

Python 데이터 분석과 이미지 처리

나동빈

KNN 숫자 인식 예제

숫자 인식 예제

- 이미지 다운로드 (digits.png)
- https://raw.githubusercontent.com/ndb796/Python-OpenCV-Basic/master/knn_example/digits.png

KNN 숫자 인식 예제

숫자 인식 예제 코드 knn_trainer.py ①

```
import cv2
import numpy as np

img = cv2.imread('digits.png')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 세로로 50줄, 가로로 100줄로 사진을 나눕니다.
cells = [np.hsplit(row, 100) for row in np.vsplit(gray, 50)]
x = np.array(cells)

# 각 (20 X 20) 크기의 사진을 한 줄(1 X 400)으로 바꿉니다.
train = x[:, :].reshape(-1, 400).astype(np.float32)

# 0이 500개, 1이 500개, ... 로 총 5,000개가 들어가는 (1 x 5000) 배열을 만듭니다.
k = np.arange(10)
train_labels = np.repeat(k, 500)[ :, np.newaxis]

np.savez("trained.npz", train=train, train_labels=train_labels)
```

KNN 숫자 인식 예제

숫자 인식 예제 코드 knn_trainer.py ②

```
print(x[0, 5].shape)

# 다음과 같이 하나씩 글자를 출력할 수 있습니다.
cv2.imshow('Image', x[0, 5])
cv2.waitKey(0)

# 다음과 같이 하나씩 글자를 저장할 수 있습니다.
cv2.imwrite('test0.png', x[0, 0])
cv2.imwrite('test1.png', x[5, 0])
cv2.imwrite('test2.png', x[10, 0])
cv2.imwrite('test3.png', x[15, 0])
cv2.imwrite('test4.png', x[20, 0])
cv2.imwrite('test5.png', x[25, 0])
cv2.imwrite('test6.png', x[30, 0])
cv2.imwrite('test7.png', x[35, 0])
cv2.imwrite('test8.png', x[40, 0])
cv2.imwrite('test9.png', x[45, 0])
```

KNN 숫자 인식 예제

숫자 인식 예제 코드 run.py ①

```
import cv2
import numpy as np
import glob

FILE_NAME = 'trained.npz'

# 파일로부터 학습 데이터를 불러옵니다.
def load_train_data(file_name):
    with np.load(file_name) as data:
        train = data['train']
        train_labels = data['train_labels']
    return train, train_labels

# 손 글씨 이미지를 (20 x 20) 크기로 Scaling합니다.
def resize20(image):
    img = cv2.imread(image)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    gray_resize = cv2.resize(gray, (20, 20))
    # 최종적으로는 (1 x 400) 크기로 반환합니다.
    cv2.imshow('Image', gray_resize)
    return gray_resize.reshape(-1, 400).astype(np.float32)
```

KNN 숫자 인식 예제

숫자 인식 예제 코드 run.py ②

```
def check(test, train, train_labels):
    knn = cv2.ml.KNearest_create()
    knn.train(train, cv2.ml.ROW_SAMPLE, train_labels)
    # 가장 가까운 5개의 글자를 찾아, 어떤 숫자에 해당하는지 찾습니다.
    ret, result, neighbours, dist = knn.findNearest(test, k=5)
    return result

train, train_labels = load_train_data(FILE_NAME)

for file_name in glob.glob('./test*.png'):
    test = resize20(file_name)
    result = check(test, train, train_labels)
    print(result)
    cv2.waitKey(0)

cv2.destroyAllWindows()
```