x

K-Digital Training 스마트 팩토리 3기

OpenCV 2

흐림효과

- cv2.blur(src, ksize[, anchor=(-1,-1)[,
  borderType=BORDER_DEFAULT]]) -> dst
  - src : 입력 이미지
  - dst : 출력 이미지
  - ksize : 커널 크기
  - anchor : anchor point, (-1,-1)은 ancho가 커널의 중앙에 있다는 뜻
  - borderType : border mode

Edge Detection(가장자리 검출)

- cv2.Sobel(src, ddepth, dx, dy[, dst[, ksize=3[, scale=1[, delta=0[, borderType=BORDER_DEFAULT]]]]]) -> dst
  - src : 입력 이미지
  - dst : 출력 이미지
  - ddepth: 출력 이미지 depth
  - dx : 도함수 x의 순서
  - dy : 도함수 y의 순서
  - ksize : Sobel kernel의 크기, 1,3,5,7중 하나
  - scale : scale factor
  - delta : delta value
  - borderType : BorderType

Edge Detection(가장자리 검출)

- cv2.Laplacian(src, ddepth[dst[, ksize[, scale[, delta[, borderType]]]]]) -> dst

- cv2.Canny(src, threshold1, threshold2[,edges[, apetureSize=3, L2gradient=false]]]) -> edges
  - src : 입력 이미지
  - edges : 출력 edge 맵
  - threshold1 : 히스테리시스 절차의 첫 번째 임계값
  - threshold2 : 히스테리시스 절차의 두 번째 임계값
  - apetureSize : Sobel 연산자의 조리개 크기
  - L2gradient : L2 norm 을 사용하여 좀 더 자세하게 할지

이미지 합성

- cv2.addWeighted(src1, alpha, src2, beta, gamma[,dst[,dtype=-1]]) -> dst
  - src1 : 첫 번째 입력 배열
  - alpha : 첫 번째 배열의 비율
  - src2 : 두 번째 입력 배열
  - beta : 두 번째 배열의 비율
  - gamma : 두 배열의 합에 추가로 더할 값
  - dst : 출력 배열
  - dtype : 출력에 추가할 depth

  - dst = src1*alpha + src2*beta + gamma;

이미지 합성

- cv2.bitwise_and(src1, src2[,dst[, mask]]) -> dst
  - 두 배열의 비트 단위 결합 (dst = src1 & src2)
  - src1 : 첫 번째 입력 배열
  - src2 : 두 번째 입력 배열
  - dst : 출력 배열
  - mask : 출력 배열의 요소를 지정하는 선택적 연산 마스크

- cv2.bitwise_or(src1, src2[,dst[, mask]]) -> dst

이미지 합성

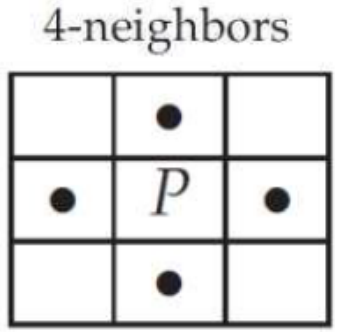
- cv2.bitwise_not(src[,dst[, mask]]) -> dst
  - 모든 비트 반전
  - src : 입력 배열
  - dst : 출력 배열
  - mask : 출력 배열의 요소를 지정하는 선택적 연산 마스크

그림 그리기

- cv2.line(img, pt1, pt2, color[, thickness[,
  lineType=LINE_8[,shift=0]]]) -> img
  - img : 이미지
  - pt1 : 선의 첫 번째 지점
  - pt2 : 선의 두 번째 지점
  - color : 선의 색
  - thickness : 선의 굵기

그림 그리기

- cv2.line(img, pt1, pt2, color[, thickness[,
  lineType=LINE_8[,shift=0]]]) -> img
  - lineType : 선의 타입
    - FILLED
    - LINE_4 : 4방향 연결
    - LINE_8 : 8방향 연결
    - LINE_AA : 안티앨리어싱 적용된 선
  - pt에서 얼마나 벗어날지



