

# AI 화재감지 카메라 제작 매뉴얼

## 1) 개요

본 시스템은 카메라 기반 AI로 **화재(fire)**를 인식하여 **MQTT 브로커**로 이벤트를 전송하고, 수신 측(무드등·진동 베개)에서 즉시 경고를 발생시키도록 설계되었다. 운영 모드는 두 가지다.

- **.pt 모드:** Ultralytics YOLO(.pt) + PyTorch
- **Hailo 모드:** Hailo-8(M.2) + HEF + 자체 후처리(DFL + NMS)

두 모드 모두 CPU 온도 MQTT 송신과 감지 이벤트 MQTT 송신을 지원한다.

## 2) 시스템 구성

### 하드웨어

- Raspberry Pi 5 + USB 카메라(또는 CSI 카메라)
- Hailo-8 (Hailo 모드일 때)

### OS/런타임

- Ubuntu 24.04 (라즈베리파이)
- Python 3.8+

### 필수 패키지

- 공통: opencv-python, paho-mqtt, flask, numpy
- .pt 모드: ultralytics, torch
- Hailo 모드: Hailo SDK, Hailo 드라이버 `#!/AuditoryAssist/AI/Hailo8/whl`

### 3) 데이터셋 & 학습

#### 3.1 데이터 구성 가이드

- **양성:** 실내 불꽃 이미지 + 다양한 조명/원근/배경
- **음성:** 화재 유사 패턴
- **해상도:** 640x640 권장(학습/추론 imgsz 일관)

#### 3.2 Ultralytics 학습 명령 예시

```
yolo detect train \  
  model=yolov8n.pt \  
  data=fire.yaml \   # names: ["fire"]  
  imgsz=640 \  
  epochs=100 \  
  batch=16
```

(.pt 모드는 여기서 학습된 .pt 파일 사용)

### 4) YOLOv8(.pt) → Hailo(.hef) 변환 절차

#### 4.1 .pt → ONNX 내보내기 (NMS 제외)

```
yolo export \  
  model=fire.pt \  
  format=onnx \  
  opset=12 \  
  imgsz=640 \  
  simplify=True \  
  dynamic=False
```

#### 4.2 ONNX → har → quantized\_model.har → hef

- ONNX → .har # ~/AI/Hailo8/step/[parse.py](#)
- .har → quantized\_model.har # ~/AI/Hailo8/step/[optimize.py](#)
- quantized\_model.har → .hef # ~/AI/Hailo8/step/[compile.py](#)

## 5) 추론 런타임 세팅

### 5.1 라벨 & 런타임 설정 파일

- 라벨 매핑(labels.json)
  - YOLOv8n(.pt) 런타임:  
/AuditoryAssist/AI/YOLOv8n/model/labels.json
  - Hailo8(HEF) 런타임:  
/AuditoryAssist/AI/Hailo8/model/labels.json
  - 예시: {"labels": ["fire"]} 또는 {"labels": ["fire", "smoke"]}

```
1  {
2      "detection_threshold": 0.5,
3      "max_boxes": 200,
4      "labels": [
5          "fire"
6      ]
7  }
```

- Hailo 전용 후처리 설정(yolov8n\_nms\_config.json)

```
1  {
2      "nms_scores_th": 0.5,
3      "nms_iou_th": 0.5,
4      "image_dims": [
5          640,
6          640
7      ],
8      "max_proposals_per_class": 100,
9      "classes": 1,
10     "regression_length": 16,
11     "background_removal": false,
12     "bbox_decoders": [
13         {
14             "name": "fire_only2/bbox_decoder41",
15             "stride": 8,
16             "reg_layer": "fire_only2/conv41",
17             "cls_layer": "fire_only2/conv42"
18         },
19         {
20             "name": "fire_only2/bbox_decoder52",
21             "stride": 16,
22             "reg_layer": "fire_only2/conv52",
23             "cls_layer": "fire_only2/conv53"
24         },
25         {
26             "name": "fire_only2/bbox_decoder62",
27             "stride": 32,
28             "reg_layer": "fire_only2/conv62",
29             "cls_layer": "fire_only2/conv63"
30         }
31     ]
32 }
```

## 5.2 MQTT 사양

### 공통 사양

- 브로커: 192.168.0.24:1883
- 감지 이벤트 토픽/페이로드
  - 화재: AI\_fire\_alert → {"sensor\_id": "AI\_D\_fire", "event": "fire\_detected"}
  - 연기: AI\_smoke\_alert → {"sensor\_id": "AI\_D\_smoke", "event": "smoke\_detected"} (추가 예정)
- 온도 토픽/페이로드
  - system/temperature/pi5 → {"sensor\_id": "raspi\_temp\_pi5", "value": "<degC>", "timestamp": "YYYY-MM-DD HH:MM:SS"}

## 6) 추론

### 6.1 .pt 모드 동작

```
python /AuditoryAssist/AI/YOLOv8n/model/pt_video.py
```

### 6.2 Hailo 모드 동작

```
python /AuditoryAssist/AI/Hailo8/model/hailo_video.py
```