

# 軟體需求規格書 (SRS)

---

## AI Hub - 智慧視覺串流平台

---

文件版本: v1.0

建立日期: 2025-12-29

狀態: 初版

---

### 目錄

- 簡介
  - 系統概述
  - 使用案例
  - 功能性需求
  - 非功能性需求
  - 介面需求
  - 資料需求
  - 測試案例
  - 附錄
-

# 1. 簡介

## 1.1 目的

本文件為 **AI Hub - 智慧視覺串流平台** 的軟體需求規格書 (Software Requirements Specification, SRS) ，旨在詳細描述系統的功能性與非功能性需求，作為系統設計、開發與測試的依據。

**預期讀者：**軟體開發團隊、品質保證團隊、專案經理、系統架構師、客戶技術代表

## 1.2 範圍

AI Hub 是一套以影像串流處理為核心的智慧視覺平台，提供：

- 多路視訊串流管理、轉碼與分發
- AI 物件偵測與行為辨識
- 虛擬圍欄告警與事件管理
- 事件錄影與影片裁剪
- Web 管理介面

## 1.3 術語定義

術語	定義
RTSP	Real Time Streaming Protocol，即時串流協議
HLS	HTTP Live Streaming，HTTP 直播串流
MQTT	Message Queuing Telemetry Transport，訊息佇列協議
YOLO	You Only Look Once，即時物件偵測演算法
VAAPI	Video Acceleration API，Intel 視訊加速介面
虛擬圍欄	畫面上定義的多邊形區域，用於偵測入侵
事件片段	事件觸發前後自動裁剪的影片
冷卻期	同一告警重複觸發的最小間隔

## 1.4 參考文件

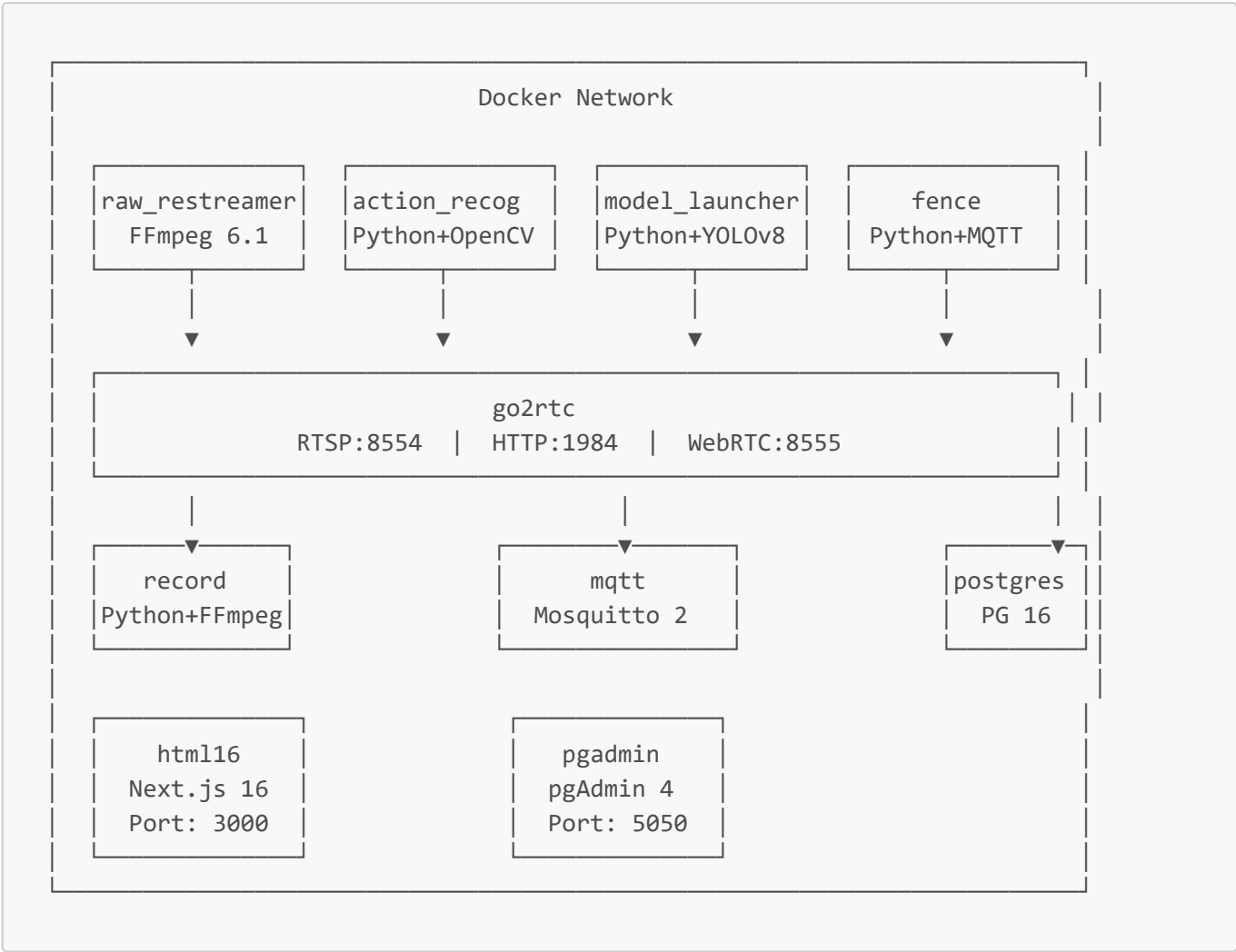
文件	版本	說明
prd_customer.md	v1.0	產品需求文件 - 客戶版
prd_internal.md	v1.0	產品需求文件 - 內部開發版

