

비정형 데이터를 위한 **개인정보 비식별화 플랫폼**

목차

- 01 프로젝트 배경 및 개요
- 02 시스템 아키텍처
- 03 주요 기능
- 04 핵심 알고리즘
- 05 보안 기능
- 06 프로젝트 차별성
- 07 활용분야
- 08 시연
- 09 향후 발전 방향
- 10 기대효과



IT/과학

"행인 찍히면 밤새 모자이크"...규제에 막힌 자율차 AI

한국경제 원문 | 기사전송 2023-04-20 18:25 최종수정 2023-04-27 20:00

AI챗으로 요약

↑ ↓ ⌂ ⌃ ⌄

얼굴 포함된 AI 학습 데이터
개인정보보호법 위반 소지

"자율주행차 개발하려 들어온 인력이 포토샵만
만지고 있으니 줄줄이 퇴사자가 나올 수밖에
요."

자율주행 스타트업 언맨드솔루션 개발진은 요
새 도로 주행 과정에서 찍힌 사람 얼굴을 일일
이 블러(흐리게 하는 필터) 처리하느라 밤을 새
우기 일쑤다. 회사 관계자는 "원본 영상을 그대
로 인공지능(AI) 학습 프로그램에 넣으면 법 위
반 소지가 있다"며 "중국에 ⑤ 계획한 가명처리 방법 및 수준에 따라 실제 가명처리를 수행하였는지 확인
판"이라고 토로했다.

주요 비정형데이터 가명처리 수행 결과 (예시)

연번	항목명	가명처리 전	가명처리 후
2	사람 얼굴		
4	차량 번호판		

※ 이해를 돋기 위해 해당 부분을 확대하여 표시하였음

프로젝트 배경

자율주행 기술 발전과 학습용 데이터 활용의 어려움

"행인 찍히면 밤새 모자이크"...규제에 막힌 자율차 AI

한국경제 원문 | 기사전송 2023-04-20 18:25 최종수정 2023-04-27 20:00

AI챗으로 요

자율주행 스타트업 언맨드솔루션 개발진은 요새 도로 주행 과정에서 찍힌 사람 얼굴을 일일이 블러(흐리게 하는 필터) 처리하느라 밤을 새우기 일쑤다. 회사 관계자는 "원본 영상을 그대로 인공지능(AI) 학습 프로그램에 넣으면 법 위반 소지가 있다"며 "중국에 고객을 다 뱃길 판"이라고 토로했다.

자율주행 스타트업 **언맨드솔루션**

참여 인원 1300여명

채널 내부에 전국 70여개 대학 이름 단 단체대화방

© AnonyData ALL RIGHTS RESERVED.
각 단체대화방에 피해자 신상 전송

서로 아는 피해자 발견 뒤 개인 메시지
주고받으며 불법합성물 제작



3 유료 불법합성물 제작 채널

참여 인원 22만여명

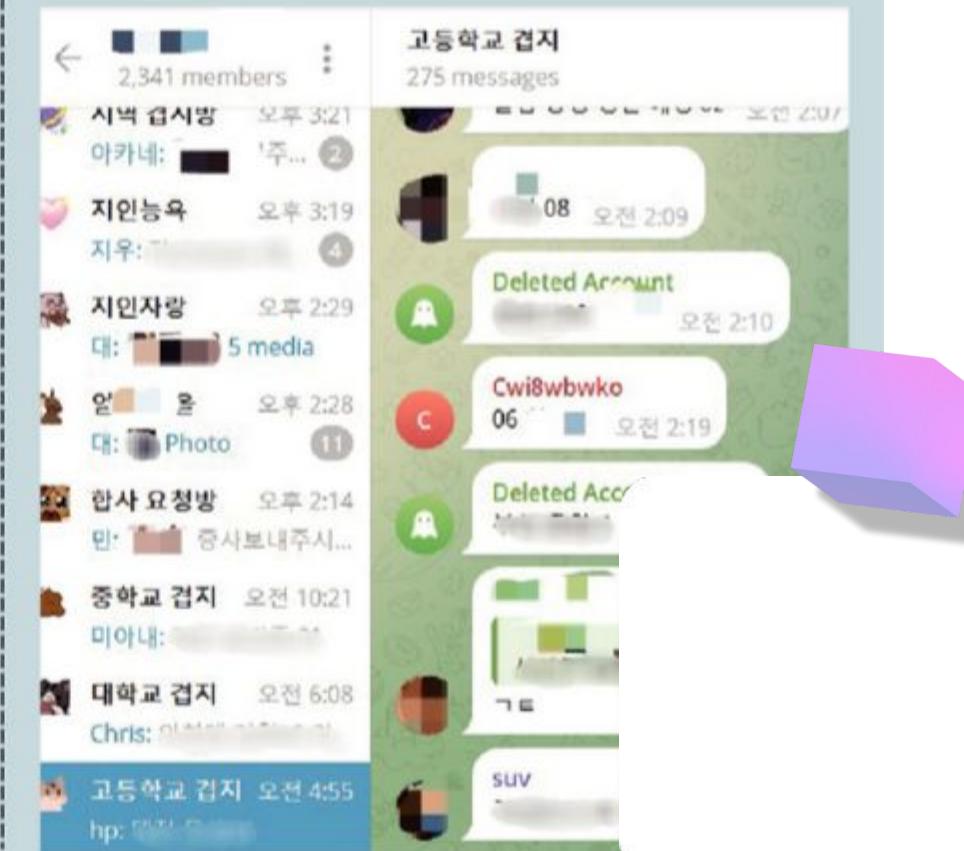
봇 프로그램 통해 불법합성물 제작. 3개 이상
제작 시 유료 전환 또는 친구 초대 요청



