
영상처리 프로그래밍

8주차

소개

- 조교 : 김형훈
 - 연구실 : A1409 (운영체제 연구실)
 - 이메일 : khh8996@naver.com
 - 전화번호 : 010-5316-7953
- 담당 교수 : 김백섭 교수님

목차

- 하르 캐스케이드 얼굴 검출
- 이미지 파일 업로드
- 실습
- 과제

하르 캐스케이드 얼굴 검출 (Haar-cascade Face Detection)

- 캐스케이드 분류기(cascade classifier)는 객체를 검출할 수 있도록 OpenCV가 제공하는 API
 - 얼굴 인식을 목적으로 시작함. 이후 물체 인식까지 가능하게 됨
- OpenCV는 캐스케이드 분류기에 사용할 수 있는 훈련된 검출기를 xml 파일 형태로 제공함
- OpenCV는 훈련된 검출기를 이용해서 객체를 검출하기 위한 캐스케이드 분류기 API를 제공함

하르 캐스케이드 얼굴 검출 (Haar-cascade Face Detection)

- `classifier = cv2.CascadeClassifier([filename])`
 - `filename` : 검출기 저장 파일 경로
 - `classifier` : 캐스케이드 분류기 객체
- `rect = classifier.detectMultiScale(img, scaleFactor, minNeighbors[, flags, minSize, maxSize])`
 - `img` : 입력 이미지
 - `scaleFactor` : 이미지 확대 크기에 제한, 1.3 ~ 1.5
 - `minNeighbors` : 요구되는 이웃 수
 - `flags` : 구식 API를 위한 것, 지금은 사용 안 함
 - `minSize, maxSize` : 해당 사이즈 영역을 넘으면 검출 무시
 - `rect` : 검출된 영역 좌표 (x, y, w, h)

하르 캐스케이드 얼굴 검출 (Haar-cascade Face Detection)

- OpenCV는 동영상 파일이나 카메라로부터 연속된 이미지 프레임을 읽을 수 있는 API 제공함
- `cap = cv2.VideoCapture(file_path 또는 index)`
 - `file_path` : 동영상 파일 경로
 - `index` : 카메라 장치 번호 (0, 1, 2, ...)
 - `cap` : VideoCaptrue 객체
- `ret, img = cap.read()`
 - `ret` : 프레임 읽기 성공 또는 실패 여부, True/False
 - `img` : 프레임 이미지, Numpy 배열 또는 None

하르 캐스케이드 얼굴 검출 (Haar-cascade Face Detection)

- OpenCV는 이미지 또는 비디오에 그림을 그릴 수 있는 API를 제공함
 - 직선, 사각형, 다각형, 원, ...
- `cv2.rectangle(img, start, end, color[, thickness, lineType])`
 - `img` : 그림 그릴 대상 이미지, Numpy 배열
 - `start` : 사각형 시작 꼭짓점 (x, y)
 - `end` : 사각형 끝 꼭짓점 (x, y)
 - `color` : 색상 (Blue, Green, Red)
 - `thickness` : 선 두께, -1 : 채우기
 - `lineType` : 선 그리기 형식
 - `cv2.LINE_4`, `cv2.LINE_8`, `cv2.LINE_AA`

하르 캐스케이드 얼굴 검출 (Haar-cascade Face Detection)

- OpenCV는 이미지 또는 비디오에 그림을 그릴 수 있는 API를 제공함
 - 직선, 사각형, 다각형, 원, ...
- `cv2.circle(img, center, radius, color[, thickness, lineType])`
 - `img` : 그림 그릴 대상 이미지, Numpy 배열
 - `center` : 원점 좌표 (x, y)
 - `radius` : 원의 반지름
 - `color` : 색상 (Blue, Green, Red)
 - `thickness` : 선 두께, -1 : 채우기
 - `lineType` : 선 그리기 형식
 - `cv2.LINE_4`, `cv2.LINE_8`, `cv2.LINE_AA`

하르 캐스케이드 얼굴 검출 (Haar-cascade Face Detection)

