Python 데이터 분석과 이미지 처리

나동빈

중급 Captcha Hacking 4 – KNN 모델 학습

KNN 모델 학습하기: (knn_trainer.py)

```
import os
import cv2
import numpy as np

file_names = list(range(0, 13))
train = []
train_labels = []

for file_name in file_names:
    path = './training_data/' + str(file_name) + '/'
    file_count = len(next(os.walk(path))[2])
    for i in range(1, file_count + 1):
        img = cv2.imread(path + str(i) + '.png')
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        train.append(gray)
        train_labels.append(file_name)

x = np.array(train)
train = x[:, :].reshape(-1, 400).astype(np.float32)
train_labels = np.array(train_labels)[:, np.newaxis]

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

중급 Captcha Hacking 4 – KNN 모델 학습

학습된 모델로 테스트 하기: 학습된 KNN 모델 불러오기 (run.py)

```
import numpy as np
import cv2
import utils

FILE_NAME = "trained.npz"

# 각 글자의 (1 x 400) 데이터와 정답 (0 ~ 9, +, *)
with np.load(FILE_NAME) as data:
    train = data['train']
    train_labels = data['train_labels']

knn = cv2.ml.KNearest_create()
knn.train(train, cv2.ml.ROW_SAMPLE, train_labels)


def check(test, train, train_labels):
    # 가장 가까운 K개의 글자를 찾아, 어떤 숫자에 해당하는지 찾습니다. (테스트 데이터 개수에 따라서 조절)
    ret, result, neighbours, dist = knn.findNearest(test, k=1)
    return result
```

중급 Captcha Hacking 4 - KNN 모델 학습

학습된 모델로 테스트 하기: 테스트 결과 확인하기 (run.py)

```
def get_result(file_name):
    image = cv2.imread(file_name)
    chars = utils.extract_chars(image)
    result_string = ""
    for char in chars:
        matched = check(utils.resize20(char[1]), train, train_labels)
        if matched < 10:
            result_string += str(int(matched))
            continue
        if matched == 10:
            matched = '+'
        elif matched == 11:
            matched = '-'
        elif matched == 12:
            matched = '*'
        result_string += matched
    return result_string

print(get_result("image_2.png"))
```