest cat 🐱 - BBC

조회수 4854만회 · 4년 전 #bbc

전송 대상

raspberrypi

구독

TV 코드로 연결

댓글

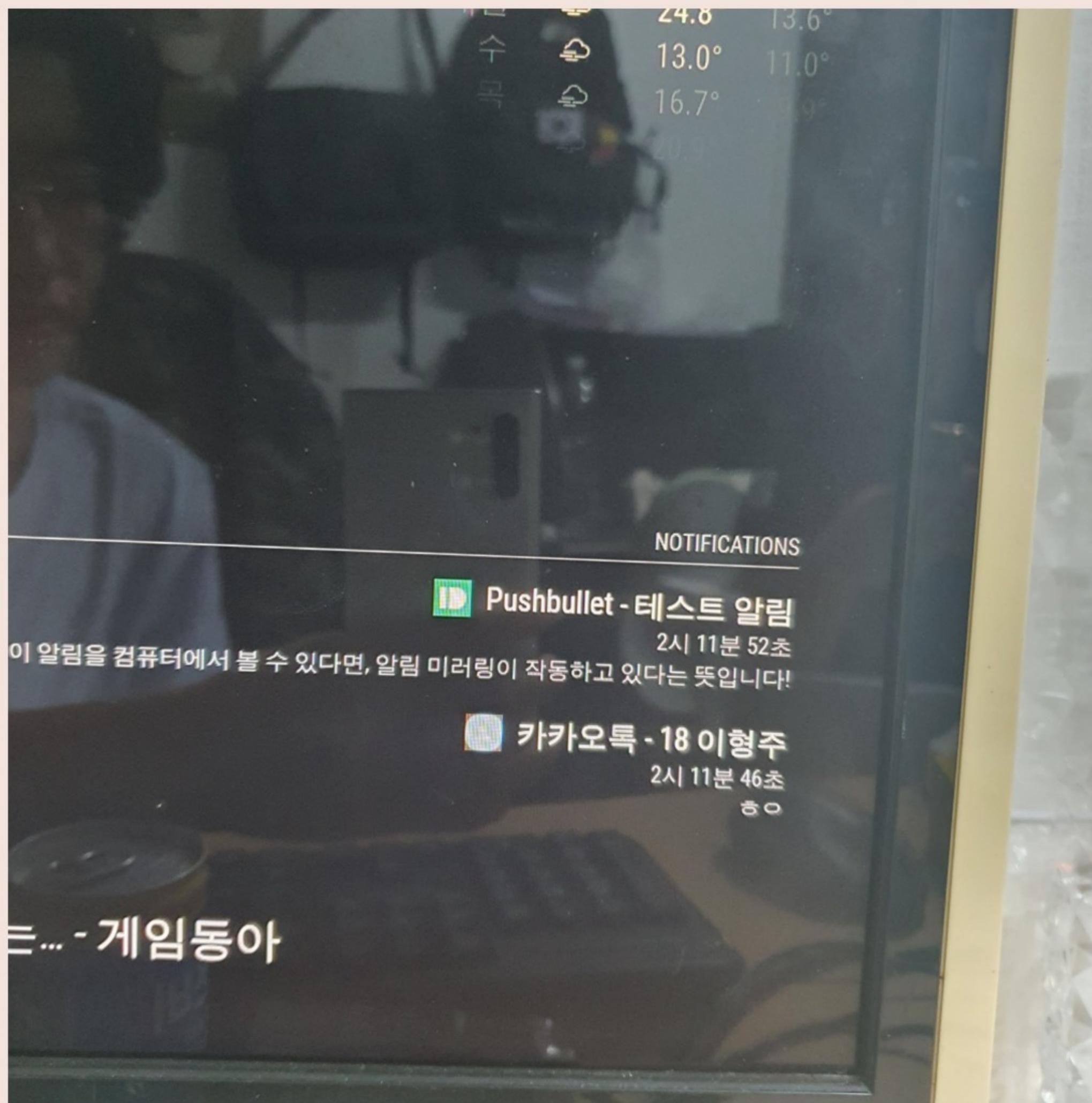
핸드폰 알림 시스템 (pushbullet API)

MMM-PushBulletNotifications

MMM-PushBulletNotifications

This [MagicMirror²](#) module displays phone notifications via the PushBullet API. Use this module to display notifications, send (some) commands to your Magic Mirror or make your Magic Mirror speak out loud.

```
module: 'MMM-PushBulletNotifications',
header: 'Notifications',
position: 'bottom_right', // This can be any of the regions.
config: {
    // See 'Configuration options' for more information.
    accessToken: "o.kPLnoYHE8Ry8TrMRGgLfL5UvRePqvPgC", //PushBullet API Access Token
    endToEndPassword: null,
    numberOfNotifications: 3,
    filterTargetDeviceName: "",
    showPushesSentToAllDevices: true,
    onlyAllowCommandsFromSourceDevices: [],
    fetchLimitPushBullet: 50,
    showPushes: true,
    showPushesOnLoad: true,
    showDismissedPushes: true,
    showMirroredNotifications: true,
    onlyShowLastNotificationFromApplication: false,
    showIndividualNotifications: false,
    showSMS: true,
    showMessage: true,
    showIcons: true,
    showDateTime: true,
    localesDateTime: 'ko-KR',
    playSoundOnNotificationReceived: true,
    soundFile: 'modules/MMM-PushBulletNotifications/sounds/new-message.mp3',
    maxMsgCharacters: 50,
    maxHeaderCharacters: 32,
    showModuleIfNoNotifications: true,
    noNotificationsMessage: "No new notifications",
    debugMode: false,
}
```



월요일, 2022년 4월 11일
02:43⁵⁹

일정

■ 불행 3시간 후
■ 점심 악속 수요일 11:00
■ 저녁 악속 목요일 18:00

현재 날씨

⇒ 1 ℉ 06:02

12.7°

체감온도 11.7°

날씨 예보 ANSEONG, KR

오늘	25.6°	12.7°
내일	24.8°	13.6°
수	13.0°	11.0°
목	16.7°	9.9°
금	20.9°	10.2°

오늘 하루 화이팅

NOTIFICATIONS

■ 카카오톡 - 18 이월주
24시간 43초
테스트 메시지→

■ Postbullet - 테스트 알림
2시간 38초
이 알림을 컴퓨터에서 볼 수 있다면, 잘된 미려증이 작동하고 있다는 뜻입니다!

