“ Your Drive, Our Watch ”

# AUTO AWARE

나를 지키는 드라이빙케어 솔루션

{ TEAM GuardianTech }

장준용 김원진 곽정우 김기재 황수환 변지윤

# CONTENTS

# BACKGROUND

최근 5년간 자동차 도난사건 발생 현황 (시도별 구분) 단위: 건

| 구분 | 2017년 | 2018년 | 2019년 | 2020년 | 2021년 |
|---|---|---|---|---|---|
| 총계 | 2,733 | 2,706 | 2,652 | 2,771 | 2,404 |
| 서울 | 302 | 248 | 293 | 309 | 222 |
| 부산 | 151 | 143 | 151 | 135 | |
| 대구 | 133 | 109 | | | |

오토바이 훔친 중학생들 '낡은 오토바이' 노려 (2023.0

배달 오토바이 절도 기승. 직접 찾(

자전거 도둑 느는데 검거율은 저조..대책은?

YouTube · 존티비JONTV

2024. 6. 5

3초면 범행 끝...귀신도 놀라고 간 자전

YouTube · KBS News

줄어들지 않는 절도

절도 이후의 처리방안 필요성 인식

# SERVICE

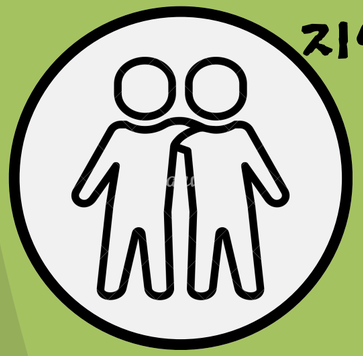
01
운전 중
발생하는 피로

02
감정상태 변화

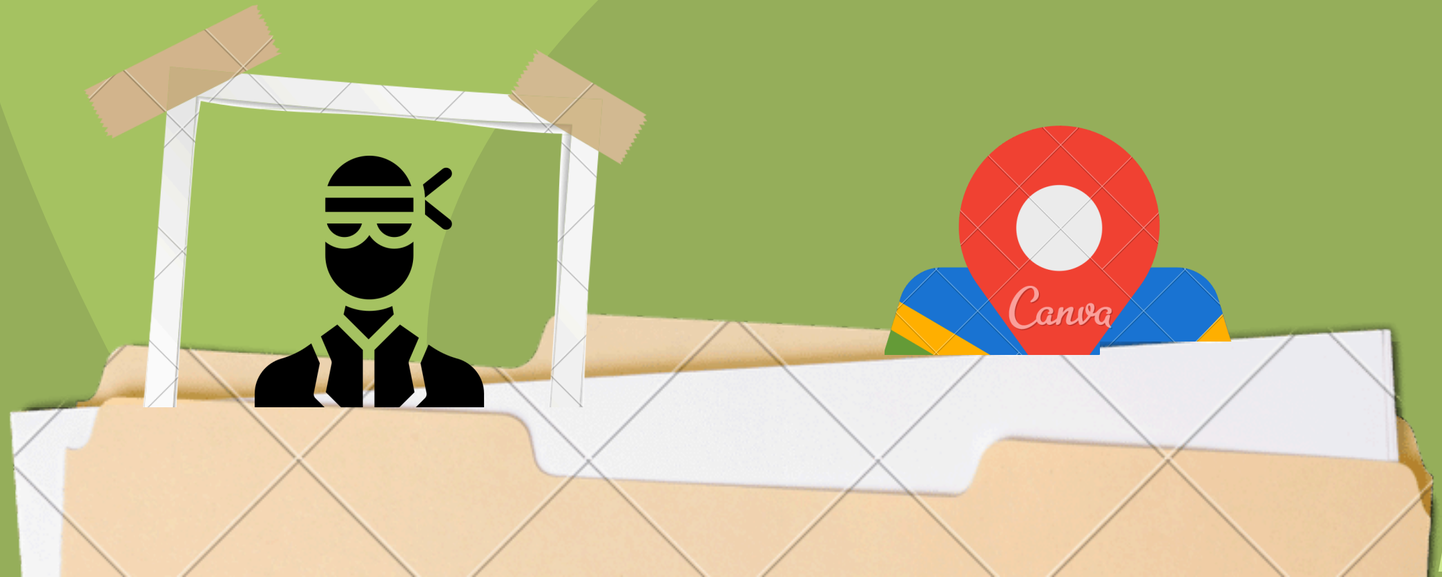
03
주의력 저하

