

사이버보안 캡스톤 디자인

# 프로젝트 최종 발표

-얼굴 인식을 활용한 주기적 사용자 인식 프로그램-

1771114 박현철

1771202 이주혁

# INDEX

0 1

프로젝트  
소개

0 2

프로젝트  
핵심 기능

0 3

주요 코드

0 4

기능 시연

# 01

## 프로젝트 소개

### 주제 선정 이유 및 배경

#### 女동료 노트북 해킹한 30대男, 한 달 내내 전부 훔쳐봤다

[데일리안] 입력 2021.03.15 21:57 | 수정 2021.03.15 16:47 | 이지희 기자 (ljh4749@dailian.co.kr)



직장 동료의 노트북을 해킹해 메신저 대화 기록을 엿보고 사진 등을 저장한 30대 남성이 실형을 선고받고 법정 구속됐다.



A씨는 2018년 8월부터 한 달간 사무실에서 피해자 B씨가 사용하는 노트북 컴퓨터에 해킹프로그램을 설치했다.

A씨는 프로그램을 작동 시켜 B씨 네이트온과 카카오톡, 구글 아이디 및 비밀번호 등을 알아냈다고 조사됐다.

그는 이런 방법으로 알아낸 정보를 이용해 B씨 네이트온 계정에 접속하는 등 총 40회에 걸쳐 정보통신망에 침입했다고 알려졌다. 또 B씨 카카오톡 대화, 사진, 메시지 등을 자신의 휴대전화에 다운로드해 보관하기도 했다고 파악됐다.

정 부장판사는 "피고인은 범행 당시 한 달 동안 수습회에 걸쳐 별다른 죄책감 없이 피해자의 노트북 컴퓨터를 해킹했다"며 "피고인은 이렇게 알아낸 카카오톡 대화 내용이나 메시지, 사진 등을 자신의 휴대폰 등에 보관함으로써 피해자의 사생활의 비밀과 인격권이 매우 심각하게 침해된 것으로 보인다"고 지적했다.

# 01

## 프로젝트 소개

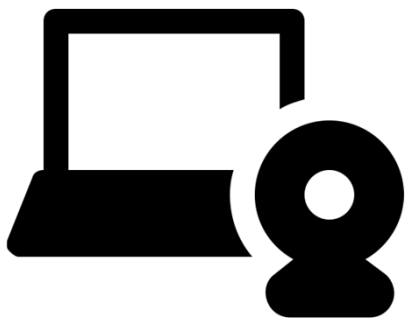
### 주제

- 얼굴인식 기능을 활용한 주기적 사용자인증 프로그램

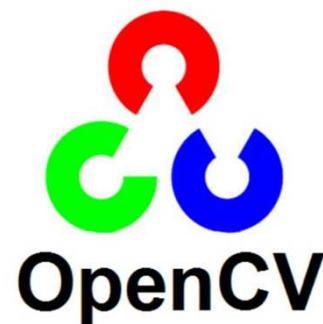
### 주제 선정 이유 및 배경

- 회사 등에서 컴퓨터를 사용하다 자리를 비울 경우 다른 사람들의 접근에 대처 불능
- 사용자의 부재 시 해킹 프로그램 설치나 사용자의 개인 정보를 염탐하는 사례가 급증
- 해킹 프로그램에 공격당한 경우 타인이 사용자의 컴퓨터를 자유자재로 염탐 가능

## 프로젝트 소개



노트북의 웹캠



OpenCV

- OpenCV를 활용하여 주기적으로 사용자의 얼굴을 인식
- 주기적 얼굴인식 과정에서 다른 사람의 얼굴이 인식될 경우 화면 잠금
- 허가 받지 않은 타인의 얼굴이 인식될 경우 사용자에게 알림

개발에 필요한 S/W 및 기술

01

## 프로젝트 소개



MediaPipe



Firebase



# 01

## 프로젝트 소개



파이썬 [프로젝트의 기반이 되는 프로그래밍 언어]



MediaPipe [얼굴을 감지하기 위해 사용되는 플랫폼]



Firebase [사용자의 데이터를 저장]



Pyqt [간편한 사용을 위해 GUI로 표현]