구글 뉴스, 4시간 전:
후 SNS에 영상 유포_러軍 만행에 네티즌 분노_조선일보 - The Chosunilbo

JOOYONTECH



커스텀 모듈 (custom modules)

커스텀 모듈 설계 : MMM-Test

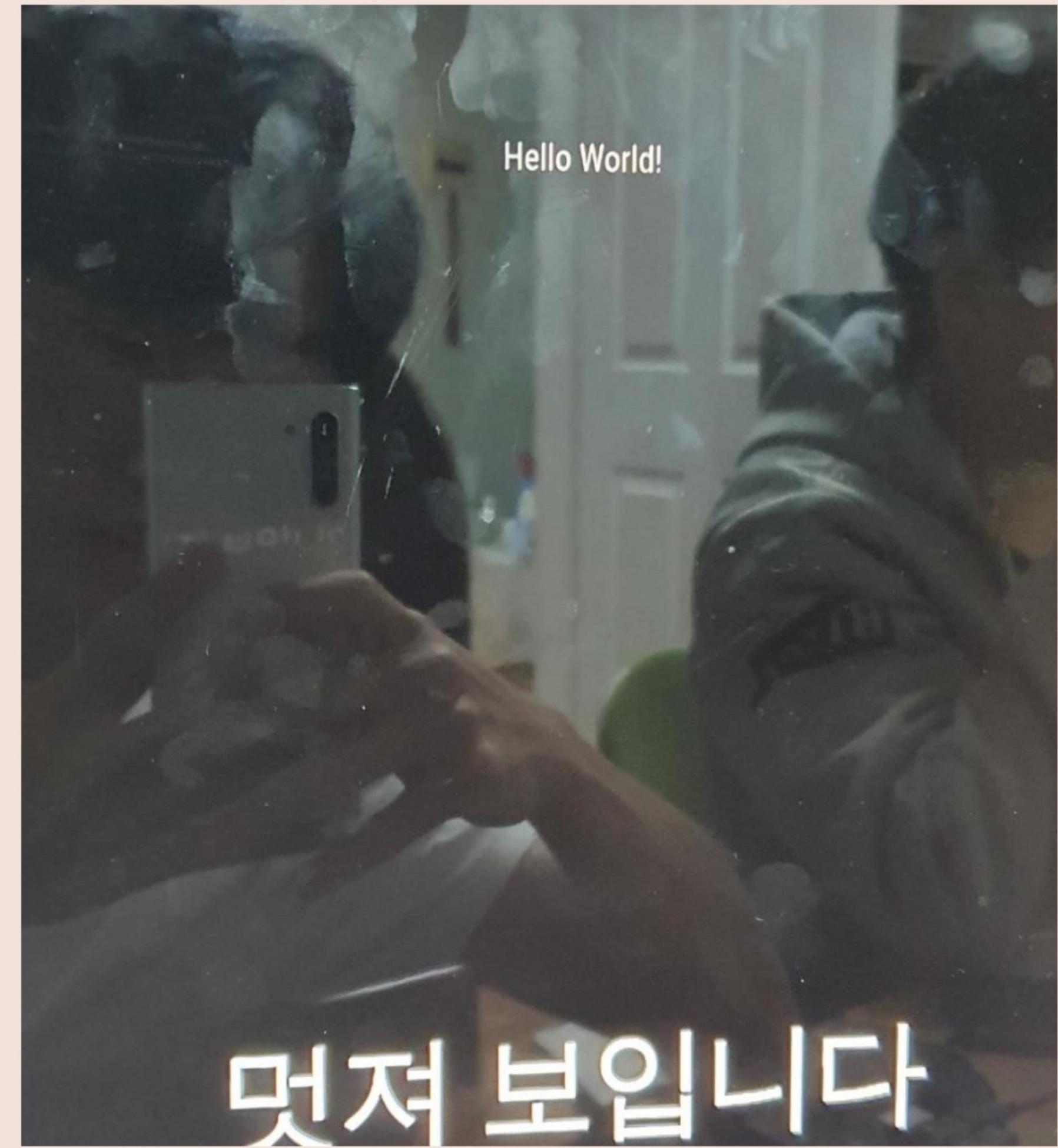
```
//helloworld.js:

Module.register("helloworld",{
    // Default module config.
    defaults: {
        text: "Hello World!"
    },
    // Override dom generator.
    getDom: function() {
        var wrapper = document.createElement("div");
        wrapper.innerHTML = this.config.text;
        return wrapper;
    }
});
```

```
Module.register("MMM-test", {
  defaults: {
    ext : "Hello World!"
  },
  getDom : function(){
    var wrapper = document.createElement("div");
    wrapper.innerHTML = this.config.text;
    return wrapper;
  }
});
```

```
modules: [
  {
    module: 'MMM-test',
    position: 'top_center',
  },
]
```

```
pi@raspberrypi:~ $ cd MagicMirror/
pi@raspberrypi:~/MagicMirror $ cd modules
pi@raspberrypi:~/MagicMirror/modules $ cd MMM-test
pi@raspberrypi:~/MagicMirror/modules/MMM-test $ npm install
```



```
Module.register("MMM-STT", {
    getDom: function(){
        const wrapper = document.createElement("div");
        wrapper.className = "wrapper";

        const leftBox = document.createElement("td");
        leftBox.className = "leftBox";
        wrapper.appendChild(leftBox);

        const img = document.createElement("img");
        img.className = "img";
        img.src = "./modules/MMM-STT/jarvis.jpeg";
        leftBox.appendChild(img);

        const rightBox = document.createElement("div");
        rightBox.className = "rightBox";
        wrapper.appendChild(rightBox);

        const text = document.createElement("span");
        text.innerText = "음성 인식 중...";
        text.className = "text";
        rightBox.appendChild(text);

        return wrapper;
    },
    getStyles: function(){
        return[
            "MMM-STT.css",
        ];
    },
    notificationReceived: function(notification, payload, sender){
        switch(notification){
            case "ALL_MODULES_STARTED":
                var timer = setInterval(()=>{
                    this.sendSocketNotification("DO_PYTHON")
                },10000)
            },
            socketNotificationReceived: function(notification, payload){
                switch(notification){
                    case "PYTHON_DONE":
                        var wrapper = document.getElementById("wrapper")
                }
            },
        })
    }
})
```

MMM-STT

```
Module.register("MMM-STT", {
    getDom: function(){
        const wrapper = document.createElement("div");
        wrapper.className = "wrapper";

        const leftBox = document.createElement("td");
        leftBox.className = "leftBox";
        wrapper.appendChild(leftBox);

        const img = document.createElement("img");
        img.className = "img";
        img.src = "./modules/MMM-STT/jarvis.jpeg";
        leftBox.appendChild(img);

        const rightBox = document.createElement("div");
        rightBox.className = "rightBox";
        wrapper.appendChild(rightBox);

        const text = document.createElement("span");
        text.innerText = "음성 인식 중...";
        text.className = "text";
        rightBox.appendChild(text);

        return wrapper;
    },
});
```

