

Best Team One

초급 프로젝트 발표



AI 6기 1팀

김영욱, 김효중, 장우정, 최무영, 최지영

목차

1. 프로젝트 소개
2. 핵심 기능 및 아키텍처
3. 모델 학습
4. 일정표 및 트러블 슈팅
5. 결과 및 시연
6. 회고 및 향후 계획



프로젝트 소개

1

프로젝트 배경

컴퓨터비전 학습 과정에서 딥러닝 기반 이미지 분류 및 객체 인식 기술 습득

2

프로젝트 목표

: 학습을 기반하여, YOLO 모델을 활용한 약물 탐지 시스템 구축, Git 협업 경험

3

기대 효과

: 의약품 식별 자동화를 통해 약물 상호 작용 및 주의 사항을 맞춤형으로 제공

Part 2 핵심 기능 및 아키텍처 (Key Features & Architecture)

약물 탐지 기능 (Object Detection)

- YOLOv8n 을 활용한 약물 객체 탐지
- 신뢰도 기반 바운딩 박스 및 생성 클래스 분류

GPT 프롬프트 엔지니어링

- 약물 정보 설명
- 사용자 상황에 맞는 1차적 진단 가능
(단, 의약품은 의료진과의 상담 후 진행)

Streamlit

- Python 기반 데이터 중심 웹 애플리케이션
- 실시간 이미지 업로드 및 예측 결과 시각화

약물 탐지 기능

GPT
프롬프트
엔지니어링

자동화 평가 시스템

Streamlit
웹 애플리
케이션

데이터
분석
(EDA)

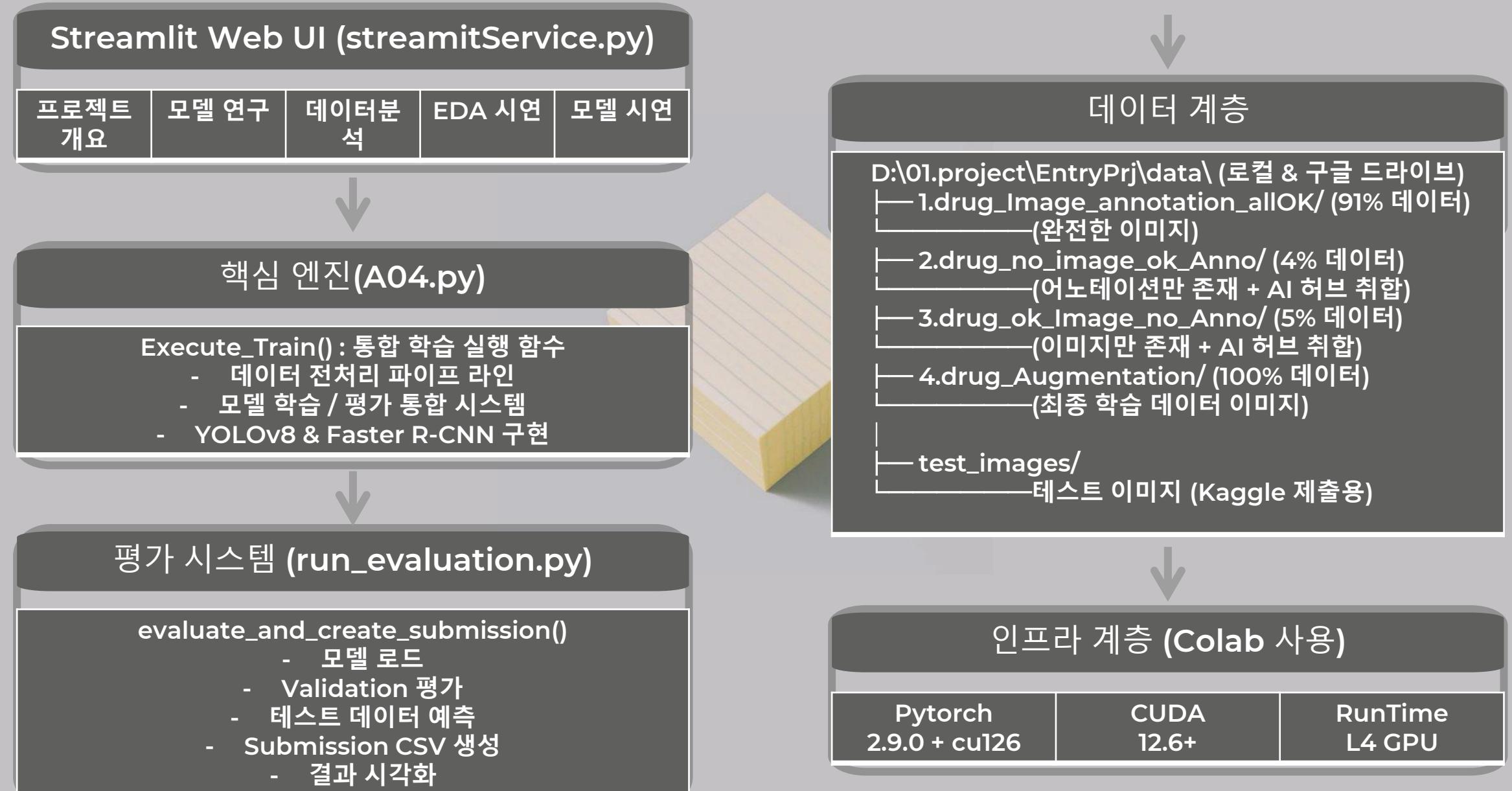
데이터 분석 (EDA)