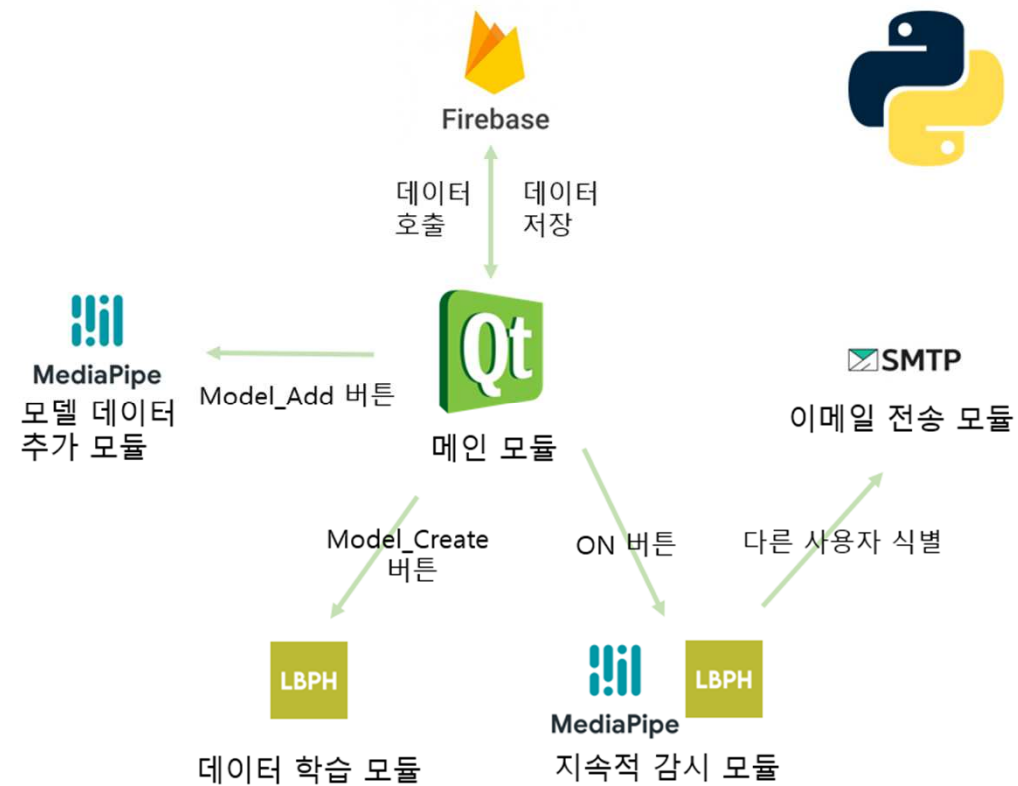


LBPH [이미지 비교에 사용되는 알고리즘]



SMTP [이메일 전송에 사용되는 프로토콜]

# 프로젝트 소개





## 02

### 프로젝트 핵심기능

#### MediaPipe

- > MediaPipe 얼굴 감지는 6 개의 랜드마크를 설정해 수행됨
- > 다중 얼굴 지원이 함께 제공되는 초고속 얼굴 감지 솔루션
- > 모바일 GPU 추론에 맞게 조정된 가볍고 성능이 우수한 얼굴 감지기인 BlazeFace를 기반으로 함
- ✓ 주기적으로 얼굴 인식 과정을 수행하기 적합한 가볍고 우수한 성능을 가진 솔루션
- ✓ TensorFlow보다 가벼우며, Haar Cascades보다 인식률이 좋기에 선정