## 2. 系統概述

### 2.1 產品願景

提供企業級智慧視覺監控解決方案，結合高效能串流處理與 AI 辨識能力，實現即時監控、智慧告警與事件管理。

### 2.2 系統架構

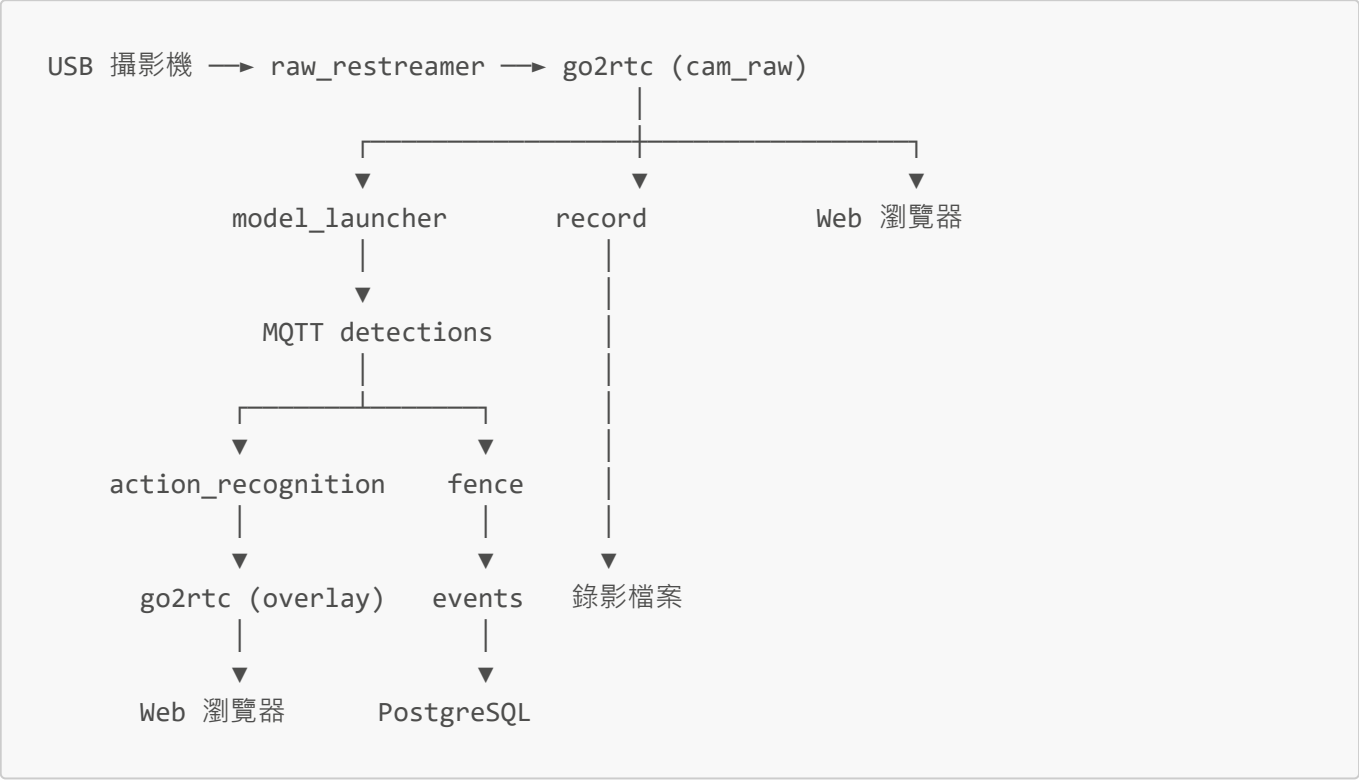


### 2.3 服務組成

服務	技術棧	職責
raw_restreamer	FFmpeg 6.1	USB 攝影機擷取轉推
action_recognition_server	Python + OpenCV	AI 結果疊加串流
model_launcher	Python + YOLOv8	動態模型推論
fence	Python + MQTT	虛擬圍欄入侵偵測
record	Python + FFmpeg	持續錄影與事件裁剪
go2rtc	Go	多協議串流伺服器
mqtt	Mosquitto 2	MQTT Broker

服務	技術棧	職責
postgres	PostgreSQL 16	事件資料持久化
html16	Next.js 16	Web 管理介面

2.4 資料流程



2.5 使用者與權限

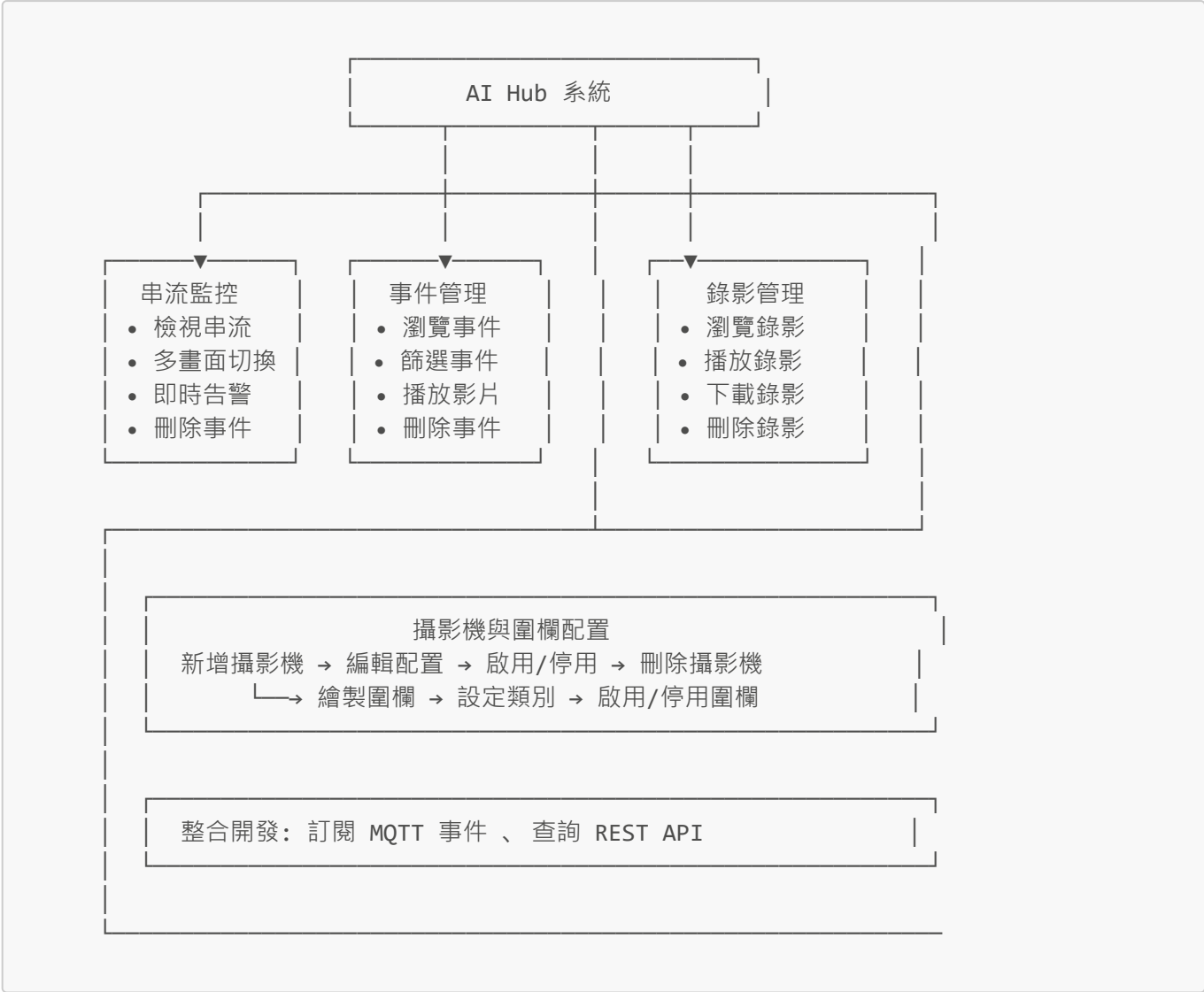
使用者	描述	權限
系統管理員	IT 人員	完全存取
監控人員	保全人員	串流檢視、事件瀏覽
管理階層	主管人員	報表查閱、紀錄回顧
整合開發者	第三方開發	API/MQTT 存取

### 3. 使用案例

#### 3.1 使用案例總覽

ID	名稱	參與者	優先級	相關需求
UC-01	檢視即時串流	所有使用者	P1	FR-STREAM-001~003, FR-WEB-001
UC-02	切換多畫面監控	系統管理員、監控人員	P1	FR-WEB-001
UC-03	接收即時告警通知	系統管理員、監控人員	P1	FR-FENCE-004, FR-WEB-001
UC-04	新增攝影機	系統管理員	P2	FR-WEB-002
UC-05	編輯攝影機配置	系統管理員	P2	FR-WEB-002, FR-AI-001
UC-06	啟用/停用攝影機	系統管理員	P2	FR-WEB-002
UC-07	刪除攝影機	系統管理員	P2	FR-WEB-002
UC-08	繪製虛擬圍欄	系統管理員	P1	FR-FENCE-001, FR-WEB-002
UC-09	設定觸發物件類別	系統管理員	P1	FR-FENCE-001
UC-10	啟用/停用圍欄	系統管理員	P1	FR-FENCE-001
UC-11	瀏覽事件列表	所有使用者	P1	FR-WEB-003
UC-12	篩選事件	所有使用者	P1	FR-WEB-003
UC-13	播放事件影片	所有使用者	P1	FR-RECORD-003, FR-WEB-003
UC-14	刪除事件	系統管理員	P2	FR-WEB-003
UC-15	瀏覽錄影檔案	所有使用者	P2	FR-RECORD-001
UC-16	播放歷史錄影	所有使用者	P2	FR-RECORD-001
UC-17	下載錄影檔案	所有使用者	P2	FR-RECORD-001
UC-18	刪除錄影檔案	系統管理員	P2	FR-RECORD-001
UC-19	訂閱 MQTT 事件	整合開發者	P1	FR-AI-003, FR-FENCE-004
UC-20	查詢 REST API	整合開發者	P3	FR-WEB-003