참여 인원 2340여명

채널 내부에 '중·고교 겹지인방', '지인능욕방',
'합사(합성사진)요청방' 등 단체대화방

불법합성물 제작·가공 뒤 성희롱



4 기타

■ 링크 공유방: 특정 피해자 1명의 불법합성
반복적으로 올리는 링크 공유방

■ 소수정예 지인방: 100~200명 인원으로
면접 보고 방 참여 허용

프로젝트 배경

개인정보 보호 필요성 증가

사회 사회일반

[단독] 'OOO 능욕방' 딥페이크,
겹지인 노렸다...지역별·대학별·
미성년까지

이들은 우선 ①‘겹지인방’이라고 불리는 텔레그램 채널을 통해 지역이나 대학교로 중심으로 모이고 ②특정 여성을 동시에 아는지 확인하고, 함께 아는 여성이 있으면 ③그가 소셜미디어에 올린 평범한 사진을 공유한 뒤 이를 악용해 불법합성물을 제작하는 것으로 드러났다.

평소 에스엔에스를 즐겨 이용하는 ㄱ(24)씨는 “계정을 비공개로 돌려도 내 주변 사람들이 언제든 내 사진으로 범죄를 저지를 수 있다는 생각에 소름이 끼친다”고 말했다.

누구나 쉽고, 안전하고, 책임감 있게
AI를 활용할 수 있다면 좋겠다

프로젝트 목표

1. 윤리적이고 안전한 데이터 사용

: 개인정보 보호와 법적 준수를 보장하는 가명처리된 데이터 제공

2. 진입 장벽 낮추기

: 복잡한 데이터 전처리 과정을 자동화하여 초보자도 쉽게 AI 학습 시작 가능

3. 신뢰성 있는 학습 환경 제공

: 불법적이거나 유해한 콘텐츠를 배제한 신뢰할 수 있는 데이터셋 제공

4. 실용적 AI 응용 확대

: 일상생활과 다양한 산업 분야에서 AI 기술 활용 증진

프로젝트 정의

- 이미지, 영상, PDF 내
개인정보 자동 탐지 및 비식별화 웹 서비스
- 비식별 데이터의 안전한 저장, 관리, 공유를
위한 통합 플랫폼
- 사용자 친화적 인터페이스로
전문가/비전문가 모두 사용 가능

주요기능

1. 비식별화 기능

- 얼굴 및 차량 번호판 자동탐지
- 다양한 마스킹 옵션(완전 마스킹, 픽셀화, 블러링)
- PDF 암호화 및 보안 열람 기능

2. 데이터 플랫폼 기능

- 비식별 처리된 데이터의 중앙 집중식 저장 및 관리
- 메타데이터 추가 및 검색 기능

프로젝트 개요

Technology

- 백엔드 : Django, Python
- 프론트엔드 : React Next
- AI : YOLO v5
- 클라우드 : AWS, EC2
- 추가 라이브러리 :
OpenCV, PyMuPDF,
Cryptography

