

# 醫療影像診斷系統說明書

日期: 2026-02-10 | 版本: 1.1.0

## 1. 系統概述

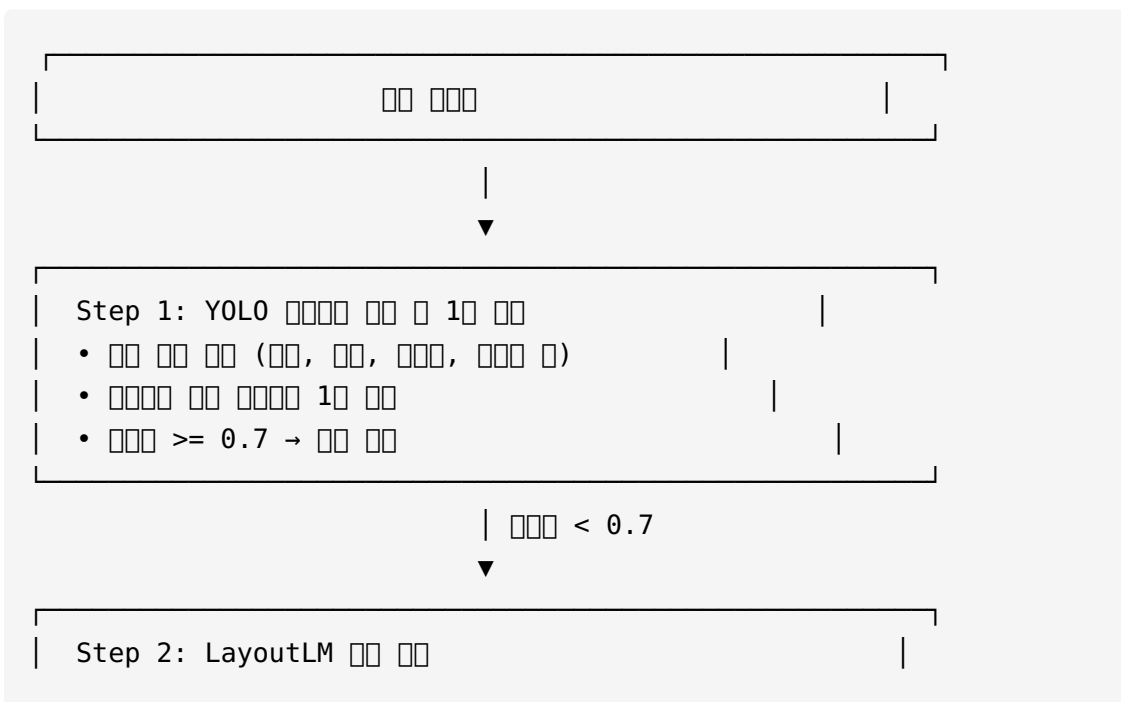
### 1.1 目的

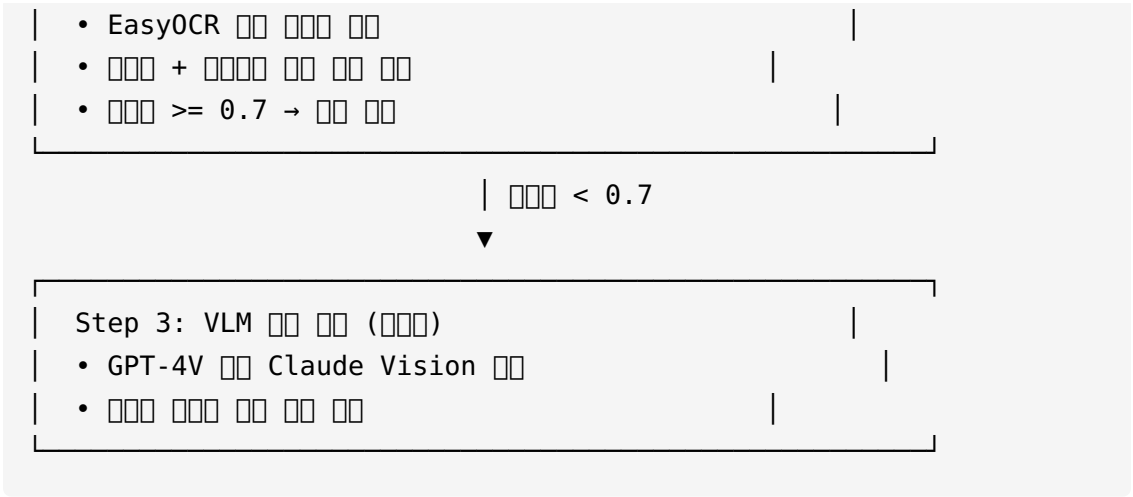
本系統旨在協助醫師進行影像診斷，提高診斷效率和準確性。

### 1.2 系統功能 (6項)

序號	功能名稱	描述	序號	功能名稱	描述
1	影像上傳	Diagnosis	4	影像分析	Admission/Discharge
2	影像查看	Medical Opinion	5	影像分析	Medical Receipt
3	影像下載	Insurance Claim	6	影像分析	Prescription