04
**사고**

운전자의 상태 **실시간 모니터링** 필요


# Process1



운전자 인식 및
표정 상태 감지

상태별 알림

# Process2

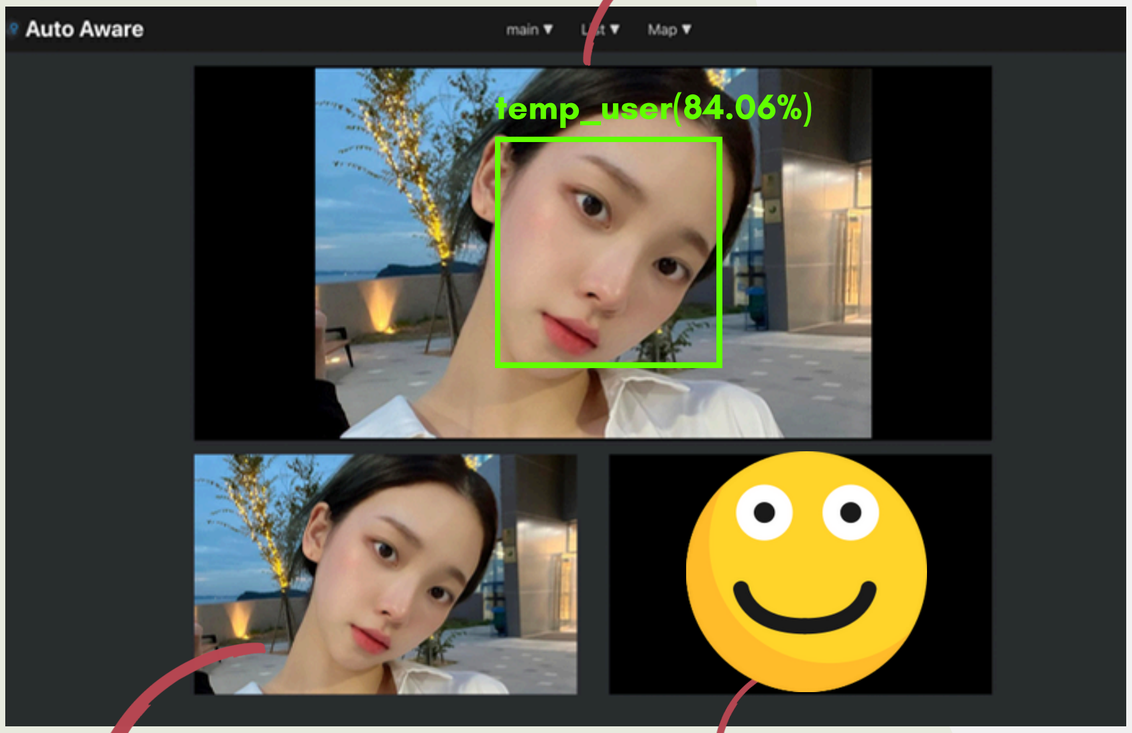
# Information Architecture

# 1 미등록 운전자 감지

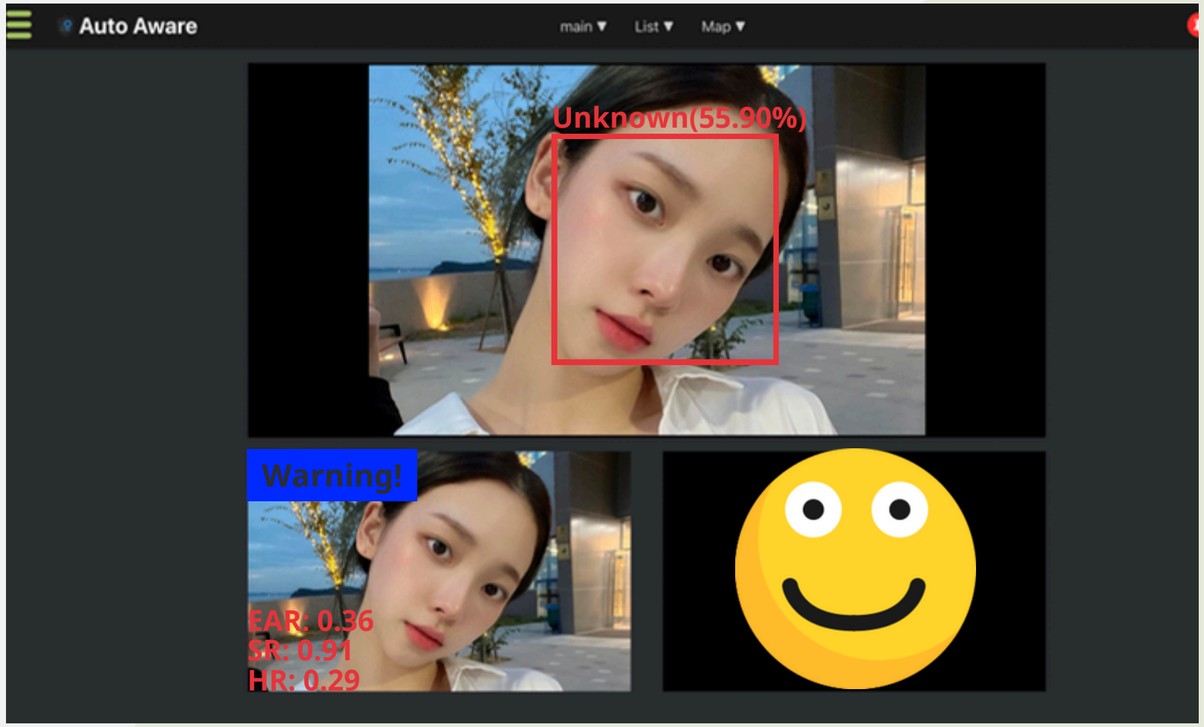
Detection of
unregistered drivers

얼굴객체탐지결과

등록된 운전자일 경우




미등록 운전자




실시간 상태 인식

원본영상

# Algorithm • 얼굴탐지

## Face Detection: Haar Cascade Classifier

```python
face_cascade_name = './haarcascades/haarcascade_frontalface_alt.xml'
face_cascade = cv2.CascadeClassifier()

if not face_cascade.load(cv2.samples.findFile(face_cascade_name)):
    print('--(!) 얼굴 인식 캐스케이드 로딩 오류')
    exit(0)

encoding_file = 'encodings.pickle'
with open(encoding_file, "rb") as f:
    data = pickle.load(f)
```