Development Environment

- 개발 도구 : Visual Studio Code, PyCharm
- 버전 관리 : Git Hub
- 컨테이너화 : Docker

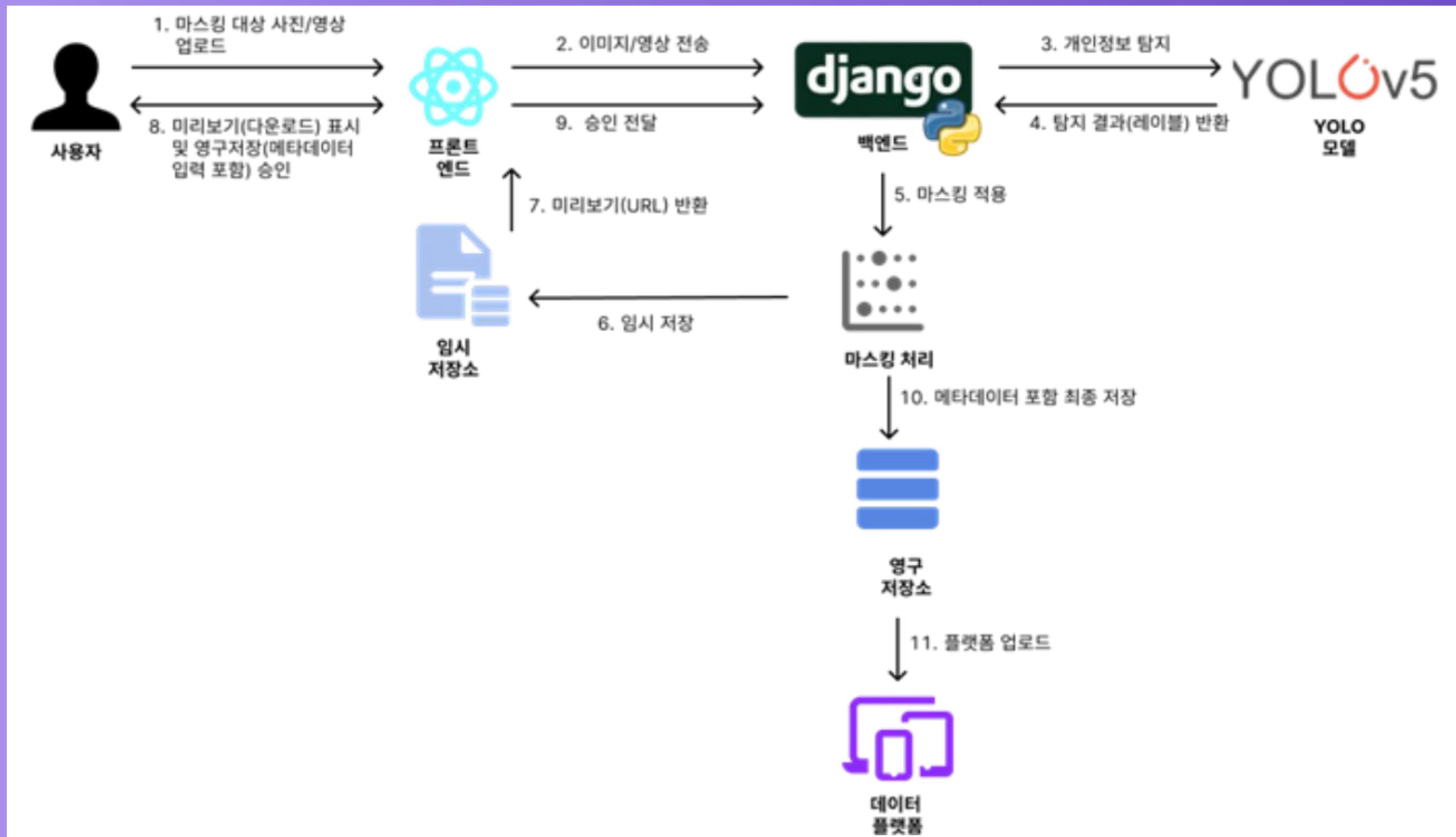
Target User

- AI 연구자 및 개발자
- 데이터 과학자
- 개인정보 보호 담당자
- AI 학습에 관심 있는 일반 사용자

Differentiation

- 쉬운 접근성
- 다양한 파일 형식 지원(이미지, 영상, PDF)
- 강력한 보안 기능 탑재

System Architecture



System Architecture



남들과는 다른 **플랫폼** 가능

이미지/영상 마스킹



PDF 마스킹



데이터 플랫폼



PDF 공유 및 배포



이미지 마스킹



Bandissoft.com
이미지 마스킹 대시보드

The screenshot shows the 'Image Masking Dashboard' from Bandissoft.com. It features a central image of a person wearing glasses. On the left, there's a file selection area with a placeholder 'Drop files here or click to select'. On the right, there are several configuration options: 'Masking Options' (checkboxes for ' 얼굴' and ' 번호판'), 'Masking Method' (radio buttons for 'Blurring', 'Masking' (which is selected), and 'Mosaic'), and a 'Masking Intensity' slider set at 18%. A large purple button at the bottom right says 'Start Processing All Files'.

기능
01

YOLO v5 모델 사용

기능
02

다양한 마스킹 옵션

- 마스킹/픽셀화/블러링
- 마스킹 강도 조절 가능.

기능
03

S3 자동 업로드 및 URL 반환

비디오 마스킹

Bandisoft.com
비디오 마스킹 대시보드

파일 목록

비디오 파일을 여기에 놓거나 클릭하여 선택하세요

비디오 선택

최근 추가된 비디오:



