



가톨릭대학교  
THE CATHOLIC UNIVERSITY OF KOREA

# OpenCV 기본연산2

- 기하변환, 연산함수, 수학 및 통계함수

미디어기술콘텐츠학과  
강호철

# 영상 크기 변환과 회전

---

- 변환 함수

- cv2.resize

- 영상의 크기 변환 – 확대/축소
    - parameter: src, dsize[, dst[, fx[, fy[, interpolation ]]]]) → dst
      - interpolation: cv2.INTER\_NEAREST, cv2.INTER\_LINEAR, cv2.INTER\_CUBIC 등

- cv2.rotate

- 영상을 90도 간격으로 회전 시킴
    - parameter: src, rotateCode[, dst]) → dst
      - rotateCode: cv2.ROTATE\_90\_CLOCKWISE, cv2.ROTATE\_180, cv2.ROTATE\_90\_COUNTERCLOCKWISE 등



# 영상 크기 변환과 회전

- 변환 함수

- cv2.getRotationMatrix2D

- Affine transformation matrix 반환
    - parameter: center, angle, scale) → M
    - angle > 0 이면 반 시계방향 회전

$$\begin{bmatrix} \alpha & \beta & (1 - \alpha) \cdot \text{center.x} - \beta \cdot \text{center.y} \\ -\beta & \alpha & \beta \cdot \text{center.x} + (1 - \alpha) \cdot \text{center.y} \end{bmatrix}$$

$$\begin{aligned} \alpha &= \text{scale} \cdot \cos \text{angle}, \\ \beta &= \text{scale} \cdot \sin \text{angle} \end{aligned}$$

- cv2.warpAffine

- 변환을 영상에 적용
    - parameter: src, M, dsize[,dst[,flags[, borderMode[, borderValue]]]]) → dst
    - borderMode: cv2.BORDER\_CONSTANT



# 영상 크기 변환과 회전

---

- `imutils`
  - `imutils.translate(image, tx, ty)`
  - `imutils.rotate(image, angle=deg, center=(cx, cy))`
  - `imutils.resize(image, width=w, height=h, inter = interpolation)`

# 산술, 비트, 비교범위, 수치 연산 함수

---

## ■ 사칙연산

- `cv2.add (src1, src2, ) → dst`
- `cv2.addWeighted (src1, alpha, src2, beta, gamma, ) → dst`
- `cv2.subtract(src1, src2, ) → dst`
- `cv2.scaleAdd(src1, alpha, src2) → dst`
- `cv2.multiply(src1, src2, ) → dst`
- `cv2.divide(src1, src2, ) → dst`

## ■ 비트연산

- `cv2.bitwise_not (src, ) → dst`
- `cv2.bitwise_and (src1, src2, ) → dst`
- `cv2.bitwise_or (src1, src2, ) → dst`
- `cv2.bitwise_xor (src1, src2, ) → dst`



# 산술, 비트, 비교범위, 수치 연산 함수

---

- 비교범위연산

- `cv2.compare (src1, src2, ) → dst`
- `cv2.inRange(src, lowerb, upperb) → dst`

- 수치연산

- `cv2.absdiff (src1, src2, ) → dst`
- `cv2.converScaleAbs (src, ) → dst`
- `cv2.exp(src) → dst, cv2.log(src) → dst`
- `cv2.pow(src, power, ) → dst, cv2.sqrt(src) → dst`
- `cv2.magnitude(x, y, ) → magnitude, cv2.phase(x, y, ) → angle`
- `cv2.cartToPolar(x, y, ) → magnitude, angle`
- `cv2.polarToCart(magnitude, angle, ) → x, y`



# 수학 및 통계 함수

---

- 정규화

- `cv2.norm (src1, src2, ) → retval`
- `cv2.normalize (src, dst, ) → dst`

- 최대 최소

- `cv2.min (src1, src2, ) → dst`
- `cv2.max (src1, src2, ) → dst`
- `cv2.minMaxLoc (src, ) → minVal, maxVal, minLoc, maxLoc`



# 수학 및 통계 함수

---

## ■ 통계

- `cv2.countNonzero (src) → retval`
- `cv2.reduce(src, dim, ) → dst`
- `cv2.mean(src, ) → retval`
- `cv2.meanStdDev(src, ) → mean, stddev`
- `cv2.calcCovarMatrix(samples, ) → covar, mean`

## ■ 난수

- `cv2.randu (dst, low, high) → dst`
- `cv2.randn (dst, mean, stddev) → dst`
- `cv2.randShuffle (dst) → dst`





# 수학 및 통계 함수

---

- 선형대수

- `cv2.eigen (src, ) → retval, eigenvalues, eigenvectors`
- `cv2.PCACompute (data, mean, ) → mean, eigenvectors`
- `cv2.PCAProject (data, mean, ) → result`
- `cv2.PCABackProject (data, mean, ) → result`

- 정렬

- `cv2.sort (src, ) → dst`



# 화이트 보드

---



# 영상처리 프로그래밍 기초

---

- Python으로 배우는 OpenCV 프로그래밍
  - 김동근 지음
  - 가메출판사, 2018