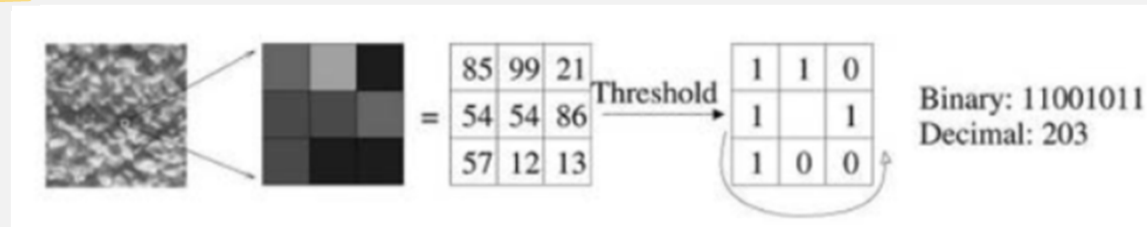
## 02

### 프로젝트 핵심 기능

#### LBPH FaceRecognizer

- > 대량으로 찍은 사용자의 사진을 LBPH FaceRecognizer를 활용해 모델링
- > 모델링에는 LBP 알고리즘 수행

#### LBP 알고리즘

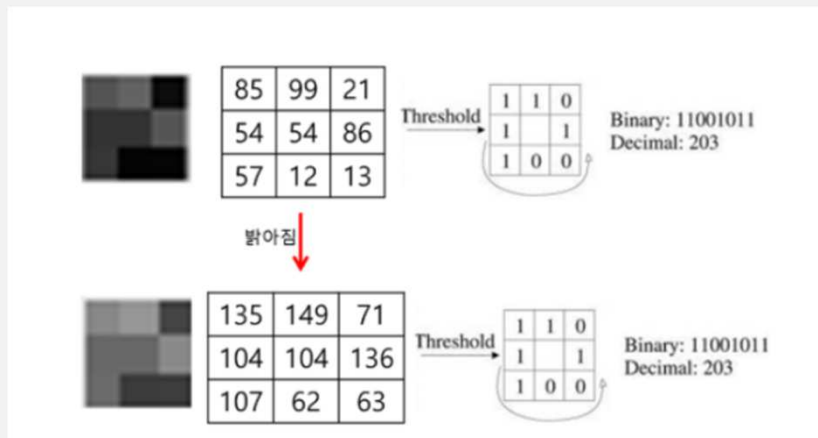


- 가운데에 위치하는 픽셀을 기준으로 픽셀 값을 비교해서 이진수로 변환(Threshold)
- 이진수를 10진수로 변환하여 히스토그램을 만들고, 이 과정을 모든 픽셀에 대해 수행

## 02

### 프로젝트 핵심 기능

#### LBP 알고리즘



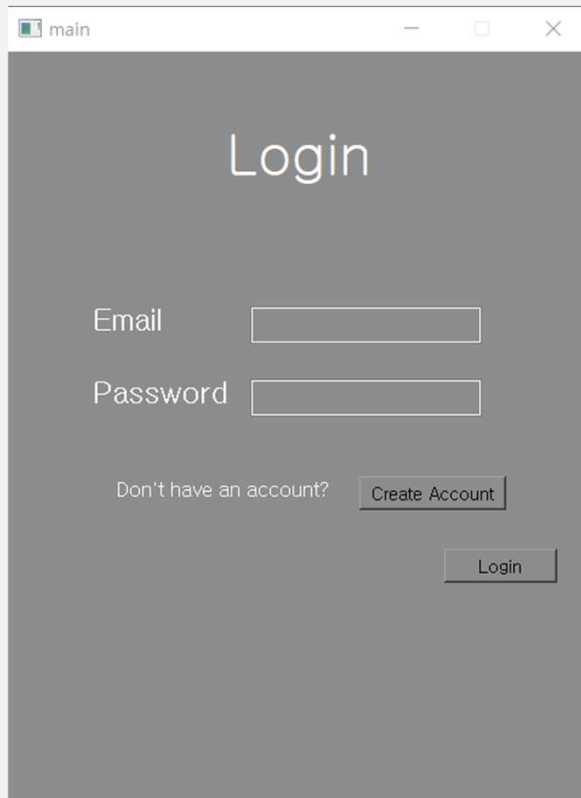
- 다른 밝기에서 촬영된 히스토그램과 비교해도 같은 값을 나타낼 수 있음
- 밝기 변화에 Robust한 알고리즘

✓ 밝기가 다른 여러 장소에서 사용될 것으로 추정되는 얼굴인식 과정에 적합한 알고리즘

## 프로젝트 핵심 기능

- OpenCV를 통해 사용자 얼굴인식 기능 구현
  - 얼굴인식 기능을 통해 사용자의 얼굴을 저장하고 다른 사람의 얼굴과 구별하도록 구현
- 얼굴인식 여부에 따라 노트북의 화면을 잠그는 기능을 구현
  - 주기적 얼굴인식 과정에서 다른 사람의 얼굴이 인식될 경우 화면을 잠금
- 등록되지 않은 사용자가 인식될 경우 사용자에게 알림
  - 웹캠으로 사진을 촬영하여 사용자에게 이메일로 경고 알림

