

# 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. 트러블 슈팅 및 데이터 분석
5. eda 시연 & 모델 시연
6. 모델 분석 개선 전략



# 프로젝트 소개

1

## 프로젝트 배경

컴퓨터비전 학습과정에서 딥러닝 기반 이미지 분류 및 객체 인식 기술 습득

2

## 프로젝트 목표

미션에 부합하는 최적의 모델 선정 및 파이프라인 개발 경험 축적

3

## 프로젝트 후 기대 효과

실무 경험 축적 및 수업 이외 지식 습득, 분석 기술, 협업 기술 향상, 및 기타 소요 기술 습득

## 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

### 프롬프트 엔지니어링

- 약물 정보 설명
- 사용자 상황에 맞는 1차적 진단 가능

### Streamlit

- Python 기반 데이터 중심 웹 애플리케이션
- 실시간 이미지 업로드 및 실 검증

### 약물 탐지 기능

### GPT 프롬프트 엔지니어링

### Streamlit 웹 애플리 케이션

### 데이터 분석 (EDA)

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

### 데이터 분석 (EDA)

### 자동화 평가 시스템

### 자동화 평가 시스템

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

## Part 2

# 프로젝트 Workflow

### 데이터 셋 정의

D:\01.project\EntryPrj\data\ (로컬 & 구글 드라이브)  
├─ 1.drug\_Image\_annotation\_allOK/ (91% 데이터)  
│ └─ (이미지 vs annotation 매핑이 된)  
├─ 2.drug\_no\_image\_ok\_Anno/ (4% 데이터)  
│ └─ (어노테이션만 존재 + AI 허브 취합)  
├─ 3.drug\_ok\_Image\_no\_Anno/ (5% 데이터)  
│ └─ (이미지만 존재 + AI 허브 취합)  
└─ test\_images/  
 └─ 테스트 이미지 (Kaggle 제출용)

### 인프라 계층 (Colab 사용)

Pytorch  
2.9.0 + cu126

CUDA  
12.6+

RunTime  
T4 GPU

### Streamlit Web UI (streamlitService.py)

프로젝트 개요	모델 연구	데이터분 석	EDA 시연	모델 시연
------------	-------	-----------	--------	-------

### 평가 시스템 (run\_evaluation.py)

evaluate\_and\_create\_submission()  
- 모델 로드  
- Validation 평가  
- 테스트 데이터 예측  
- Submission CSV 생성  
- 결과 시각화

### 핵심 파이프라인(A04.py)

Execute\_Train() : 통합 학습 실행 함수  
- 데이터 전처리  
- 모델 학습 / 평가 통합 시스템  
- YOLOv8 & Faster R-CNN 구현

