FVE3011

OpenCV in Python

담당교수: 최 학남

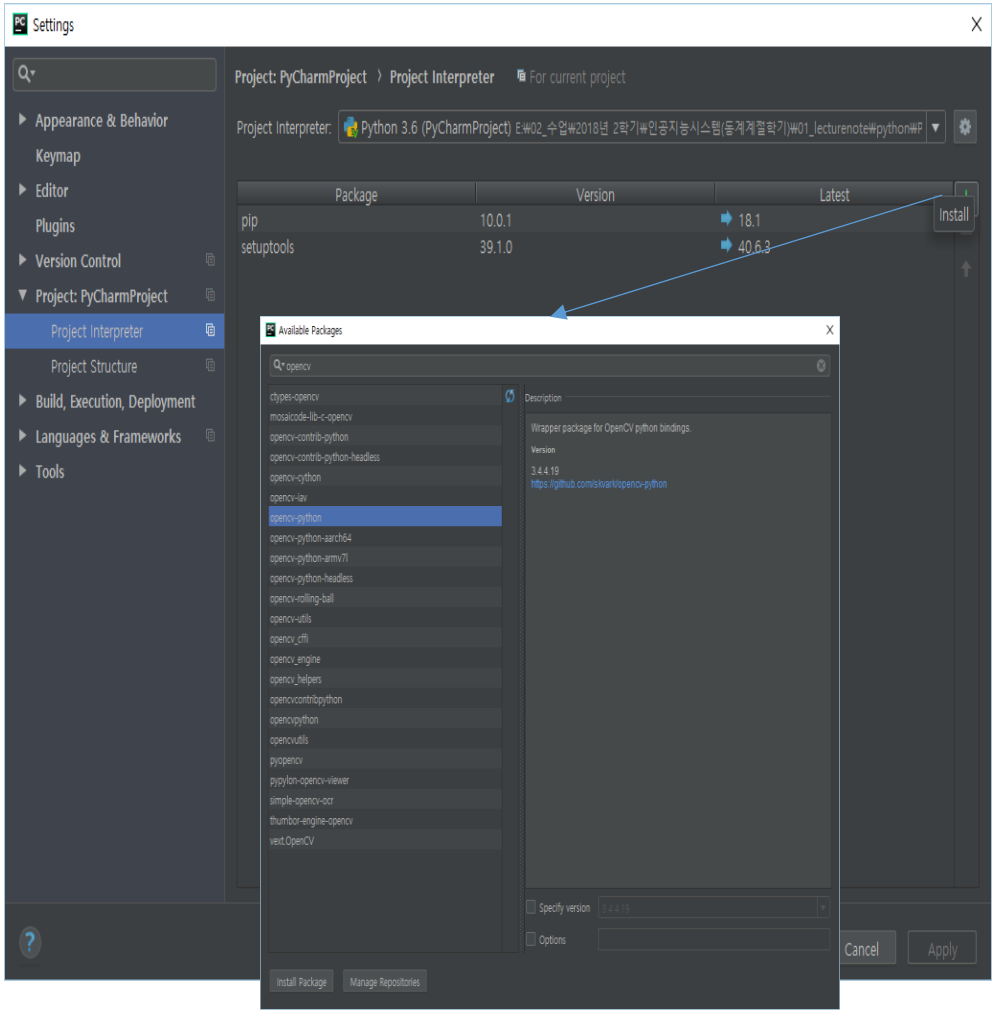
인하대학교

Contents

- OpenCV 설치
- Read and display images using OpenCV
- Data augmentation using OpenCV