## 프로젝트 핵심 기능



main

Login

Email

Password

Don't have an account? [Create Account](#)

[Login](#)

- 프로그램의 초기 화면
- 계정 생성, 로그인 등의 과정 수행

핵심 기능

02

## 프로젝트 핵심 기능



main

### Sign Up

Email

Password

Confirm Pass

Name

- 사용자 계정 생성 과정
- 이메일, 비밀번호, 이름 입력

## 프로젝트 핵심 기능

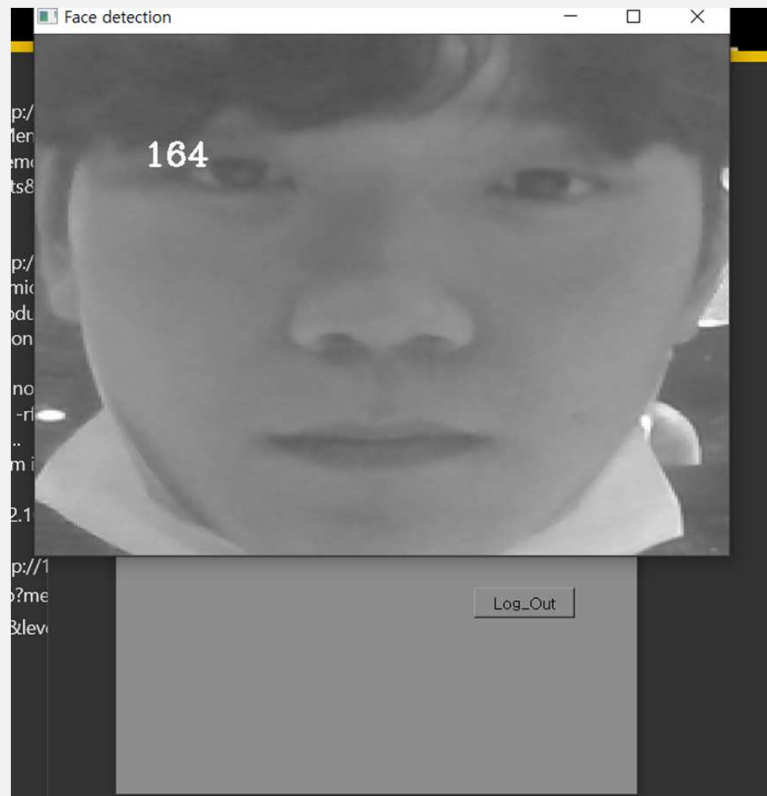


- 로그인 이후의 화면
- 얼굴 모델 생성 및 편집이 가능
- 프로그램을 작동시키고 끌 수 있음

핵심 기능

## 02

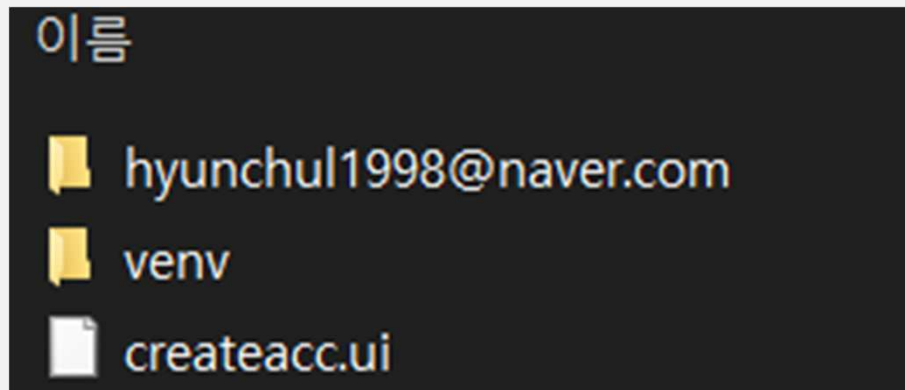
### 프로젝트 핵심 기능



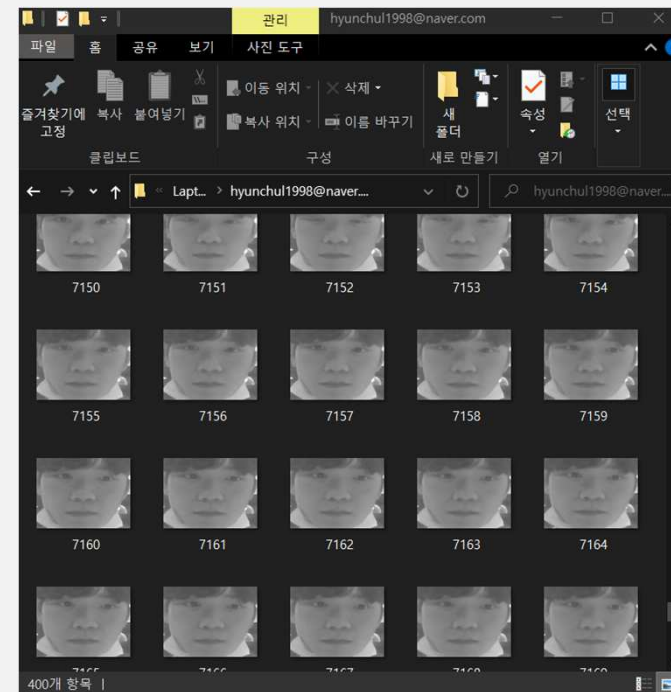
- 새로운 사용자 얼굴 모델 추가 과정
- 사용자의 사진을 200장 촬영
- 촬영된 200장으로 모델 파일 생성