MMM-STT-CSS

```
.wrapper{
    width:250px;
    height:50px;
    display: flex;
    align-items:center;
}

.leftBox{
    width:30%;
    height:100%;
    padding: 15px;
    display: flex;
    justify-content: center;
    align-items: center;
}

.img{
    width: 50px;
    height: 50px;
    border-radius:50%;
}

.rightBox{
    width:70%;
    height:100%;
    padding: 15px;
    display: flex;
    justify-content: flex-end;
    align-items: center;
}

.text{
    font-size: 20px;
    font-weight: bold;
    color :green;
}
```



MMM-STT

```
notificationReceived: function(notification, payload, sender){  
    switch(notification){  
        case "ALL_MODULES_STARTED":  
            var timer = setInterval(()=>{  
                this.sendSocketNotification("DO PYTHON")  
            }, 10000)  
        }  
    },  
    socketNotificationReceived: function(notification, payload){  
        switch(notification){  
            case "PYTHON_DONE":  
                var wrapper = document.getElementById("wrapper")  
            }  
        },  
    })
```

node_helper

```
const NodeHelper = require("node_helper");
const spawn = require("child_process").spawn;

module.exports = NodeHelper.create({
  socketNotificationReceived(notification, payload){
    if(notification === "DO_PYTHON"){
      this.job();
    }else{
    }
  },
  job(){
    const process = spawn("python3", [/home/pi/MagicMirror/modules/MMM-STT/Kakao_STT.py"])

    process.stdout.on('data',(data)=>{
      console.log("stdoutOn"+data);
    })
    process.stderr.on('data',(data)=>{
      console.log("stderrON"+data);
    })
    process.on('exit',(code)=>{
      console.log("exit"+code);
      this.sendSocketNotification("PYTHON_DONE",data)
    })
  }
})
```



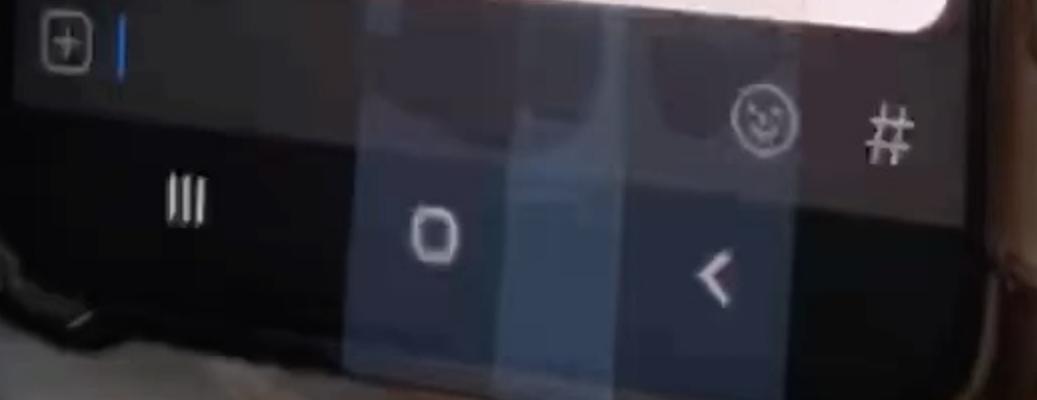
아
자
차
카
타
파
하

오류 0000

일일 확진자 수: 43925
누적 확진자수: 17658794

자세히 보기

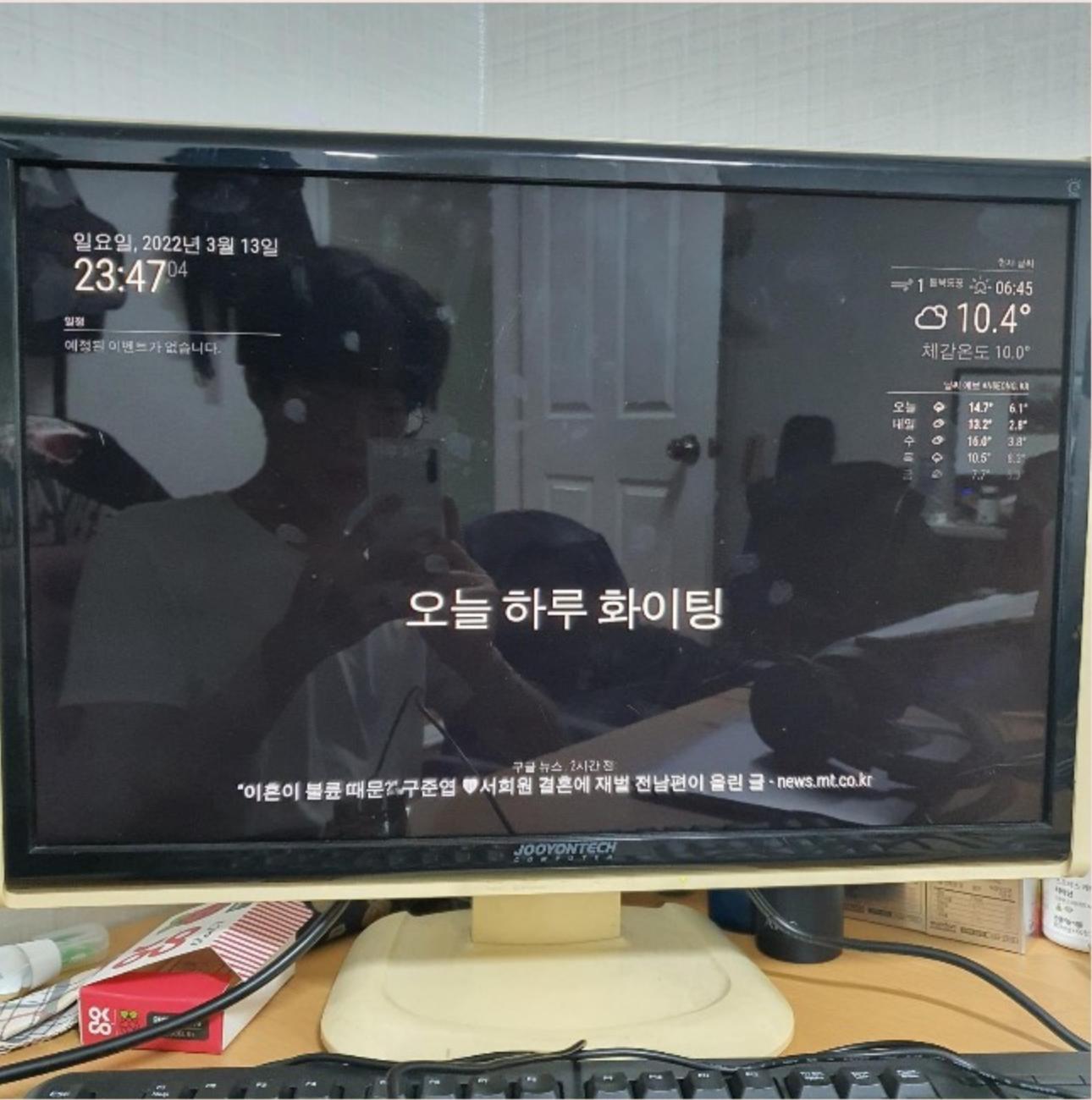
20 167
스마트미러 서비스



하드웨어 (HardWare)

