



# Streamlit 기반 MNIST ONNX 숫자 인식 서비스

## 1 프로젝트 개요

이번 미션 목표는 ONNX(Open Neural Network Exchange) 포맷의 MNIST 모델을 활용해 Streamlit 웹 애플리케이션 상에서 사용자가 직접 손글씨 숫자를 입력하고 그 결과를 실시간으로 예측하는 AI 추론 서비스를 구현하는 것이다.

사용자는 웹 브라우저의 캔버스 영역에 숫자를 그리고, 그 이미지를 모델 입력 규격(28x28, 1채널)에 맞게 전처리한 뒤 ONNX Runtime을 통해 추론을 수행한다. 예측된 결과는 막대차트로 시각화되며, 모델이 출력한 각 클래스(0~9)의 확률을 확인할 수 있다.

또한 추론된 이미지는 자동으로 저장되어, 사용자가 과거에 입력했던 그림과 예측 결과를 하단에서 한눈에 확인할 수 있도록 구현하였다.

## 주요 구현 기능

구분	기능 설명
입력 캔버스	<code>streamlit-drawable-canvas</code> 라이브러리를 활용하여 사용자가 직접 손글씨 숫자를 그릴 수 있는 인터페이스 제공
전처리 이미지 표시	사용자가 그린 이미지를 28×28 크기의 흑백 이미지로 변환 및 시각화
모델 추론 결과 시각화	0~9 클래스별 예측 확률을 막대 차트( <code>st.bar_chart</code> )로 시각화
이미지 저장소 (Gallery)	예측 완료 시 자동으로 이미지와 예측 결과( <code>.json</code> )를 저장하고, 저장된 결과를 목록 형태로 시각화
모델 관리 기능	MNIST ONNX 모델을 자동 다운로드( <code>utils/download_model.py</code> ) 및 캐싱( <code>@st.cache_resource</code> )하여 효율적인 세션 관리
도커화 및 배포	<code>Dockerfile</code> 을 이용해 서비스를 컨테이너화하고, Docker Hub를 통해 배포 완료

## 2 코드 설명

### 디렉토리 구조

```
mission17/team5_yena/
├── app.py
├── requirements.txt
├── Dockerfile
├── models/
├── saved_images/
└── utils/
```

```
|— path.py
|— preprocess.py
|— download_model.py
└— onnx_runtime.py
```

## 주요 코드

### app.py

Streamlit 메인 애플리케이션 파일로, 다음 영역으로 구성된다.

1. 입력 캔버스: 사용자가 숫자 그림
2. 전처리 이미지 표시: 모델 입력용 28x28 이미지 변환
3. 모델 추론 결과: onnxruntime.InferenceSession으로 예측 수행 및 확률 시각화
4. 갤러리: 저장된 이미지 및 예측 결과 확인

### utils/preprocess.py

이미지를 MNIST 모델에 맞게 전처리하는 함수가 정의되어있다.

- 흑백 변환 → 여백 제거 → 정사각형 패딩 → 28x28 리사이즈
- 배경/선 색 자동 반전 → 정규화 및 텐서 변환
- 출력: (1,1,28,28) numpy tensor, 시각화용 이미지

### utils/onnx\_runtime.py

ONNX 모델 세션을 캐싱해 효율적으로 로드하였다.

### utils/download\_model.py

필요한 ONNX 모델이 없을 경우 자동 다운로드 코드이다.

- MNIST-7, MNIST-8, MNIST-12 버전 지원
- GitHub의 ONNX Model Zoo에서 직접 fetch 후 `models/` 폴더에 저장

### Dockerfile

Python 3.12-slim 기반 경량 이미지로 빌드하였다.

1. 필수 라이브러리 설치 ( `libgl1`, `libsm6`, `libxext6`, `libxrender1` )
2. `requirements.txt` 기반 Python 패키지 설치
3. 모델 폴더( `models/` ) 및 소스 복사
4. Streamlit 서버 실행

### ◆ 주요 Docker 명령어

```
# 1. 이미지 빌드
docker build -t yena1/mnist-streamlit .
```

# 2. 컨테이너 실행

```
docker run -p 8501:8501 yena1/mnist-streamlit
```

# 3. 로컬 코드와 연동 (수정 즉시 반영)

```
docker run -p 8501:8501 -v $(pwd):/app yena1/mnist-streamlit
```

## 3 Docker Hub URL

👉 <https://hub.docker.com/r/yena1/mnist-streamlit>

# 실행 예시

```
docker pull yena1/mnist-streamlit
```

```
docker run -p 8501:8501 yena1/mnist-streamlit
```

본 프로젝트는 단순 모델 학습을 넘어, **ONNX 모델을 실 서비스 형태로 통합하는 AI 엔지니어링 실습**을 완성했다.

Streamlit, Docker, ONNX Runtime을 활용한 경량화된 추론 파이프라인을 통해 AI 모델 배포 및 시각화의 전 과정을 경험할 수 있었다.