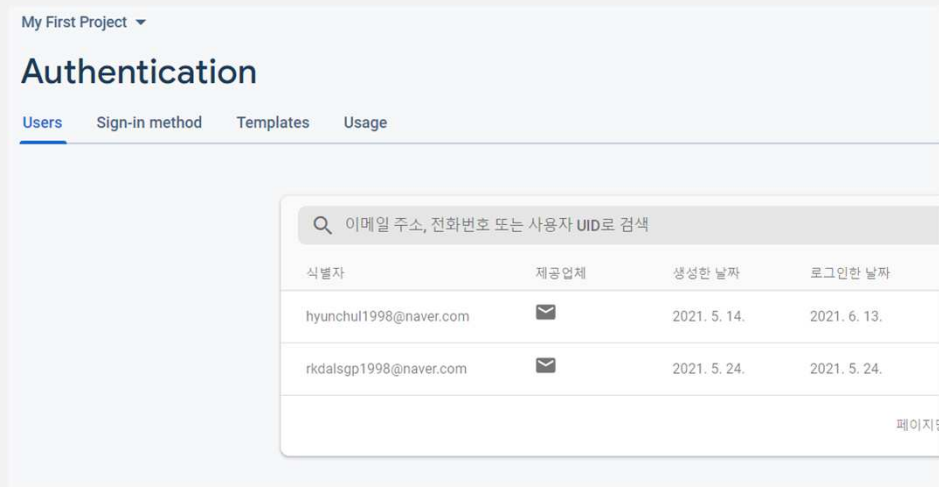
## 프로젝트 핵심 기능



- 사용자 얼굴 사진 촬영 시 폴더가 생성되며 사진이 저장됨



## 프로젝트 핵심 기능



- Firebase에 새로운 사용자 계정이 추가됨을 확인 가능

## 프로젝트 핵심 기능



Storage

Files Rules Usage

결제 사기나 피싱과 같은 악용으로부터 Storage 리소스를 보호하세요. 앱 체크 구성 X

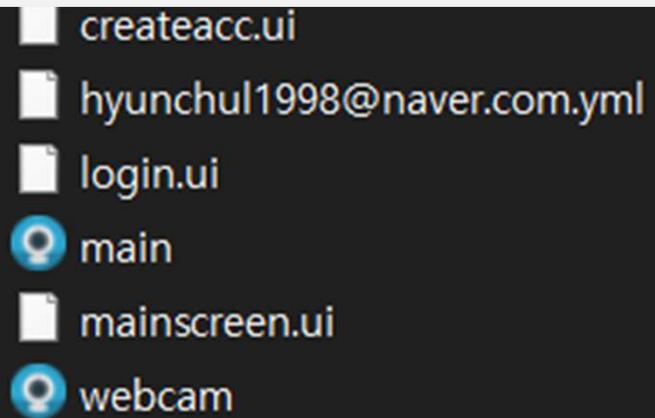
gs://subtle-seer-310104.appspot.com

파일 업로드

이름	크기	유형	최종 수정 날짜
gc159gc@naver.com	34.32 MB	application/x-msdos-program	2021. 6. 13.
hyunchul1998@naver.com	34.39 MB	application/x-msdos-program	2021. 6. 13.
rkdalsgp1998@naver.com	36.12 MB	application/x-msdos-program	2021. 5. 24.

