

OpenCV 주요 함수 정리 (기능별)

이미지 입출력

- `cv2.imread(filename, flags=1)`

지정된 경로의 이미지 파일을 불러와 NumPy 배열 형식으로 반환한다. flags 인자를 통해 흑백, 컬러 여부를 설정할 수 있다.

- `cv2.imwrite(filename, img)`

NumPy 배열 형태의 이미지를 지정된 파일 이름으로 저장한다. 파일 포맷은 확장자에 따라 자동 결정된다.

- `cv2.imshow(winname, mat)`

이미지를 새로운 윈도우에 표시한다. 디버깅이나 시각화를 위해 자주 사용된다.

- `cv2.waitKey(delay=0)`

특정 시간(ms) 동안 키보드 입력을 기다리며, 0 이면 무한 대기한다. 주로 imshow 이후에 사용된다.

- `cv2.destroyAllWindows()`

imshow 로 띄운 모든 창을 닫는다.

색상 변환

- `cv2.cvtColor(src, code)`

BGR, RGB, GRAY, HSV, LAB 등 다양한 색상 공간 간의 변환을 수행한다. 모델 입력 전 전처리에 자주 사용된다.

기하 변환

- `cv2.resize(src, dsize, fx=0, fy=0)`

이미지를 지정한 크기로 리사이징하거나 fx, fy 배율을 통해 크기를 조절한다.

- `cv2.getRotationMatrix2D(center, angle, scale)`

이미지를 회전시키기 위한 2D 변환 행렬을 생성한다.

- `cv2.warpAffine(src, M, dsize)`

직선 유지하는 변환을 통해 회전, 이동, 확대/축소, 기울이기 등을 구현한다.

- `cv2.warpPerspective(src, M, dsize)`

이미지를 사각형 영역 기준으로 원근감 있게 변형한다.

- `cv2.flip(src, flipCode)`

flipCode 값에 따라 좌우/상하/대각선 방향으로 이미지를 뒤집는다.

필터링

- `cv2.GaussianBlur(src, ksize, sigmaX)`

가우시안 커널을 이용해 노이즈 제거 및 부드러운 블러 효과를 적용한다.

- `cv2.medianBlur(src, ksize)`

커널 내 중앙값을 사용하여 노이즈(특히 소금-후추 노이즈)를 제거한다.

- `cv2.bilateralFilter(src, d, sigmaColor, sigmaSpace)`

엣지를 보존하면서 블러 처리하는 필터로, 피사체의 경계가 잘 유지된다.

- `cv2.blur(src, ksize)`

커널 내 평균값을 사용하여 이미지를 흐리게 만든다.

- `cv2.boxFilter(src, ddepth, ksize)`

커널 내 모든 요소가 동일한 값인 고정 필터를 적용한다.

- `cv2.filter2D(src, ddepth, kernel)`

사용자가 지정한 커널을 사용해 원하는 방식으로 필터링을 적용할 수 있다.

에지 검출

- `cv2.Canny(image, threshold1, threshold2)`

이미지의 그래디언트를 계산하여 경계를 검출하는 알고리즘. 가장 일반적으로 사용된다.

- `cv2.Sobel(src, ddepth, dx, dy, ksize)`

x, y 방향 미분을 통해 이미지의 경계와 변화량을 검출한다.

- `cv2.Laplacian(src, ddepth)`

2 차 미분을 통해 보다 민감하게 경계를 검출한다.

이진화

- `cv2.threshold(src, thresh, maxval, type)`

지정한 임계값 기준으로 픽셀 값을 흑백(0 또는 maxval)으로 변환한다.

- `cv2.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C)`

이미지의 지역 특성을 고려해 픽셀마다 다른 임계값을 사용해 이진화한다.

윤곽선

- `cv2.findContours(image, mode, method)`

이진 이미지에서 객체의 외곽선을 추출한다.

- `cv2.drawContours(image, contours, contourIdx, color, thickness)`

검출한 윤곽선을 이미지에 선으로 시각화한다.