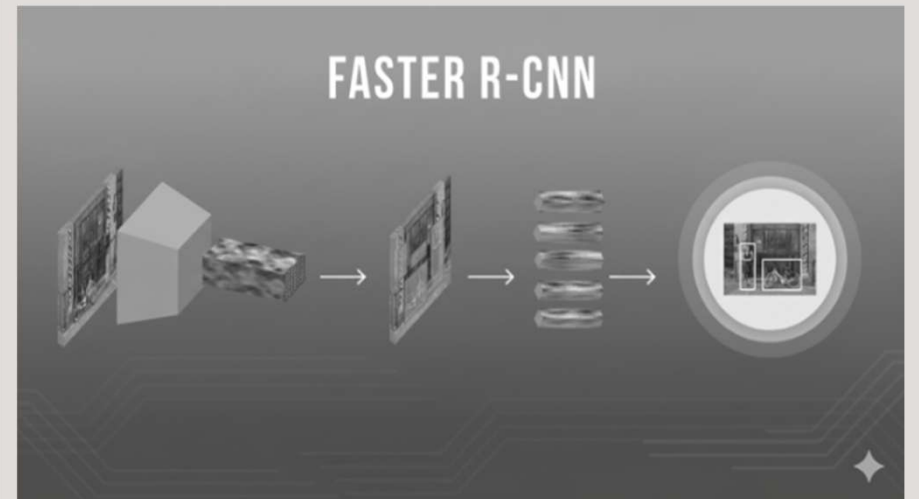
## Part 3 모델 연구

# ” 모델 선택 방향

### YOLOv8



### Faster R-CNN



## Part 3 모델 연구

# YOLOv8

```
from A04 import Execute_Train

if __name__ == "__main__":
    # data_dir = r"D:\01.project\EntryPrj\data\oraldrug\2.drug_no_image_ok_Anno"
    # data_dir = r"D:\01.project\EntryPrj\data\oraldrug\3.drug_ok_Image_no_Anno"

    data_dir = r"D:\01.project\EntryPrj\data\oraldrug\1.drug_Image_annotation_allOK"

    trans_type = ["default", "A", "B"]
    for transform_type in trans_type:
        # -----
        # 예제 : YOLOv8 Nano - 모든 파라미터 명시
        # -----
        Execute_Train(
            model_type="yolov8",          # 모델 타입: "yolov8"
            data_dir=data_dir,            # 데이터 디렉토리 경로
            model_size="n",               # YOLOv8 모델 크기: "n", "s", "m", "l", "x"
            epochs=1,                     # 학습 에포크 수
            batch_size=16,                # 배치 크기
            lr=0.001,                     # 학습률 (YOLOv8 권장: 0.001)
            bestload=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"
        )
```

## 장점

1. 추론 속도 우수 (Real-time 가능) : 단일 단계 구조로 지연 시간이 짧고 처리량이 높음.
2. 배포 및 운영 용이 : 모델 구조와 추론 파이프라인이 비교적 단순  $\Rightarrow$  서비스 적용(웹/모바일)에 유리함.
3. 튜닝 포인트가 명확하여 개선 사이클이 빠름 : 개선 방향을 실험적으로 빠르게 반복하기 용이함.

## 단점

1. 정밀한 박스 품질에서의 한계 : 전반적으로 빠르지만, 객체 경계를 타이트하게 맞춰야 하는 상황에는 2-Stage 계열보다 박스 정밀도가 불리할 수 있음.
2. 객체 밀집 장면에서의 성능 저하 기능
3. 촬영 환경 변화(조명/배경)에 민감  $\Rightarrow$  도메인 다양화와 증강 전략 필요.

## Part 3 모델 연구

# “Faster R-CNN

```
from A04 import Execute_Train

if __name__ == "__main__":
    # data_dir = r"D:\01.project\EntryPrj\data\oraldrug\2.drug_no_image_ok_Anno"
    # data_dir = r"D:\01.project\EntryPrj\data\oraldrug\3.drug_ok_image_no_Anno"

    data_dir = r"D:\01.project\EntryPrj\data\oraldrug\1.drug_image_annotation_allOK"

    # -----
    # 예제 : FasterRCNN - 모든 파라미터 명시
    # -----
    trans_type = ["default", "A", "B"]
    for transform_type in trans_type:
        Execute_Train(
            model_type="faster",          # 모델 타입: "faster"
            data_dir=data_dir,           # 데이터 디렉토리 경로
            backbone="resnet50",         # FasterRCNN 백본: "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"
        )
```

### 장점

1. 높은 검출 정확도 : 2-Stage 구조⇒ 전반적으로 안정적인 정확도 확보 가능.
2. 작은 물체 검출에 유리한 설계 가능, 복잡한 배경/조명 변화에서 상대적으로 견고.
3. 확장성 : 동일한 프레임워크에서 Mask R-CNN(세그멘테이션), Cascade(정밀도 강화) 등으로 확장 가능하며, 요구 성능에 맞춰 고도화하기 좋음.