- Firebase에 사용자의 계정 정보와 얼굴 모델 파일이 저장됨

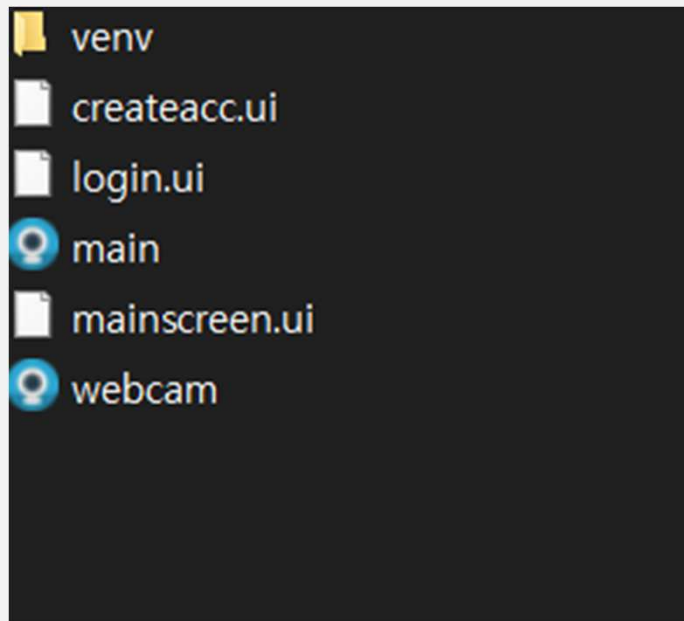
## 프로젝트 핵심 기능



createacc.ui  
hyunchul1998@naver.com.yml  
login.ui  
main  
mainscreen.ui  
webcam