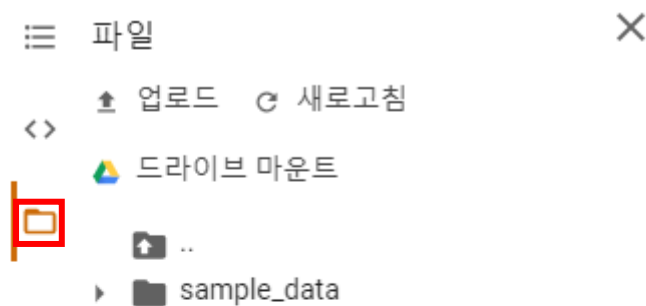
- OpenCV에서는 스레시홀딩(thresholding)을 해주는 API 제공
 - 스레시홀딩(thresholding)이란 여러 값을 경계점을 기준으로 두 가지 부류로 나누는 것
 - 바이너리 이미지(Binary Image; 검은색과 흰색 만으로 구성된 이미지)를 만들 때 사용
- `ret, out = cv2.threshold(img, threshold, value, type_flag)`
 - `img` : 변환할 이미지
 - `threshold` : 경계 값
 - `value` : 경계 값 기준에 만족하는 픽셀에 적용할 값
 - `type_flag` : 스레시홀드 적용 방법 지정
 - `cv2.THRESH_BINARY` : 픽셀 값이 경계 값을 넘으면 `value`, 넘지 못하면 0
 - `cv2.THRESH_BINARY_INV` : `cv2.THRESH_BINARY`의 반대
 - `cv2.THRESH_TRUNC` : 픽셀 값이 경계 값을 넘으면 `value`, 넘지 못하면 원래의 값 유지

하르 캐스케이드 얼굴 검출 (Haar-cascade Face Detection)

- OpenCV는 두 영상의 각 픽셀에 대한 비트와이즈(bitwise, 비트 단위) 연산 기능
 - 비트와이즈 연산은 두 영상을 합성할 때 특정 영역만 선택하거나 제외하는 등의 수행을 할 때 사용
 - AND, OR, XOR, NOT 연산
- `cv2.bitwise_and(img1, img2, mask=None)`
- `cv2.bitwise_or(img1, img2, mask=None)`
- `cv2.bitwise_xor(img1, img2, mask=None)`
- `cv2.bitwise_not(img1, img2, mask=None)`
 - `img1, img2` : 연산 대상 이미지 (동일한 shape)
 - `mask` : 0이 아닌 픽셀만 연산