- 이미지 : 어노테이션 매핑 분석
- 전수 식별을 통해 bbox 박스 및 약 이름 확인(약학 정보원)

자동화 평가 시스템

- Kaggle Submission 파일 생성
- mAP@[0.75:0.95] 평가 지표 계산

Part 2 핵심 기능 및 아키텍처 (Key Features & Architecture)



Part 2 핵심 기능 및 아키텍처 (Key Features & Architecture)

EntryPrj/

- |
 | data/
 | |
 | | oraldrug/
 | | |
 | | | 4.drug_Augmentation/ # 원본 이미지 + 어노테이션 없는 이미지 취합 데이터
 | | | |
 | | | | train_images/ # 학습 이미지
 | | | | train_annotations/ # 어노테이션
 | | | |
 | | | test_images/ # 테스트 이미지
- |
 | modelfiles/
 | |
 | | yolo_best.pt # 저장된 모델
 | |
 | submission/
 | |
 | | submission_*.csv # 베스트 모델 저장
 # 케글 제출 파일
 # 제출용 CSV
- |
 | src/
 | |
 | | A04.py # 소스 코드
 | |
 | | eda.py # 메인 학습 코드
 | |
 | | streamlitService.py # 데이터 분석
 # Streamlit 앱
- |
 | doc/
 | |
 | | report.pdf # 프로젝트 리포트



Part 3 모델 학습

”