0:00 / 0:07

파일명	상태	작업
myface.mp4	pending	삭제

마스킹 옵션

마스킹 대상

얼굴 번호판

마스킹 방법

블러링 마스킹 모자이크

마스킹 강도

50%

모든 파일 처리 시작

기능
01

동영상 내 얼굴 및 번호판 자동 탐지

기능
02

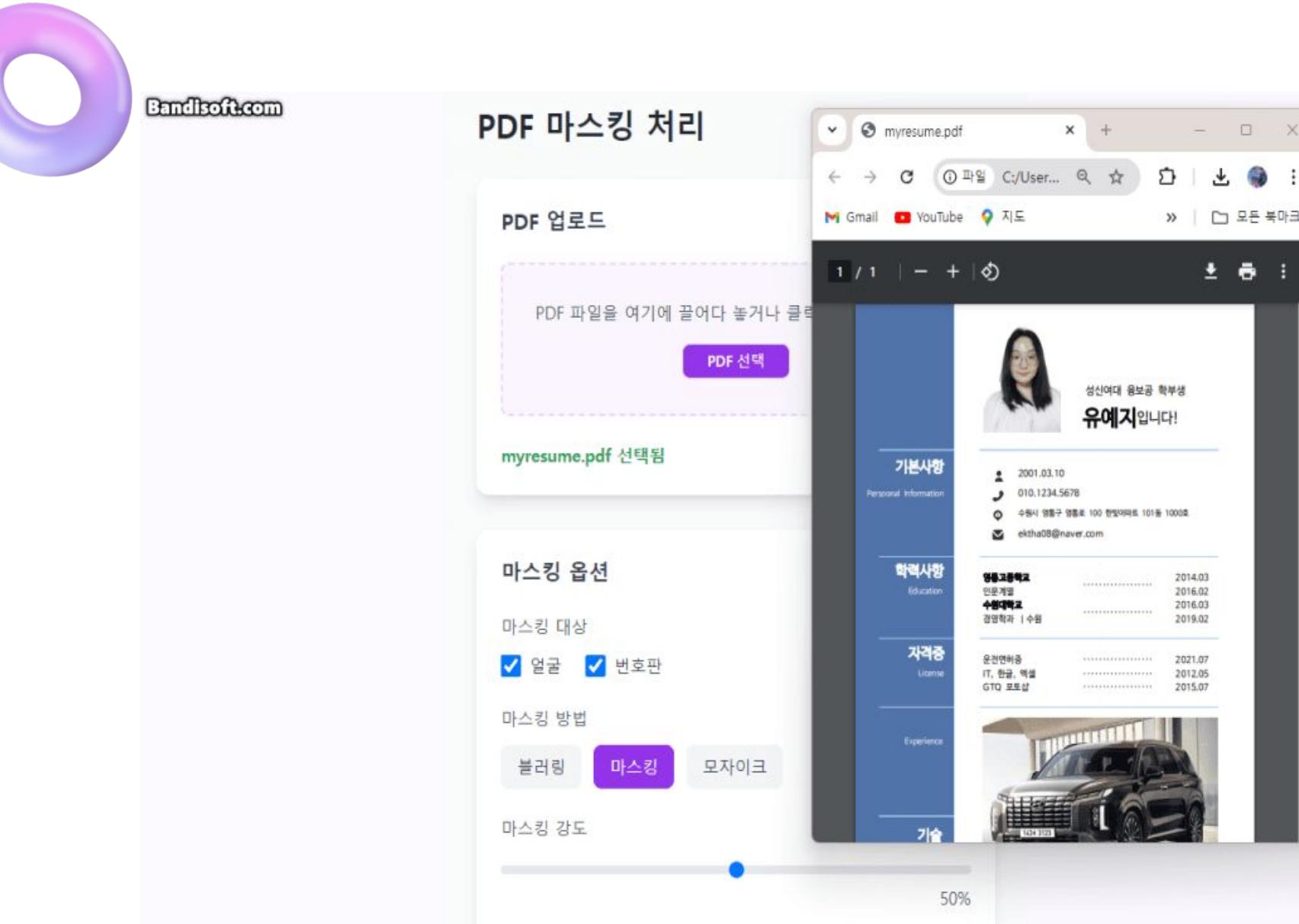
이미지와 동일한 마스킹 옵션 제공

- 마스킹/픽셀화/블러링
- 마스킹 강도 조절 가능.
- 처리 이미지 S3 자동 업로드 및 URL 반환

기능
03

프레임별 처리 후 비디오 재조합

PDF 마스킹

기능
01

PDF 내 이미지 변환&처리

기능
02

이미지 마스킹 처리 후 PDF로 반환

- 마스킹/픽셀화/블러링
- 마스킹 강도 조절 가능.

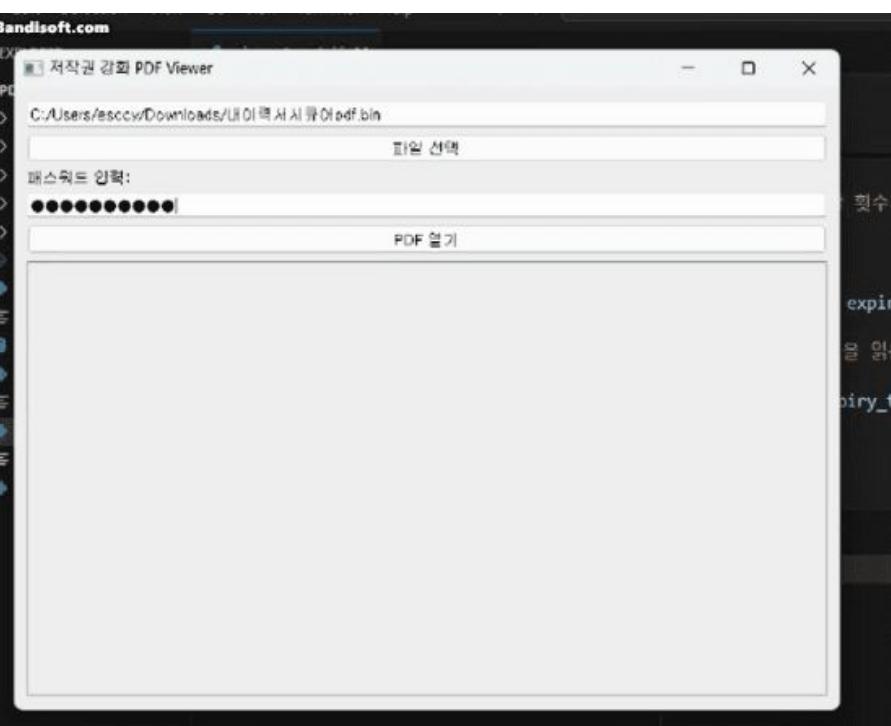
기능
03

처리된 PDF S3 업로드 및 URL 반환

파일 암호화

기능
01

PDF 암호화
(Fernet 암호화 사용)



기능
02

만료 시간, 최대 조회 횟수 설정

- 만료시간 초과, 최대 조회 횟수 초과시 파일 파괴.

