## ■ numpy 모듈

- python list: 여러 데이터 형의 값을 저장할 수 있는 강력한 구조
- 연산 속도가 빠른 기본 리스트 선호 및 동일 데이터 형으로 구성
  예) 데이터 분석, 영상 처리 등
- list 요소는 index로 참조, numpy 차원은 axis

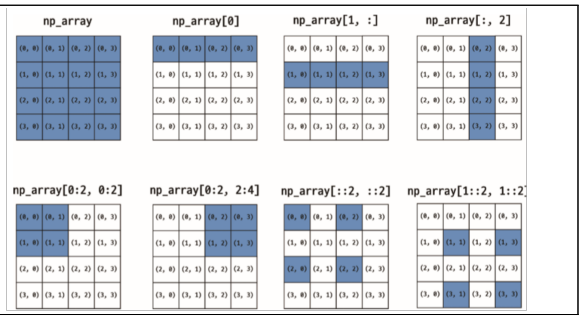
| python list | numpy module |
|---|---|
| scores1 = [10, 20, 30]<br>scores2 = [70, 80, 90]<br>total = scores1 + scores2<br># [10, 20, 30, 70, 80, 90] | import numpy as np<br>scores1 = np.array([10, 20, 30])<br>scores2 = np.array([70, 80, 90])<br>total = scores1 + scores2<br># [80, 100, 120] |

- numpy 핵심: 다차원배열 ndarray
- ndarray 정보 조회: ndim(축 개수), shape(m x n), size(원소 개수), dtype(원소 데이터형), itemize(바이트 단위), data(버퍼), stride(다음 요소 접근을 위한 바이트 크기)
- 연산: 데이터형에 기초 python 연산자 그대로 사용, *, **, /, ...
- Indexing과 Slicing: 예) scores[2], scores[1:4]
- logical indexing: 주어진 조건 하에 원하는 값을 추려내는 것

| | |
|---|---|
| ages = np.array([18, 19, 25])<br>y = ages > 20<br># y = array([False, False, True]) | y = ages[ages>20]<br># y = array([25]) |

- 2D array value & slicing

```
import numpy as np
y = [ [1,2,3], [4,5,6], [7,8,9]]
np_array = np.array(y)
np_array[0,0] = 12 # okay
np_array[2,2] = 1.234 # 값 변환: 1
np_array[0:2, 1:3] #[[2,3],[5,6]]
```




- arange(), range()

| | |
|---|---|
| np.arange(1,6) #[1,2,3,4,5]<br>np.arange(1,10,2) #[1,3,5,7,9] | np.array(range(5))<br>#[0,1,2,3,4] |

- linspace(), logspace()

| | |
|---|---|
| np.linespace(0,10,100)<br>#0, 10까지 총 100개의 수들 생성 | np.logspace(0,5,10)<br>#log0부터 log5까지 10개의 수 생성 |

- reshape(), flatten()

| | |
|---|---|
| y=np.arange(12) #[0, ... 11]<br>y.reshape(3,4)<br>#[[0,1,2,3],[4,5,6,7],[8,9,10,11]] | y.flatten()<br># 2차원 배열을 1차원 배열로 |

- 난수와 통계

| | |
|---|---|
| np.random.seed(100)<br>np.random.rand(5) # randn, randint | np.mean(...) # 평균<br>np.median(...) # 중앙값<br>np.corrcoef(x, y, ...) # 상관계수 |



동의대학교 김남규(ngkim@deu.ac.kr)

## ■ OpenCV

- concole 창에서 opencv python install:
  pip install opencv-python / pip install opencv-contrib-python
- 이미지 그리기

```
import cv2

img_gray = cv2.imread('./images/mandrill.jpg', cv2.IMREAD_GRAYSCALE)
img_color = cv2.imread('./images/mandrill.jpg', cv2.IMREAD_COLOR)

cv2.imshow('grayscale', img_gray)    # 회색조 이미지로 화면에 표시
cv2.imshow('color image', img_color) # 컬러 이미지로 화면에 표시

# 다음 두 행은 키보드 입력을 기다렸다가 모든 창을 끄고 종료하는 코드
cv2.waitKey(0)
cv2.destroyAllWindows()
# cv2.waitKey(1)  # 화면 Holding 되어 있을 때
```

- 이미지 위에 그리기, 글자 등

```
cv2.line(img, (0,0), (200,200), (0,0,255), 5) # 직선 시작/끝점, 색상, 두께
cv2.rectangle(img, (0,200), (200,20), (0,0,0), 5)  # 사각형의
cv2.putText(img, "hello", (70,70), fontFace = 2,
            fontScale = 1, color = (0,0,0)) # 텍스트
cv2.imshow('lined', img)
```