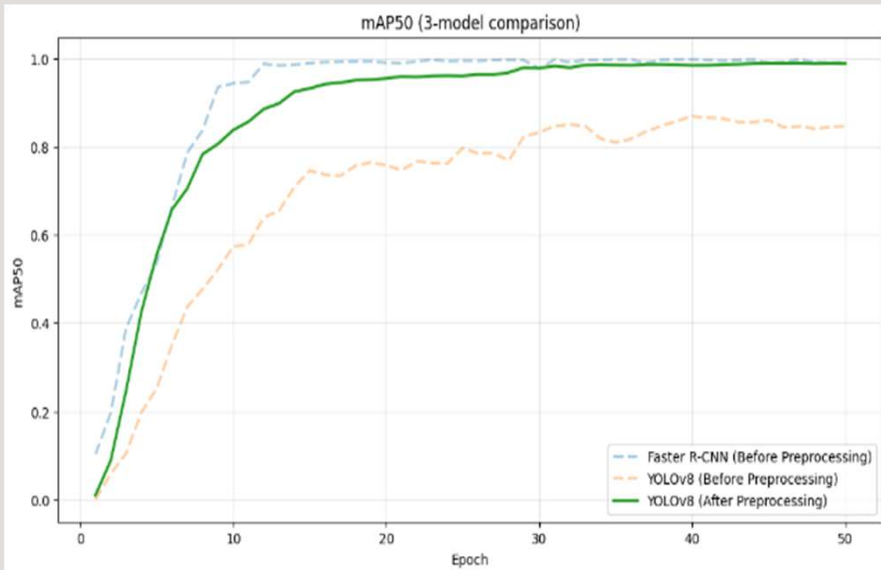
### 단점

1. 추론 속도 및 연산 비용 부담 : YOLO 대비 추론 지연과 운영 비용이 큼
2. 파이프라인 복잡도 및 튜닝 난이도
3. 실시간 서비스 제약 : 실시간 처리 요구가 높은 서비스(모바일/웹 실시간)에서는 단독 메인 검출기로 쓰기 어려움.



## Part 3 모델 연구

# ”모델 선택 전략



- YOLO는 추론 속도와 경량성이 뛰어나 실시간 서비스 및 대량 이미지 처리에서 1차 알약 검출기로 적합하다.
- 반면 **Faster R-CNN**은 **2-Stage** 구조로 후보 영역을 정밀하게 판별하여, 작은 알약이나 복잡한 배경에서 정확도 향상을 기대할 수 있다.
- 따라서 운영 관점에서는 YOLO를 기본 검출기로 사용하고, 신뢰도가 낮거나 겹침·난이도가 높은 사례에 한해 **Faster R-CNN**으로 2차 검증 하는 구성이 효율적이라고 판단된다. ⇒ 이 방식은 전체 처리 속도를 유지하면서도, 어려운 샘플에서의 오탐·미탐을 줄여 최종 성능을 끌어올릴 수 있다.
- 본 프로젝트에서는 제공된 데이터에 따라 속도 기반 YOLO와 정확도 보강용 **Faster R-CNN** 두가지 모델을 비교하여 최적의 결과를 도출할 수 있는 모델을 선택하였다.

Part 4

## 트러블 슈팅 (Issue & Solution)

문제 정의 :  
이미지 데이터와 어노테이션 정보 간  
미 매칭

원인 추론  
: 케글 제공 데이터 자체가 미 매칭

- 조사 및 해결 과정:
- 원본 데이터에서 미매칭 데이터 선별
  - AI 허브를 통한 어노테이션 재작업 진행
  - 미매칭 외 bbox 겹침 등 추가 이슈 발견 → 데이터 전수 수정

결과:  
데이터 품질 및 신뢰도 확보 완료



## Part 4 데이터 통계

▶ 전체 통계

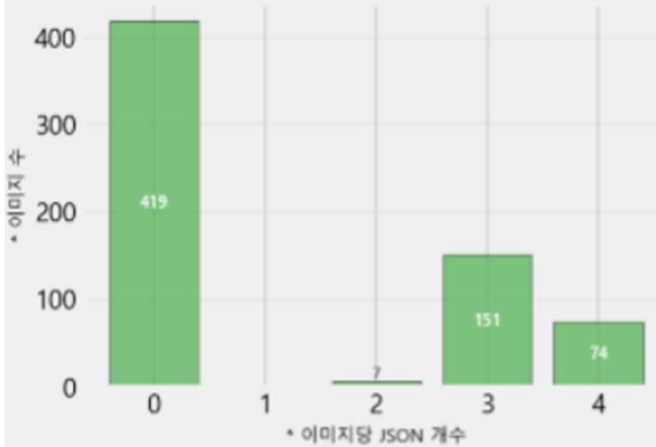
■ 전체 통계 요약

- \* 총 이미지: 651개
- \* JSON 있는 이미지: 232개
- \* JSON 없는 이미지: 419개

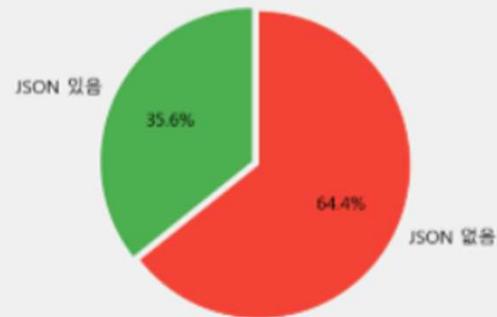
총 JSON: 1,001개  
이미지 있는 JSON: 763개  
이미지 없는 JSON: 238개