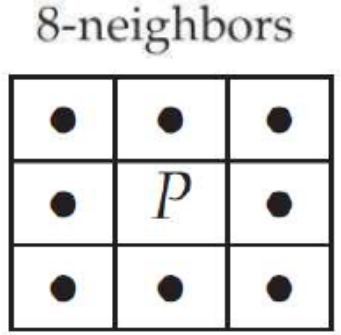


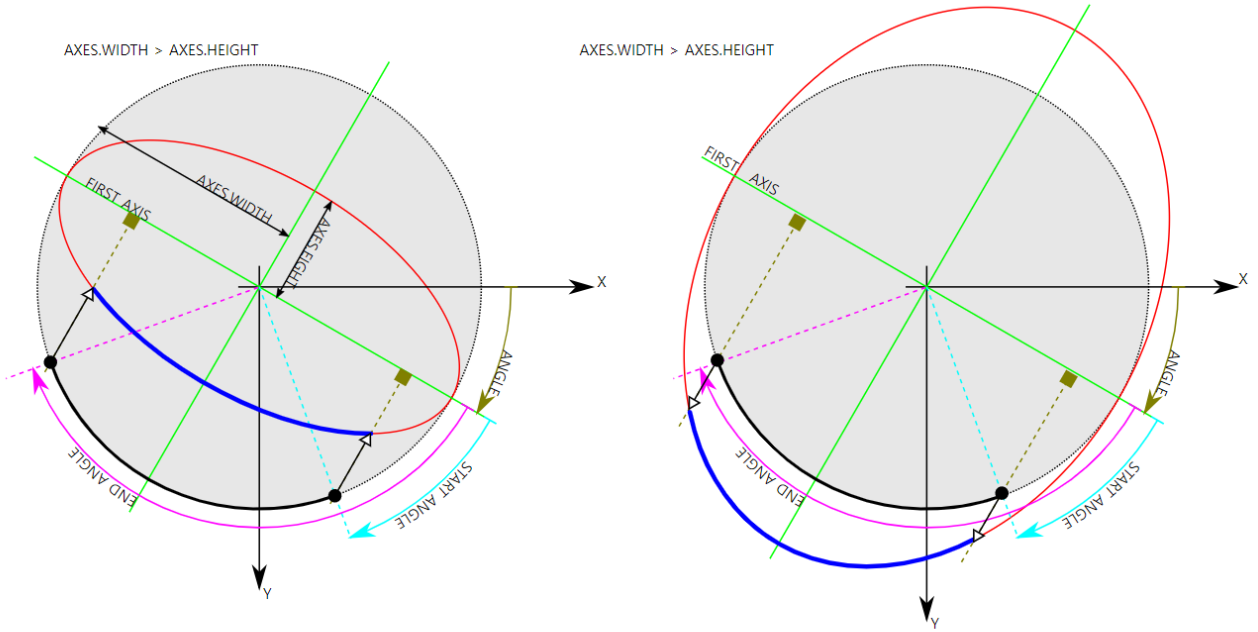
그림 그리기

- cv2.circle(img, center, radius, color[, thickness[, lineType=LINE_8[, shift=0]]]) -> img
  - center : 원의 중심
  - radius : 원의 반지름

- cv2.rectangle(img, pt1, pt2, color[, thickness=1[, lineType=LINE_8[, shift=0]]]) -> img
  - pt1 : 사각형의 한 꼭지점
  - pt2 : pt1의 반대 꼭지점

그림 그리기

- cv2.ellipse(img, center, axes, angle, startAngle, endAngle, color[, thickness=1[, lineType=LINE_8[, shift=0]]]) -> img
  - 타원
  - axes : 타원의 주축 크기의 절반
  - angle : 타원 회전 각도
  - startAngle : 타원 호의 시작 각도
  - endAngle : 타원 호의 끝 각도