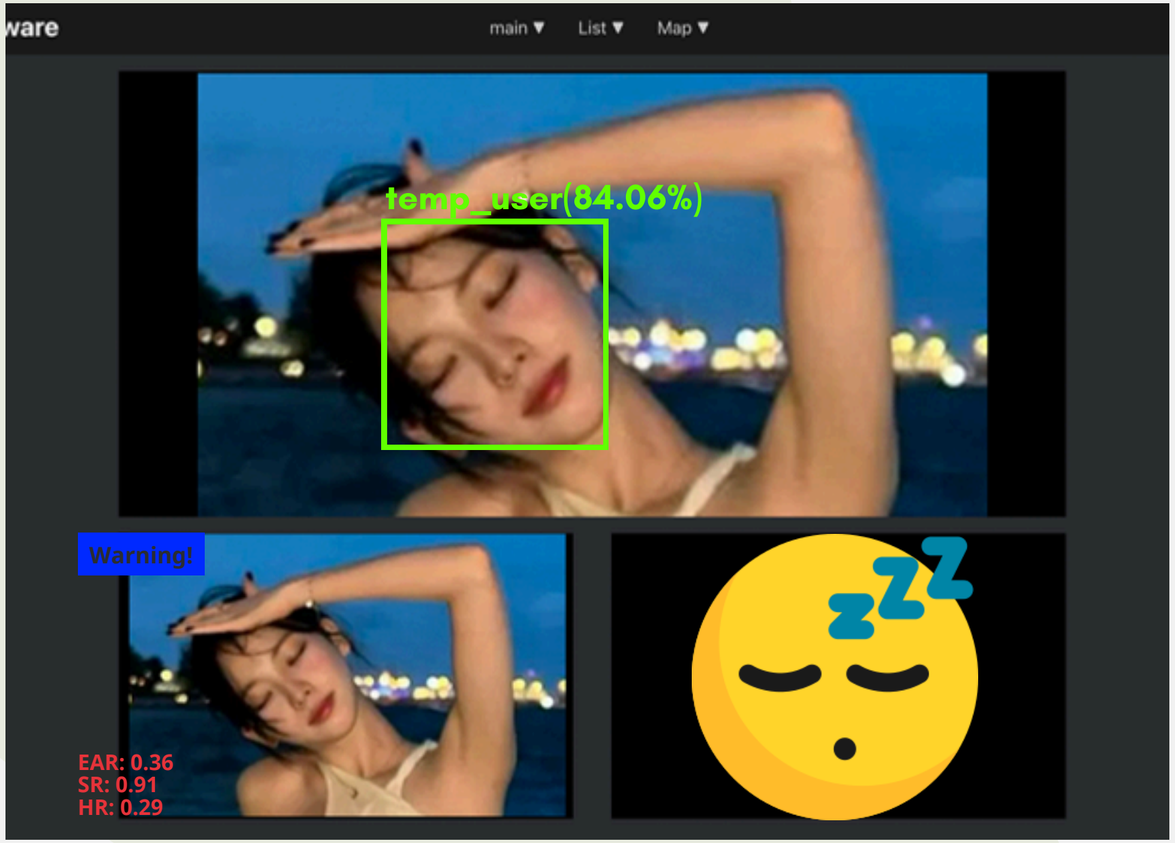
```python
for file_name in filter(lambda f: f.endswith('.jpg'),images_list):
    name = human_name
    image_path = os.path.join(capture_folder, file_name)
    image = cv2.imread(image_path)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    boxes = face_recognition.face_locations(rgb, model='cnn')
    encodings = face_recognition.face_encodings(rgb, boxes)

    knownEncodings.extend(encodings)
    knownNames.extend([name] * len(encodings))
    print(f"인코딩됨 {image_path} 사용자 이름: {name}")

data = {"encodings": knownEncodings, "names": knownNames}
with open(file_path, "wb") as f:
    f.write(pickle.dumps(data))
```

```
{'encodings': [array([-2.46226937e-02,  3.93431373e-02, -7.13933259e-04, -2.36742776e-02,
        -9.91125256e-02,  7.22405175e-03, -4.20636274e-02, -8.59756470e-02,
         1.33779928e-01, -9.53071266e-02,  2.30500430e-01, -6.46997541e-02,
        -2.45042324e-01, -7.57248104e-02, -7.17187226e-02,  2.04386443e-01,
        -1.62169039e-01, -1.37742311e-01, -3.70325595e-02,  3.69541347e-02,
         1.48125798e-01, -6.96968473e-03, -3.83071490e-02,  1.04670106e-02,
        -6.48032427e-02, -3.05492908e-01, -9.97473374e-02, -3.61687206e-02,
        -1.03266677e-02, -8.15404430e-02, -1.46004260e-02,  4.90644202e-02,
        -1.69980556e-01, -5.08730970e-02,  4.26725633e-02,  3.66324410e-02,
        -2.27578599e-02, -7.80399442e-02,  2.12569609e-01,  1.33192306e-02,