3.2 使用案例關係圖



3.3 UC-01: 檢視即時串流

項目	內容
ID	UC-01
名稱	檢視即時串流
參與者	所有使用者 (系統管理員、監控人員、管理階層)
優先級	P1 (必要)
觸發條件	使用者開啟 Web 儀表板
前置條件	1. 使用者已開啟瀏覽器 2. 至少一台攝影機已配置並啟用 3. 串流服務正常運行
後置條件	使用者可在瀏覽器中看到即時影像
相關需求	FR-STREAM-001, FR-STREAM-002, FR-STREAM-003, FR-WEB-001

主要流程 (Main Flow):

步驟	使用者動作	系統回應
1	開啟瀏覽器輸入 http://host:3000	載入 Web 儀表板頁面
2	-	讀取 cameras.json 取得攝影機清單
3	-	篩選 enabled=true 的攝影機
4	-	為每個攝影機建立 HLS 播放器實例
5	-	連線至 go2rtc HLS 端點 /api/stream.m3u8?src={streamId}
6	-	顯示串流畫面，狀態標籤顯示「線上」(綠色)
7	觀看即時影像	持續更新畫面 (15 FPS)

替代流程 (Alternative Flows):

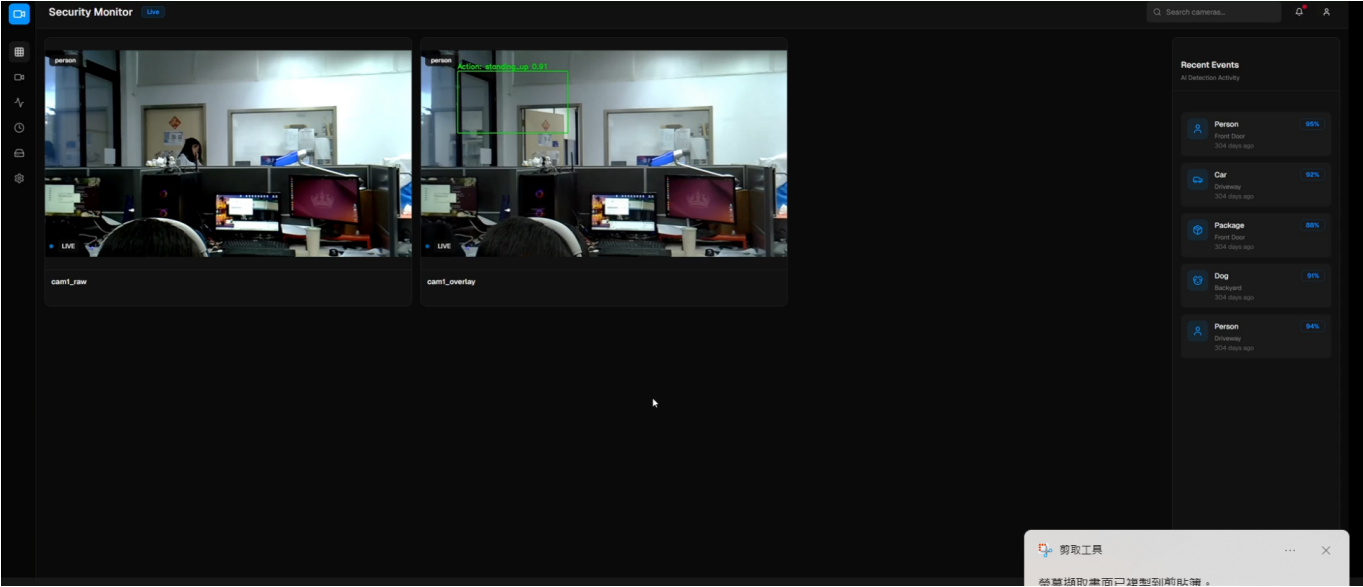
代碼	條件	處理方式
3a	無啟用的攝影機	顯示「尚未配置攝影機」提示訊息
5a	HLS 連線失敗	顯示「連線中…」，每 5 秒自動重試
5b	連續失敗 3 次	顯示「連線失敗」(紅色)，提供「重試」按鈕
6a	串流延遲 > 3 秒	顯示「延遲過高」警告

例外流程 (Exception Flows):

代碼	條件	處理方式
E1	go2rtc 服務未啟動	顯示「串流服務不可用」錯誤
E2	網路斷線	顯示「網路連線中斷」，自動重連

UI 設計:


即時監控儀表板



即時監控畫面：多畫面監控、串流狀態、事件通知



Event Details



Event ID

evt\_003

Object Type

package

Zone

entrance

Camera ID

cam1

Camera

Front Door

Confidence


88%


Timestamp

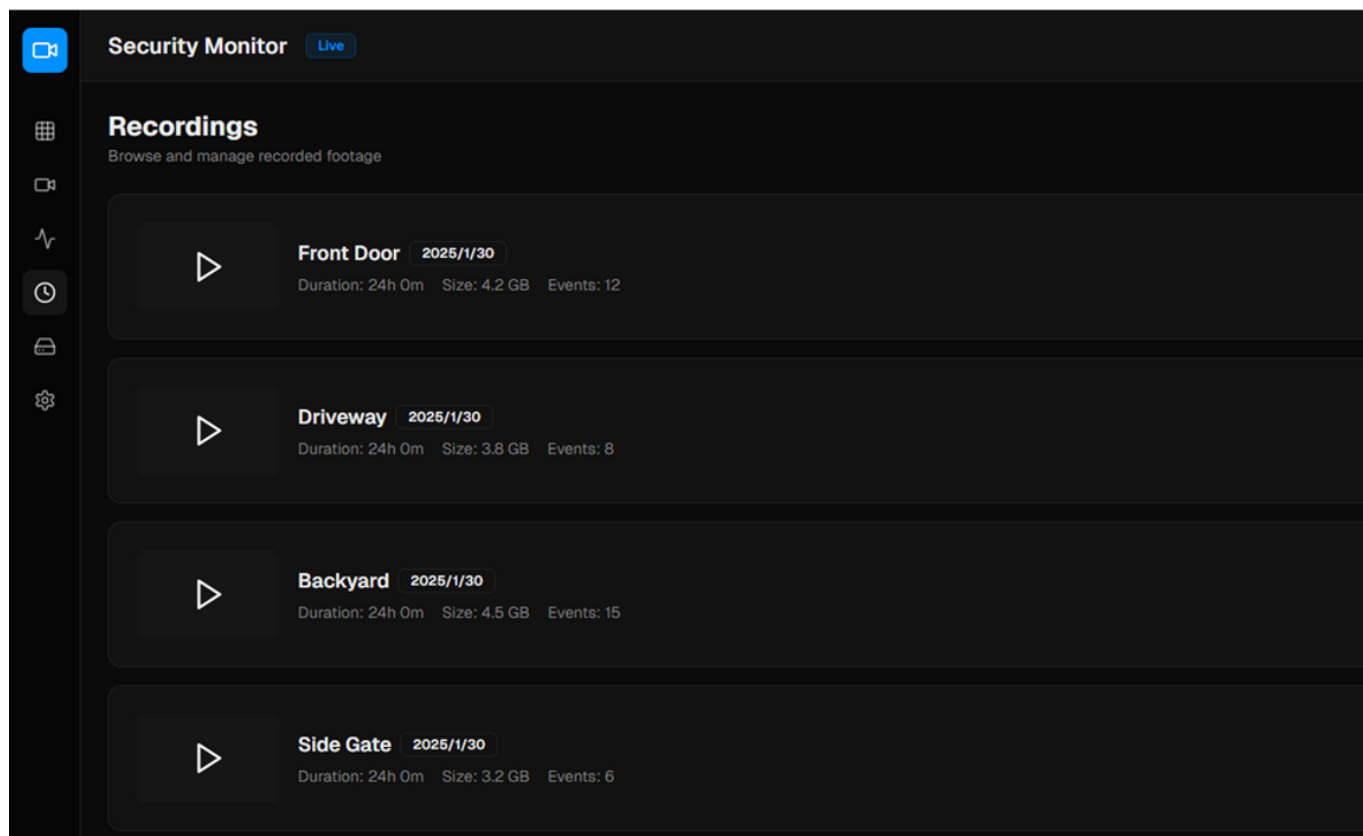
2025/1/30 下午8:10:45

Relative Time

2025/1/30

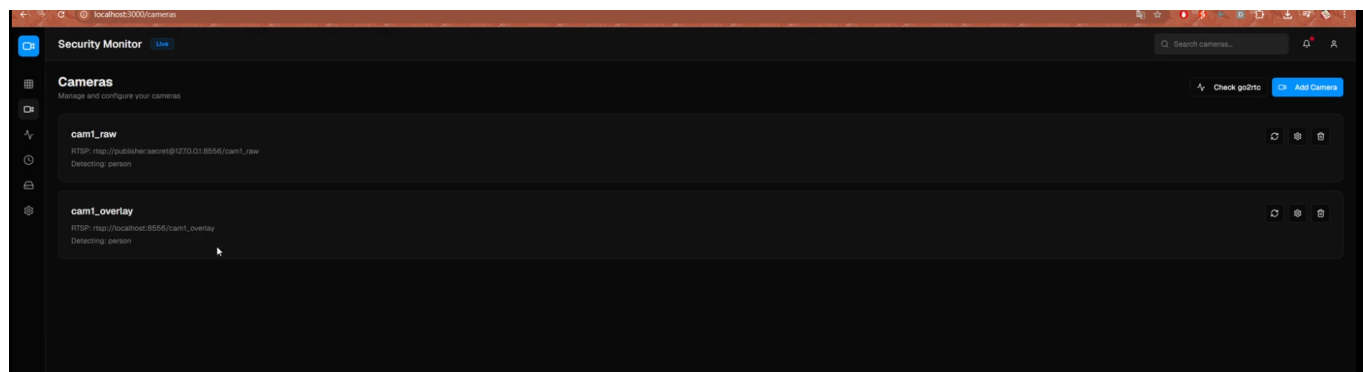
 Download

 Delete



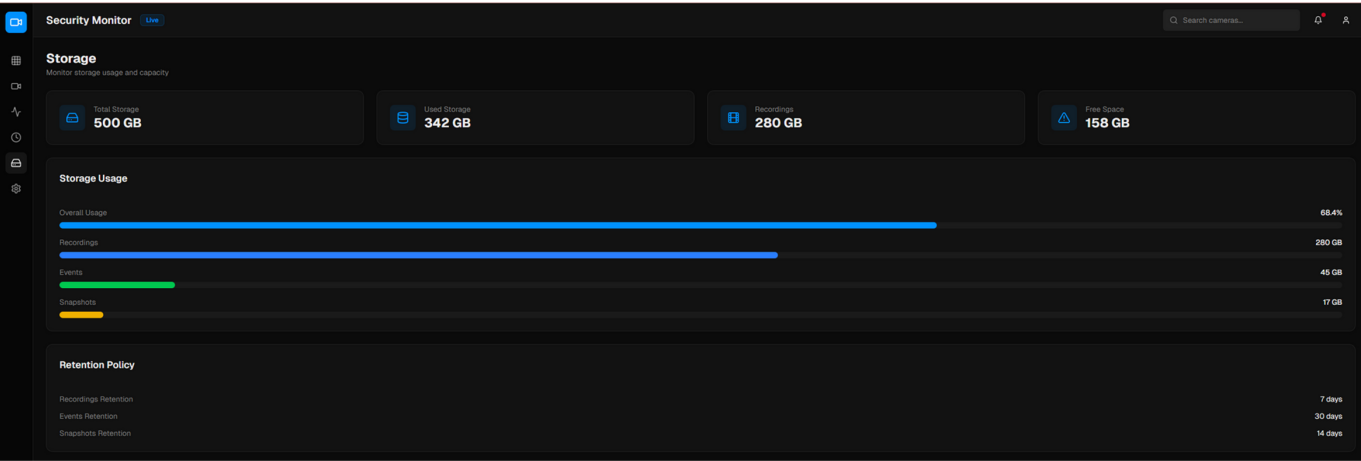
事件瀏覽：時間軸、事件縮圖、快速播放

## 攝影機配置



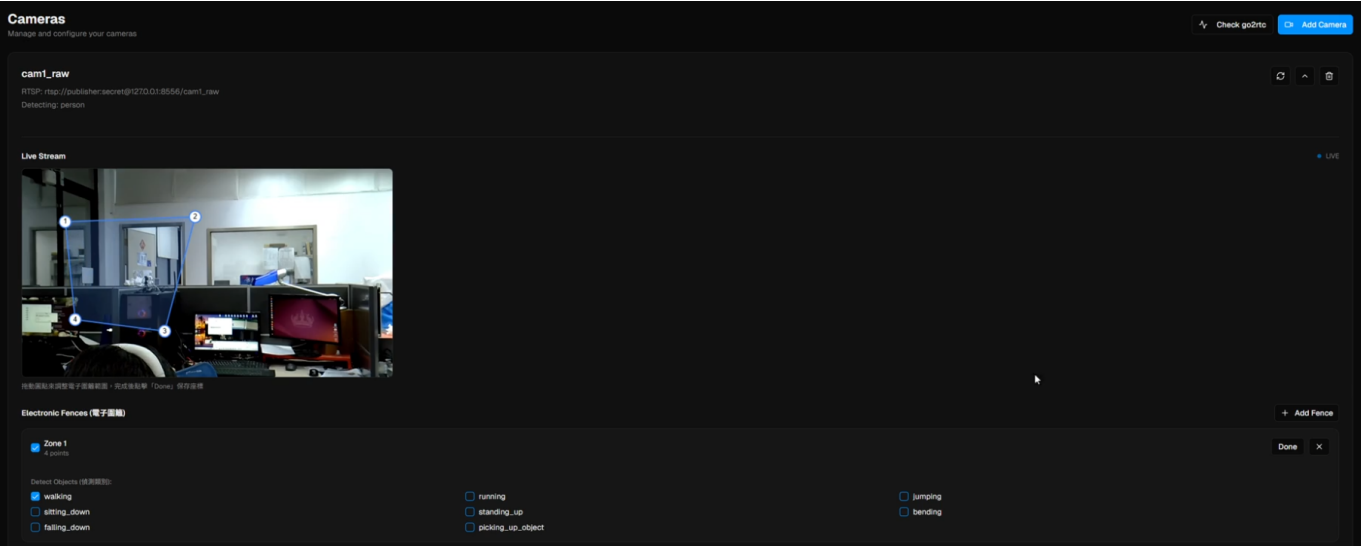
攝影機設定：參數配置、虛擬圍欄繪製

狀態檢查



狀態檢查：系統狀態、已使用空間

電子圍籬



電子圍籬：虛擬圍欄繪製、圍籬設定

3.4 UC-02: 切換多畫面監控

項目	內容
ID	UC-02
名稱	切換多畫面監控
參與者	系統管理員、監控人員
優先級	P1 (必要)
前置條件	已有多台攝影機配置
後置條件	畫面顯示選定的佈局模式