모니터 + 아크릴판 + 하프필름 & 나무액자

밝은 환경

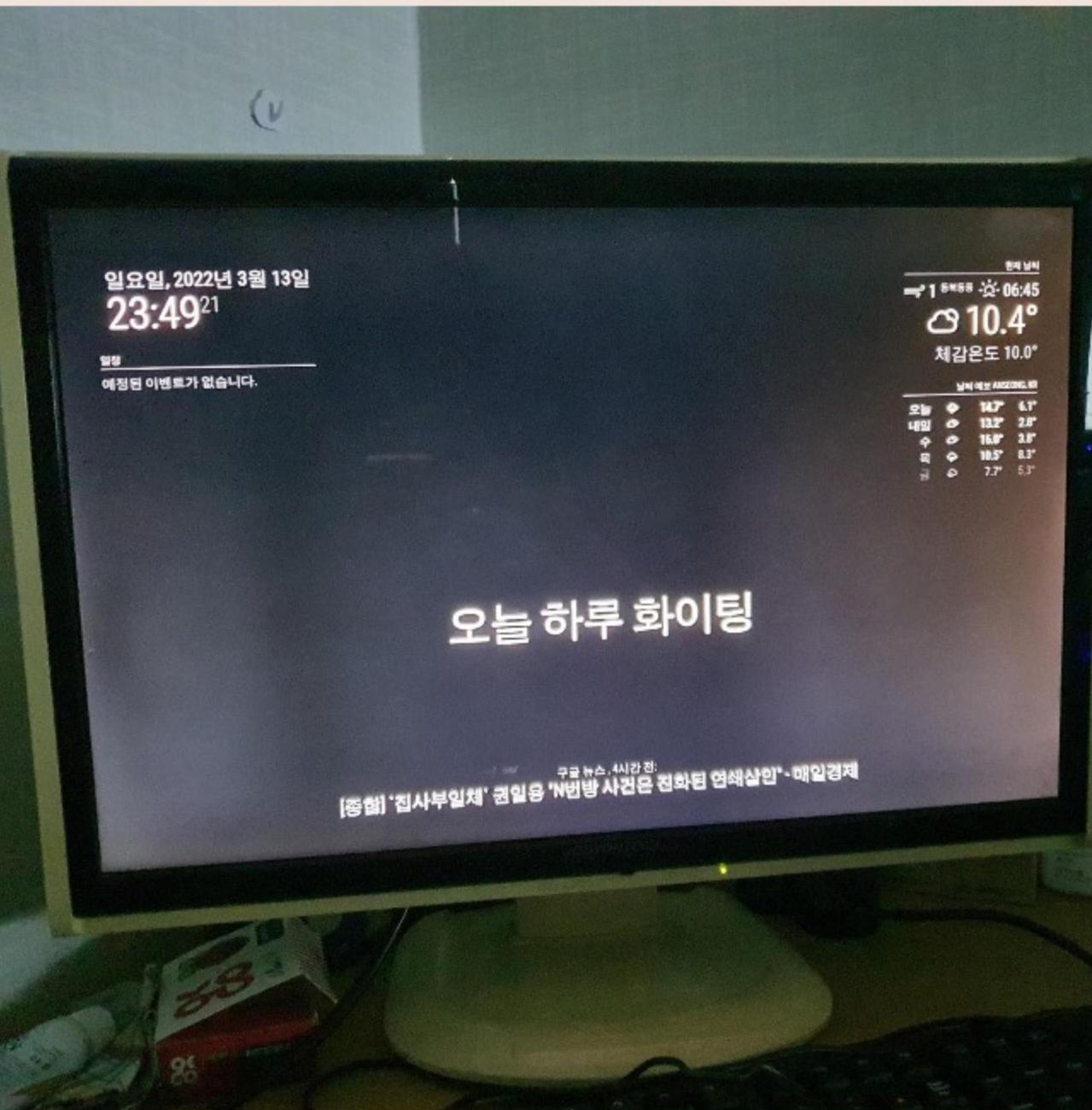


필름 붙이기 전