타원호의 매개변수

그림 그리기

- cv2.polylines(img, pts, isClosed, color[, thickness[, lineType=LINE_8[, shift=0]]]) -> img
  - 다각형
  - pts : 다각형 선분들의 배열
  - isClosed : 닫힌 도형인지 아닌지 여부

- cv2.fillPoly(img, pts, color[, lineType=LINE_8[, shift=0[, offset=Point()]]]) -> img
  - 채워져 있는 다각형
  - offset : 모든 점들의 오프셋

그림 그리기

- cv2.putText(img, text, org, fontFace, fontScale, color[, thickness=1[, lineType=LINE_8[, bottomLeftOrigin=0]]]) -> img
  - 이미지에 글자 출력
  - text : 그려질 글자 배열
  - org : 글자의 좌하단 구석
  - fontFace : font type
  - fontSacle : font scale factor
  - color : 글자 색깔
  - bottomLeftOrigin : true -> origin 은 이미지의 좌 하단, false -> 좌 상단

그림 그리기

- cv2.clipLine(imgRect, pt1, pt2) -> retval, pt1, pt2
  - 사각형에 직선을 긋는 함수
  - imgRect : 이미지 사각형
  - pt1 : 선의 시작 위치
  - pt2 : 선의 종료 위치

얼굴 검출

- cv2.CascadeClassifier(filename)
  - filename : 로드 할 분류기 파일 이름
  - https://github.com/opencv/opencv/tree/master/data/haarcascades

XML 파일 이름	검출 대상
haarcascade_frontalface_default.xml haarcascade_frontalface_alt.xml haarcascade_frontalface_alt2.xml haarcascade_frontalface_alt_tree.xml	정면 얼굴 검출
haarcascade_profileface.xml	측면 얼굴 검출
haarcascade_smile.xml	웃음 검출
haarcascade_eye.xml haarcascade_eye_tree_eyeglasses.xml haarcascade_lefteye_2splits.xml haarcascade_righteye_2splits.xml	눈 검출
haarcascade_frontalcatface.xml haarcascade_frontalcatface_extended.xml	고양이 얼굴 검출
haarcascade_fullbody.xml	사람의 전신 검출
haarcascade_upperbody.xml	사람의 상반신 검출
haarcascade_lowerbody.xml	사람의 하반신 검출
haarcascade_russian_plate_number.xml haarcascade_licence_plate_rus_16stages.xml	러시아 자동차 번호판 검출

얼굴 검출

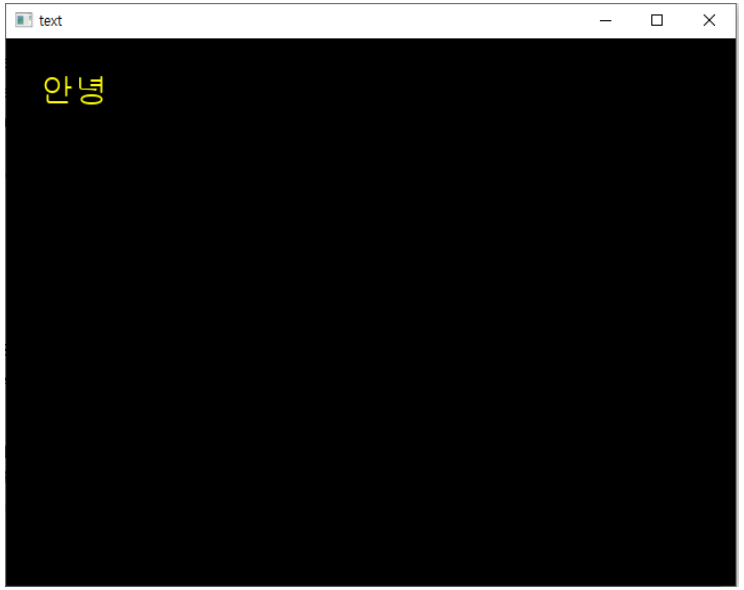
- {classifier}.detectMultiScale(img[, scaleFactor=1.1[, minNeighbors=3[, flags=None[, minSize=None[, maxSize=None]]]]]) -> objects
  - scaleFactor : 영상 축소 비율
  - minNeighbors : 최소 탐지 개수(슬라이딩 윈도우시에 얼마나 많이 찾는지)
  - flags : 사용되지 않음
  - minSize : 최소 객체 크기, (w,h)
  - maxSize : 최대 객체 크기, (w,h)
  - 결과값 : [(x,y,w,h)] : 검출된 객체의 사각형 정보의 배열