## 2. 系統架構





類別	物件	物件	物件
Header	Header, Title, Subtitle	-	Header, QR
Body	Text, Table	Text, Table	QR
Footer	Text, QR	Text, Table, Text	Text

2.3 YOLO 物件偵測

- 類別: YOLO 物件 (Header + Body Table)
- 物件 ID: 0: stamp (Header) 1: signature (Header) 2: table (Body) 3: barcode (Body) 4: qrcode (QR) 5: hospital\_logo (Header)

2.4 LayoutLM 物件偵測

- 類別: Header + OCR Table + Body Table
- OCR 類別: 物件 ID Mock OCR 類別
- 類別: "Header", "Table", "Text", "Table", "Text" 物件
- 類別: "Table", "Table", "Table", "Table" 物件
- 類別: "Table", "Table", "Table", "Table" 物件
- 類別: "Table", "Table", "Table", "Table" 物件
- 類別: "Table", "Table", "Table", "Table" 物件
- 類別: "Table", "Table", "Table", "Table" 物件

4. Step 1: YOLO 物件偵測

3.1 物件偵測

- 物件 ID: YOLOv8n (ultralytics)
- 物件 ID: 50
- 物件 ID: 640×640
- 物件 ID: 16

3.2 物件偵測

Epoch 50/50 物件

- Box Loss: 0.5765

- Class Loss: 0.3342

- DFL Loss: 0.8823

3.3 验证 (Validation)

验证结果

验证项目	验证结果
mAP50	99.5%
mAP50-95	96.0%
Precision	98.7%
Recall	98.8%

验证 AP50

验证项目	AP50
stamp	99.5%
signature	99.5%
table	99.5%
barcode	99.5%
qrcode	99.5%
hospital_logo	99.5%

3.4 1. 验证结果

验证结果

验证项目	验证结果	验证结果
验证项目	10/10 (100%)	-
验证项目	0/10 (0%)	验证项目
验证项目	10/10 (100%)	-
验证项目	10/10 (100%)	-
验证项目	10/10 (100%)	-
验证项目	10/10 (100%)	-
验证项目	50/60 (83.3%)	-

📄

- 📄/📄 📄: 📄 📄 📄 📄 📄 📄 (📄+📄, 📄 📄)📄 📄 YOLO 📄 📄 📄 📄
- 📄 📄: Step 2 (LayoutLM)📄 📄 📄 📄

## 5. Step 2: LayoutLM 📄

### 4.1 📄 📄

- 📄 📄: microsoft/layoutlmv3-base
- 📄 📄: 5
- 📄 📄: 2
- 📄 📄 📄: 512
- 📄: 5e-5 (warmup steps: 30)

### 4.2 📄 📄

📄 📄

Epoch	Train Loss	Eval Loss	Eval Accuracy
1	0.0030	0.0011	100%
2	0.0020	0.0011	100%
3	0.0014	0.0007	100%
4	0.0012	0.0006	100%
5	0.0010	0.0006	100%

### 4.3 📄 📄

📄 📄 📄

📄 📄	📄
📄	10/10 (100%)
📄	10/10 (100%)
📄📄📄	10/10 (100%)
📄📄📄	10/10 (100%)

Item	Count
Header	10/10 (100%)
Text	10/10 (100%)
Total	60/60 (100%)

#### 4.4 Mock OCR Results

v1.1 Mock OCR Results EasyOCR Results:

Item	Mock OCR	Real OCR (EasyOCR)
Header	10/10 (100%)	10/10 (100%)
Text	10/10 (100%)	10/10 (100%)
Table	10/10 (100%)	10/10 (100%)
Form	10/10 (100%)	10/10 (100%)
Image	10/10 (100%)	10/10 (100%)
Text	10/10 (100%)	10/10 (100%)
Total	60/60 (100%)	60/60 (100%)

- Mock OCR Results EasyOCR Results 100% Match
- EasyOCR Results Match Mock OCR Results

#### 4.5 Summary

- Header/Text Match: Step 1 Mock OCR Results EasyOCR Results 100% Match
- Table Match: Mock OCR Results EasyOCR Results 0.99 Match
- Form OCR Results: Mock OCR Results OCR Results Match

### 6. Step 1 vs Step 2

Item	Step 1 (YOLO)	Step 2 (LayoutLM)
Header	Header Match	EasyOCR Header + Table Match
Table	83.3%	100%
Header/Text Match	Match	Match

단계	Step 1 (YOLO)	Step 2 (LayoutLM)
추론 속도 (FPS)	~0.1	~2
모델 크기	~100MB (CPU)	~1GB (GPU)

## 7. API 배포

### 6.1 배포 방법

- 프레임워크: FastAPI
- 서버: Uvicorn
- UI: Swagger UI (자동 생성)

### 6.2 엔드포인트

엔드포인트	HTTP 메소드	설명
/	GET	API 정보
/health	GET	서비스 상태 확인
/classes	GET	인식 가능한 클래스 목록
/classify	POST	단일 이미지 분류
/classify/batch	POST	배치 이미지 분류
/docs	GET	Swagger UI

