
영상처리 프로그래밍

9주차

소개

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

목차

- Python + PyCharm
- GUI
- Video
- Cartoonify
- 실습
- 과제

Python + PyCharm

- Python은 1990년 암스테르담의 구도 반 로섬(Guido Van Rossum)이 개발한 인터프리터 언어
- Python 설치 과정은 아래 링크 참고
 - <https://mingtrace.tistory.com/485>
- PyCharm은 Python에 사용되는 JetBrains사에 의해 개발된 IDE(Integrated Development Environment)
- PyCharm 설치 과정은 아래 링크 참고
 - <https://mingtrace.tistory.com/500>

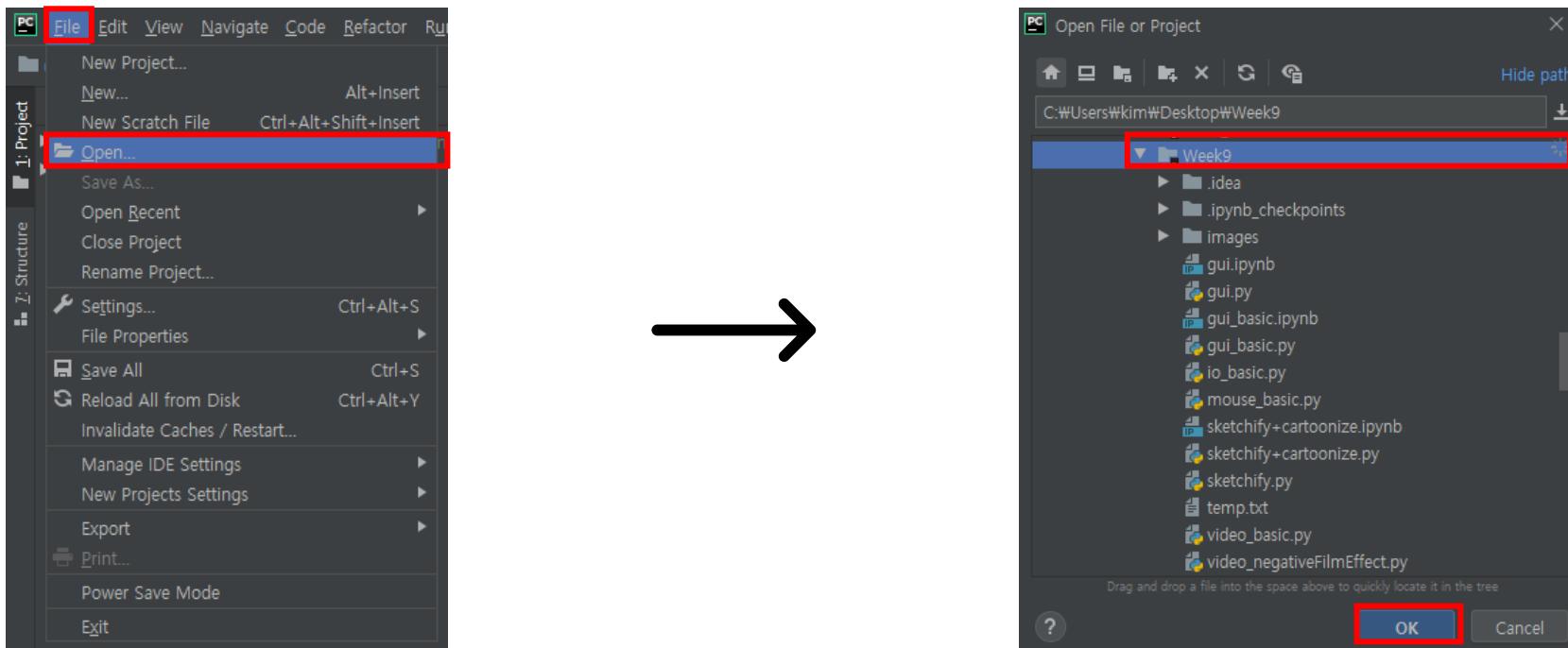
Python + PyCharm

- PyCharm에서 실습 실행 방법
 - 스마트 캠퍼스에서 Week9.zip 다운로드 및 압축 풀기



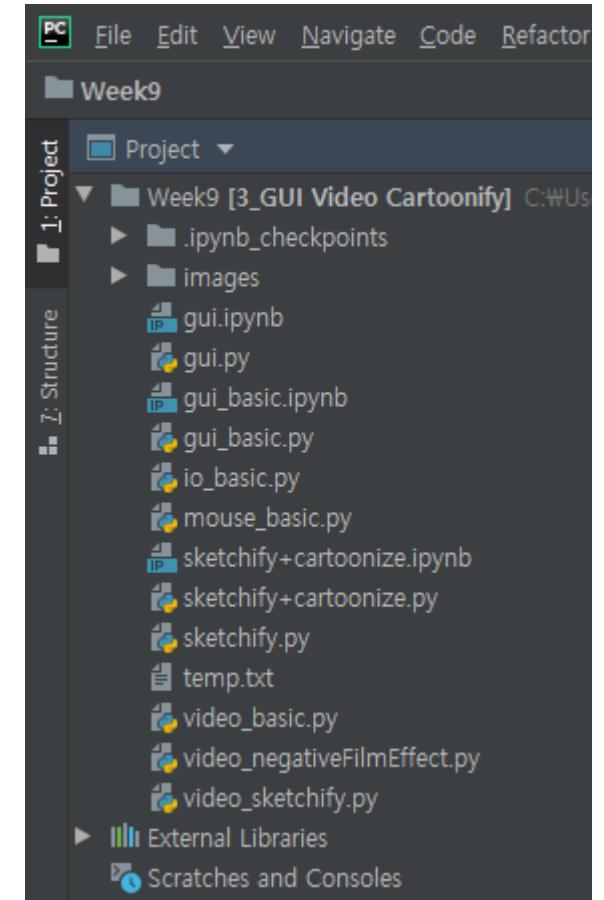
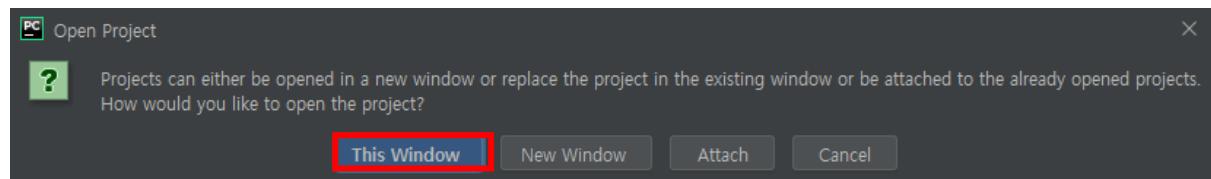
Python + PyCharm

- PyCharm에서 실행 방법
 - 앱축 파일 경로 파악한 후, PyCharm 실행
 - File에서 Open 클릭 후, 앱축 파일 경로를 찾아가서 앱축 파일 폴더(Week9) 클릭



