

front.py: 캡차 데이터 만드는 웹의 프론트엔드가 구현된 코드

- prob200(): main.html을 렌더링 하는 함수
- img(path): 이미지의 파일 이름을 입력받아 서버에 있는 해당 이미지를 리턴해주는 함수
- start(): 캡차 데이터 만드는 과정을 실행하는 함수, 프론트엔드에 만들어진 캡차 이미지 띄움
- check(): 사용자가 입력한 캡차 문자열이 만들어진 캡차 이미지와 맞는지 확인하는 함수

gener.py: 캡차 이미지 데이터 만드는 코드

- generate(string, path): string을 입력으로 받아 문자열에 맞는 이미지를 제작 후 경로에 저장하는 함수
- gen(length, path): 랜덤한 length 길이의 캡처 문자열을 만드는 함수

test.py: utils에 구현한 함수들이 제대로 작동하는지 확인하는 코드

utils.py: 캡차 이미지 해킹을 위한 함수들이 모여 있는 코드

- remove_first_0(string): 문자열에서 맨 앞에 오는 0을 제거하는 함수
 - get_chars(image, color): 캡차 이미지에서 color에 해당하는 색의 문자들을 추출하여 grayscale 하는 함수
 - extract_chars(image): 캡차 이미지에서 문자 하나하나 분리하는 함수.
1. get_chars를 이용해 문자들을 gray_scale
 2. findContours 함수를 통해 각 문자들의 경계선을 추출하여
 3. 해당 경계선에 맞게 각 문자들을 자른 뒤 문자들의 list를 반환
- resized20(image): 해당 이미지를 리사이즈 하는 함수 20x20 배열의 이미지로 리사이즈 한 다음 (400)의 1차원 배열을 반환

hack.py: 캡차 해킹 자동화 코드, localhost:10000/start (캡차 데이터 만드는 웹)을 통해 만들어진 캡차 이미지의 답을 자동으로 맞춰주는 코드

1. 먼저 localhost:10000/start 로 post를 보내 캡차 이미지 만들기를 실행
2. run.py의 get_result 함수를 통해 만들어진 이미지의 정답을 자동으로 예측하여 localhost:10000/check 로 전송
3. 위 두 과정을 100번 반복

knn_trainer.py: knn 모델을 학습하는 코드

- training_data 폴더에 있는 0~9(숫자), 10~12(operator)를 읽어와 knn 모델 학습을 위한 데이터 셋을 만드는 코드

make_train_data.py: knn 모델 학습을 위한 트레인 데이터를 만드는 코드

1. 캡차 데이터 만드는 웹에서 생성된 이미지를 불러와 `utils.py`의 `extract_chars`를 이용해 한 문자씩 분할
2. 사용자가 분할된 문자를 보고 직접 해당하는 숫자 혹은 operator를 키보드로 입력하여 `training_data/0~12/`에 저장하는 코드

run.py: 캡차 데이터 만드는 웹의 백엔드가 구현된 코드

- row28-34: `knn_trainer.py`을 통해 만들어진 데이터셋인 `trained.npz`를 불러와 knn모델을 학습하는 과정
- `check(test)`: test 이미지를 불러와 knn을 이용해 캡차 이미지의 한 문자의 정답을 리턴
- `get_result(file_name)`: 캡차 이미지의 정답을 리턴하는 함수
 1. `file_name` 이미지(캡차 이미지)를 불러와 `utils.py`의 `extract_chars`를 이용해 문자들을 분리
 2. 분리된 문자들을 `check` 함수를 통해 정답을 리턴
 3. `check` 함수를 통해 얻은 정답을 취합하여 캡차 이미지의 정답을 리턴
- `app.run()`: 서버 실행 함수