글씨 쓰기

- 한글은 cv2.putText에서 지원 안함
- PIL(Python Image Library) 이용
  - pip install pillow

```python
from PIL import ImageFont, ImageDraw, Image

def pilPutText(src, text, pos, font_size, font_color):
    img_pil = Image.fromarray(src)
    draw = ImageDraw.Draw(img_pil)
    font = ImageFont.truetype("fonts/gulim.ttc", font_size)
    draw.text(pos, text, font=font, fill=font_color)
    return np.array(img_pil)
```



영상 처리

- 영상 출력
  - cv2.VideoWriter_fourcc(c1,c2,c3,c4) -> retval
    - 4개의 fourcc code를 합쳐주는 함수
    - fourcc code란? 4글자 코드, 데이터의 형식을 구분하는 고유 글자
    - retval : forcc code

  - cv2. VideoWriter(filename, forcc, fps, frameSize, isColor=true)
    - VideoWriter 클래스 생성자
    - filename : 출력 파일 이름
    - forcc
    - fps : frame per seconds
    - frameSize : 비디오 프레임의 사이즈
    - isColor : color frame을 출력할지, false 면 grayscale

영상 처리

- 영상 출력
  - {VideoWriter}.write(image)
    - 프레임을 녹화
    - image : 녹화할 프레임
  - {VideoWriter}.release()
    - 녹화 중지

영상 처리

- 영상 캡쳐
  - cv2.imwrite(filename, img[, params]) -> retval
    - filename : 출력 이미지 파일 이름
    - img : 출력할 이미지
    - params : flags

마우스 처리

- 영상 캡쳐
  - cv2.setMouseCallback(winname, onMouse, userdata = 0)
    - winname : 윈도우 이름
    - onMouse : 마우스 이벤트를 처리할 콜백함수
    - userdata : 콜백함수에 사용할 변수
  - cv::MouseCallback(event, x, y, flags, userdata)
    - event : mouse event type 값
    - x : x좌표값
    - y : y좌표값
    - flags : mouse event flag 값
    - userdata : (optional) 함수 호출시 입력값

마우스 처리

- 마우스 이벤트
  - cv::MouseEventTypes {
    cv::EVENT_MOUSEMOVE = 0,
    cv::EVENT_LBUTTONDOWN = 1,
    cv::EVENT_RBUTTONDOWN = 2,
    cv::EVENT_MBUTTONDOWN = 3,
    cv::EVENT_LBUTTONUP = 4,
    cv::EVENT_RBUTTONUP = 5,
    cv::EVENT_MBUTTONUP = 6,
    cv::EVENT_LBUTTONDBLCLK = 7,
    cv::EVENT_RBUTTONDBLCLK = 8,
    cv::EVENT_MBUTTONDBLCLK = 9,
    cv::EVENT_MOUSEWHEEL = 10,
    cv::EVENT_MOUSEHWHEEL = 11
    }

마우스 처리

- 마우스 이벤트
  - cv::MouseEventFlags {
    cv::EVENT_FLAG_LBUTTON = 1,
    cv::EVENT_FLAG_RBUTTON = 2,
    cv::EVENT_FLAG_MBUTTON = 4,
    cv::EVENT_FLAG_CTRLKEY = 8,
    cv::EVENT_FLAG_SHIFTKEY = 16,
    cv::EVENT_FLAG_ALTKEY = 32
    }