총 매핑 수: 1,182개

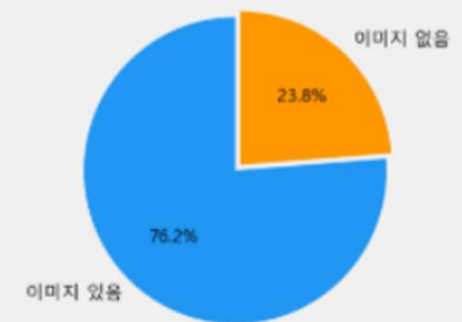
▶ 이미지당 JSON 개수 분포



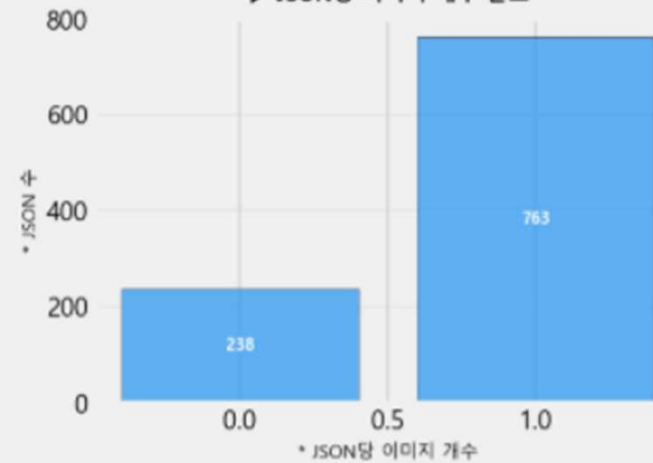
▶ 이미지별 JSON 매핑 비율



▶ JSON별 이미지 매핑 비율



▶ JSON당 이미지 개수 분포



## Part 4 Image-Annotation 매핑 분석

### ▣ 이미지당 JSON 개수 통계

- 최소: 0개
- 최대: 4개
- 평균: 1.17개

### ▣ 분포

- 0개: 419개 (64.4%)
- 2개: 7개 (1.1%)
- 3개: 151개 (23.2%)
- 4개: 74개 (11.4%)

### ▣ JSON당 이미지 개수 통계

- 최소: 0개
- 최대: 1개
- 평균: 0.76개

### ▣ 분포

- 0개: 238개 (23.8%)
- 1개: 763개 (76.2%)

### ▣ 각 상태별 이미지/어노테이션 숫자

#### ▪ 이미지(O)/어노테이션(O)

image : 232  
anno : 763

#### ▪ 이미지(X)/어노테이션(O)

image : 137  
anno : 365

→ AI HUB 데이터로 부터 같은 이름의 이미지를 복사 후 매칭.  
→ 또한 이 이미지에 해당하는 추가 **annotation**을 복사.

#### ▪ 이미지(O)/어노테이션(X)

image : 419  
anno : 1676

→ MD5 checksum으로 이미지를 확인 후 이에 해당하는 **annotation** 복사 후 매칭

### ▣ 예외 분석

#### ▪ category\_id가 1인 (dl\_idx 와 category\_id가 다른 경우)

1950 개 → 모두 category\_id 를 dl\_idx 로 수정.

#### ▪ 같은 image에서 bbox의 좌표가 같은 경우.

137개. 이미지(X)/어노테이션(O) 에서 추가 annotation 복사 중 생긴 것으로, category\_id와 bbox가 동일한 경우 하나는 삭제.

#### ▪ box 좌표가 잘못된 경우

6 개

K-003351-003832-020238\_0\_2\_0\_2\_70\_000\_200.json, BBox: []

K-003351-013900-020238\_0\_2\_0\_2\_70\_000\_200.json, BBox: []

K-003351-016688-018357\_0\_2\_0\_2\_90\_000\_200.json, BBox: [70, 438, 584]

K-003351-031863-036637\_0\_2\_0\_2\_90\_000\_200.json, BBox: []

K-003351-031863-038162\_0\_2\_0\_2\_90\_000\_200.json, BBox: [428, 199, 295]

K-003544-006563-016548-033878\_0\_2\_0\_2\_75\_000\_200.json, BBox: [133.183, 202, 200]