- `cv2.boundingRect(array)`

윤곽선을 둘러싸는 최소 크기의 직사각형을 계산한다.

- `cv2.contourArea(contour)`

윤곽선으로 둘러싸인 영역의 면적을 계산한다.

- `cv2.arcLength(curve, closed)`

윤곽선을 따라 측정한 총 길이를 반환한다.

- `cv2.minEnclosingCircle(points)`

윤곽선을 완전히 포함하는 최소 반지름 원을 계산한다.

- `cv2.approxPolyDP(curve, epsilon, closed)`

곡선 형태의 윤곽선을 직선 다각형으로 근사화한다.

- `cv2.convexHull(points)`

점 집합의 볼록한 외곽을 계산하여 외곽 윤곽선을 생성한다.

그리기

- `cv2.line(img, pt1, pt2, color, thickness)`

두 점 좌표를 기준으로 선을 이미지에 그린다.

- `cv2.rectangle(img, pt1, pt2, color, thickness)`

두 점 좌표로 사각형을 이미지에 그린다.

- `cv2.circle(img, center, radius, color, thickness)`

중심 좌표와 반지름을 기반으로 원을 이미지에 그린다.

- `cv2.ellipse(img, center, axes, angle, startAngle, endAngle, color, thickness)`

중심, 축 길이, 회전 각도 등을 기반으로 타원을 그린다.

- `cv2.putText(img, text, org, fontFace, fontScale, color, thickness)`

이미지에 문자열을 원하는 위치와 스타일로 삽입한다.

비트 연산

- `cv2.bitwise_and(src1, src2, mask=None)`

두 이미지 간 AND 비트 연산을 수행하여 공통된 부분을 추출한다.

- `cv2.bitwise_or(src1, src2, mask=None)`

두 이미지 간 OR 비트 연산을 수행하여 전체 영역을 병합한다.

- `cv2.bitwise_xor(src1, src2, mask=None)`

두 이미지 간 서로 다른 부분만 추출하는 비트 XOR 연산을 수행한다.

- `cv2.bitwise_not(src)`

이미지의 색상을 반전시킨다.

영상 연산

- `cv2.add(src1, src2)`

두 이미지 또는 배열을 픽셀 단위로 더한다.

- `cv2.subtract(src1, src2)`

두 이미지 또는 배열을 픽셀 단위로 뺀다.

- `cv2.multiply(src1, src2)`

두 이미지 또는 배열을 픽셀 단위로 곱한다.

- `cv2.divide(src1, src2)`

두 이미지 또는 배열을 픽셀 단위로 나눈다.

히스토그램

- `cv2.calcHist(images, channels, mask, histSize, ranges)`

픽셀 밝기 분포를 계산하여 이미지의 색상 또는 명암 통계를 얻는다.

- `cv2.equalizeHist(src)`

명암 분포를 고르게 조절하여 대비를 향상시킨다.

- `cv2.normalize(src, dst, alpha, beta, norm_type)`

데이터의 범위를 일정한 범위로 재조정하여 비교 가능하게 만든다.

비디오

- `cv2.VideoCapture(index)`

웹캠이나 비디오 파일에서 프레임을 읽을 수 있는 객체를 생성한다.

- `cap.read()`

비디오 캡처 객체에서 현재 프레임을 읽어온다.

- `cap.release()`

비디오 장치를 해제하고 리소스를 반환한다.

- `cv2.VideoWriter(filename, fourcc, fps, frameSize)`

프레임을 비디오로 저장할 수 있는 객체를 생성한다.

- `out.write(frame)`

단일 프레임을 비디오 파일로 저장한다.

- `cv2.VideoWriter_fourcc(*'XVID')`

비디오 압축에 사용할 코덱을 설정한다.

객체 검출

- `cv2.CascadeClassifier('path').detectMultiScale(img, ...)`

미리 학습된 분류기를 이용해 얼굴이나 눈 등의 객체를 검출한다.