主要流程:

步驟	使用者動作	系統回應
1	點擊佈局選擇器 (如 [2×2])	-
2	-	重新計算畫面網格
3	-	調整每個播放器尺寸為 (容器寬 / 列數) × (容器高 / 行數)
4	-	根據啟用攝影機數量填充格位
5	-	儲存偏好至 LocalStorage

佈局規格:

佈局	格數	適用場景
1×1	1	單攝影機放大檢視
2×2	4	一般監控
3×3	9	多點監控
4×4	16	大型場域

3.5 UC-03: 接收即時告警通知

項目	內容
ID	UC-03
名稱	接收即時告警通知
參與者	系統管理員、監控人員
優先級	P1 (必要)
觸發條件	虛擬圍欄偵測到入侵
前置條件	1. 儀表板已開啟 2. 虛擬圍欄已配置並啟用 3. WebSocket 連線正常
後置條件	使用者收到告警並可查看詳情

主要流程:

步驟	系統動作	說明
1	AI 偵測到物件進入圍欄	model_launcher → MQTT detections
2	fence 服務判斷入侵	Ray Casting 演算法
3	產生事件並發布 MQTT	vision/{cam}/events

步驟	系統動作	說明
4	寫入 PostgreSQL events 表	持久化
5	Web UI 透過 WebSocket 接收	MQTT over WS
6	右上角顯示告警通知	Toast 元件
7	播放告警音效	(可配置)
8	通知徽章數字 +1	未讀計數

### 3.6 UC-04: 新增攝影機

項目	內容
<b>ID</b>	UC-04
<b>名稱</b>	新增攝影機
<b>參與者</b>	系統管理員
<b>優先級</b>	P2 (重要)
<b>前置條件</b>	使用者具有管理權限
<b>後置條件</b>	新攝影機設定儲存至 cameras.json

主要流程:

步驟	使用者動作	系統回應
1	進入攝影機管理頁面	顯示攝影機列表
2	點擊「新增攝影機」	開啟新增表單 Modal
3	填寫基本資訊	即時驗證欄位
4	點擊「測試連線」	嘗試連線 RTSP
5	-	顯示連線結果 (成功/失敗)
6	點擊「儲存」	驗證必填欄位
7	-	寫入 cameras.json
8	-	通知相關服務重載配置
9	-	關閉 Modal · 刷新列表

表單欄位:

欄位	類型	必填	驗證規則	預設值
ID	text	✓	^[a-z0-9_-]{1,32}\$, 唯一	-
名稱	text	✓	長度 1-100	-

欄位	類型	必填	驗證規則	預設值
位置	text	○	長度 0-200	-
RTSP URL	text	✓	有效 URL 格式	rtsp://127.0.0.1:8554/
解析度	select	✓	640x480/1280x720/1920x1080	1280x720
幀率	number	✓	1-60	15
AI 模型	select	○	models.json 中的模型 ID	-
啟用錄影	checkbox	○	-	true
啟用快照	checkbox	○	-	true

### 3.7 UC-05: 編輯攝影機配置

項目	內容
<b>ID</b>	UC-05
<b>名稱</b>	編輯攝影機配置
<b>參與者</b>	系統管理員
<b>優先級</b>	P2 (重要)

主要流程:

步驟	使用者動作	系統回應
1	在列表中點擊攝影機	進入詳情頁面
2	點擊「編輯」按鈕	表單變為可編輯狀態
3	修改欄位值	即時驗證
4	點擊「儲存」	驗證所有欄位
5	-	更新 cameras.json
6	-	顯示「儲存成功」提示

### 3.8 UC-06: 啟用/停用攝影機

項目	內容
<b>ID</b>	UC-06
<b>名稱</b>	啟用/停用攝影機
<b>參與者</b>	系統管理員

主要流程:

步驟	使用者動作	系統回應
1	點擊攝影機的啟用開關	顯示確認對話框
2	確認操作	更新 cameras.json 的 enabled 欄位
3	-	通知服務重載
4	-	更新 UI 狀態

影響範圍:

- 停用攝影機會：停止串流、停止 AI 偵測、停止錄影
- 啟用攝影機會：恢復所有相關服務

### 3.9 UC-07: 刪除攝影機

項目	內容
ID	UC-07
名稱	刪除攝影機
參與者	系統管理員

主要流程:

步驟	使用者動作	系統回應
1	點擊「刪除」按鈕	顯示警告對話框
2	輸入攝影機名稱確認	驗證輸入
3	點擊「確認刪除」	從 cameras.json 移除
4	-	(選用) 刪除相關錄影檔案
5	-	顯示「刪除成功」

注意事項:

- 刪除為不可逆操作
- 相關事件紀錄保留 (可選是否一併刪除)

### 3.10 UC-08: 繪製虛擬圍欄

項目	內容
ID	UC-08
名稱	繪製虛擬圍欄

項目	內容
參與者	系統管理員
優先級	P1 (必要)
前置條件	已選擇目標攝影機，串流正常顯示
後置條件	圍欄配置儲存至 cameras.json
相關需求	FR-FENCE-001

主要流程:

步驟	使用者動作	系統回應
1	進入攝影機配置頁面	載入攝影機設定與現有圍欄
2	點擊「新增圍欄」按鈕	進入繪製模式，游標變為十字
3	在畫面上點擊第一個頂點	顯示頂點標記 (圓點)
4	依序點擊 2-9 個頂點	即時顯示邊線預覽
5	雙擊結束或點擊「完成」	封閉多邊形，顯示填充效果
6	-	開啟圍欄屬性表單
7	填寫名稱、選擇觸發類別	-
8	點擊「儲存」	驗證配置
9	-	寫入 cameras.json
10	-	通知 fence 服務重載

座標轉換邏輯:

```
// 點擊座標轉正規化座標
function normalizePoint(clientX, clientY, canvas) {
  const rect = canvas.getBoundingClientRect();
  return {
    x: (clientX - rect.left) / rect.width, // 0-1
    y: (clientY - rect.top) / rect.height // 0-1
  };
}
```

### 3.11 UC-09: 設定觸發物件類別

項目	內容
ID	UC-09



項目	內容
名稱	設定觸發物件類別
參與者	系統管理員

主要流程:

步驟	使用者動作	系統回應
1	在圍欄屬性表單中	顯示可選類別清單 (多選)
2	勾選/取消勾選類別	更新 detectObjects 陣列
3	儲存	寫入 cameras.json

可選類別來源:

- 1. 攝影機的 `availableDetectionObjects` 配置
- 2. AI 模型支援的類別 (從 `class_file` 讀取)