## Part 5 결과 시연 자료

EDA tool 시연 탭

Annotation 파일 선택  
K-003351-013900-016262\_0\_2\_0\_2\_75\_000\_200.json

dl\_name: 알약하이트린정 2mg, dl\_idx: 3350, category\_id: 3350

left: 354, top: 182, width: 186, height: 189

전체 그리기, 선택 그리기, 저장, 추가, 삭제

학습 이미지 샘플02

이전, 다음

10 sample images with IDs:

- K-003351-003832-016232\_0\_2\_0\_2\_70\_000\_200.png
- K-003351-003832-016688\_0\_2\_0\_2\_75\_000\_200.png
- K-003351-003832-016688\_0\_2\_0\_2\_90\_000\_200.png
- K-003351-003832-020238\_0\_2\_0\_2\_70\_000\_200.png
- K-003351-003832-020238\_0\_2\_0\_2\_75\_000\_200.png
- K-003351-003832-029667\_0\_2\_0\_2\_70\_000\_200.png
- K-003351-003832-029667\_0\_2\_0\_2\_90\_000\_200.png
- K-003351-003832-036637\_0\_2\_0\_2\_70\_000\_200.png
- K-003351-003832-036637\_0\_2\_0\_2\_75\_000\_200.png
- K-003351-003832-038162\_0\_2\_0\_2\_90\_000\_200.png

## Part 5 결과 시연 자료

The screenshot shows a Streamlit web application interface for drug detection and interaction analysis. The browser address bar shows the URL: `entryprj-rbrmekfjfm5ldqvcqofgc.streamlit.app`.

**Left Sidebar:**

- Files: `./files/yolo_class_mapping.json` 에서 74개 클래스 매핑 로드 완료!
- 식품의약품안전처 의약품개요정보(e약은요) 데이터 로드 완료
- 10개 약물 상세 정보 (`./files/drug_API_info.json`)
- 설정**
  - 검출 신뢰도 임계값: 0.65
  - 모델 선택 방식
    - ☒ 기본 모델 사용
    - ☐ 커스텀 모델 업로드
  - 모델 선택: `yolobest.pt`
  - 모델 로드 완료: `yolobest.pt`

**Main Content Area:**

- Navigation tabs: 프로젝트개요, 모델 연구, 데이터 분석, EDA tool 시연, **모델 시연**
- 모델 시연 탭**
- 약 검출 및 상호작용 분석 시스템**
- YOLO 모델로 약을 검출하고, GPT를 통해 약물 상호작용을 분석합니다.
- 이미지 업로드**
  - 이미지를 선택하세요
  - Drag and drop file here (Limit 200MB per file • JPG, JPEG, PNG)
  - Browse files
  - 100.png 1.6MB
- 검출 결과**
  - Two images showing drug detection results with bounding boxes and confidence scores:
    - Left image: A white pill labeled "BSP" with a blue bounding box and confidence score `1899 0.99`.
    - Right image: A yellow pill labeled "GC" with a green bounding box and confidence score `16547 0.99`.

## Part 5

# 결과 시연 자료

Streamlit

entryprj-rbrmekfjfm5ldqvcqofgc.streamlit.app

Fork

./files/yolo\_class\_mapping.json' 에서 74개 클래스 매핑 로드 완료!

식품의약품안전처 의약품개요정보(e약은요) 데이터 로드 완료

10개 약물 상세 정보 (./files/drug\_API\_info.json)

설정

검출 신뢰도 임계값  
0.25

모델 선택 방식  
☒ 기본 모델 사용  
☐ 커스텀 모델 업로드

모델 선택  
yolobest.pt

모델 로드 완료:  
yolobest.pt

검출 상세 정보

총 3개의 약이 검출되었습니다.

	약 인덱스	약물명	신뢰도
0	29666	리바로정 4mg	95.16%
1	3350	일양하이트린정 2mg	92.02%
2	18356	중근당글리아티린연질캡슐(콜린알포세레이트)	73.34%

GPT API 키를 입력하세요

1단계: 검출된 약물 분석 시작

GPT 1단계 분석 결과

약물명	주요 성분	핵심 효능/효과	대표적인 부작용
리바로정 4mg	리바스타틴	콜레스테롤 수치 감소	근육 통증, 간효소 증가
일양하이트린정 2mg	로란타딘	알레르기 증상 완화	졸림, 구강 건조
중근당글리아티린연질캡슐	콜린알포세레이트	인지기능 개선, 신경 보호	소화불량, 두통

## Part 5 결과 시연 자료

Streamlit

entryprj-rbrmekfjfm5ldqvcqofgc.streamlit.app

Fork

['/files/yolo\_class\_mapping.json' 에서 74개 클래스 매핑 로드 완료!

