

손글씨 문장 인식 프로그램 파일 설명

시스템 요구사항

- Python 버전: Python 3.10 권장 (3.10.18에서 테스트됨)
- 운영체제: Windows 10/11 (Tesseract OCR 경로 설정)

폴더 구조

- matlab/: EMNIST 데이터셋 파일 저장 폴더
- 문서/: 프로젝트 문서 및 보고서 저장 폴더
- 테스트용 이미지/: 손글씨 인식 테스트용 이미지 파일들

핵심 실행 파일

EMNIST drawing board.py (6KB)

- 기능: 마우스를 이용한 그림판 GUI 손글씨 인식 프로그램
- 설명: 검은 배경에 흰색 펜으로 직접 손글씨를 그려 EMNIST 모델로 인식
- 조작법:
 - 마우스 드래그로 글자 작성
 - 스페이스바: 예측 실행
 - c키: 화면 지우기
 - x키: 프로그램 종료

EMNIST webcam.py (6KB)

- 기능: 웹캠을 이용한 실시간 손글씨 인식 프로그램
- 설명: 웹캠 화면 중앙 ROI 영역에 손글씨를 보여주면 자동으로 문자 인식
- 조작법:
 - ROI 박스 안에 손글씨 표시
 - 스페이스바: 수동 예측 및 상위 3개 결과 출력
 - r키: ROI 박스 표시 토글
 - x키: 프로그램 종료
- 자동 기능: 1초 간격으로 실시간 자동 예측

EMNIST_tesseract_opencv.py (12KB)

- 기능: 이미지 파일에서 문장 단위 손글씨 인식
- 설명: Tesseract OCR + OpenCV를 이용해 문자를 개별 분리 후 EMNIST 모델로 일괄 인식

- **인터랙티브 메뉴:**
 - 1번: 단일 문자 이미지 예측
 - 2번: 여러문자 이미지 추출 후 일괄 예측
 - 3번: 종료
- **처리 과정:**
 1. Tesseract로 문자 좌표 추출
 2. OpenCV로 개별 문자 이미지 분리
 3. EMNIST 모델로 각 문자 예측
- **백업 기능:** Tesseract 실패 시 OpenCV 윤곽선 기반 문자 분리

모델 학습 파일

`emnist_save_import.py` (3KB)

- **기능:** import 로 얻은 EMNIST 데이터셋 모델 학습 및 저장
- **설명:** import EMNIST 데이터셋을 불러와 기본적인 신경망 모델 학습

`emnist_save_mat_balanced.py` (4KB)

- **기능:** 클래스 불균형 해결된 NIST EMNIST 모델 학습
- **특징:**
 - NIST에서 얻은 .mat 형식의 데이터셋을 불러와 모델 학습
 - 숫자/대문자/소문자별 가중치 조정
 - Dropout을 통한 과적합 방지
 - 더 복잡한 신경망 구조 (512→256→128 노드)

모델 파일 (.h5)

`emnist_model.h5` (6.768KB)

- import EMNIST 학습 모델

`emnist_model_2.h5` (6.768KB)

- 개선된 import EMNIST 모델 버전 2 (성능적으로 우수함)

`emnist_model_balanced_1.h5` (6.768KB)

- NIST 데이터셋 클래스 균형 조정 모델 버전 1

`emnist_model_balanced_2.h5` (6.768KB)

- NIST 데이터셋 클래스 균형 조정 모델 최신 (권장 사용)

유틸리티 파일

`모델간 성능 비교.py` (8KB)

- **기능:** 다양한 EMNIST 모델들의 성능 비교 및 분석
- **출력:** 각 모델별 정확도, 신뢰도, 예측 결과 비교표

`requirements.txt` (1KB)

- **내용:** 프로젝트 실행에 필요한 Python 패키지 목록
- **주요 의존성:** tensorflow, opencv-python, pytesseract, keras 등

실행 방법

1. **환경 설정:** `pip install -r requirements.txt` (파이썬 버전 3.10 권장, 그 이상의 버전은 모델 호환 불가능)
2. **Tesseract 설치:** OCR 기능 사용을 위한 별도 설치 필요
3. **모델 파일 확인:** .h5 파일들이 같은 경로에 위치해야 함
4. **프로그램 실행:** 원하는 .py 파일 실행

사용 시나리오

- **실시간 웹캠 인식:** `EMNIST_webcam.py` 사용
- **직접 그리기:** `EMNIST_drawing_board.py` 사용
- **이미지 파일 분석:** `EMNIST_tesseract_opencv.py` 사용