**교육일지**

**교육 제목 : 비전**

**교육 장소 : YGL C6 강의실**

**교육 일시 : 2021/11/16**

**기하학적 변환 (Geometric transformation)**

**# warpAffine(src, M, dsize[, dst[, flags[, borderMode[, borderValue]]]]) -> dst**

**# src: 입력영상**

**# M: affine transform matrix (size: 2 x 3)**

**# dsize: 출력영상 크기, (0, 0) = 입력영상크기로 출력**

**# borderValue: 값이 없는 영역을 채우는 값, default = 0**

**src = cv2.imread('fig/puppy.bmp')**

**if src is None:**

**print('Image load failed!')**

**sys.exit()**

**affine = np.array([[1, 0, 200],**

**[0, 1, -100]], dtype=np.float32)**

**dst = cv2.warpAffine(src, affine, (0, 0))**

**cv2.imshow('src', src)**

**cv2.imshow('dst', dst)**

**cv2.waitKey()**

**cv2.destroyAllWindows()**

**영상확대(Scaling)**

**# resize(src, dsize[, dst[, fx[, fy[, interpolation]]]]) -> dst**

**# src: 입력영상**

**# dsize: 출력영상의 크기**

**# fx, fy: 0, 0**

**# interpolation**

**src = cv2.imread('fig/rose.jpg') # src.shape=(320, 480)**

**print(src.shape)**

**if src is None:**

**print('Image load failed!')**

**sys.exit()**

**dst1 = cv2.resize(src, (1920, 1280), interpolation=cv2.INTER\_NEAREST)**

**dst2 = cv2.resize(src, (1920, 1280)) # cv2.INTER\_LINEAR (Bilinear)**

**dst3 = cv2.resize(src, (1920, 1280), interpolation=cv2.INTER\_CUBIC)**

**dst4 = cv2.resize(src, (1920, 1280), interpolation=cv2.INTER\_LANCZOS4)**

**cv2.imshow('src', src)**

**cv2.imshow('INTER\_NEAREST', dst1[800:1600, 800:1200])**

**cv2.imshow('Bilinear', dst2[800:1600, 800:1200])**

**cv2.imshow('INTER\_CUBIC', dst3[800:1600, 800:1200])**

**cv2.imshow('INTER\_LANCZOS4', dst4[800:1600, 800:1200])**

**cv2.waitKey()**

**cv2.destroyAllWindows()**

**회전변환 (Rotation)**

**# warpAffine(src, M, dsize[, dst[, flags[, borderMode[, borderValue]]]]) -> dst**

**# src: 입력영상**

**# M: affine transform matrix (size: 2 x 3)**

**# dsize: 출력영상 크기, (0, 0) = 입력영상크기로 출력**

**# borderValue: 값이 없는 영역을 채우는 값, default = 0**

**# getRotationMatrix2D(center, angle, scale) -> retval**

**# center: 영상의 center**

**# angle: 회전각도**

**# scale: 확대율**

**src = cv2.imread('fig/puppy.bmp')**

**if src is None:**

**print('Image load failed!')**

**sys.exit()**

**# rad = 20 \* np.pi / 180**

**# affine = np.array([[np.cos(rad), np.sin(rad), 0],**

**# [-np.sin(rad), np.cos(rad), 0]], dtype=np.float32)**

**cp = (src.shape[1] / 2, src.shape[0] / 2)**

**affine = cv2.getRotationMatrix2D(cp, 20, 1)**

**dst = cv2.warpAffine(src, affine, (0, 0))**

**cv2.imshow('src', src)**

**cv2.imshow('dst', dst)**

**cv2.waitKey()**

**cv2.destroyAllWindows()**