        -2.26474628e-01,  2.42004544e-02,  4.05550525e-02,  2.09843367e-01,
         1.07068822e-01,  2.74427664e-02,  6.55417517e-03, -1.54786259e-01,
         1.10453337e-01, -1.77512035e-01,  2.84467172e-02,  1.44935548e-01,
         1.07078850e-01,  7.36137927e-02,  4.52821888e-03, -1.04093969e-01,
         3.18571180e-02,  1.00810677e-01, -1.02199927e-01,  1.55573897e-02,
         8.16121176e-02, -9.51032043e-02, -4.38000485e-02, -1.23422325e-01,
         2.09867209e-01,  1.93834342e-02, -9.99949053e-02, -2.11380035e-01,
         1.31258175e-01, -7.65423104e-02, -1.54948458e-01,  4.97719795e-02,
        -1.39855340e-01, -1.46189749e-01, -3.21117818e-01, -5.62776439e-03,
         4.11284417e-01,  5.98733798e-02, -2.48641551e-01,  6.74213395e-02,
        -3.78577225e-02, -6.97906688e-03,  1.42613575e-01,  1.51719183e-01,
         3.50821391e-02,  4.03708629e-02, -1.00111239e-01, -9.64969024e-03,
         2.24868551e-01, -1.02979876e-01, -5.55116907e-02,  1.95030227e-01,
         4.20343876e-03,  1.31688863e-01,  3.85113508e-02,  7.35057890e-03,
        -2.46686023e-03,  3.62375230e-02, -1.30908296e-01, -2.56371759e-02,
...
        -0.13104156, -0.04521126,  0.12028705, -0.23572333,  0.18822911,
         0.13584715,  0.07918852,  0.04345482,  0.13641228,  0.11334223,
        -0.0044436 , -0.01905295, -0.25227633, -0.03029839,  0.12047943,
         0.02913003,  0.03513784, -0.01587448])],
 'names': ['Kwak', 'Kwak', 'Kwak', 'Kwak']}
```