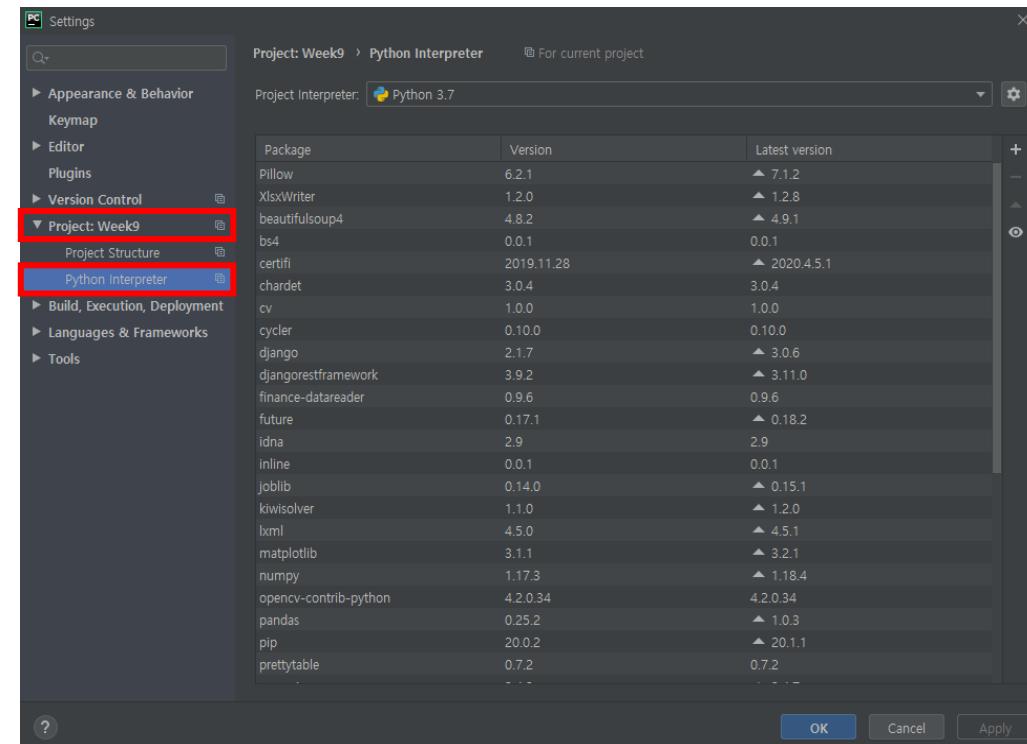
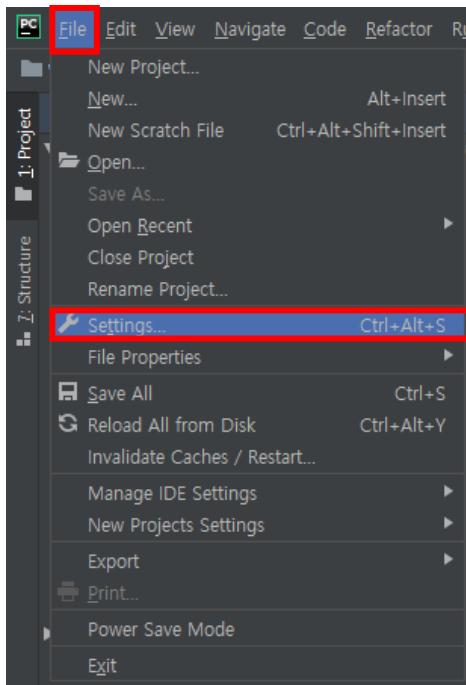
Python + PyCharm

- PyCharm에서 실행 방법
 - 왼쪽 그림에서 This Window 클릭
 - 오른쪽 그림과 같이 PyCharm에서 Week9 파일을 여는 것을 볼 수 있음