- 로그인 시에 사용자의 얼굴 모델 파일인 yml파일을 Firebase에서 다운로드

## 프로젝트 핵심 기능

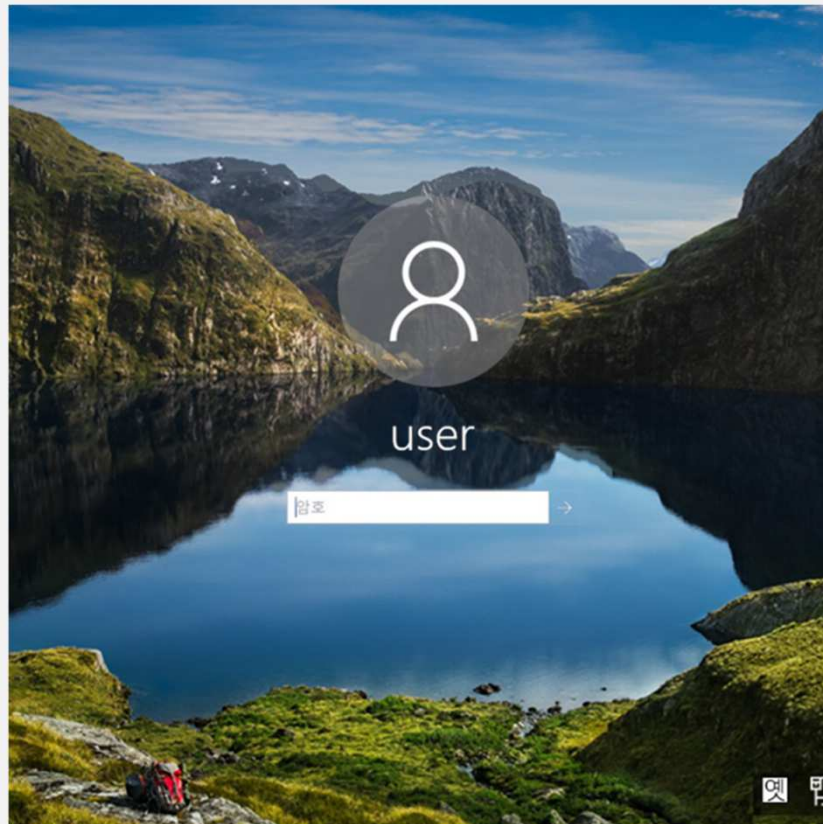


- 로그아웃 시에 사용자의 얼굴 모델 파일인 yml파일을 자동으로 삭제

핵심 기능

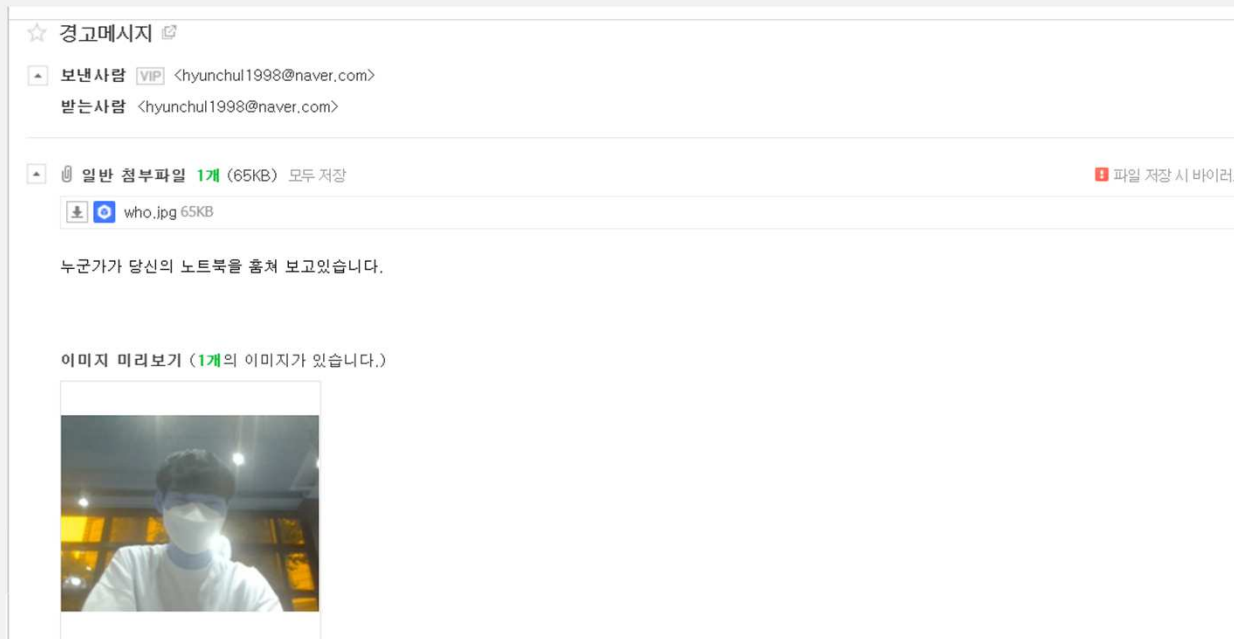
02

## 프로젝트 핵심 기능



- 프로그램 동작 시 타인의 얼굴이 인식될 경우 노트북의 화면을 잠금

## 프로젝트 핵심 기능



- 타인이 사용자의 노트북을 무단으로 사용할 시 사진과 함께 경고 메일 전송

## 03

### 주요 코드

```
class Login(QDialog): # 로그인 페이지
    def __init__(self):
        super(Login, self).__init__()
        loadUi("login.ui", self)
        self.loginbutton.clicked.connect(self.loginfunction)
        self.password.setEchoMode(QtWidgets.QLineEdit.Password)
        self.createaccbutton.clicked.connect(self.gotocreate)
        self.invalid.setVisible(False)

    def loginfunction(self): #이메일과 패스워드를 입력하여 로그인한다.
        global email
        email = self.email.text()
        password = self.password.text()
        print("Successfully logged in with email: ", email, "and password", password)
        try:
            auth.sign_in_with_email_and_password(email, password)
            self.label.setText("Loading...")
            self.download() # 인증 절차에 성공하면 모델파일을 다운로드하는 스레드 실행
            data = db.child(email.split('@')[0]).get().val() #데이터베이스에서 개인정보를 가져온다.
            global name
            name = data['name'] # 해당 이메일에 맞는 이름을 메인 페이지에 넘기기위해 전역변수로 설정
            mainscreen = Mainscreen() # 메인화면을 불러오고, download()가 끝나면 페이지를 전환한다.
            widget.addWidget(mainscreen)
        except:
            self.invalid.setVisible(True) # 로그인에 실패할 경우 실패 문구 GUI에 표시
```