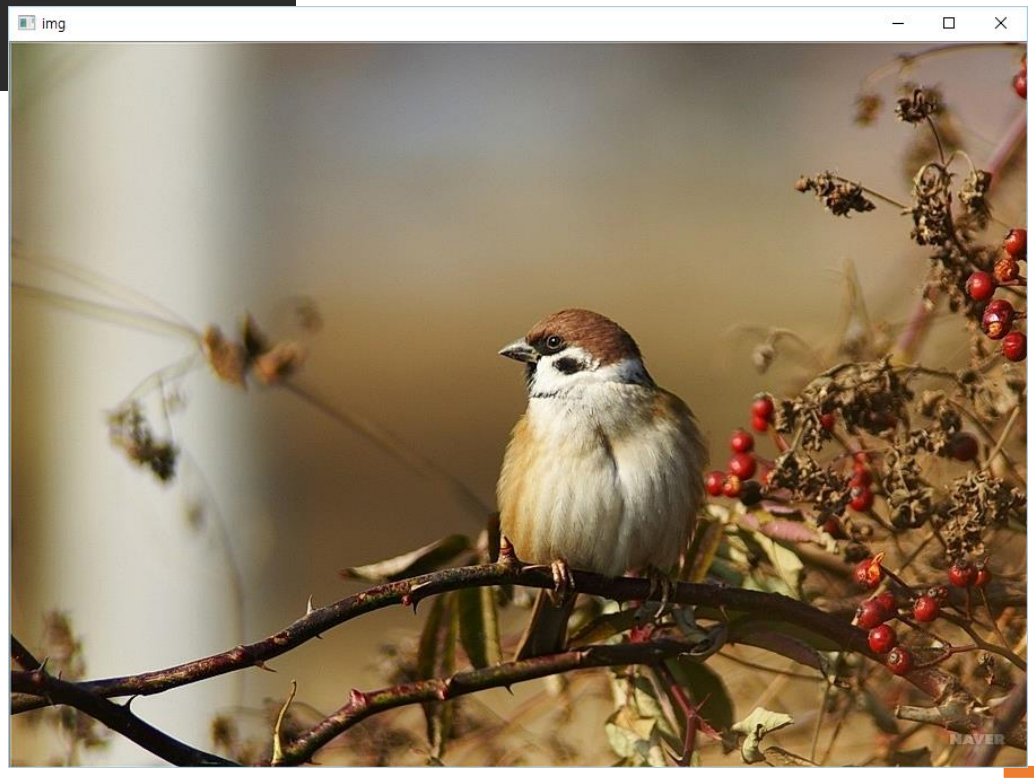
OpenCV 설치

- Anaconda 설치
  - https://www.anaconda.com/download/
- Anaconda prompt
  - pip install opencv-python
  - conda install -c menpo opencv3
- Pycharm
  - File → settings → project: → project interpreter
- Python program
  - import cv2



Read and display image

```
import cv2
img = cv2.imread('img.jpg')
cv2.imshow('img',img)
cv2.waitKey(0)
```



Smoothing images

- Filter2D function
  - Need to design the filter
  - Apply the convolution with designed filter

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

img = cv.imread('opencv_logo.png')

kernel = np.ones((5,5),np.float32)/25
dst = cv.filter2D(img,-1,kernel)

plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(dst),plt.title('Averaging')
plt.xticks([]), plt.yticks([])
plt.show()
```

Smoothing images

- Averaging
  - blur = cv.blur(img,(5,5))
- Gaussian filter
  - blur = cv.GaussianBlur(img,(5,5),0)
- Median filter
  - median = cv.medianBlur(img,5)

Gaussian Noise

- Img = cv.imread(‘ ’);
- row,col,ch= img.shape
- mean = 0
- var = 0.5
- sigma = var**0.5
- gauss = np.random.normal(mean,sigma,(row,col,ch))
- noisyImg = Img + gauss

```
img = img + noise.astype(np.uint8)
```

Mean, var을 이용하여 노이즈 레벨을 조절

Geometric transformations of images

- Scaling
  - Resizing the image
  - cv.resize()

```
import numpy as np
import cv2 as cv

img = cv.imread('messi5.jpg')

res = cv.resize(img,None,fx=2, fy=2, interpolation = cv.INTER_CUBIC)

#OR

height, width = img.shape[:2]
res = cv.resize(img,(2*width, 2*height), interpolation = cv.INTER_CUBIC)
```

Geometric transformations of images

- Translation

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

```
import numpy as np
import cv2 as cv

img = cv.imread('messi5.jpg',0)
rows,cols = img.shape

M = np.float32([[1,0,100],[0,1,50]])
dst = cv.warpAffine(img,M,(cols,rows))

cv.imshow('img',dst)
cv.waitKey(0)
cv.destroyAllWindows()
```

image

Geometric transformations of images

- Rotation
  - Rotation of an image for an angle theta is achieved by the transformation matrix of the form

$$M = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