- 이미지 합성

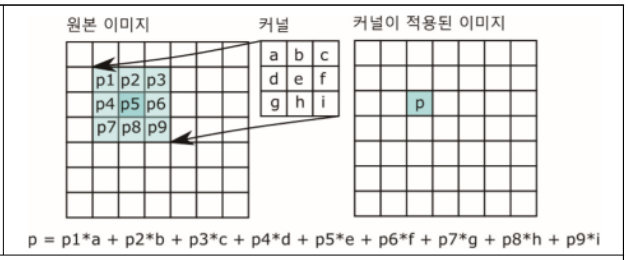
```
cv2.addweighted(image_a, weight_for_a, image_b, weight_for_b, gamma)
```

- 픽셀 영역 설정 → 마스크 이미지 → 필터링된 이미지

```
cv2.inRange(image, low_value, high_value)
cv2.bitwise_and(image1, image2, dst, mask)
```

- 이미지 필터링

```
kernel = np.ones((3, 3),
                 np.float32) / 9
averaged33 = cv2.filter2D(org, -1,
                          kernel)
cv2.imshow('original', org)
cv2.imshow('filtered',
           averaged33)
```




```
cv2.GaussianBlur(original_image, (3,3), 1)
cv2.medianBlur(original_image,  5)
cv2.bilateralFilter(original_image, 9, 50, 50)
cv2.threshold(src_image, thresh_value, maxValue, thresh_option)
cv2.adaptiveThreshold(img,value, method, thresholdType, blocksize, C)
```



동의대학교 김남규(ngkim@deu.ac.kr)

## ■ Python 영상 I/O

```
import numpy as np
from PIL import Image, ImageFont, ImageDraw
from PIL.ImageChops import add, subtract, multiply, difference, screen
import PIL.ImageStat as stat
from skimage.io import imread, imsave, imshow, show,
     imread_collection, imshow_collection
from skimage import color, viewer, exposure, img_as_float, data
from skimage.transform import SimilarityTransform, warp, swirl
from skimage.util import invert, random_noise, montage
import matplotlib.image as mpimg
import matplotlib.pylab as plt
from scipy.ndimage import affine_transform, zoom
from scipy import misc
```

- PIL 사용하기: Python Image Library → Pillow

```
im = Image.open("parrot.png") # read the image, provide the correct path
print(im.width, im.height, im.mode, im.format, type(im))
     # 453 340 RGB PNG <class 'PIL.PngImagePlugin.PngImageFile'>
im.show() # display the image
```

```
im_g = im.convert('L')  # convert the RGB color image to a grayscale image
im_g.save('parrot_gray.png') # save the image to disk
Image.open("parrot_gray.png").show() # read the grayscale image
```

- Matplotlib 사용하기

```
im = mpimg.imread("hill.png") # read the image as a numpy ndarray
print(im.shape, im.dtype, type(im))
      # (960, 1280, 4) float32 <class 'numpy.ndarray'>
plt.figure(figsize=(10,10))
plt.imshow(im)   # display the image
plt.axis('off'), plt.show()
```

```
im1 = im
im1[im1 < 0.5] = 0 # make the image look darker
plt.imshow(im1)
```

```
im = mpimg.imread("lena_small.jpg") # read the image as a numpy ndarray
methods = ['none', 'nearest', 'bilinear', 'bicubic', 'spline16', 'lanczos']
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(15, 12),
                         subplot_kw={'xticks': [], 'yticks': []})
fig.subplots_adjust(hspace=0.05, wspace=0.05)
for ax, interp_method in zip(axes.flat, methods):
    ax.imshow(im, interpolation=interp_method)
    ax.set_title(str(interp_method), size=20)
plt.tight_layout(), plt.show()
```

- scikit-image 사용하기

```
im = imread("parrot.png") # read image
hsv = color.rgb2hsv(im) # from RGB to HSV color space
hsv[:, :, 1] = 0.5 # change the saturation
im1 = color.hsv2rgb(hsv) # from HSV back to RGB
imsave('parrot_hsv.png', im1) # save image to disk
im = imread("parrot_hsv.png")
plt.axis('off'), imshow(im), show()
```

```
im = data.astronaut(), imshow(im), show()
```

- scipy misc

```
im = misc.face() # load the raccoon's face image
imsave('face.png', im) # uses the Image module (PIL)
plt.imshow(im), plt.axis('off'), plt.show()
```

```
import imageio
im = imageio.imread('../images/pepper.jpg')
print(type(im), im.shape, im.dtype)
plt.imshow(im), plt.axis('off'), plt.show()
```

## ■ 영상 데이터 구조 전환

- PIL → numpy ndarray(skimage, scipy, scikit-image)

```
im = Image.open('flowers.png') # read image into an Image object with PIL
im = np.array(im) # create a numpy ndarray from the Image object
imshow(im) # use skimage imshow to display the image
plt.axis('off'), show()
```