- 메인 모듈의 로그인 페이지와 로그인 함수



## 03

### 주요 코드

```
if results.detections: #얼굴이 감지되었으면 다음을 실행
    for detection in results.detections:
        a = detection.location_data.relative_bounding_box
        #얼굴이 감지된 영역의 좌표를 구해서 얼굴만 나오도록 이미지를 자른다.
        ymin = int(a.ymin * 480)
        height = int(a.height * 480)
        xmin = int(a.xmin * 640)
        width = int(a.width * 640)
        # print(ymin, ymin + height, xmin, xmin + width)
        img = image.copy()
        cut = image[ymin: ymin + height, xmin:xmin + width]
        img = cv2.rectangle(img, (xmin, ymin), (xmin + width, ymin + height), (255, 0, 0), 3)
        img = cv2.resize(cut, dsize=(640, 480), interpolation=cv2.INTER_AREA)
        #이미지를 학습시키기 위해서는 이미지들의 크기가 모두 같아야하기 때문에 사이즈를 재조정한다.
if time.time() - start_time >= 0.001:
    cv2.imwrite(data_path, img) #편집된 이미지를 저장한다.
    start_time = time.time()
    count += 1
cv2.putText(img, str(count), (100, 120), cv2.FONT_HERSHEY_COMPLEX, 1, (250, 120, 255), 2)
cv2.imshow("Face detection", img)
if (cv2.waitKey(5) & 0xFF == 27) or count == 200: #이미지가 200장 저장되면 종료
    break
return cap.release()
```

- 모델을 만들기 위한 사진을 촬영하는 모듈

## 03

### 주요 코드

```
# 폴더에 있는 파일 리스트 열기
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)
    print(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))
```

- 사진을 학습하는 모듈

## 03

### 주요 코드

```
try:
    img = cv2.resize(cut, dsize=(640, 480), interpolation=cv2.INTER_AREA)
    result = model.predict(img)
    # 모델과 비교했을 때의 유사도를 설정
    if result[1] < 500:
        # 0~100 표시하기위함
        confidence = int(100 * (1 - (result[1]) / 300))
        confidence = int(result[1])
    # confidence가 클수록 유사도가 낮음
    # 유사도가 조금만 낮아도 화면이 잠기는 것을 방지하기 위해 50 카운트의 한도를 설정함
    if confidence > 40:
        count += 1
        if count >= 50: # 카운트가 50 이상이 되면 캡처하여 이메일로 전송하고, 화면을 잠금
            cv2.imwrite("who.jpg", image1)
            ctypes.windll.user32.LockWorkStation()
            sendmail.warning_message(id)
            count = 0
    else: # 유사도가 정상이면 카운트 0으로 초기화
        count2 += 1
        if count2 >= 10:
            count = 0
    cv2.putText(img, "confidence : " + str(confidence) + ", count : " + str(count), (100, 120),
                cv2.FONT_HERSHEY_COMPLEX, 1, (255, 120, 255), 2)
    print(confidence)
except:
```

- 주기적으로 감시하는 모듈

04

## 시연 영상

사이버 보안 캡스톤 디자인

기능 시연 영상

Youtube 링크: <https://youtu.be/E3wI3jgaPkA>

## 05 기타

### 프로젝트 개선 사항 및 아쉬운 점

- 지속적인 웹캠 사용으로 전력 소모량 ↑  
-> 사용수단을 웹 캠으로 한정 짓는 것이 아닌 외부 캠과 연동하여 사용
- 하나의 계정에 여러 인물의 얼굴 모델 파일을 추가하는 것도 활용 가치가 있을 것으로 추정
- 이미지 비교연산만 따로 수행할 수 있다면 배터리 소모를 많이 줄일 수 있을 것으로 예상됨