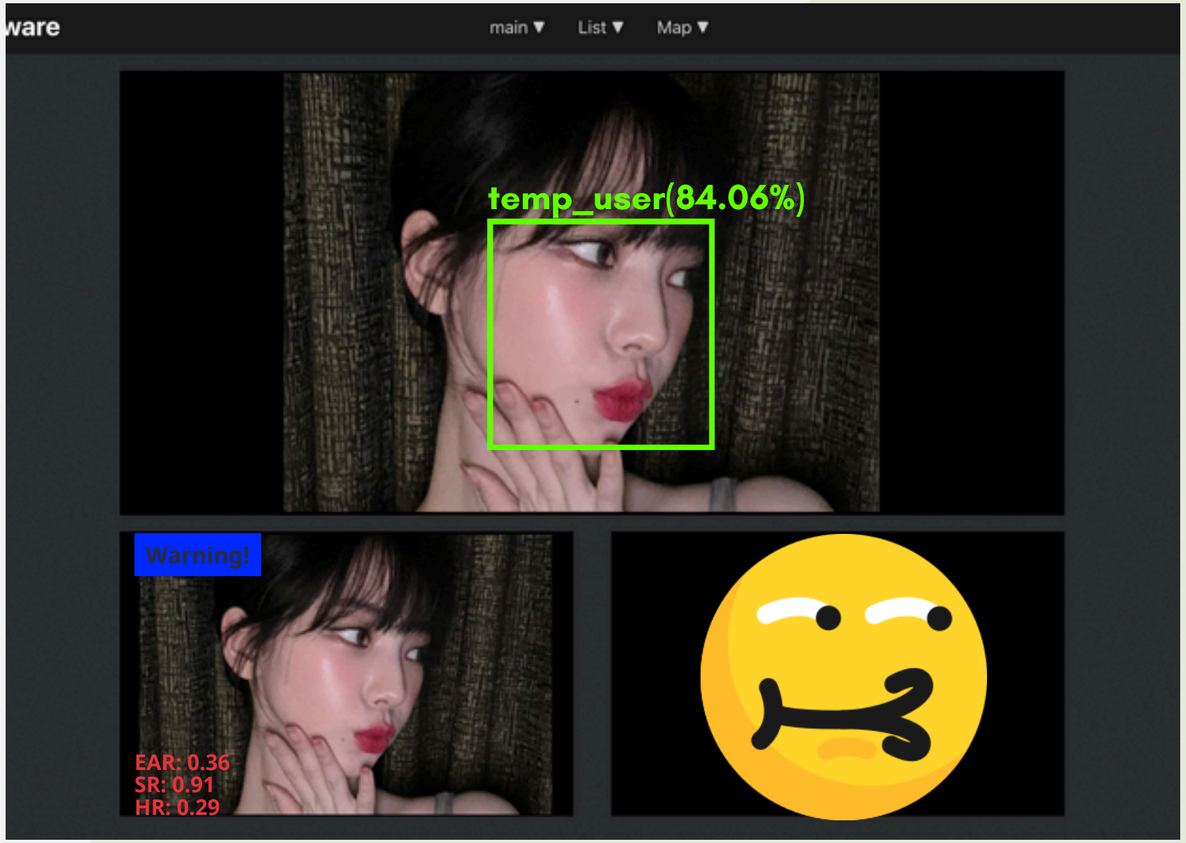
# 2 운전자 상태 인식

Real-time
facial recognition

## CASE 2. 졸음운전 경고



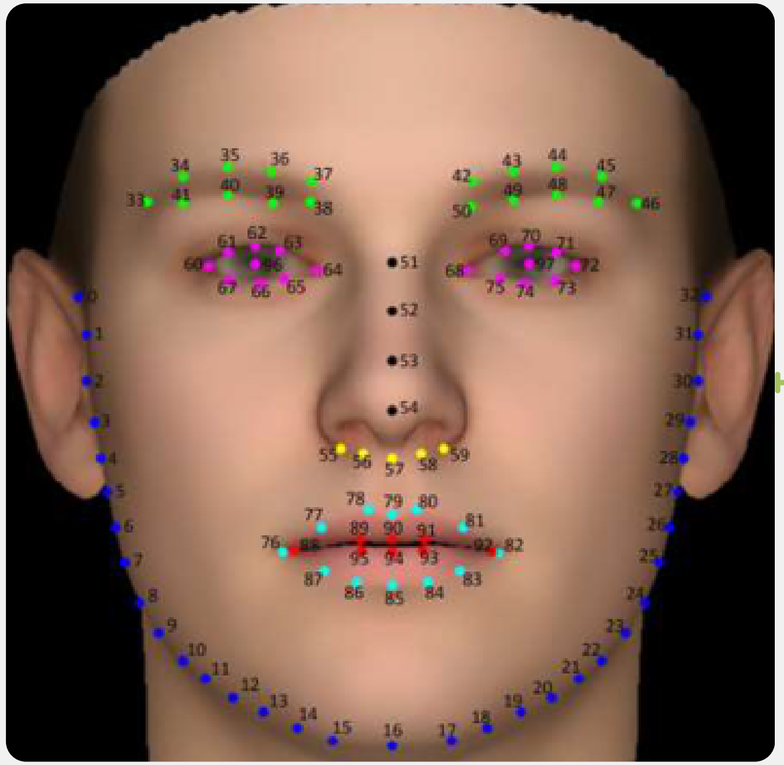

## CASE 3. 전방 주시 경고



# Algorithm • 졸음 및 전방미주시 탐지

## 🤖 주요메커니즘 🤖

- 얼굴 탐지 및 특징점 추출
- 얼굴 메트릭 계산(EAR, HR, SR)
- 눈 깜빡임 감지
- 머리 움직임 감지
- 최종상태 확인
- 경고 상태 업데이트)



```python
# 눈 종횡비(EAR) 계산 함수
def ear(pts):
    A = dist.euclidean(pts[1], pts[7])
    B = dist.euclidean(pts[3], pts[5])
    C = dist.euclidean(pts[0], pts[4])
    return (A + B) / (2.0 * C)

# 머리 회전율 계산 함수
def head_rate(pts):
    A = dist.euclidean(pts[51], pts[54])
    B = dist.euclidean(pts[54], pts[16])
    return A / B

# 시선 이동률 계산 함수
def stir_rate(pts):
    A = dist.euclidean(pts[4], pts[54])
    B = dist.euclidean(pts[28], pts[54])
    return A / B
```