YOLOv8n 구성



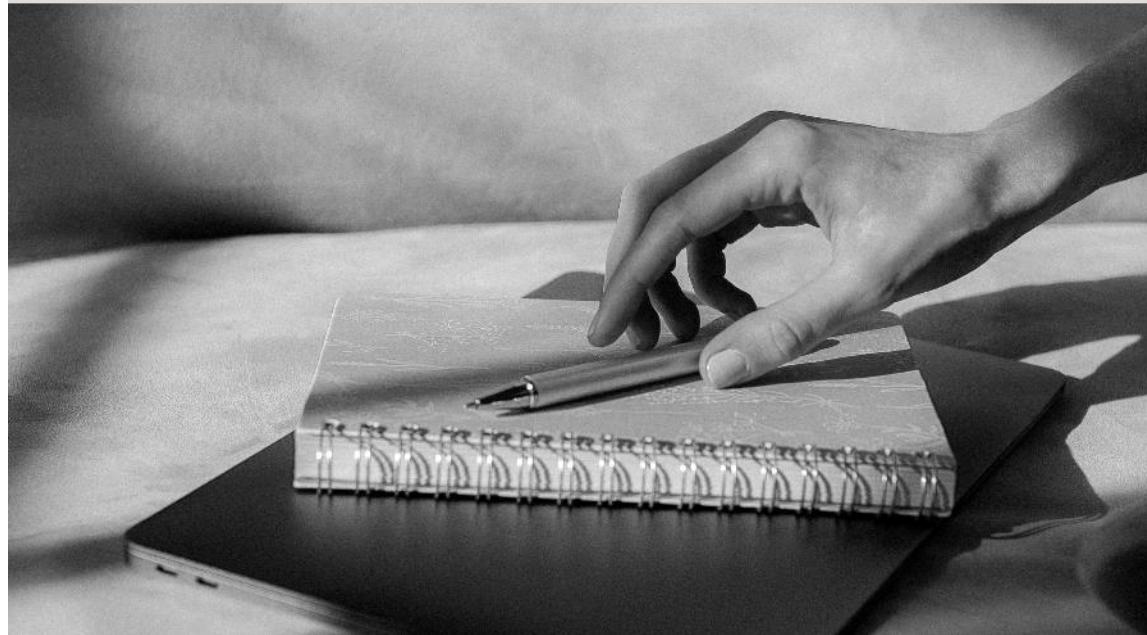
```
from A04 import Execute_Train

if __name__ == "__main__":
    data_dir = r"D:\01.project\EntryPrj\data\oraldrug\4.drug_Augmentation"

    # _____
    # 예제 : FasterRCNN - 모든 파라미터 명시
    # _____
    trans_type = ["default"] # , "A", "B"]
    for transform_type in trans_type:
        Execute_Train(
            model_type="yolov8",           # 모델 타입: "yolov8" 또는 "faster"
            data_dir=data_dir,           # 데이터 디렉토리 경로
            model_size="n",              # YOLOv8 모델 크기: "n", "s", "m", "l", "x"
            epochs=50,                   # 학습 에포크 수
            batch_size=16,               # 배치 크기
            lr=0.001,                    # 학습률 (YOLOv8 권장: 0.001)
            bBestLoad=True,              # Best 모델 로드 여부 (yolobest.pt)
            imgsz=640,                   # 이미지 크기
            patience=10,                 # Early stopping patience (에포크 수)
            train_ratio=0.8,             # 학습/검증 데이터 분할 비율
            num_workers=4,               # 데이터 로더 워커 수
            transform_type=transform_type # 데이터 증강 타입: "default", "A", "B"
        )
```

”