### 3.12 UC-10: 啟用/停用圍欄

項目	內容
ID	UC-10
名稱	啟用/停用圍欄
參與者	系統管理員

主要流程:

步驟	使用者動作	系統回應
1	點擊圍欄的啟用開關	更新 <code>virtualFences[].enabled</code>
2	-	儲存 <code>cameras.json</code>
3	-	通知 <code>fence</code> 服務

效果:

- 停用圍欄：該圍欄不再觸發事件
- 啟用圍欄：恢復事件偵測

3.13 UC-11: 瀏覽事件列表

項目	內容
ID	UC-11
名稱	瀏覽事件列表
參與者	所有使用者
優先級	P1 (必要)
前置條件	資料庫中有事件紀錄
相關需求	FR-WEB-003

主要流程:

步驟	使用者動作	系統回應
1	點擊側邊欄「事件」	跳轉至 /events 頁面
2	-	查詢最近 20 筆事件 (ORDER BY ts DESC)
3	-	以卡片形式顯示事件列表
4	向下捲動	觸發 Infinite Scroll · 載入下 20 筆
5	點擊單一事件卡片	開啟事件詳情 Modal

事件卡片資訊:

欄位	說明
縮圖	影片第一幀 (若有)
攝影機名稱	從 camera_id 對應
類別	class_name
信心度	score (百分比顯示)
時間	ts (相對時間: "5 分鐘前")
圍欄名稱	fence_name

3.14 UC-12: 篩選事件

項目	內容
ID	UC-12
名稱	篩選事件
參與者	所有使用者

篩選條件:

篩選項	類型	說明
攝影機	下拉選單	全部 / 指定攝影機
類別	多選下拉	person, car, ...
時間範圍	日期選擇器	開始日期 ~ 結束日期
信心度	滑桿	最低信心度 (0-100%)

SQL 查詢範例:

```
SELECT * FROM events
WHERE camera_id = 'cam1'
  AND class_name IN ('person', 'car')
  AND ts BETWEEN '2025-01-01' AND '2025-01-31'
  AND score >= 0.8
ORDER BY ts DESC
LIMIT 20 OFFSET 0;
```

### 3.15 UC-13: 播放事件影片

項目	內容
<b>ID</b>	UC-13
<b>名稱</b>	播放事件影片
<b>參與者</b>	所有使用者
<b>前置條件</b>	事件已裁剪完成 (video_path 不為空)

主要流程:

步驟	使用者動作	系統回應
1	點擊事件卡片的播放按鈕	開啟影片播放 Modal
2	-	載入 /api/events/{id}/video
3	-	使用 HTML5 video 播放 .mkv
4	點擊播放/暫停	控制播放
5	關閉 Modal	停止播放

### 3.16 UC-14: 刪除事件

項目	內容
<b>ID</b>	UC-14

項目	內容
名稱	刪除事件
參與者	系統管理員

主要流程:

步驟	使用者動作	系統回應
1	勾選要刪除的事件	顯示批次操作列
2	點擊「刪除選中」	顯示確認對話框
3	確認刪除	從資料庫刪除記錄
4	-	刪除對應的影片檔案
5	-	刷新列表

### 3.17 UC-15: 瀏覽錄影檔案

項目	內容
<b>ID</b>	UC-15
<b>名稱</b>	瀏覽錄影檔案
<b>參與者</b>	所有使用者

主要流程:

步驟	使用者動作	系統回應
1	進入錄影頁面	顯示攝影機列表
2	選擇攝影機	顯示日期列表 (有錄影的日期)
3	選擇日期	顯示該日錄影片段列表
4	-	顯示檔案名、大小、時長

### 3.18 UC-16: 播放歷史錄影

項目	內容
<b>ID</b>	UC-16
<b>名稱</b>	播放歷史錄影
<b>參與者</b>	所有使用者

主要流程:

步驟	使用者動作	系統回應
1	點擊錄影檔案	開啟播放器頁面
2	-	載入 .mkv 檔案
3	播放/暫停/拖曳進度條	控制播放

### 3.19 UC-17: 下載錄影檔案

項目	內容
<b>ID</b>	UC-17
<b>名稱</b>	下載錄影檔案
<b>參與者</b>	所有使用者

主要流程:

步驟	使用者動作	系統回應
1	點擊「下載」按鈕	觸發 HTTP GET
2	-	設定 Content-Disposition: attachment
3	-	瀏覽器下載 .mkv 檔案

### 3.20 UC-18: 刪除錄影檔案

項目	內容
<b>ID</b>	UC-18
<b>名稱</b>	刪除錄影檔案
<b>參與者</b>	系統管理員

主要流程:

步驟	使用者動作	系統回應
1	勾選錄影檔案	-
2	點擊「刪除」	顯示確認對話框 (含檔案大小總計)
3	確認	刪除檔案系統中的檔案
4	-	刷新列表

### 3.21 UC-19: 訂閱 MQTT 事件

項目	內容
ID	UC-19
名稱	訂閱 MQTT 事件
參與者	整合開發者
優先級	P1 (必要)
前置條件	1. MQTT Broker 運行中 2. 開發者有連線憑證
相關需求	FR-AI-003, FR-FENCE-004

主要流程:

步驟	開發者動作	系統回應
1	建立 MQTT 連線	Broker 回應 CONNACK
2	訂閱 Topic	Broker 回應 SUBACK
3	等待訊息	當有事件時推送 PUBLISH
4	處理訊息	-

連線範例 (Python):

```
import paho.mqtt.client as mqtt
import json

def on_connect(client, userdata, flags, rc):
    print(f"Connected with result code {rc}")
    client.subscribe("vision/+/events") # 訂閱所有攝影機的事件

def on_message(client, userdata, msg):
    event = json.loads(msg.payload.decode())
    print(f"Event: {event['id']}")
    print(f"  Camera: {event['camera_id']}")
    print(f"  Class: {event['class_name']}")
    print(f"  Score: {event['score']:.2%}")
    print(f"  Time: {event['ts']}")
    print(f"  Fence: {event['fence_name']}")

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

# 連線至 Broker
client.connect("localhost", 1883, 60)
```

```
# 持續監聽
client.loop_forever()
```

#### Topic 清單:

Topic	說明	QoS	頻率
<code>vision/{cam}/detections</code>	即時偵測結果	0	每幀
<code>vision/{cam}/events</code>	入侵事件	1	事件驅動
<code>vision/{cam}/status</code>	攝影機狀態	0	每 30 秒

#### 事件訊息完整格式:

```
{
  "id": "evt_20251229163542_a1b2c3",
  "camera_id": "cam1_raw",
  "class_name": "person",
  "ts": "2025-12-29T16:35:42.123Z",
  "score": 0.952,
  "bbox": [0.12, 0.25, 0.45, 0.88],
  "center": [0.285, 0.565],
  "fence_id": "fence_001",
  "fence_name": "入口區域",
  "alert_level": "high"
}
```

### 3.22 UC-20: 查詢 REST API

項目	內容
ID	UC-20
名稱	查詢 REST API
參與者	整合開發者
優先級	P3 (選用)

#### API 端點清單:

端點	方法	說明
<code>/api/cameras</code>	GET	取得攝影機列表
<code>/api/cameras/:id</code>	GET	取得單一攝影機
<code>/api/events</code>	GET	查詢事件
<code>/api/events/:id</code>	GET	取得事件詳情