### 6.3 배포 설정 (v1.1)

- OCR 모드 설정: `use_mock_ocr` 파라미터로 `False`로 설정하여 EasyOCR을 사용
- 모델 로드: YOLO, LayoutLM, EasyOCR 3개 모델을 동시에 로드
- OCR 엔진: EasyOCR을 사용하여 텍스트 인식
- OCR 해상도: OCR을 위한 이미지를 422 픽셀로 조정

### 6.4 API 응답 예시

```
{
  "predicted_class": "문서",
  "confidence": 0.9972,
  "final_step": 2,
```

```

"processing_time": 4.11,
"step1_result": {
  "predicted_class": "□□□",
  "confidence": 1.0,
  "requires_step2": false
},
"step2_result": {
  "predicted_class": "□□□",
  "confidence": 0.9972,
  "all_probabilities": {
    "□□□": 0.9972,
    "□□□": 0.0008,
    "□□□□□": 0.0005,
    "□□□□□": 0.0006,
    "□□□□□": 0.0005,
    "□□□": 0.0004
  },
  "ocr_words_count": 21,
  "mock_ocr_used": false
}
}

```

## 8. □□□□ □□

```

□□□□□/
├─ api/
│   ├── __init__.py
│   └─ app.py                # FastAPI □
├─ config/
│   └─ config.yaml          # □□ □□
├─ src/
│   ├── pipeline.py          # □□ □□□□□
│   ├── preprocessor/         # □□□ (□□□□ □□ □)
│   ├── step1_yolo/           # YOLO □□ □ □□
│   ├── step2_layoutlm/       # LayoutLM □□ □ OCR
│   ├── step3_vlm/            # VLM □□ □□ (placeholder)
│   └─ utils/                 # □□□□
├─ scripts/
│   ├── train_yolo.py          # YOLO □□
│   ├── train_layoutlm_simple.py # LayoutLM □□
│   ├── test_step1_classifier.py # Step 1 □□□
│   └─ test_step2_layoutlm.py  # Step 2 □□□

```



```
|   |─ test_real_ocr.py      # OCR vs Mock OCR 테스트
|   └─ test_api.py         # API 테스트
└─ data/
   |─ models/              # 모델
   |─ yolo_dataset/        # YOLO 데이터셋
   └─ sample/              # 샘플
└─ run_api.py              # API 실행
└─ requirements.txt        # requirements
```

---

## 9. 실행

### 8.1 설치

```
pip install -r requirements.txt
```

### 8.2 API 테스트

```
python run_api.py
```

### 8.3 테스트

```
# Step 1 테스트
python scripts/test_step1_classifier.py

# Step 2 테스트
python scripts/test_step2_layoutlm.py

# OCR vs Mock OCR 테스트
python scripts/test_real_ocr.py

# API 테스트
python scripts/test_api.py
```

---

## 10. **OCR 처리**

---

### 9.1 **OCR**

- [ ] Step 3 (VLM) **모델**: GPT-4V **또는** Claude Vision **사용**
- [x] ~~**OCR** **도구**~~: EasyOCR **사용** (v1.1), **정확도** 100% **확인**
- [ ] **문서** **정제** **도구**: **PDF** **도구**, **HTML** **도구**

### 9.2 **API**

- [ ] **API** **호출** **도구** **사용** **도구** **사용**
- [ ] **API** **호출** **도구** **사용** **도구** **사용**
- [ ] Docker **컨테이너**

### 9.3 **배포**

- [ ] **API** **호출** **도구** **사용**
- [ ] **API** **호출** **도구**
- [ ] **클라우드** **서비스** (AWS/GCP)

## 11. **결과**

---

3개 **모델** **평가** **결과** **표** **결과** **표** **결과** **표**.

**평가** **결과**: 1. **YOLO** **모델** **정확도** **표**: mAP50 99.5% **표** 2. **LayoutLM** **모델** **정확도** **표**: 100% **표** **표** 3. **문서**/**문서** **정제** **도구** **정확도** **표**: Step 2 **정제** **도구** **정확도** **표** **표** 4. **OCR** **도구**: EasyOCR **정확도** **표** **표** 6 **정제** **도구** 100% **정제** **도구** 5. **REST API** **도구**: FastAPI **정제** **도구** API **정제** (**정제** OCR **정제** **표**)

**평가** **결과**: | **정제** | **정제** | |-----|-----| | Step 1 **정제** **정제** | ~0.1**정제**/**정제** | | Step 2 **정제** **정제** | ~2**정제**/**정제** **정제** | | Step 1 **정제** | 83.3% (**정제**/**정제** **정제**) | | Step 2 **정제** (Mock OCR) | 100% | | Step 2 **정제** (Real OCR) | 100% |

**정제** **정제** **정제** **정제**, **정제** **정제** **정제** **정제** **정제** **정제** **정제**.

---

**정제**: Claude Opus 4.6 (Co-authored with BongwooChoi)