Faster R-CNN 구성



```
from A04 import Execute_Train

if __name__ == "__main__":
    data_dir = r"D:\01.project\EntryPrj\data\oraldrug\4.drug_Augmentation"

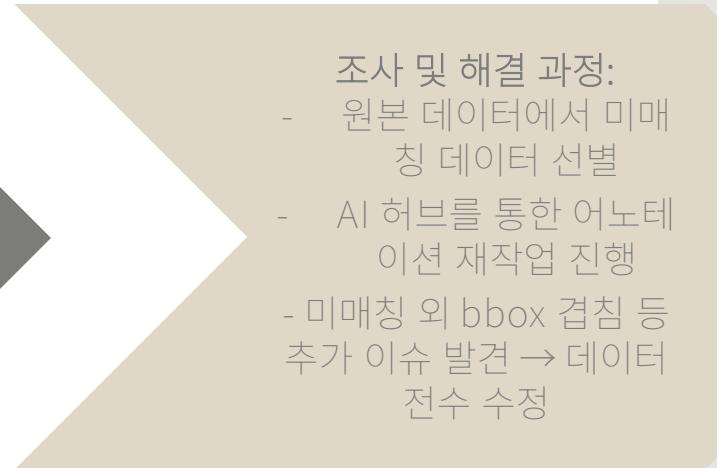
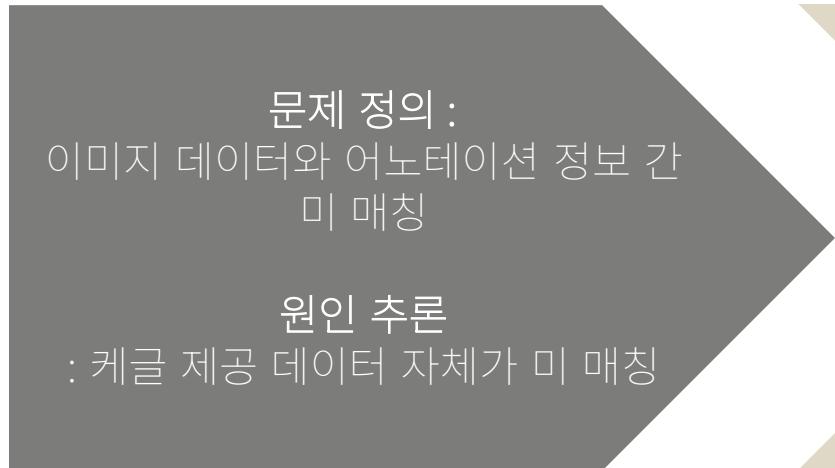
    #
    # 예제 : FasterRCNN - 모든 파라미터 명시
    #
    trans_type = ["default"] # , "A", "B"]
    for transform_type in trans_type:
        Execute_Train(
            model_type="faster",           # 모델 타입: "faster" 또는 "yolov8"
            data_dir=data_dir,             # 데이터 디렉토리 경로
            backbone="resnet50",           # FasterRCNN 백분: "resnet50" 또는 "mobilenet"
            epochs=50,                     # 학습 에포크 수
            batch_size=16,                 # 배치 크기
            lr=0.005,                      # 학습률 (FasterRCNN 권장: 0.005)
            bBestLoad=True,                # Best 모델 로드 여부 (fasterbest.pt)
            imgsz=640,                     # 이미지 크기
            patience=10,                   # Early stopping patience (에포크 수)
            gubun="partial",               # 최적화 방식: "freeze", "partial", "all"
            train_ratio=0.8,                # 학습/검증 데이터 분할 비율
            num_workers=4,                  # 데이터 로더 워커 수
            transform_type=transform_type   # 데이터 증강 타입: "default", "A", "B"
        )
```

Part 4 전체 일정표

| 12.01 | 12.02 | 12.03 | 12.04 | 12.05 | 12.08 | 12.09 | 12.10 | 12.11 | 12.12 |
|-------|-------|-------|-------|--|---|---|---|--|---|
| | | | | <ul style="list-style-type: none"> • 프로젝트 킥 오프 • 데이터 준비 • 환경구축 • 베이스라인 구축 <p>1주차</p> | <ul style="list-style-type: none"> • 이미지와 AI 허브 자료 비교 • 체크섬 유사도 매칭 <p>2주차</p> | <ul style="list-style-type: none"> • 이미지 + 어노테이션 전수 bbox 오버레이 확인 및 좌표 수정 | <ul style="list-style-type: none"> • 이미지 + 어노테이션 전수 bbox 오버레이 확인 및 좌표 수정 | <ul style="list-style-type: none"> • 이미지 구간별 Yolo 모델 학습 | <ul style="list-style-type: none"> • 케글 제출 • Score 0.982 점 달성 |

| 12.15 | 12.16 | 12.17 | 12.18 | 12.19 | 12.22 | 12.23 | 12.24 | 12.25 | 12.26 |
|--|--|---|---|--|---|--|-------|-------|-------|
| <ul style="list-style-type: none"> • RCNN 학습 및 확인 | <ul style="list-style-type: none"> • 스트림잇 서버 구축 • 프롬프트 엔지니어링 | <ul style="list-style-type: none"> • 스트림잇과 GPT 프롬프트 엔지니어링 • 자료 제작 | <ul style="list-style-type: none"> • 리포트 자체 pdf, eda 구현 확인 | <ul style="list-style-type: none"> • 코드 통합 테스트 • 리포 stream 으로 구현 확인 <p>3주차</p> | <ul style="list-style-type: none"> • 자료 정리 취합 <p>4주차</p> | <ul style="list-style-type: none"> • 발표 및 시뮬레이션 | | | |

Part 4 트러블 슈팅 (Issue & Solution)



Part 5 결과 시연 자료

Part 5 결과 시연 자료

Part 6 회고 및 향후 계획

감사합니다.