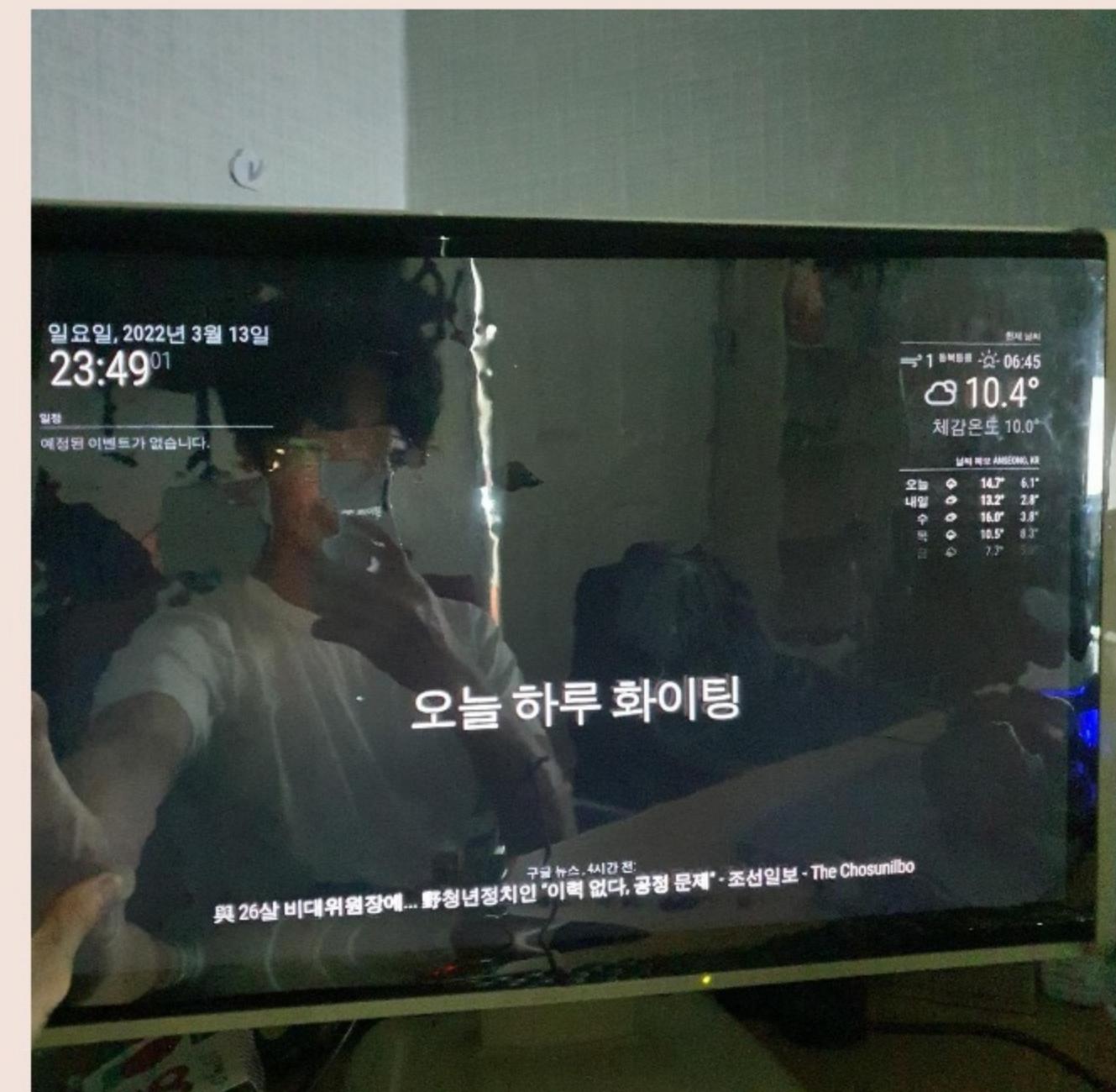


필름 붙인 후

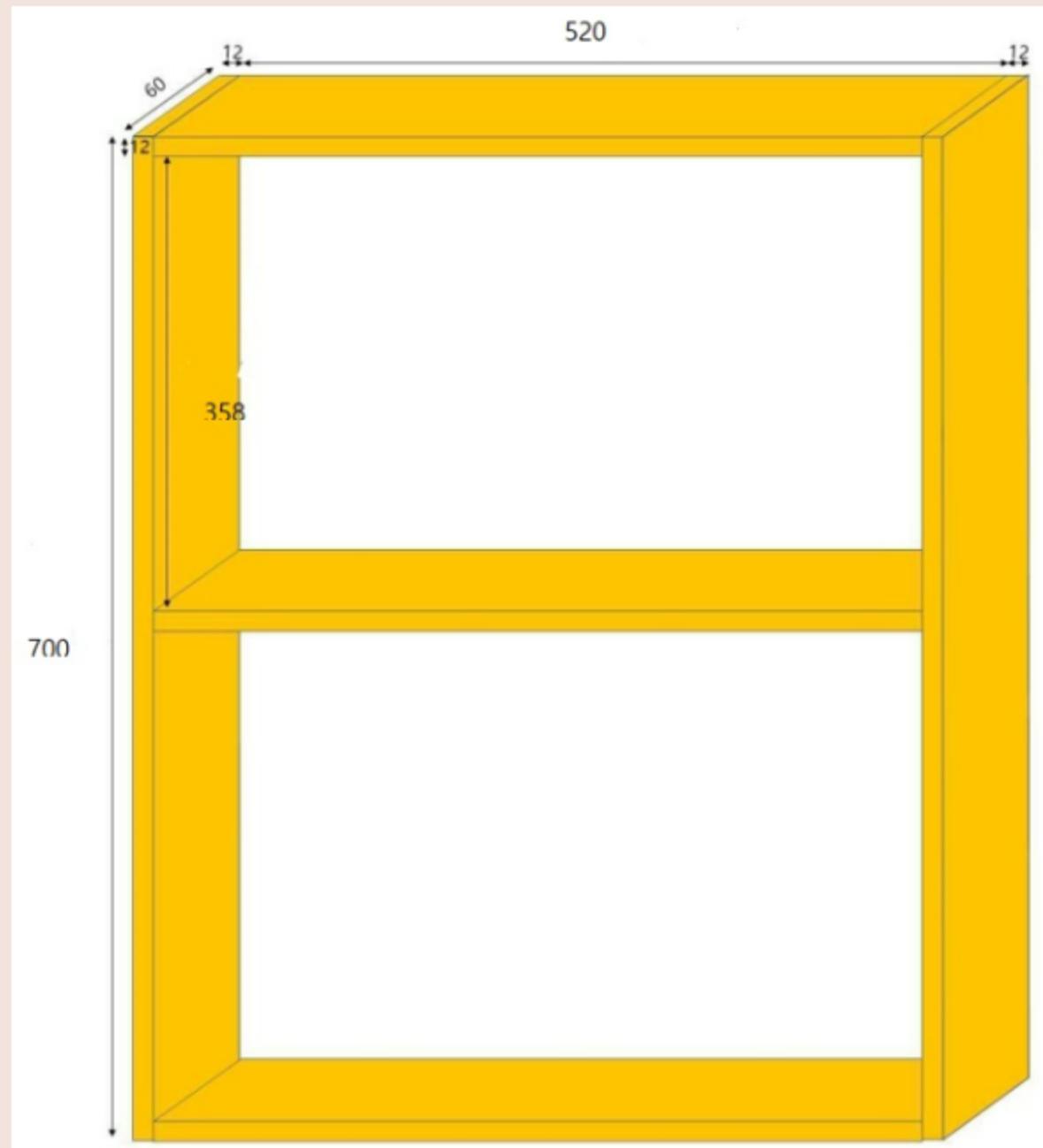
