

제목 이름

프로젝트 계획서(연구내용)

과제명	국문	실시간 얼굴 인식 기반 모자이크 처리 시스템 개발
	영문	Development of a Real-Time Face Detection-Based Mosaic Processing System

1. 프로젝트 필요성

최근 개인정보 보호에 대한 관심이 높아지면서, 영상 속 인물의 신원을 보호하기 위한 기술이 중요해지고 있다. 특히 CCTV, 방송, SNS 영상 등 다양한 미디어에서 얼굴 정보를 자동으로 감지하고 모자이크 처리하는 기술은 개인 프라이버시 보호의 핵심 도구로 부상하고 있다. 본 프로젝트는 이러한 사회적 요구에 부응하여, 실시간으로 얼굴을 감지하고 자동으로 모자이크 처리하는 프로그램을 구현하는 것을 목표로 한다.

2. 프로젝트 목표 및 내용

1) 프로젝트 최종 목표

웹캠 영상을 실시간으로 분석하여 얼굴을 자동 감지하고, 해당 영역에 모자이크 처리 효과를 적용하는 시스템을 개발한다. 이를 통해 영상 내 인물의 프라이버시를 보호하면서도 실시간 영상 처리의 효율성과 정확성을 확인한다.

2) 프로젝트 단계별 목표 및 내용

1단계 - 목표 : OpenCV 환경 설정

내용 : Python 및 OpenCV 설치 및 테스트

2단계 - 목표 : 얼굴 감지 기능 구현

내용 : Haar Cascade 또는 DNN 기반 얼굴 인식 모델 적용

3단계 - 목표 : 모자이크 처리 구현

내용 : 얼굴 영역을 지정하고, 화소 축소/확대 방식으로 모자이크 처리

4단계 - 목표 : 실시간 영상 처리

내용 : 웹캠 입력 프레임에서 실시간으로 처리 적용

5단계 - 목표 : 성능 테스트 및 개선

내용 : 처리 속도, 정확도 측정 및 최적화 수행

3. 연구과제의 추진전략·방법

1) 연구과제의 추진전략·방법

라이브러리 활용: Python 기반 OpenCV 라이브러리를 중심으로 구현. Haar Cascade, Dlib 또는 Mediapipe 모델을 비교하여 적절한 얼굴 검출 방법 선정.

프레임 처리 전략: 웹캠 입력을 일정 FPS 단위로 처리하고, 감지된 얼굴에만 모자이크 효과 적용하여 성능 최적화.

모자이크 알고리즘: 얼굴 영역 추출 후 해상도를 일시적으로 낮춘 후 확대하여 픽셀화 효과 구현.

모듈화: 얼굴 감지, 모자이크 처리, 영상 출력 기능을 각각 모듈로 분리하여 유지보수 및 확장성 확보.

4. 연구과제의 활용방안 및 기대효과

1) 연구과제의 활용방안

방송 영상 처리: 인터뷰 등에서 제3자의 얼굴을 자동 모자이크 처리

CCTV 영상의 개인정보 보호: 실시간으로 지나가는 사람 얼굴을 자동으로 흐리게 처리

모바일 앱 연동: SNS 영상 공유 시 얼굴 보호 기능으로 연동 가능

2) 기대효과

- 영상 기반 개인정보 보호 기술에 대한 이해도 증진
- 실시간 영상 처리 시스템 구현 경험 습득
- OpenCV 및 얼굴 인식 알고리즘 활용 능력 향상
- 관련 분야(영상처리, 인공지능, 보안)의 실무 프로젝트 경험 확보

5. 기타 – 참고문헌(Reference)

AI HUB - 인공지능 허브 (한국지능정보사회진흥원)

- URL: <https://www.aihub.or.kr>
- 설명: 다양한 AI 관련 데이터셋 및 활용 예제 제공. 얼굴 이미지, 영상 데이터 등 공공 데이터 다수 존재.
- 활용 예시: 얼굴 인식 학습용 데이터나 기술 참고에 유용.

공공데이터 포털 - 영상처리/AI 관련 데이터 제공

- URL: <https://www.data.go.kr>
- 설명: 정부 주도 공개 데이터 제공 사이트. 얼굴 관련 CCTV 데이터, 이미지 데이터 등 활용 가능.

한빛미디어 - OpenCV, Python 영상처리 관련 기술 블로그 및 출판사 자료

- URL: <https://www.hanbit.co.kr>
- 설명: 국내 IT 서적 출판사이며, OpenCV나 영상처리 관련 블로그 글 및 예제 코드 다수 존재.
- 예시 페이지: “OpenCV로 영상처리 시작하기” 같은 기술 연재글

Inflearn (인프런) - 영상처리 및 OpenCV 강의

- URL: <https://www.inflearn.com>
- 설명: 한국어로 된 실습 중심의 OpenCV 강의가 다수 있음.
- 활용 예시: 프로젝트 구현 단계에서 참고 가능.



OpenCV 공식 문서

- URL: <https://docs.opencv.org>
- 설명: OpenCV의 모든 함수, 클래스, 모듈에 대한 공식 레퍼런스 문서 제공.

결과 작성 내용

1. 시스템 구현:

- 웹캠을 이용하여 실시간 얼굴 인식 및 모자이크 처리 시스템을 성공적으로 구현하고 Haar Cascade와 DNN 얼굴 인식 모델을 모두 적용하여 높은 정확도와 효율성을 기록

이름	수정한 날짜	유형
 deploy.prototxt	2025-05-14 오전 8:55	PROTOTXT 파일
 res10_300x300_ssd_iter_140000_fp16.c...	2025-05-14 오전 9:00	CAFFEMODEL 파일

(1)모델의 설계도

(2)이진(binary) 파일

```
# Haar Cascade를 이용한 얼굴 검출
haar_faces = haar_face_cascade.detectMultiScale(gray, 1.1, 4)

# DNN을 이용한 얼굴 검출 (DNN 모델은 이미지를 300x300 크기로 변환해야 함)
h, w = frame.shape[:2]
blob = cv2.dnn.blobFromImage(frame, 1.0, (300, 300), (104.0, 177.0, 123.0), swapRB=True, crop=False)
net.setInput(blob)
dnn_faces = net.forward()

# 얼굴 크기 조정
face = cv2.resize(face, (10, 10)) # 축소
face = cv2.resize(face, (w, h), interpolation=cv2.INTER_NEAREST) # 확대 (모자이크 효과)
```

2. 성능 평가:

- Haar Cascade 모델은 처리 속도에서 우수하고, DNN 모델은 얼굴 인식 정확도에서 더 높은 성능을 보였으며 실시간 처리 FPS는 평균 20 FPS로 원활하게 작동

3. 기술적 문제 해결:

- 조명이 부족한 환경에서도 얼굴 인식 성능을 개선하기 위해 추가적인 최적화가 필요함을 확인

4. 향후 개선 사항:

- 처리 속도 개선을 위해 딥러닝 기반 모델을 추가하고, 모바일 환경에서의 성능 최적화 작업이 필요

5. 데모영상

<https://youtu.be/pJxjFX-MjJo>