

---

---

---

---

---



# 모자이크 application 설계 계획서

1. 주제: 영상에서 사람의 얼굴을 찾아 모자이크 처리해주는 프로그램.

2. 동기: 영상 편집 시 사람 얼굴을 모자이크 해주는 시 너무 시간이 많이 걸린다.  
초상권 및 인권을 위해 공개되는 영상에서는 모자이크가 필수

## 3. 설계

1) object detection을 이용해 사람 얼굴을 찾는다.

↳ 이미지에서 위치 찾기

2) 해당 영역을 blur 처리한다.

3) 처리 후 이미지 아 영감을 제공한다.

\* 참고자료

· 얼굴 detect 하는 모델 => yolo 사용

↳ yolo는 pytorch 모델.

· opencv로 이미지 처리 및 저장.

## 4. 데이터 수집

인터넷에 공개된 데이터 셋을 사용한다.

<https://aihub.or.kr/aidata/73>

=> Aihub에 공개되어 있는 사람 얼굴 데이터 셋.

## 5. 모델 학습시키기.

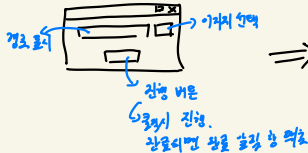
'얼굴' class만 detect 하려 때문에 yolov5에서 작은 모델 사용.

## 6. UI

UI는 python 이용

web 분야는 일반 프로그래밍으로 개발

<초기>



<최종>



21. 10. 17

AI hub서 dataset 확인 결과  
 얼굴 데이터 가진 하버드 컴퓨터 과학  
 얼굴 감출보이는 간단한 프로그램 만들기  
 적합할 것 같음

→ 새로 예제셋 찾기.

찾은 결과 마스크 애머센트 존재 (robotflood)  
 꽤 좋은 것도 있고 마스크 쓴 것도 많아서 때문에  
 이 애머센트로 일괄 학습 결정.

21. 10. 18

앞에 찾은 robotflood의 마스크 애머센트  
 이용하여 학습 진행 (이미지 수 = 105)  
 batch = 16 epoch = 200으로 학습 ⇒ 정확도 ↓  
 batch = 5 epoch = 500으로 학습 ⇒ 조금 낮은 성능부터

→ 학습 애머의 수가 적어서 정확도 하하

정확도야 더 높게 생각함

(precision, recall의 성능이 생각보다 낮음)

~0.8 ~0.5

(F2 score도 생각함)

0.6이상이면 긍정적인 평가

(데이터 뒤 부족으로 판단  
 학습은 애머의 더 찾기.

못 찾았으니 코로틀 등등 이용해 데이터 확보 (최소 300이상)

21. 10. 17

적당한 애머셋을 찾기복합  
 찾아보니 직접 학습 아닌 얼굴 감출된 학습된 라이브러리를 활용  
 → cvlib는 tensorflow, opencv를 이용한 vision library 처.  
 yolov4 처의 같이 사용 가능하고, 영상 처리도 가능함.

테스트 해보니 잘 작동하는 것을 확인함

→ 우선순위로 cvlib를 이용해 계산을 진행함.

yolov4를 학습한 좋은 모델로 dataset을 찾거나

프로그램을 실행한 후에 library가 아닌 직접 학습된

모델을 넣는 것 고려하기로 결정



공간과 ~ ~ ~ ~ ~

21. 10. 26

모자이크 코드 작성  
 이미지에게 원하는 위치에 모자이크 진행  
 → 거기에 해당 부분은 해당 색으로 채워줄

(UI 코딩 시작  
 Pythani으로 UI를 구현해본 후 여러 매개변수 공부함)

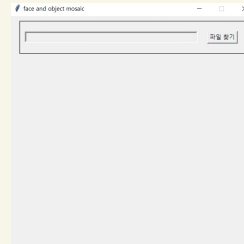
→ blur 처리 및 다른 처리가 되는지 확인

연구실 작업으로 인해 늦게...

21. 11. 14

python으로 UI 구현하기 위해 tkinter 라이브러리 사용  
 필요는 부분 ⇒ 파일 가져오기 / 옵션 선택 / 처리 버튼.

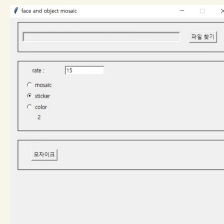
아래는 간단히 파일 가져오는 부분들 구현해서 실행은 하면.



(파일 가져오는 것까지 확인.  
 처리되는 부분 세게 구현하기)

21. 11. 16

처리를 옵션으로는 모자이크 비율 처리  
 처리 방식은 3종 크기로 모자이크, 스티커, 색칠 방식으로 구현  
 아래가 같이 rate는 default 값으로 15.  
 처리 방식은 라디오 버튼을 구현



(처리 버튼 클릭시 가져오기(명) 처리가 되지만 select, 처리는 비록 구현 X.  
 이후 구현하여 결과 출력)

21. 11. 20

(3종 방식 모두 구현하여 결과 확인)



(결과와 exe로 변환해서 확인 완료)

21.11.25

(이제 파일 서명새 문제 발생  
코드가 잘못 간됨.)

21.11.29

(이제 처리 후에 이어 다음 코드가 아닌 영상제작 코드로 연결됨.  
↳ 그래서 코드가 동작 X.  
실용하게 영상 등록함을 확인.)

(이제 exe 만들기)

21.12.01

(pyinstaller로 exe 만들.  
exe 처치한 onefile 옵션으로 만들었지만 실행 불가.  
↳ 고함 pref 파일 등 패치해야 되므로.  
(onefile 옵션 사용 X)  
정말 실행 가능함을 확인.)

21.12.02

(프로그래밍 이름은 "Mosaic Img"로 결정  
아름 만들.)



21.12.06

(github에 code upload 및 readme 정리 완료.  
exe 돌아는 용량) 너무 커 upload 안함.

\* 프로젝트는 여전히 진행. (기밀레 및 허깅페이스로 인해)  
후속 작업 가능함으로.

1. 이제 프로그램에 추가하고 싶은 기능은 무엇이 추가 하나.

2. cvlib 대신에 object detection 모델로 변경

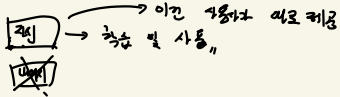
↳ 거기에 허버인 yoloV5 등.

DNN - openCV → 차분 강화

비영의 강자 관련 얼굴 보러오기.  
↳ 구분

리얼타임으로 검증하는 모델 따로 실행

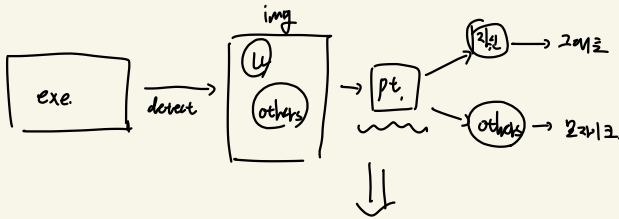
↳ 블록 모델은 CNN 사용



detect 및 face 블록에 detect 결과 따로 넣

↳ 이를 모델에 다시 넣어  
최상의 얼굴 인이 판단.

정확도가 90% 이상 가능.



DNN 에 입력 해라.  
(블록 모델)