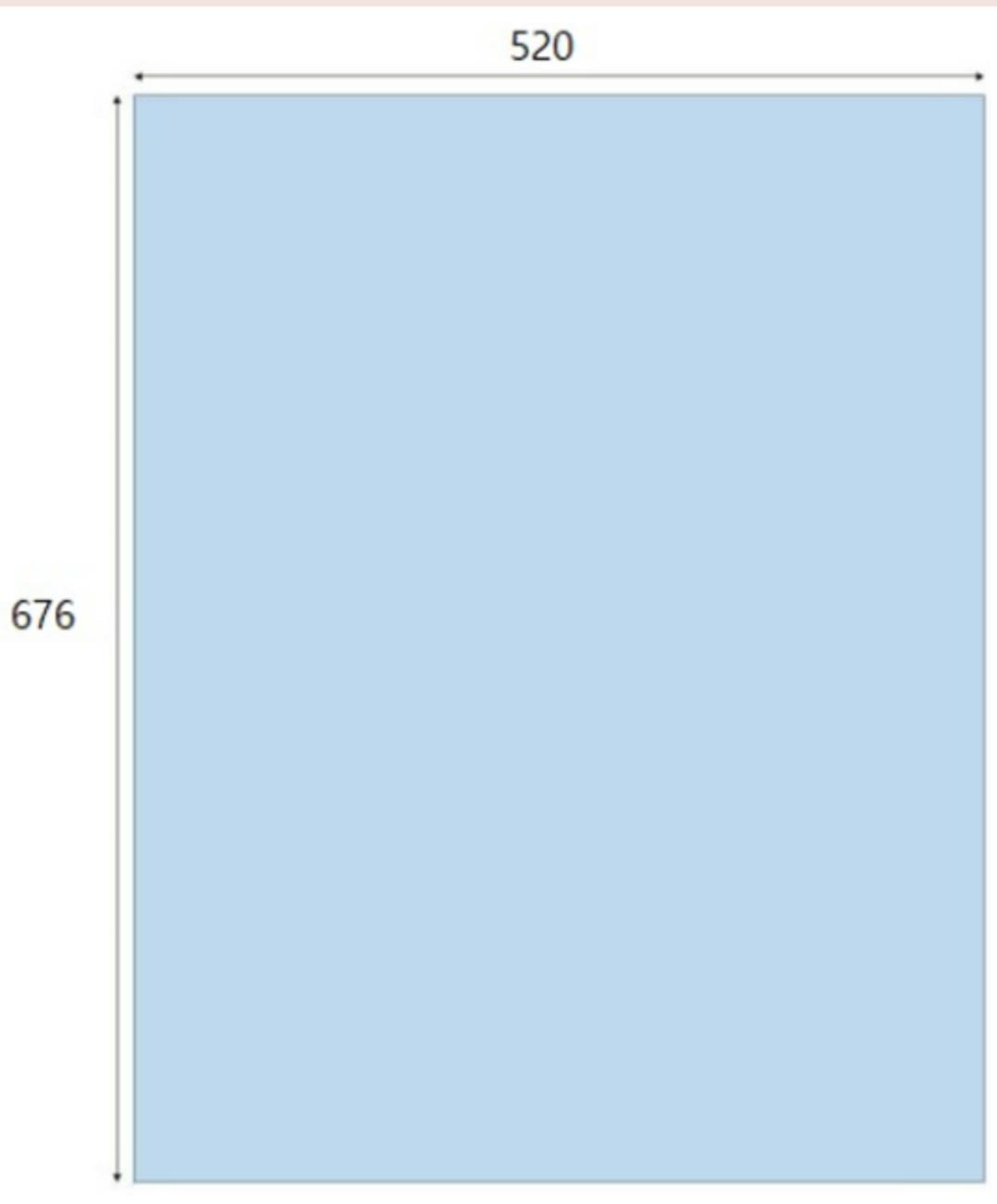
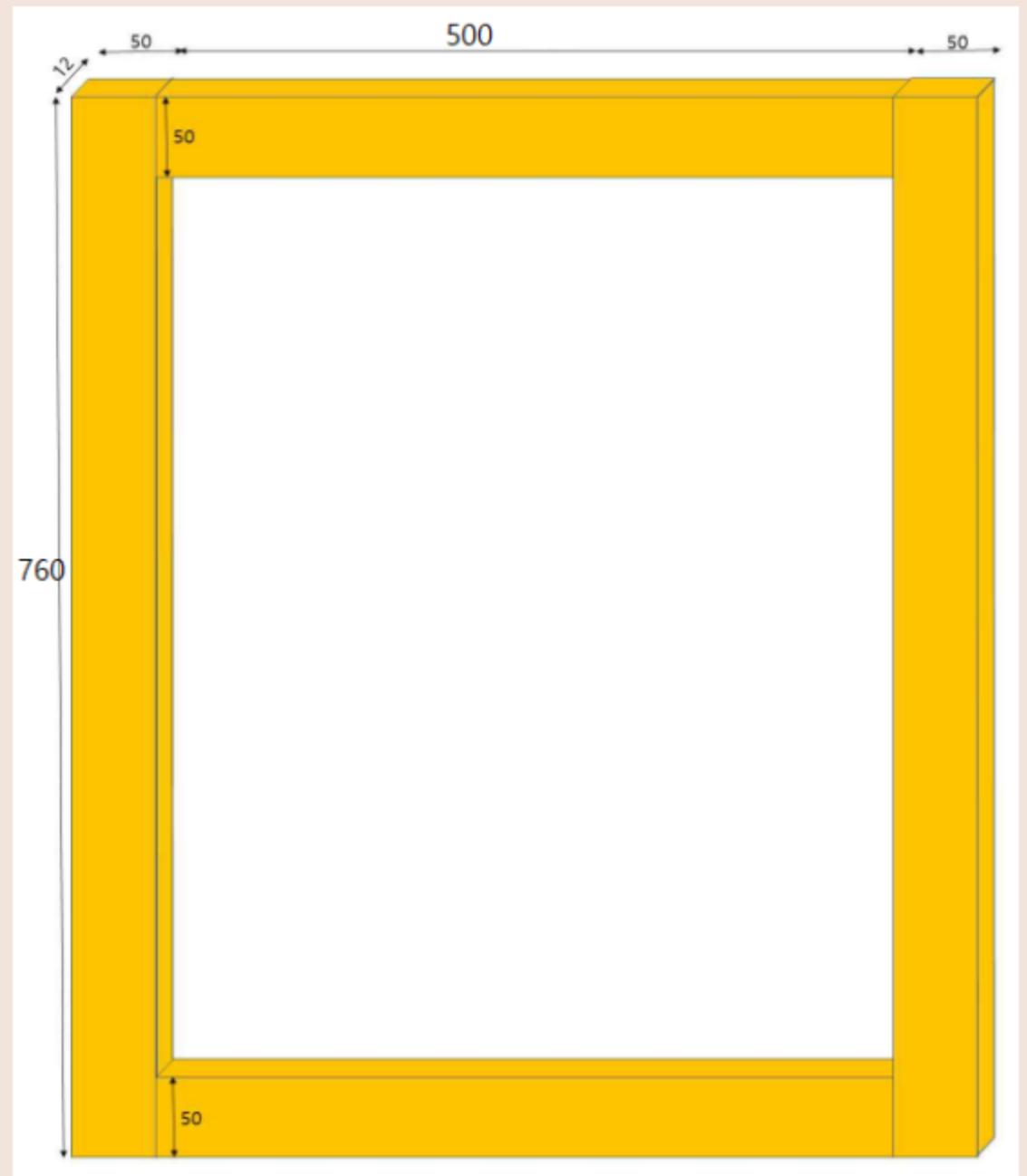
어두운 환경