- numpy ndarray → PIL

```
im = imread('flowers.png') # read image into numpy ndarray with skimage
im = Image.fromarray(im) # create a PIL Image object from the numpy ndarray
im.show() # display the image with PIL Image.show() method
```

- numpy ndarray을 활용한 이미지/데이터 slicing

```
lena = mpimg.imread("lena.jpg") # read the image as a numpy ndarray
                                # read-only lena = lena.copy()
lx, ly, _ = lena.shape
X, Y = np.ogrid[0:lx, 0:ly]
mask = (X - lx / 2) ** 2 + (Y - ly / 2) ** 2 > lx * ly / 4
lena[mask,:] = 0 # masks
plt.figure(figsize=(10,10))
plt.imshow(lena), plt.axis('off'), plt.show()
```

## ■ 파일 및 컬러 형식 전환

- png → jpg

```
im = Image.open("parrot.png")
print(im.mode) # RGB
im.save("parrot.jpg")
```

- RGBA → RGB

```
im = Image.open("hill.png")
print(im.mode) # RGBA
im.convert('RGB').save("hill.jpg") # first convert to RGB mode
```

- RGB → Gray

```
im = imread("parrot.png", as_gray=True)
```

```
im = imread("Ishihara.png")
im_g = color.rgb2gray(im)
plt.subplot(121), plt.imshow(im, cmap='gray'), plt.axis('off')
plt.subplot(122), plt.imshow(im_g, cmap='gray'), plt.axis('off')
plt.show()
```

## ■ 변환 경험하기 및 변환의 종류

- Alpha blending

```
im1 = mpimg.imread("messi.jpg") / 255 # scale RGB values in [0,1]
im2 = mpimg.imread("ronaldo.jpg") / 255
i = 1
plt.figure(figsize=(18,15))
for alpha in np.linspace(0,1,20):
    plt.subplot(4,5,i)
    plt.imshow((1-alpha)*im1 + alpha*im2)
    plt.axis('off')
    i += 1
plt.subplots_adjust(wspace=0.05, hspace=0.05)
plt.show()
```

- crop(잘라내기), resize(크기변환), 화소변환: negating, log, power-law, reflecting, rotating, affince transformation ...



동의대학교 김남규(ngkim@deu.ac.kr)

## ■ 영상 다루기

- 이미지의 화소값 바꾸기

```
# choose 5000 random locations inside image
im1 = im.copy() # keep the original image, create a copy
n = 5000
x, y = np.random.randint(0, im.width, n), np.random.randint(0, im.height,n)
for (x,y) in zip(x,y):
    im1.putpixel((x, y),
                 ((0,0,0) if np.random.rand() < 0.5 else (255,255,255)))
    # salt-and-pepper noise
im1.show()
```

- 이미지 위에 그리기

```
im = Image.open("parrot.png")
draw = ImageDraw.Draw(im)
draw.ellipse((125, 125, 200, 250), fill=(255,255,255,128))
del draw
im.show()
```

- 이미지 위에 텍스트

```
im = Image.open("parrot.png")
draw = ImageDraw.Draw(im)
font = ImageFont.truetype("arial.ttf", 23) # use a truetype font
draw.text((10, 5), "Welcome to image processing with python", font=font)
del draw
im.show()
```

- 썸네일 만들기

```
im = Image.open("parrot.png")
im_thumbnail = im.copy() # need to copy the original image first
im_thumbnail.thumbnail((100,100)) # now paste the thumbnail on the image
im.paste(im_thumbnail, (10,10))
im.save("parrot_thumb.jpg")
im.show()
```

- 이미지 통계

```
s = stat.Stat(im)
print(s.extrema) # maximum and minimum pixel values for each channel R, G, B
print(s.count)
print(s.mean)
print(s.median)
print(s.stddev)
```

```
pl = im.histogram()
plt.bar(range(256), pl[:256], color='r', alpha=0.5)
plt.bar(range(256), pl[256:2*256], color='g', alpha=0.4)
plt.bar(range(256), pl[2*256:], color='b', alpha=0.3)
plt.show()
```

- 채널 분리와 합치기

```
im = Image.open("parrot.png")
ch_r, ch_g, ch_b = im.split() # split the RGB image into 3 channels: R, G, B
# we shall use matplotlib to display the channels
plt.figure(figsize=(18,6))
plt.subplot(1,3,1); plt.imshow(ch_r, cmap=plt.cm.Reds); plt.axis('off')
plt.subplot(1,3,2); plt.imshow(ch_g, cmap=plt.cm.Greens); plt.axis('off')
plt.subplot(1,3,3); plt.imshow(ch_b, cmap=plt.cm.Blues); plt.axis('off')
plt.tight_layout()
plt.show() # show the R, G, B channels
```

```
im = Image.merge('RGB', (ch_b, ch_g, ch_r)) # swap the red and blue channels
obtained last time with split()
im.show()
```



동의대학교 김남규(ngkim@deu.ac.kr)