식품의약품안전처 의약품개요정보(e약은요) 데이터 로드 완료

10개 약물 상세 정보 (/files/drug\_API\_info.json)

설정

검출 신뢰도 임계값  
0.25

모델 선택 방식  
☒ 기본 모델 사용  
☐ 커스텀 모델 업로드

모델 선택  
yolobest.pt

모델 로드 완료:  
yolobest.pt

사용자 맞춤 분석

개인 정보 입력

나이  
35

성별  
여성

알고 있는 기저질환 (예: 고혈압, 당뇨)  
혈압약

약물 / 음식 알러지 (예: 아스피린)  
사과 알러지

증상을 입력하세요 (선택사항)  
복통

2단계: 맞춤형 복약 분석 시작

GPT 2단계 분석 결과

1. 종합 평가

- 안전 평가:** 현재 복통 증상을 알고 있는 환자에게 제공된 약물 (리바로정 4mg, 일양하이트린정 2mg, 중근당글리타티린연질캡슐) 의 자세한 정보가 없어, 이 약물들이 복통에 미치는 영향을 명확히 평가하기 어렵습니다. 그러나 일반적으로 복통이 있는 경우, 소화기 자극을 줄 수 있는 약물은 주의해야 하므로 복용에 신중을 기해야 합니다.
- 약물 상호작용:** 제공된 약물들에 대한 상호작용 정보가 없습니다. 따라서 특정 약물 간의 상호작용을 확인할 수 없으므로, 복용 시 주의가 필요합니다.



## Part 6

# Submission 데이터 검증 : 한 이미지에 카타로그 2 (2개)



카테고리	알약 2개 판정
홍색 장방형 코팅정	1
흰색 원형 정제	1



**신바로정 식별표시**

S300					
장축(mm)	14.5	단축(mm)	8	두께(mm)	5.3
성상	갈색의 장방형 필름코팅정				

**넥시움정40mg 식별표시**

A EI 40mg					
장축(mm)	16	단축(mm)	8	두께(mm)	5.5
성상	분홍색 장방형 필름코팅정				

**무코스타정 식별표시**

OG 33					
장축(mm)	8.1	단축(mm)	8.1	두께(mm)	3.4
성상	백색의 원형 필름코팅 정제				

**에어탈정 식별표시**

Airtal					
장축(mm)	8.0	단축(mm)	8.0	두께(mm)	3.8
성상	코팅된 흰색의 정제				

## Part 6

# Submission 데이터 검증 : score 0.75 미만 데이터 (77개)



카테고리	알약 1개 판정	알약 2개 판정
흰색 원형 코팅정	4	5
흰색 장방형 코팅제	9	6
홍색 장방형 필름제	7	3

자이프랙사정 2.5mg 식별표시

63.1%

식별정보	제품정보	제품설명서
LILLY 4112	장축(mm) 7.2 단축(mm) 7.2 두께(mm) 3.3	성상 필름코팅된 박색의 원형정제

에어탈정 식별표시

37.62%

식별정보	제품정보	제품설명서
Airtal	장축(mm) 8.0 단축(mm) 8.0 두께(mm) 3.8	성상 코팅된 흰색의 정제

타이레놀 8시간이알서방정 식별표시

89.42%

식별정보	제품정보
TYLENOL ER	장축(mm) 19.2 단축(mm) 7.8 두께(mm) 6.2

써스펜 8시간이알서방정 650mg 식별표시

39.46%

식별정보	제품정보	제품설명서
SUSPEN ER	장축(mm) 19.23 단축(mm) 7.70 두께(mm) 6.05	성상 이 약은 박색의 장방형 필름코팅정제이다.