**Face Detection 모델: RetinaFace**

**Face Landmark Detection 모델: PFLD**


!
PREVIEW
시연

# Vision

자동차 제조사와의
파트너십

상용차량, 운송업체 등을
대상으로 확장

표정인식 다양화 ->
고의사고, 보험사기 예방

데이터 분석을 통해
맞춤형 광고 제공

# Trouble Shooting

## case 1.

서버에서 웹캠을 열어 프레임단위로
리액트에 보내는데, 연산 처리등
프레임 간의 사이에서
처리할 것이 많아
웹캠이 다운되는 현상이 발생했다.

처리부분을 최대한 간소화하고,
연산처리 부분은 비동기식으로 처리하여
속도를 향상시켰더니, 정상 작동 되었다.

## case 2.

기능적인 부분을 파이썬으로 만든 다음,
그 기능을 프론트엔드와 백엔드가
연결되게끔 코드를 재구성해봤는데
작동이 잘 안되고 계속적인 오류가 존재했다.

기존에 파이썬으로 만든 기능들의
코드를 함수화 시켜 서버에 넣었더니
손쉽게 해결이 되었다.

## case 3.

사용자의 이미지를 등록하여
학습하기 위한 과정에서
로컬에 저장 후 가져와서 학습을 시키는 방법을 적용시켰지만 리스트를 확인할 때
데이터 베이스에서 유저의 아이디를 확인하고 가져와야 하는 번거로움이 있었다.

unknown 이미지가 캡쳐될 때
이미지를 바이너리 데이터로 변환 후
데이터 베이스에 userid와 함께
저장하여 해결하였다.


!
Q&A

# Thank You :)

Korea IT Academy | KDT

---

**{ TEAM GuardianTech }**

장준용 김원진 곽정우 김기재 황수환 변지윤