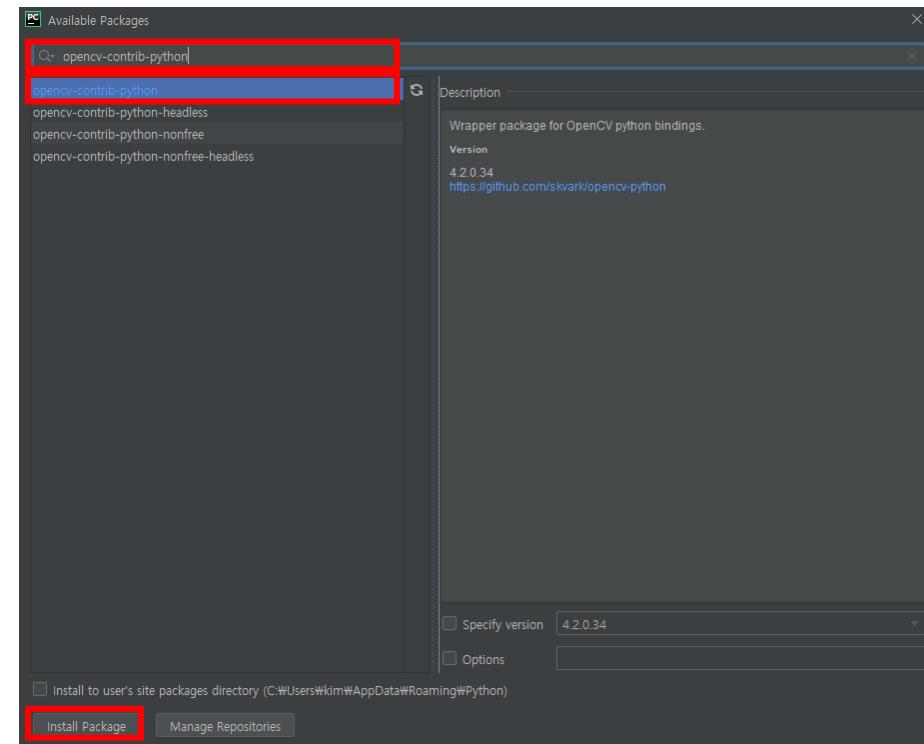
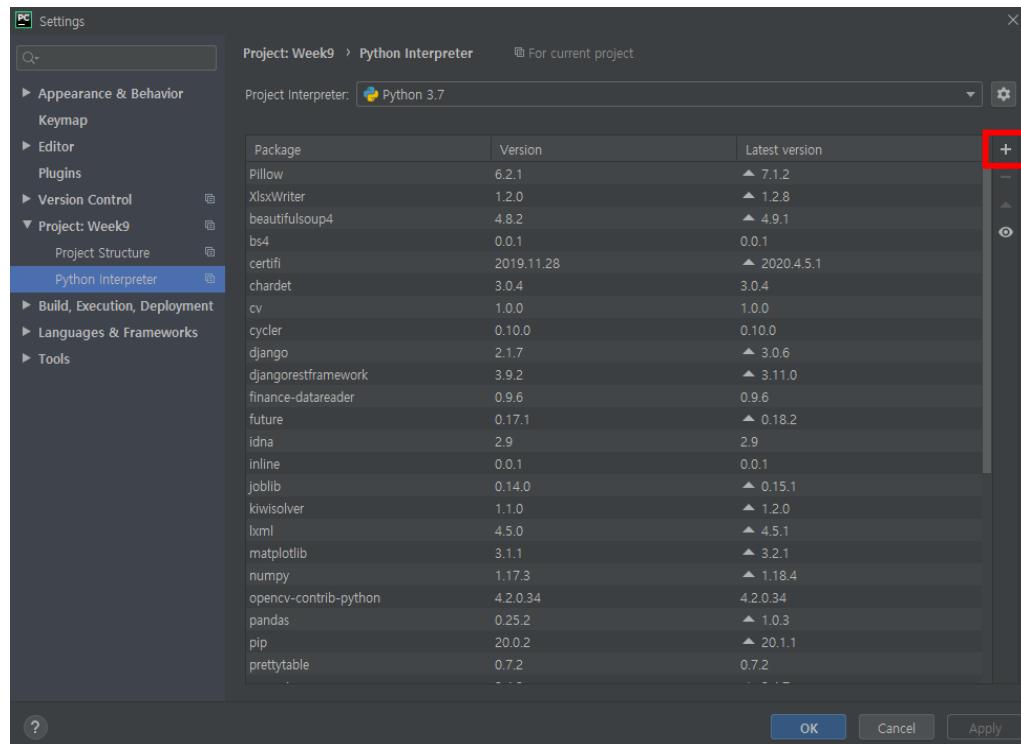
Python + PyCharm

- PyCharm에서 실습 실행 방법
 - OpenCV 라이브러리 설치
 - File에서 Settings 클릭 후, Project에서 Python Interpreter 클릭