베콤씨에프정 식별표시

96.72%

식별정보	제품정보	제품설명서
YH B-F	장축(mm) 20.1 단축(mm) 8.5 두께(mm) 7.1	성상 적갈색의 타원형 제과정

비타비백정 식별표시

37.44%

식별정보	제품정보	제품설명서
YH V100	장축(mm) 19.38 단축(mm) 9.31 두께(mm) 6.60	성상 적색의 타원형 제과정

## Part 6

# Submission 데이터 검증 : score 0.75 미만 데이터 (77개)



카테고리	알약 1개 판정
미항색 투명 연질 캡슐	8
담황색 연질 캡슐	5
홍색 원형 정제	6



## Part 6

# Submission 데이터 검증 : score 0.75 미만 데이터 (77개)



카테고리	알약 1개 판정
담황색 코팅 정제	3
흰색 경질 캡슐	2



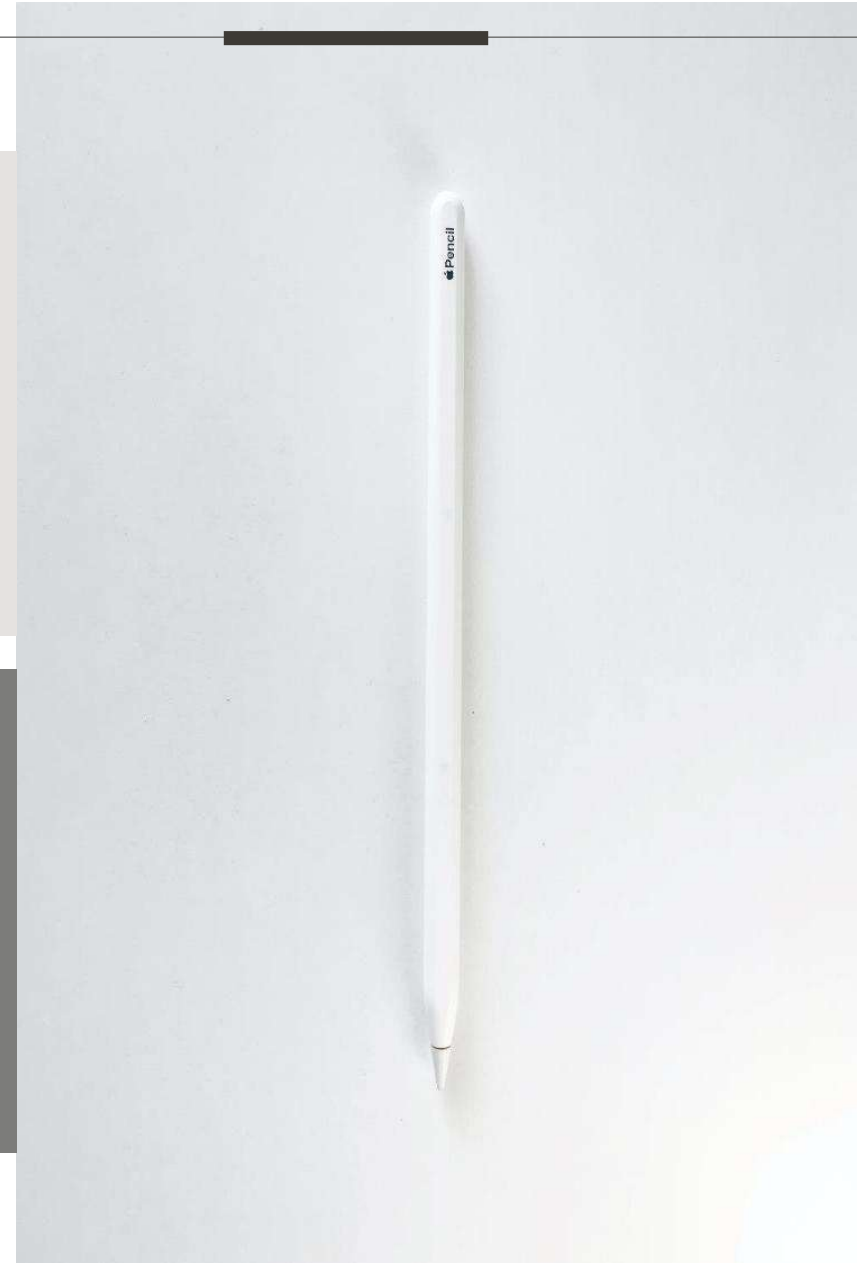


## Part 7

# 성능 개선 전략

모델 성능을 개선하기 위해서는 증강 기법을 색상마다 다르게 시도

- 흰색 알약: 대비 / 샤프닝 각인 강조
- 황색 알약: HSV 색 공간 변환, 채도 강화
- 홍색 알약: RED 채널 강화, 십자 각인 패턴



[

감사합니다.

]