기능
03

비밀키를 이용한 보안 강화

- 비밀키 설정 횟수 이상 틀리는 경우, 파일 파괴

데이터 플랫폼

Bandisoft.com



기능
01

세분화된 검색과 필터 기능

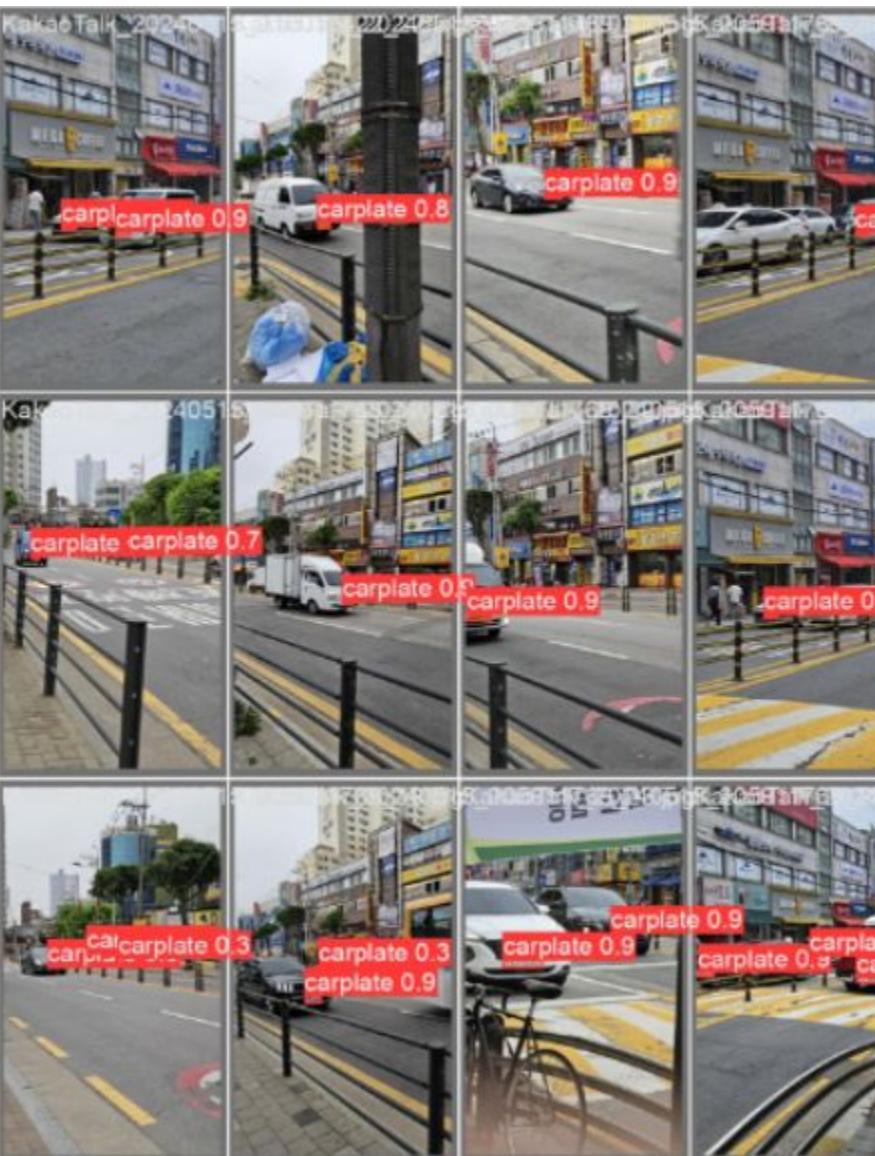
기능
02

다중 선택 일괄 처리

기능
03

데이터 분류 및 태그 지정

핵심 알고리즘 YOLO v5



개요

- object detection 계열 딥러닝 알고리즘
- 'You Only Look Once'의 약자
: 이미지를 한 번만 보고 여러 객체를 동시에 탐지