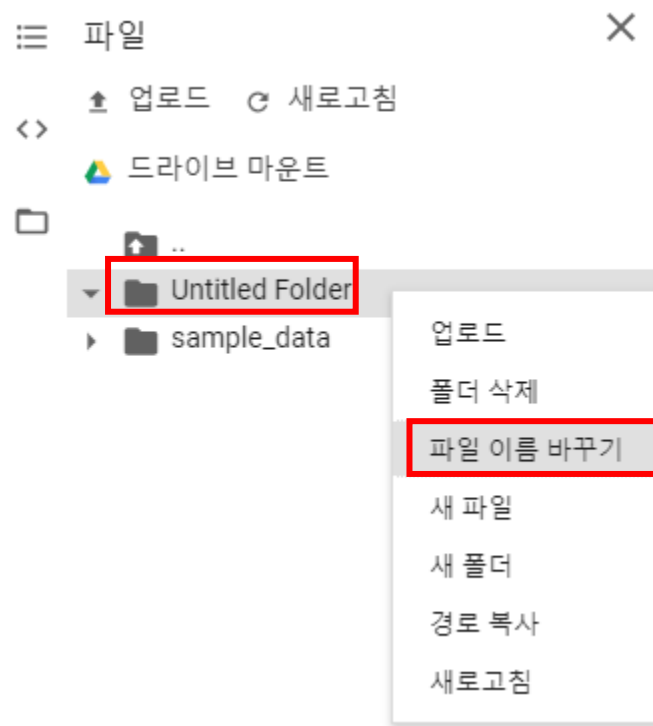
이미지 파일 업로드

- 파일 아이콘 클릭 후, 오른쪽 마우스 우 클릭 후, 새 폴더 클릭



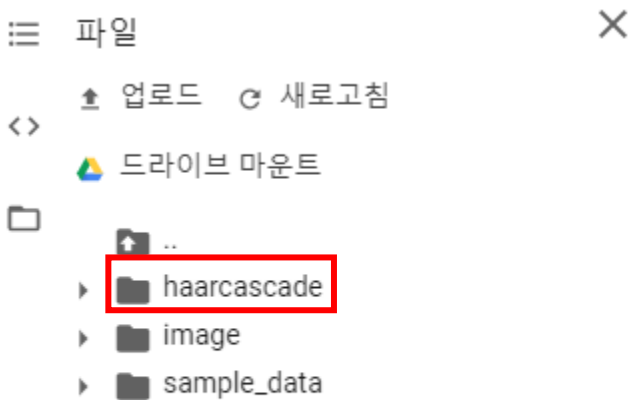
이미지 파일 업로드

- 새 폴더 생성 후, 폴더명 image로 변경



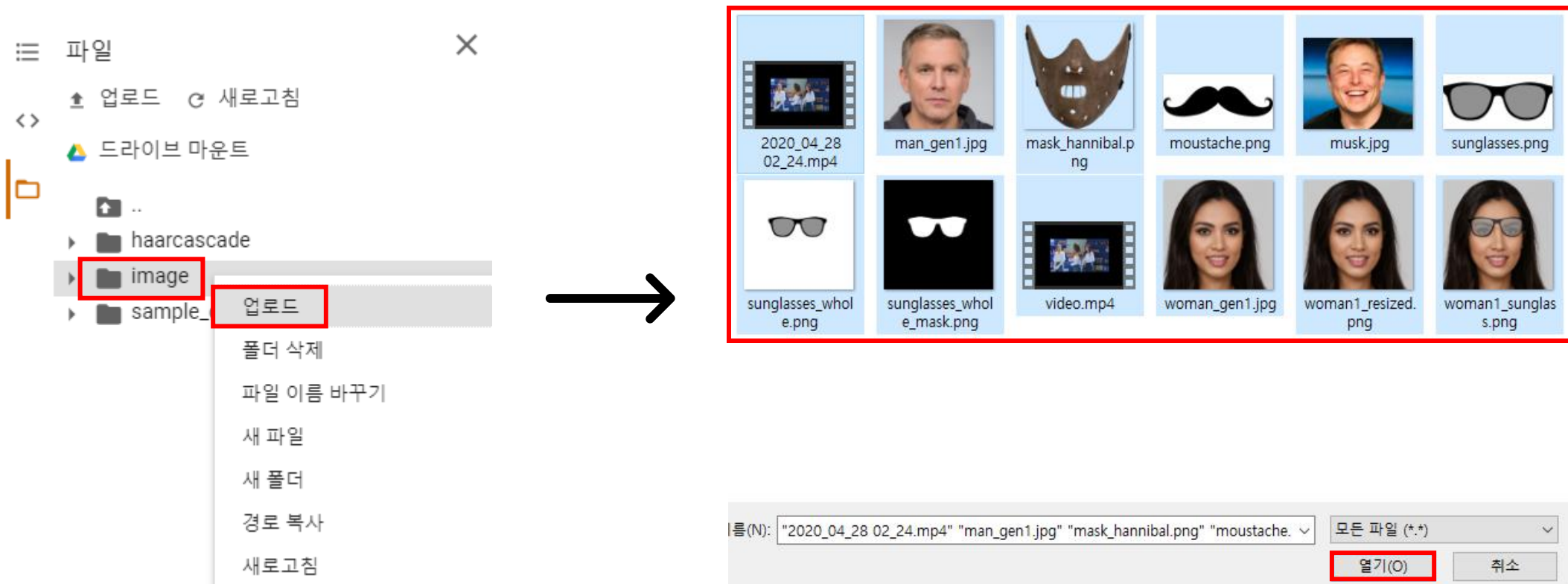
이미지 파일 업로드

- 위와 같은 방법으로, 폴더명 haarcascade인 폴더 생성



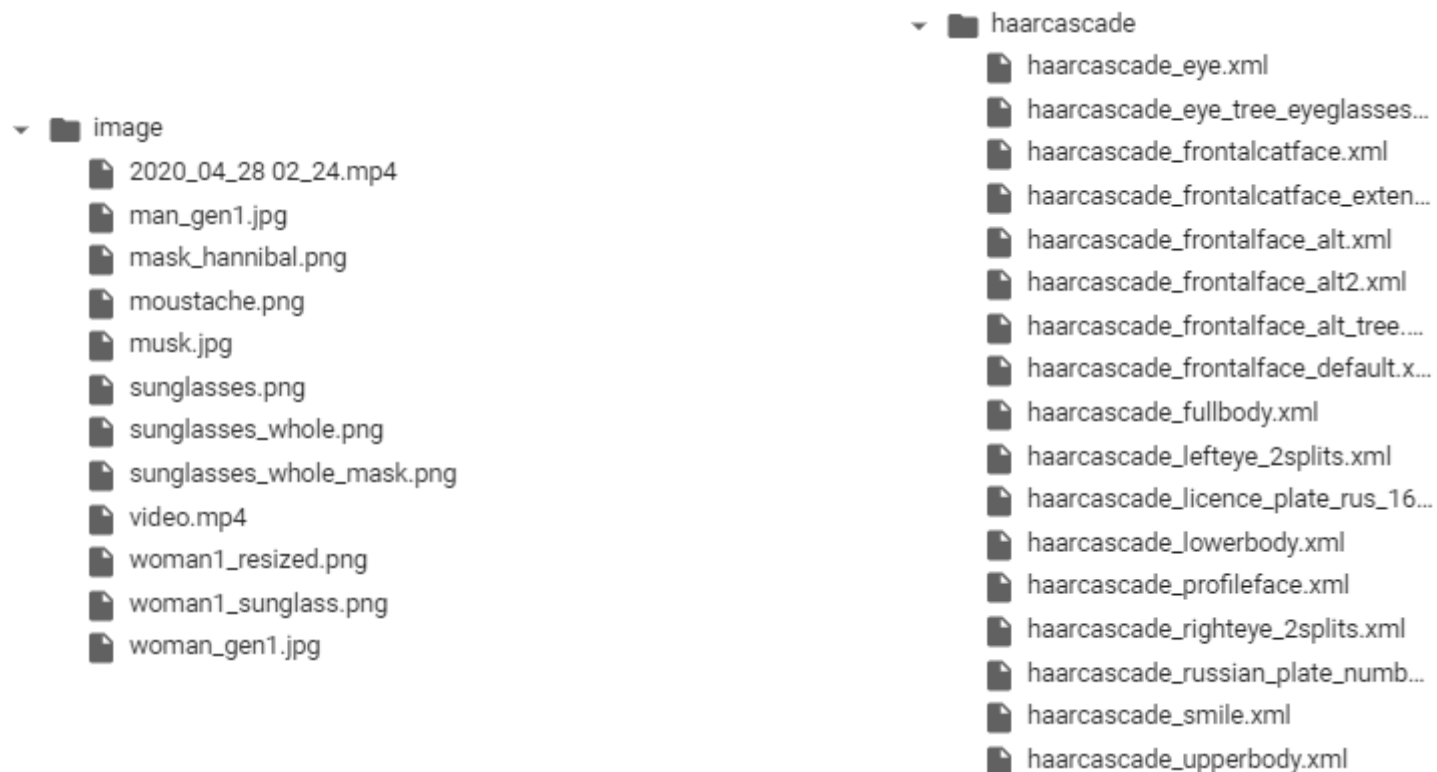
이미지 파일 업로드

- image 폴더에 images 실습 이미지 파일 업로드



이미지 파일 업로드

- 위와 같은 방법으로 haarcascade 폴더에 haarcascades 실습 이미지 파일 업로드
- 아래 그림과 같이, colab에 images와 haarcascades 실습 이미지 파일 업로드 됨



실습

- 실습은 예시를 따라 하는 것으로 대체함

과제

- 소스 코드와 결과를 캡처하여 문서(워드, 한글)에 정리한 후, 스마트 캠퍼스에 제출