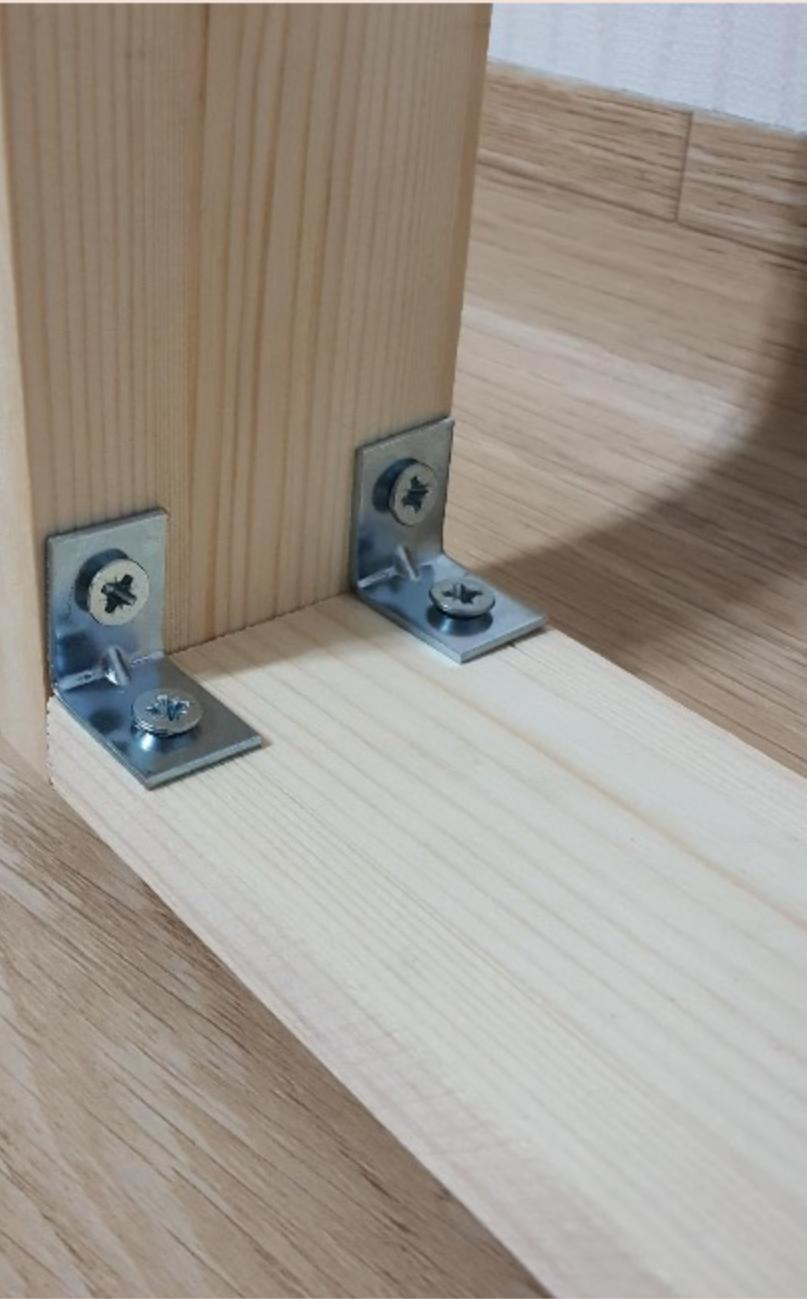


필름 붙이기 전

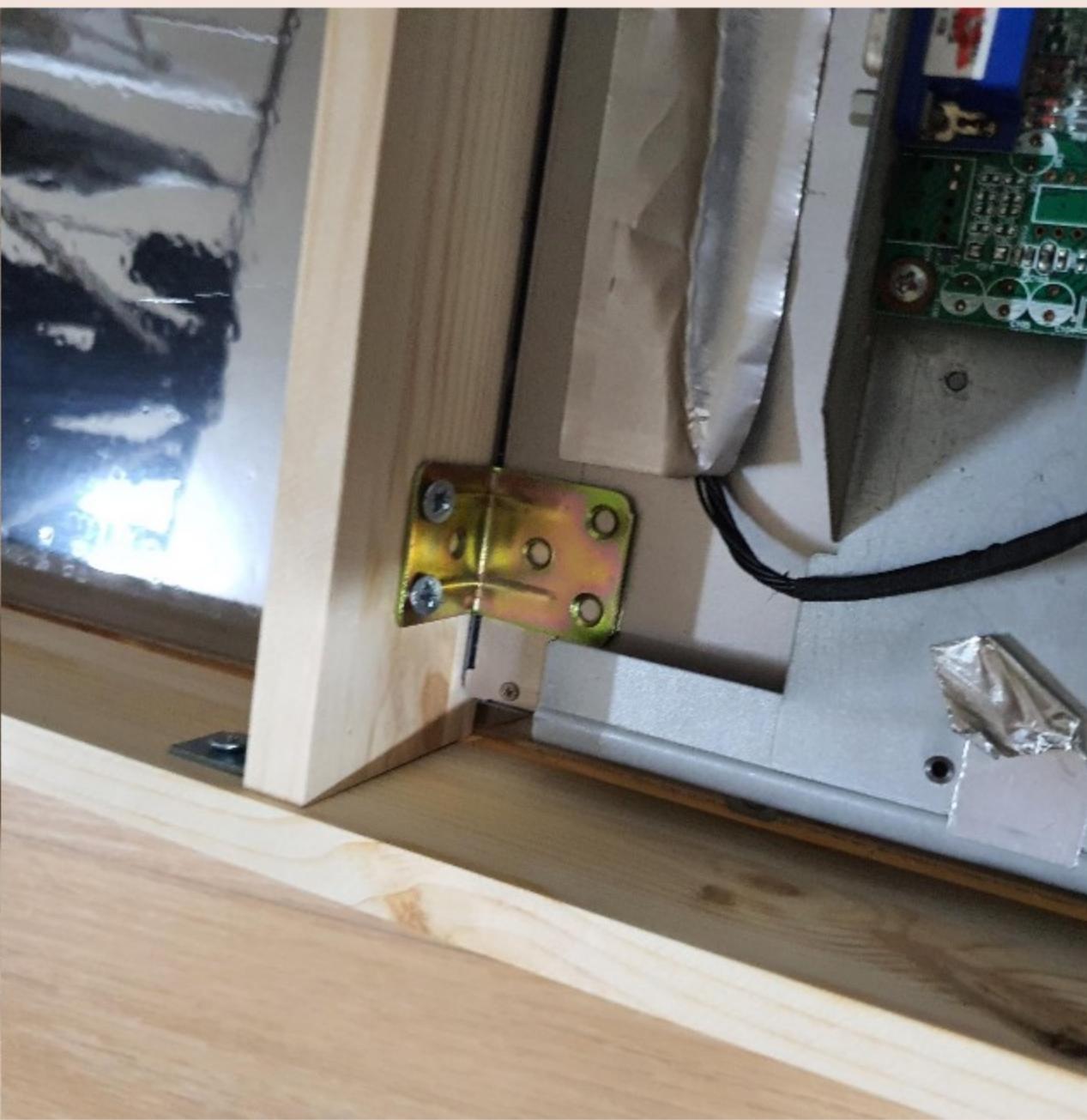
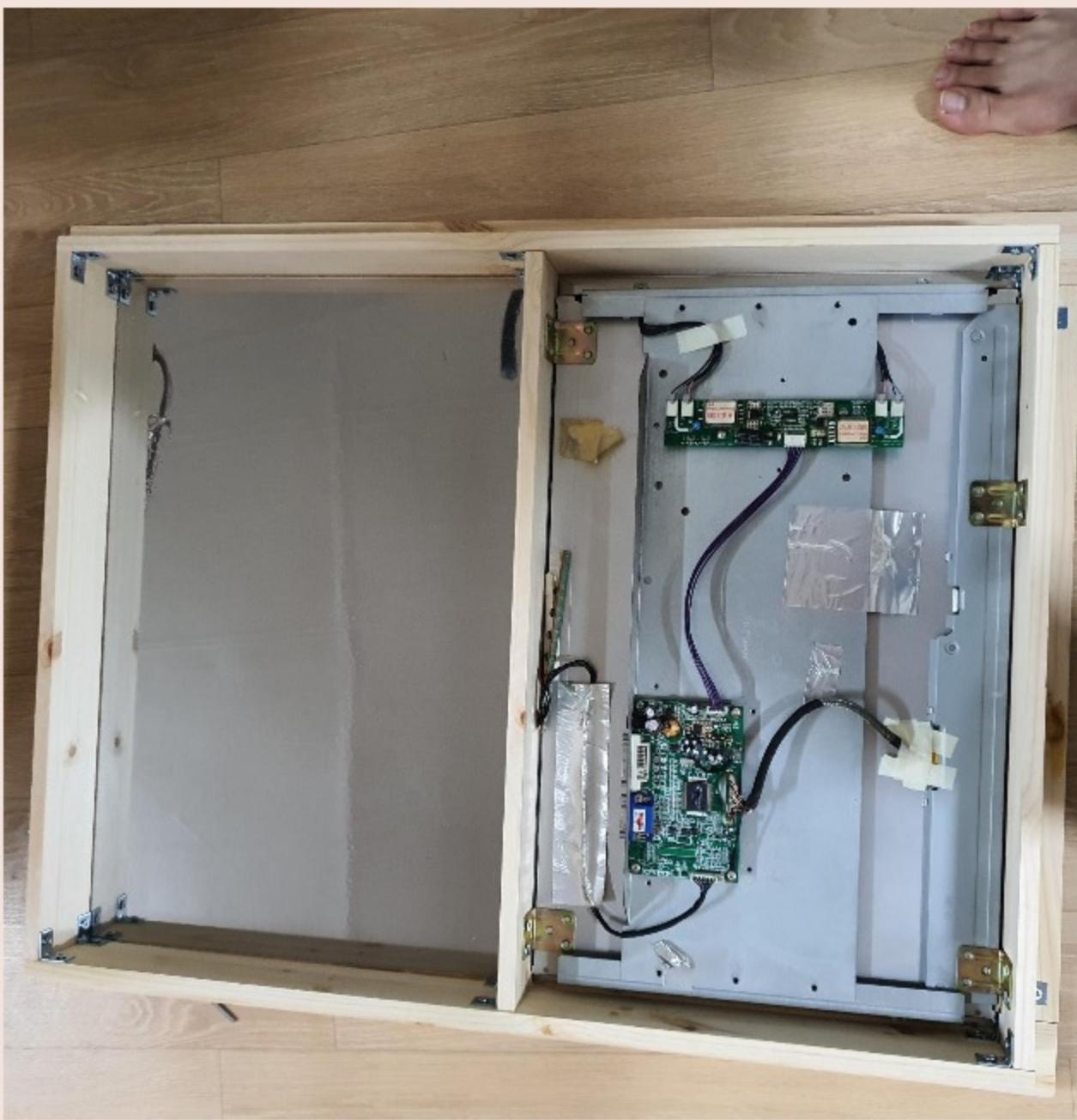


필름 붙인 후











최종 출력 영상

얼굴 인식 성공 시 GUI 화면 출력

내 풀디

Thonny - /home/pi/Facial_Recognition/Facial_Recognition.py 6

New Load Save Run Debug Over Into Out Stop

Facial_Recognition.py

```
68         # 유사도 화면에 표시
69         display_string = str(confidence) + '% Confidence it is user'
70         cv2.putText(image, display_string, (100, 120),
71                     cv2.FONT_HERSHEY_COMPLEX, 1, (250, 120, 255), 2)
72         # 80 보다 크면 동일 인물로 간주해 UnLocked!
73         if confidence > 80:
74             cv2.putText(image, "Unlocked", (250, 450),
75                         cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
76             cv2.imshow('Face Cropper', image)
77             os.system("cd ~pi \n pm2 start mm.sh")
78             break
79         else:
80             # 80 미만.. Locked!!
81             cv2.putText(image, "Locked", (250, 450),
82                         cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 2)
83             cv2.imshow('Face Cropper', image)
84     except:
```

Shell

```
>>> cd ~pi \n pm2 start mm.sh
```

/home/pi/Facial_Recognition/Facial_Recognition.py:33: SyntaxWarning: "is" with a comparison to None always evaluates as False; instead use "=="

if faces is ():

Q & A