원리

- 이미지를 그리드로 분할
- 각 그리드 셀에서 객체의 경계 상자와 클래스 확률을 예측
- 단일 신경망으로 전체 탐지 과정을 수행

특징

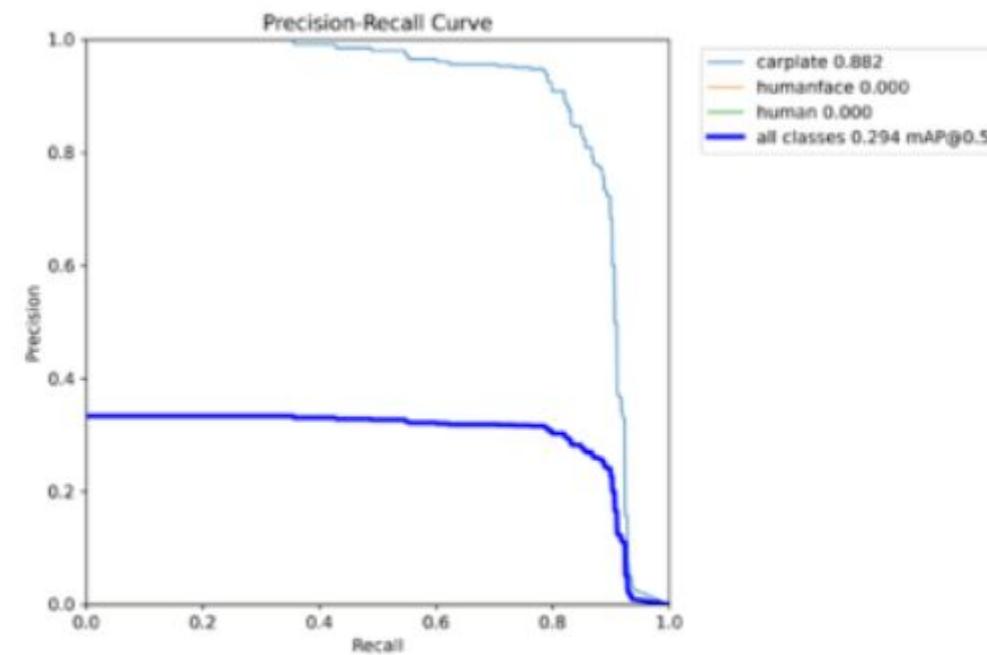
- 빠른 처리 속도: 실시간 비디오 처리 가능
- 높은 정확도: 최신 버전으로 성능 개선
- 경량화: 모바일 및 임베디드 시스템에 적합

핵심 알고리즘 YOLO v5

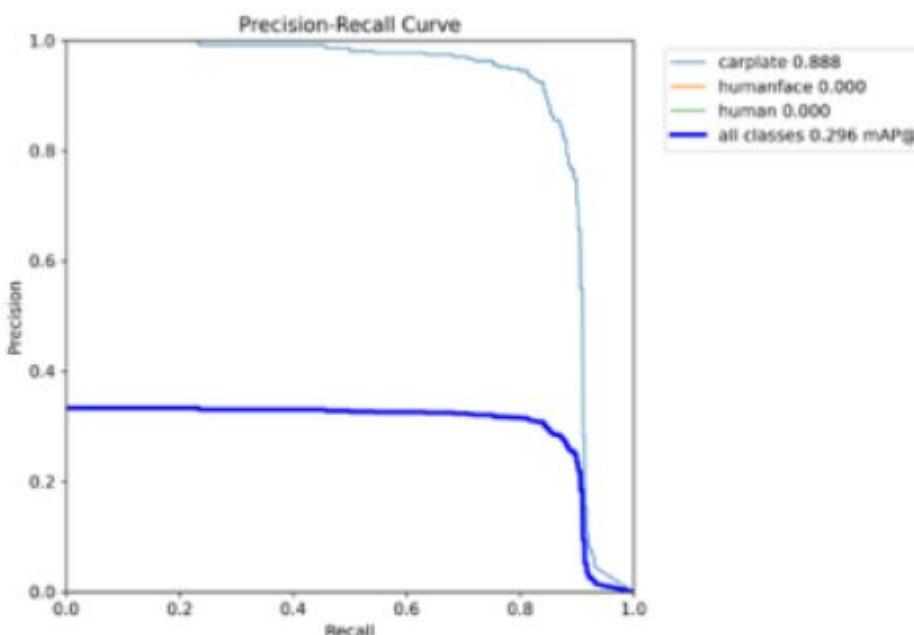


적용

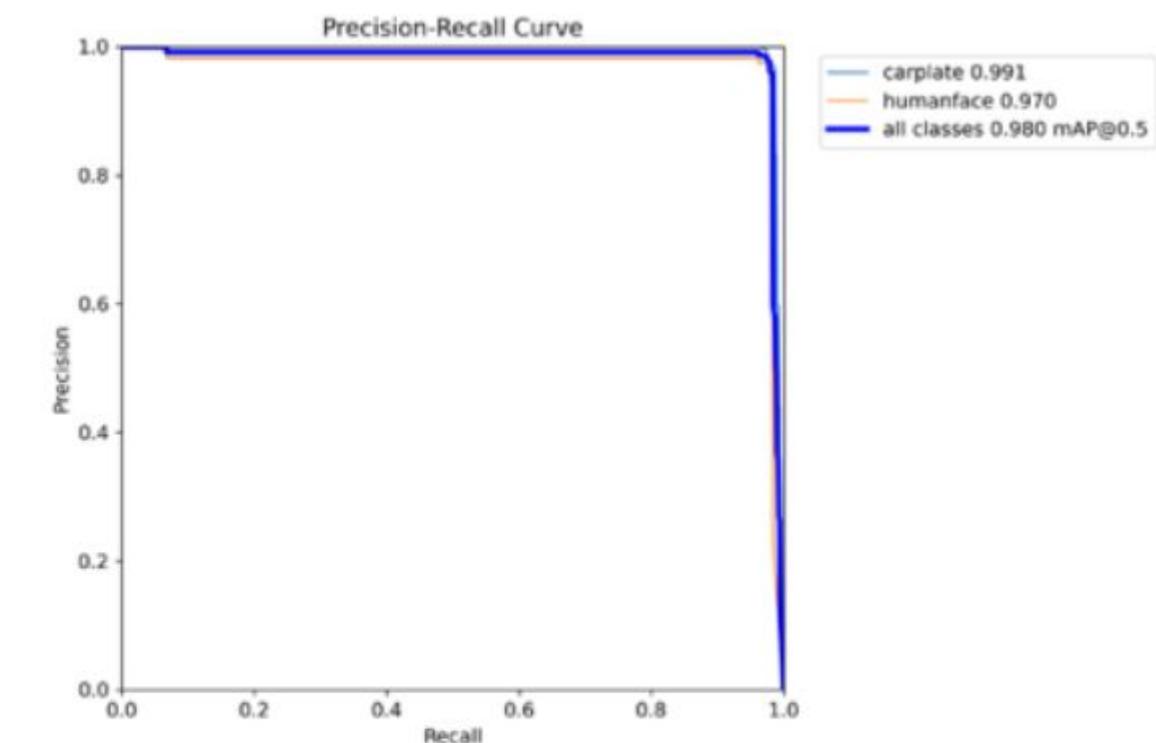
- 얼굴 및 번호판 자동 탐지에 활용
- 1만 장의 얼굴 이미지 학습
- 3천 장의 차량 번호판 이미지로 학습