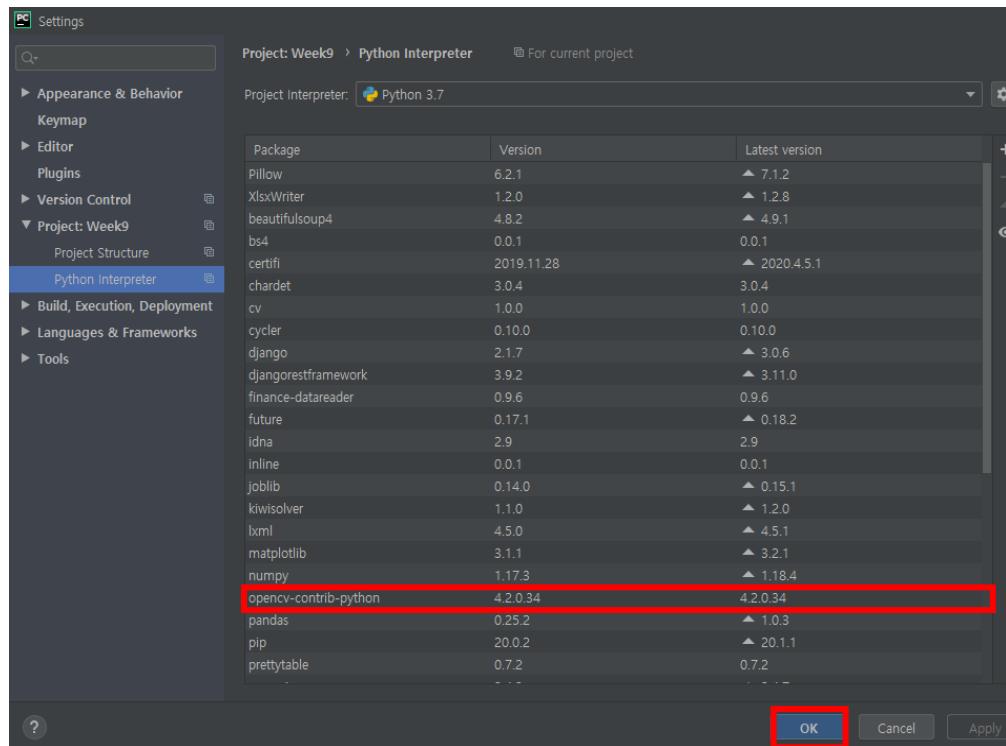
Python + PyCharm

- PyCharm에서 실습 실행 방법
 - Python Interpreter에서 + 버튼 클릭
 - opencv-contrib-python 입력 후, 클릭 및 설치



Python + PyCharm

- PyCharm에서 실습 실행 방법
 - opencv-contrib-python 설치된 것 확인 후, OK 버튼 클릭



GUI

- OpenCV는 이미지 또는 비디오에 그림을 그릴 수 있는 API를 제공함
 - 직선, 사각형, 다각형, 원, ...
- cv2.line(img, start, end, color[, thickness, lineType])
 - img : 그림 그릴 대상 이미지, Numpy 배열
 - start : 선 시작 꼭짓점 (x, y)
 - End : 선 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

GUI

- 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

GUI

- 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

GUI

- OpenCV는 이미지 또는 비디오에 그림을 그릴 수 있는 API를 제공함
 - 직선, 사각형, 다각형, 원, ...
- cv2.ellipse(img, center, axes, angle, from, to, color[, thickness, lineType])
 - img : 그림 그릴 대상 이미지, Numpy 배열
 - center : 원점 좌표 (x, y)
 - axes : 기준 축 길이
 - angle : 기준 축 회전 각도
 - from, to : 호를 그릴 시작 각도와 끝 각도
 - color : 색상 (Blue, Green, Red)

GUI

- OpenCV는 이미지 또는 비디오에 그림을 그릴 수 있는 API를 제공함
 - 직선, 사각형, 다각형, 원, ...
- cv2.polyline(img, points, isClosed, color[, thickness, lineType])
 - img : 그림 그릴 대상 이미지, Numpy 배열
 - points : 꼭짓점 좌표, Numpy 배열 리스트
 - isClosed : 닫힌 도형 여부, True/False
 - color : 색상 (Blue, Green, Red)
 - thickness : 선 두께, -1 : 채우기
 - lineType : 선 그리기 형식
 - cv2.LINE_4, cv2.LINE_8, cv2.LINE_AA

GUI

- OpenCV는 이미지 또는 비디오에 그림을 그릴 수 있는 API를 제공함
 - 직선, 사각형, 다각형, 원, ...
- cv2.putText(img, text, point, fontFace, fontSize, color[, thickness, lineType])
 - img : 글씨를 표시할 이미지
 - text : 표시할 문자열
 - point : 글씨를 표시할 좌표 (좌측 하단 기준)(x, y)
 - fontFace : 글꼴
 - cv.FONT_HERSHEY_PLAIN, cv.FONT_HERSHEY_SIMPLEX, cv.FONT_HERSHEY_DUPLEX, cv.FONT_HERSHEY_COMPLEX_SMALL, ...
 - fontSize : 글꼴 크기
 - color : 색상 (Blue, Green, Red)

GUI

- OpenCV는 창 관리할 수 있는 API를 제공함
- key = cv2.waitKey([delay]) : 키보드 입력 대기
 - delay : 키보드 입력을 대기할 시간(ms), 0 : 무한대 (기본 값)
 - key : 사용자가 입력한 키 값
- cv2.namedWindow(title, [, option]) : 이름을 갖는 창 열기
 - title : 창 이름, 제목 줄에 표시
 - option : 창 옵션
 - cv2.WINDOW_NORMAL, cv2.WINDOW_AUTOSIZE

GUI

- OpenCV는 창 관리할 수 있는 API를 제공함
- cv2.destroyWindow(title) : 창 닫기
 - title : 닫을 대상 창 이름
- cv2.destroyAllWindows() : 열린 모든 창 닫기