 - 제출 형식 : 교과목명_주차_학번_이름.docx or .hwp (제출 형식 안 맞을 시 감점)
 - 기간 내에 제출하지 못할 경우 이메일로 제출 (감점)
 - 주석 작성 필수 (주석 없을 시 감점)
 - 부정 행위 적발 시 감점 (컨닝)

과제1

- 마스크 크기 조정
 - 마스크 크기를 조정해서 마스크 쓴 얼굴에 눈과 입이 보이도록 하시오.



과제2

- 선글라스의 투명도를 반투명으로 변환하시오.
 1. 원 영상과 같은 크기로 선글라스가 위치할 곳으로 이동된 영상(선글라스_위치 이동 영상)을 만드시오.
 2. 1 번의 영상으로 선글라스가 있는 부분만 값이 255이고 나머지는 0인 마스크 영상을 만드시오.
 3. 마스크 영상과 원 영상을 혼합하되, 선글라스가 있는 부분은 혼합하고 나머지 부분은 원 영상으로 그대로 두시오.

선글래스(위치이동)영상



마스크영상



결과영상



선택 과제

- Week8.ipynb 파일에서 선택 과제 부분이 있습니다.
- 수행할 시, 추가 점수를 드릴 예정입니다.
- 기말 프로젝트 대체 과제이므로 이 과제에 대해서는 질문을 받지 않습니다.

감사합니다