초기 데이터셋 &
초기 모델



초기 데이터셋 &
신규학습 모델



신규 데이터셋 &
신규학습 모델

지표

- 차량번호판 정확도 **0.991**
- 사람 얼굴 정확도 **0.970**

핵심 알고리즘

마스킹 알고리즘

```

for image in images:
    # 이미지 읽기
    img_array = np.frombuffer(image.read(), np.uint8)
    img = cv2.imdecode(img_array, cv2.IMREAD_COLOR)

    # YOLOv5로 객체 탐지
    results = model(img)

    # 탐지된 객체에 대해 마스킹 적용
    for xyxy, conf, cls in results.xyxy[0].cpu().numpy():
        x1, y1, x2, y2 = map(int, xyxy)
        class_id = int(cls)
        if (class_id == 0 and 'licensePlate' in masking_targets) or \
            (class_id == 1 and 'face' in masking_targets):
            roi = img[y1:y2, x1:x2]
            if masking_method == 'masking':
                print('masking')
                roi[:] = (0, 0, 0)
            elif masking_method == 'pixelation':
                print('pixelation')
                h, w = roi.shape[:2]
                roi = cv2.resize(roi, (masking_intensity, masking_intensity),
interpolation=cv2.INTER_LINEAR)
                roi = cv2.resize(roi, (w, h), interpolation=cv2.INTER_NEAREST)
            elif masking_method == 'blurring':
                print('blurring')
                kernel_size = max(1, masking_intensity // 10) * 2 + 1
                roi = cv2.GaussianBlur(roi, (kernel_size, kernel_size), 0)
            img[y1:y2, x1:x2] = roi

```

방법

- 완전 마스킹: 대상 영역을 검은색으로 완전히 가림
- 픽셀화(모자이크): 대상 영역을 저해상도로 변환
- 블러링: 가우시안 블러를 적용하여 대상 영역을 흐리게 처리

대상

- 얼굴 (Face)
- 차량 번호판 (License Plate)

옵션

- 마스킹 방법 선택
- 마스킹 강도 조절 (1-100)
- 마스킹 대상 선택 (얼굴, 번호판, or 둘 다)

핵심 알고리즘

마스킹 알고리즘

```

for image in images:
    # 이미지 읽기
    img_array = np.frombuffer(image.read(), np.uint8)
    img = cv2.imdecode(img_array, cv2.IMREAD_COLOR)

    # YOLOv5로 객체 탐지
    results = model(img)

    # 탐지된 객체에 대해 마스킹 적용
    for *xyxy, conf, cls in results.xyxy[0].cpu().numpy():
        x1, y1, x2, y2 = map(int, xyxy)
        class_id = int(cls)
        if (class_id == 0 and 'licensePlate' in masking_targets) or \
            (class_id == 1 and 'face' in masking_targets):
            roi = img[y1:y2, x1:x2]
            if masking_method == 'masking':
                print('masking')
                roi[:] = (0, 0, 0)
            elif masking_method == 'pixelation':
                print('pixelation')
                h, w = roi.shape[:2]
                roi = cv2.resize(roi, (masking_intensity, masking_intensity),
interpolation=cv2.INTER_LINEAR)
                roi = cv2.resize(roi, (w, h), interpolation=cv2.INTER_NEAREST)
            elif masking_method == 'blurring':
                print('blurring')
                kernel_size = max(1, masking_intensity // 10) * 2 + 1
                roi = cv2.GaussianBlur(roi, (kernel_size, kernel_size), 0)
            img[y1:y2, x1:x2] = roi

```

과정

1. YOLO v5로 객체 탐지
2. 탐지된 객체에 선택된 마스킹 방법 적용
3. 처리된 이미지/비디오 생성

특징

- 실시간 처리 가능
- 다중 객체 동시 마스킹
- 이미지 및 비디오 모두 지원

성능

- 처리 속도: 평균 0.2초/프레임 (GPU 사용 시)
- 마스킹 정확도: 객체 탐지 정확도에 비례

핵심 알고리즘

Fernet 대칭키 암호화

```
from cryptography.fernet import Fernet
pdf_content = pdf_file.read()

# Encrypt PDF content
fernet_key = Fernet.generate_key()
fernet = Fernet(fernet_key)
encrypted_content = fernet.encrypt(pdf_content)

# Create metadata
metadata = f"{expiry_time}|{fernet_key.decode()}|{max_views}|{secret_key}".encode()

encrypted_metadata = base64.b64encode(metadata)

# Combine encrypted content and metadata
final_content = encrypted_metadata + b'\n' + encrypted_content
```