```
img = cv.imread('messi5.jpg',0)
rows,cols = img.shape

M = cv.getRotationMatrix2D((cols/2,rows/2),90,1)
dst = cv.warpAffine(img,M,(cols,rows))
```

Geometric transformations of images

- Affine transformation

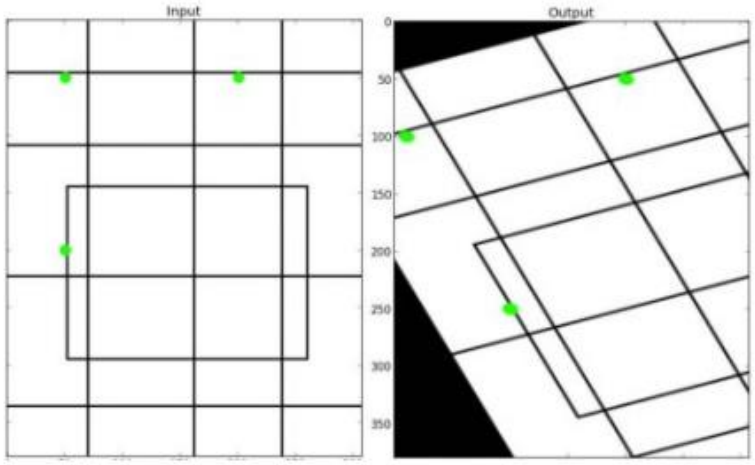
```
img = cv.imread('drawing.png')
rows,cols,ch = img.shape

pts1 = np.float32([[50,50],[200,50],[50,200]])
pts2 = np.float32([[10,100],[200,50],[100,250]])

M = cv.getAffineTransform(pts1,pts2)

dst = cv.warpAffine(img,M,(cols,rows))

plt.subplot(121),plt.imshow(img),plt.title('Input')
plt.subplot(122),plt.imshow(dst),plt.title('Output')
plt.show()
```



image

Geometric transformations of images

- Perspective transformation

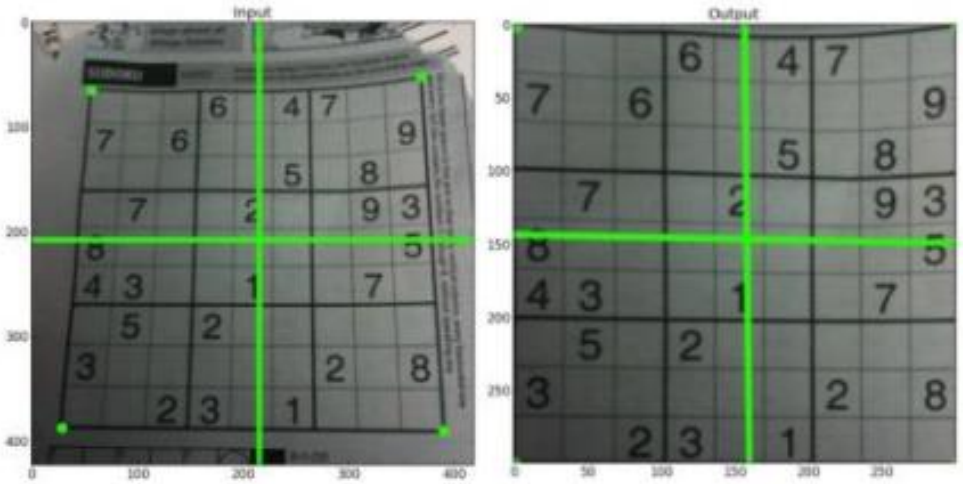
```
img = cv.imread('sudoku.png')
rows,cols,ch = img.shape

pts1 = np.float32([[56,65],[368,52],[28,387],[389,390]])
pts2 = np.float32([[0,0],[300,0],[0,300],[300,300]])

M = cv.getPerspectiveTransform(pts1,pts2)

dst = cv.warpPerspective(img,M,(300,300))