GUI

- OpenCV는 마우스 이벤트를 처리하는 API를 제공함
 - 마우스에서 입력을 받으려면 이벤트를 처리할 함수를 미리 선언해 놓고 cv2.setMouseCallback() 함수에 그 함수를 전달함
- cv2.setMouseCallback(win_name, onMouse [, param])
 - win_name : 이벤트를 등록할 윈도 이름
 - onMouse : 이벤트 처리를 위해 미리 선언해 놓은 콜백 함수
 - param : 필요에 따라 onMouse 함수에 전달할 인자

GUI

- MouseCallback(event, x, y, flags, param) : 콜백 함수 선언부
 - event : 마우스 이벤트 종류
 - cv2.EVENT_MOUSEMOVE : 마우스 움직임
 - cv2.EVENT_LBUTTONDOWN : 왼쪽 버튼 누름
 - cv2.EVENT_RBUTTONDOWN : 오른쪽 버튼 누름
 - cv2.EVENT_MBUTTONDOWN : 가운데 버튼 누름
 - cv2.EVENT_LBUTTONUP : 왼쪽 버튼 뗄
 - cv2.EVENT_RBUTTONUP : 오른쪽 버튼 뗄
 - cv2.EVENT_MBUTTONUP : 가운데 버튼 뗄
 - cv2.EVENT_LBUTTONDOWNDBLCLK : 왼쪽 버튼 더블 클릭
 - cv2.EVENT_RBUTTONDOWNDBLCLK : 오른쪽 버튼 더블 클릭
 - cv2.EVENT_MBUTTONDOWNDBLCLK : 가운데 버튼 더블 클릭
 - cv2.EVENT_MOUSEWHEEL : 휠 스크롤
 - cv2.EVENT_MOUSEHWHEEL : 휠 가로 스크롤

GUI

- MouseCallback(event, x, y, flags, param) : 콜백 함수 선언부
 - x, y : 마우스 좌표
 - flags : 마우스 동작과 함께 일어난 상태
 - cv2.EVENT_FLAG_LBUTTON : 왼쪽 버튼 누름
 - cv2.EVENT_FLAG_RBUTTON : 오른쪽 버튼 누름
 - cv2.EVENT_FLAG_MBUTTON : 가운데 버튼 누름
 - cv2.EVENT_FLAG_CTRLKEY : Ctrl 키 누름
 - cv2.EVENT_FLAG_SHIFTKEY : Shift 키 누름
 - cv2.EVENT_FLAG_ALTKEY : Alt 키 누름
 - param : cv2.setMouseCallback() 함수에서 전달한 인자

Video

- 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

Video

- OpenCV는 동영상 파일이나 카메라로부터 연속된 이미지 프레임을 읽을 수 있는 API 제공함
- `ret = cap.isOpened()` : 객체 초기화 확인
 - `ret` : 초기화 여부, True/False
- `cap.release()` : 캡처 자원 반납

Cartoonify

- 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_BINARY의 반대
 - cv2.THRESH_TRUNC : 픽셀 값이 경계 값을 넘으면 value, 넘지 못하면 원래의 값 유지

Cartoonify

- 미디언 블러링 (Median Blurring)
 - 커널 영역 픽셀 값 중에 중간 값을 대상 픽셀의 값으로 선택하는 것
- dst = cv2.medianBlur(src, ksize)
 - src : 입력 영상, Numpy 배열
 - ksize : 커널 크기

Cartoonify

- 바이레터럴 필터 (Bilateral Filter)
 - 가우시안 필터와 경계 필터를 사용하여 노이즈 없고 경계가 비교적 또렷한 영상을 얻음
 - 속도가 느림
- dst = cv2.bilateralFilter(src, d, sigmaColor, sigmaSpace[, dst, borderType])
 - src : 입력 영상, Numpy 배열
 - d : 필터의 직경(diameter), 5보다 크면 매우 느림
 - sigmaColor : 색공간 필터의 시그마 값
 - sigmaSpace : 좌표 공간의 시그마 값

실습

- 실습은 예시를 따라 하는 것으로 대체함

과제

- 소스 코드와 결과를 캡쳐하여 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)

과제 1

- Sketchify Video Frame
 - video_Twice.mp4 파일을 실행할 때, sketch 모드로 변환하여 실행
 - 마우스 왼쪽 버튼 누르면 왼쪽 상단에 Playing 글씨 표시되면서 동영상 진행
 - 마우스 오른쪽 버튼 누르면 왼쪽 상단에 Stopped 글씨 표시되면서 동영상 멈춤



감사합니다