암호화

설정

과정

- Fernet 대칭 키 암호화 사용
(AES128 in CBC mode with PKCS7 padding)
- 암호화 키: 무작위 생성된 Fernet 키

- 만료 시간 (Expiry Time)
- 최대 조회 횟수 (Max Views)
- 비밀 키 (Secret Key)
- Base64 인코딩으로 메타데이터 보호

1. PDF 내용 암호화
2. 메타데이터 생성 및 암호화된 내용과 결합
3. 암호화된 파일은 .bin 확장자로 저장

보안 기능

```
...
# jwt 토큰 인증 기능
refresh = RefreshToken.for_user(user)
return Response({
    'refresh': str(refresh),
    'access': str(refresh.access_token),
})

# csrf 보호 기능 데코레이터
@csrf_exempt

# 권한 기반 접근 제어 데코레이터
@permission_classes([AllowAny])
@permission_classes([IsAuthenticated])

# Fernet 암호화를 이용한 PDF 내용 암호화
fernet_key = Fernet.generate_key()
fernet = Fernet(fernet_key)
encrypted_content = fernet.encrypt(pdf_content)

# 만료 시간 설정 및 경종
expiry_time = int(time.time()) + (expiry_minutes * 60)
if current_time > expiry_time:
    self.self_destruct(file_path)

# 비밀번호 유통 및 로그인 시도 제한
if entered_password != correct_password:
    self.attempts += 1
    if self.attempts >= 3:
        self.self_destruct(file_path)

# 파일 자동 파기 메커니즘
def self_destruct(self, file_path):
    with open(file_path, 'wb') as f:
        f.write(b'0' * 1000000) # Overwrite with zeros
    os.remove(file_path)
```

- 1 • CSRF 보호
- 2 • S3 보안 연동 (AWS 인증 정보 사용)
- 3 • REST framework의 권한 클래스 사용 (AllowAny, IsAuthenticated)
- 4 • PDF 보안 및 접근 제어 시스템
 - DRM 기반 접근 제어
 - 3단계 인증 프로세스 (비밀번호, 만료 시간, 조회 횟수 제한)
 - 데이터 유출 방지 메커니즘 (자동 파기 기능)
- 5 • JWT(JSON Web Token) 기반 인증

남들과 다른 차별점



웹 기반
사용자 친화적
인터페이스



대용량 파일 처리
및 일괄 마스킹



다양한
데이터 유형 지원



강력한
보안기능

향후 발전 방향 의료 데이터 가명처리

⑤ 계획한 가명처리 방법 및 수준에 따라 실제 가명처리를 수행하였는지 확인

연번	항목명	세부 항목	가명처리 전	가명처리 후
1	구강 촬영데이터	⑤ 상악치		
		⑥ 하악치		

향후 발전 방향 Fawkes 오픈소스 적용

Example

내 이미지가 AI 학습이나 영상에 활용되지
않도록

```
fawkes -d ./imgs --mode low
```

or

```
python3 protection.py -d ./imgs --mode low
```

마스킹 옵션에 '[AI 학습 방해]' 옵션을 추가

Tips

- The perturbation generation takes ~60 seconds per image on a CPU machine, and it would be much faster on a GPU machine. Use `batch-size=1` on CPU and `batch-size>1` on GPUs.
- Run on GPU. The current Fawkes package and binary does not support GPU. To use GPU, you need to clone this repo, install the required packages in `setup.py`, and replace tensorflow with tensorflow-gpu. Then you can run Fawkes by `python3 fawkes/protection.py [args]`.



Original

Min

Low

Mid

향후 발전 방향 특정 영역 마스킹 제외 기능

```
...  
  
def process_image_with_exclusion(image,  
masking_method, masking_targets,  
masking_intensity, exclusion_areas):  
    # 원본 이미지 복사  
    original = image.copy()  
    mask = np.ones(image.shape[:2],  
    dtype=np.uint8) * 255  
  
    # 제외 영역 마스크 생성  
    for area in exclusion_areas:  
        x1, y1, x2, y2 = map(int, area)  
        mask[y1:y2, x1:x2] = 0  
  
    # YOLO 모델로 객체 탐지  
    results = model(image)
```

향후 발전 방향 활용 분야

- 자율주행 및 교통 데이터
- 의료 데이터 관리
- 공공 행정 문서 처리
- 기업 문서 보안
- SNS 내 개인 사진 관리 및 비식별화
업로드
- API 제공을 통한 서비스 확장

기대 효과 한 눈에 보기

EXPECTATION 01.

AI 학습 데이터의
안전한 활용 촉진
자율주행 및 교통
분야 혁신 지원
2056. 01. 01
(법적 규제 준수
용이성)

EXPECTATION 02.

개인정보 보호 강화
사회적 신뢰 구축

2056. 04. 01

EXPECTATION 03.

데이터 경제 활성화

협업 및 연구 효율성
증대

다양한 산업 분야로의
확장 가능성

2056. 06. 01

EXPECTATION 04.

AI 기술 민주화

2056. 10. 01

시연

128.134.233.77:800