plt.subplot(121),plt.imshow(img),plt.title('Input')
plt.subplot(122),plt.imshow(dst),plt.title('Output')
plt.show()
```



image

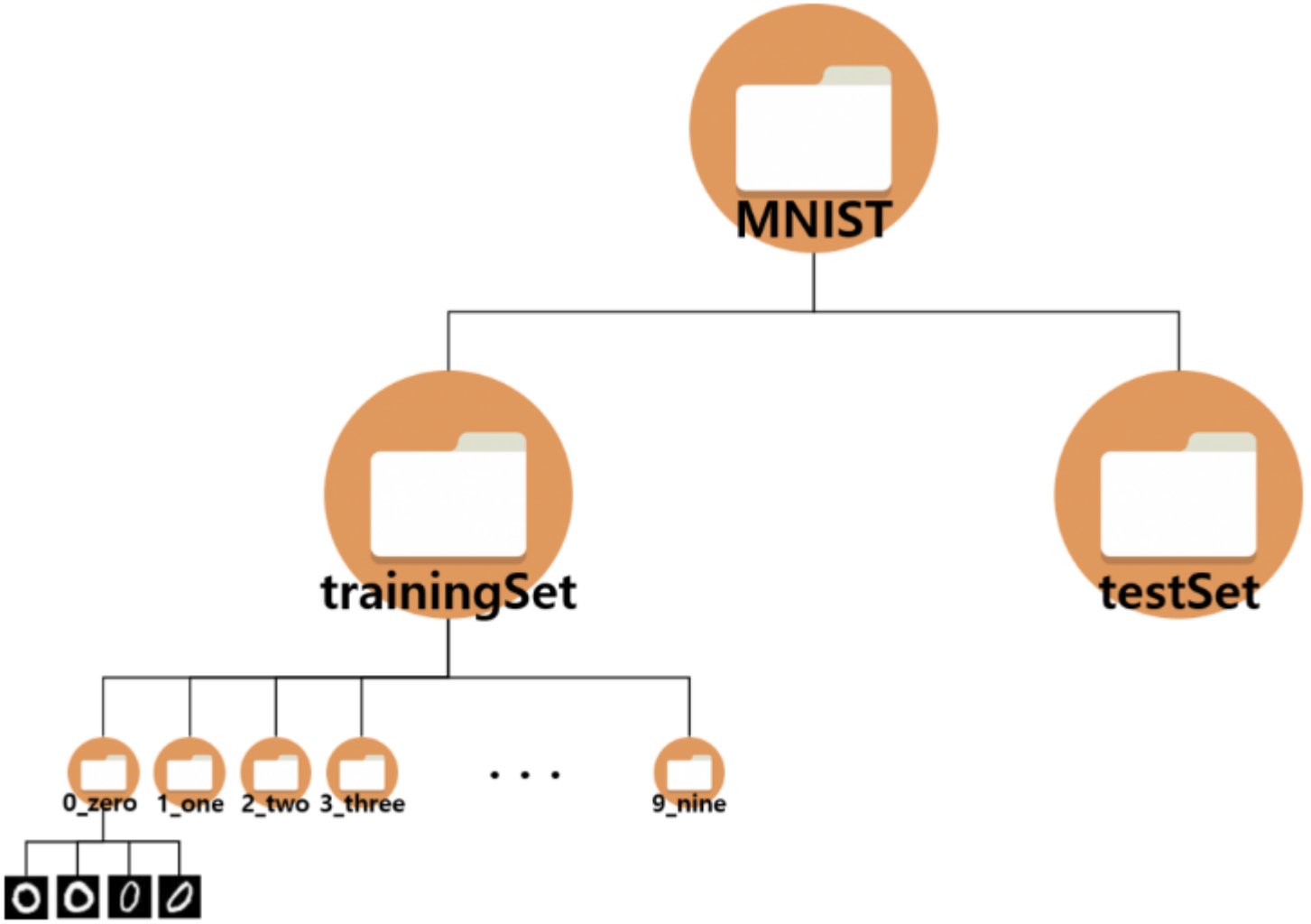
Data augmentation

- MNIST data
  - Download from http://yann.lecun.com/exdb/mnist/
  - Binary files
  - Pycharm setting
    - File → settings → project: → project interpreter
      - python-mnist 추가

```
from mnist import MNIST

mndata = MNIST('MNIST/')
mndata.gz = True
print(mndata)
images, labels = mndata.load_training()
```

MNIST 구조



Homework: data augmentation

HW1: data augmentation

- 4중 for(while)문을 사용하여 data augmentation을 진행하시오.
  - Blur → 2
    - Filter size: 3x3
    - Filter size: 5x5
  - Noise → 10
    - Gaussian (Normal) noise: mean=0, std: 1~20, 2씩 증가
  - Translation → 10
    - (0,0) → (tx, ty)
    - tx:1~10, 1씩 증가
    - ty:1~10, 1씩 증가
  - Rotation → 11
    - (-5 ~5) degree, 1도 간격으로 증가