端點	方法	說明
/api/events/:id/video	GET	下載事件影片
/api/recordings/:cam/:date	GET	取得錄影列表

查詢事件範例:

```
curl -X GET "http://localhost:3000/api/events?camera=cam1&limit=10" \
-H "Accept: application/json"
```

回應格式 (成功):

```
{
  "success": true,
  "data": [
    {
      "id": "evt_xxx",
      "camera_id": "cam1",
      "class_name": "person",
      "ts": "2025-12-29T16:35:42.000Z",
      "score": 0.95,
      "fence_name": "入口區域",
      "video_path": "/events/evt_xxx.mkv"
    }
  ],
  "meta": {
    "total": 150,
    "page": 1,
    "pageSize": 10
  }
}
```

回應格式 (錯誤):

```
{
  "success": false,
  "error": {
    "code": "INVALID_CAMERA",
    "message": "Camera 'cam99' not found"
  }
}
```

## 4. 功能性需求

### 4.1 視訊串流處理模組 (FR-STREAM)

#### FR-STREAM-001: USB 攝影機擷取

項目	內容
ID	FR-STREAM-001
優先級	P1 (必要)
描述	系統應能從 USB 攝影機擷取原始影像並編碼為 RTSP 串流

規格:

參數	規格
裝置	/dev/video0, /dev/video1, ...
輸入格式	YUYV422 / MJPEG
輸出解析度	640×480, 1280×720, 1920×1080 (可配置)
輸出幀率	10, 15, 30 FPS (可配置)
輸出編碼	H.264 (libx264 / h264_vaapi)
協議	RTSP over TCP

驗收條件:

- ☐ 系統啟動 30 秒內偵測到 USB 攝影機
- ☐ 輸出 RTSP 可透過 VLC/ffplay 播放
- ☐ 攝影機拔除後 10 秒內偵測並記錄錯誤
- ☐ 攝影機重新接入後自動恢復串流

#### FR-STREAM-002: AI 辨識結果疊加

項目	內容
ID	FR-STREAM-002
優先級	P1 (必要)
描述	系統應將 AI 偵測結果 ( 邊界框、標籤 ) 即時疊加於串流畫面

規格:

參數	規格
輸入串流	rtsp://go2rtc:8554/cam1_raw
輸出串流	rtsp://go2rtc:8554/cam1_overlay

參數	規格
偵測資料	MQTT: vision/cam1/detections
邊界框	依物件類別區分顏色，線寬 2px
標籤	"{類別} {信心度%}"
處理延遲	> 10ms

驗收條件:

- ☐ 邊界框位置與偵測結果一致
- ☐ 疊加延遲 > 10ms
- ☐ 無偵測結果時畫面正常顯示
- ☐ MQTT 斷線時不影響原始串流

**FR-STREAM-003: 多協議串流輸出**

項目	內容
ID	FR-STREAM-003
優先級	P1 (必要)
描述	go2rtc 應提供 RTSP、HLS、MSE 等多種協議輸出

埠口配置:

協議	埠口	用途
RTSP	8554	VLC, ffmpeg
HLS/HTTP	1984	瀏覽器播放
WebRTC	8555	超低延遲 (保留)

驗收條件:

- ☐ RTSP 串流可用 VLC 正常播放
- ☐ HLS 可在 Chrome/Edge 播放
- ☐ 支援  $\geq 10$  個同時連線

**FR-STREAM-004: 硬體加速編碼**

項目	內容
ID	FR-STREAM-004
優先級	P2 (重要)
描述	系統應支援 Intel VA-API 硬體加速，降低 CPU 負載

偵測邏輯:

1. 檢查 /dev/dri/renderD128 是否存在
2. 讀取環境變數 ACTION\_HWACCEL (auto/vaapi/none)
3. 選擇編碼器: VA-API 可用用 h264\_vaapi，否則 libx264

驗收條件:

- ☐ 啟用 VA-API 時 CPU 使用率降低 ≥ 30%
- ☐ 無硬體時自動 fallback 至軟體編碼

## 4.2 AI 物件偵測模組 (FR-AI)

### FR-AI-001: 模型配置管理

項目	內容
ID	FR-AI-001
優先級	P1 (必要)
描述	系統應透過 models.json 配置多個 AI 模型

配置結構:

```
{
  "models": [{
    "id": "yolov8n",
    "name": "YOLOv8 Nano",
    "type": "yolov8",
    "weights": "/models/yolov8n.pt",
    "class_file": "/class/coco.txt",
    "inputSize": [640, 640],
    "device": "cpu",
    "confidence": 0.5,
    "runner": "/app/yolov8_inference.py"
  }]
}
```

驗收條件:

- ☐ 支援同時配置 ≥ 5 個模型
- ☐ 配置檔語法錯誤時顯示明確錯誤訊息

### FR-AI-002: 動態模型啟動

項目	內容
ID	FR-AI-002
優先級	P1 (必要)

項目	內容
描述	model_launcher 應根據 cameras.json 中的 modelId 自動啟動對應模型

驗收條件:

- ☐ 攝影機配置變更後 10 秒內重新匹配模型
- ☐ 同一模型可服務多個攝影機
- ☐ 推論程序異常時 5 秒內自動重啟

#### FR-AI-003: 偵測結果發布

項目	內容
ID	FR-AI-003
優先級	P1 (必要)
描述	推論結果應透過 MQTT 發布至 vision/{cameraId}/detections

訊息格式:

```
{
  "cameraId": "cam1",
  "timestamp": "2025-01-15T14:32:15.123Z",
  "inferenceTime": 45.2,
  "detections": [{
    "class_name": "person",
    "bbox": [0.12, 0.25, 0.45, 0.88],
    "score": 0.952,
    "center": [0.285, 0.565]
  }]
}
```

驗收條件:

- ☐ 偵測結果發布延遲 < 100ms
- ☐ bbox 座標為正規化格式 [0-1]
- ☐ 無偵測時發布空 detections 陣列

### 4.3 虛擬圍欄模組 (FR-FENCE)

#### FR-FENCE-001: 多邊形圍欄定義

項目	內容
ID	FR-FENCE-001
優先級	P1 (必要)

項目	內容
描述	使用者應能在畫面上定義 3-10 個頂點的多邊形監控區域

配置結構:

```
{
  "virtualFences": [{
    "id": "fence_001",
    "name": "入口區域",
    "enabled": true,
    "points": [
      {"x": 0.2, "y": 0.3},
      {"x": 0.8, "y": 0.3},
      {"x": 0.8, "y": 0.7},
      {"x": 0.2, "y": 0.7}
    ],
    "detectObjects": ["person", "car"],
    "alertLevel": "high"
  }]
}
```

驗收條件:

- ☐ 支援 3-10 個頂點的多邊形
- ☐ 每個攝影機支援  $\geq 5$  個圍欄
- ☐ 配置變更後 5 秒內生效

## FR-FENCE-002: Ray Casting 入侵判斷

項目	內容
ID	FR-FENCE-002
優先級	P1 (必要)
描述	系統應使用 Ray Casting 演算法判斷物件中心是否在圍欄內

觸發條件:

```
IF point_in_polygon(detection.center, fence.points)
  AND detection.class_name IN fence.detectObjects
  AND fence.enabled == true
  AND NOT in_cooldown(fence.id, detection.class_name)
THEN trigger_event()
```

驗收條件:

- ☐ 判斷準確率  $\geq 99.9\%$

- ☐ 單次判斷延遲 < 1ms

### FR-FENCE-003: 告警冷卻機制

項目	內容
ID	FR-FENCE-003
優先級	P1 (必要)
描述	避免同一物件/圍欄組合頻繁觸發告警

參數:

參數	環境變數	預設值
冷卻期	FENCE_COOLDOWN_SEC	30 秒
離開時間	FENCE_LEAVE_SEC	5 秒

驗收條件:

- ☐ 冷卻期內不重複觸發
- ☐ 物件離開後經過 LEAVE\_SEC 可重新觸發

### FR-FENCE-004: 事件產生與通知

項目	內容
ID	FR-FENCE-004
優先級	P1 (必要)
描述	入侵事件應同時記錄至資料庫並發布 MQTT

事件 ID 格式: `evt_{timestamp}_{random}`

驗收條件:

- ☐ 事件 100% 寫入資料庫
- ☐ MQTT 發布延遲 < 200ms
- ☐ 事件 ID 全域唯一

---

## 4.4 錄影管理模組 (FR-RECORD)

### FR-RECORD-001: 持續錄影

項目	內容
ID	FR-RECORD-001
優先級	P2 (重要)

項目	內容
描述	系統應 24×7 持續錄製串流，按配置時長分段儲存

參數:

參數	環境變數	預設值
分段長度	SEGMENT_SECONDS	300 秒
輸出格式	固定	.mp4

目錄結構: `recordings/{cameraId}/{YYYY-MM}/{DD}/HH-MM-SS.mp4`

**FR-RECORD-002: 緩衝錄影**

項目	內容
ID	FR-RECORD-002
優先級	P1 (必要)
描述	維護短期循環緩衝用於事件裁剪

參數:

參數	環境變數	預設值
啟用	EVENT_BUFFER_ENABLED	1
緩衝長度	EVENT_BUFFER_SECONDS	10 秒
分段長度	EVENT_BUFFER_SEGMENT_SECONDS	1 秒

**FR-RECORD-003: 事件片段裁剪**

項目	內容
ID	FR-RECORD-003
優先級	P1 (必要)
描述	監聽事件 MQTT，裁剪事前 N 秒與事後 M 秒影片

參數:

參數	環境變數	預設值
事前秒數	EVENT_PRE_SECONDS	10 秒
事後秒數	EVENT_POST_SECONDS	10 秒

輸出: `share/events/{eventId}.mp4`



驗收條件:

- ☐ 裁剪影片可正常播放
- ☐ 時間精確度 ±1 秒

---

## 4.5 Web 管理介面模組 (FR-WEB)

### FR-WEB-001: 即時監控儀表板

項目	內容
ID	FR-WEB-001
優先級	P1 (必要)
描述	提供多畫面即時串流監控與告警通知

功能:

- 多畫面佈局: 1×1, 2×2, 3×3, 4×4
- 串流播放: HLS / MSE
- 狀態顯示: 線上/離線/錯誤
- 即時告警: WebSocket 推播

驗收條件:

- ☐ 頁面載入 < 3 秒
- ☐ 支援同時顯示 ≥ 4 路串流
- ☐ 告警通知即時推播 < 2 秒

### FR-WEB-002: 攝影機配置介面

項目	內容
ID	FR-WEB-002
優先級	P1 (必要)
描述	提供攝影機 CRUD 操作與圍欄繪製介面

驗收條件:

- ☐ 配置即時生效無需重啟
- ☐ 圍欄可視覺化繪製
- ☐ 刪除需二次確認

### FR-WEB-003: 事件瀏覽介面

項目	內容
ID	FR-WEB-003

項目	內容
優先級	P1 (必要)
描述	提供事件列表、篩選、播放功能

功能:

- 分頁載入: 每頁 20 筆
- 篩選: 攝影機、類別、時間、信心度
- 縮圖預覽
- 影片播放

驗收條件:

- ☐ 篩選回應 < 1 秒
  - ☐ 影片可在瀏覽器播放
-

## 5. 非功能性需求

### 5.1 效能需求 (NFR-PERF)

ID	需求	目標
NFR-PERF-001	串流延遲 (HLS)	< 500ms
NFR-PERF-002	串流延遲 (RTSP)	< 300ms
NFR-PERF-003	AI 推論速度 (YOLOv8n CPU)	≥ 10 FPS
NFR-PERF-004	事件反應時間	< 2 秒
NFR-PERF-005	並發處理 (4核8GB)	4 路 720p
NFR-PERF-006	Web 首頁載入	< 3 秒
NFR-PERF-007	API 回應 P99	< 500ms

### 5.2 可靠性需求 (NFR-REL)

ID	需求	說明
NFR-REL-001	自動重連	串流/MQTT 斷線 10 秒內重試
NFR-REL-002	服務隔離	單服務故障不影響其他服務
NFR-REL-003	資料持久化	事件/錄影不因重啟遺失
NFR-REL-004	系統可用性	≥ 99.5% (月)

#### 重連策略:

- 首次重試: 2 秒
- 最大間隔: 30 秒
- 使用 Exponential Backoff with Jitter

### 5.3 可擴展性需求 (NFR-SCALE)

ID	需求	說明
NFR-SCALE-001	水平擴展	支援多節點部署
NFR-SCALE-002	模型熱插拔	新增模型無需重啟
NFR-SCALE-003	攝影機擴展	單節點 ≥ 8 路

### 5.4 安全性需求 (NFR-SEC)

ID	需求	說明
NFR-SEC-001	RTSP 認證	publisher:secret 認證
NFR-SEC-002	MQTT 認證	username/password

ID	需求	說明
NFR-SEC-003	資料庫安全	生產環境更換預設密碼
NFR-SEC-004	網路隔離	PostgreSQL 不暴露公網

5.5 可維護性需求 (NFR-MAINT)

ID	需求	說明
NFR-MAINT-001	容器化	Docker Compose 一鍵部署
NFR-MAINT-002	統一日誌	JSON 格式，含時間戳與服務名
NFR-MAINT-003	配置分離	環境變數 + JSON 配置檔

5.6 相容性需求 (NFR-COMPAT)

瀏覽器支援:

瀏覽器	版本	支援
Chrome	90+	✓ 完全支援
Edge	90+	✓ 完全支援
Firefox	88+	✓ 完全支援
Safari	-	X 不支援

作業系統支援:

OS	版本	支援
Ubuntu	20.04+	✓
Debian	11+	✓
CentOS	8+	社群

---

## 6. 介面需求

### 6.1 使用者介面

技術規格:

項目	規格
框架	Next.js 16 + React 18
樣式	Tailwind CSS
即時通訊	WebSocket (MQTT over WS)

頁面結構:

/	# 儀表板首頁
— /cameras	# 攝影機管理
— /events	# 事件瀏覽
— /recordings	# 錄影管理
— /settings	# 系統設定
— /status	# 系統狀態

響應式斷點:

名稱	寬度	最大格數
Mobile	< 640px	2×2
Tablet	640-1024px	3×3
Desktop	> 1024px	4×4

### 6.2 硬體介面

硬體	規格
USB 攝影機	UVC 1.1+, USB 2.0/3.0
裝置路徑	/dev/video0, /dev/video1
Intel VAAPI	/dev/dri/renderD128

### 6.3 軟體介面

#### go2rtc API

端點	方法	用途
/api/streams	GET	列出所有串流
/api/stream?src={id}	GET	取得串流資訊

MQTT Topics

Topic	方向	說明
vision/{cam}/detections	Pub	偵測結果
vision/{cam}/events	Pub	入侵事件
vision/{cam}/status	Pub	攝影機狀態

6.4 通訊介面

服務	埠口	對外
go2rtc RTSP	8554	選用
go2rtc HTTP	1984	選用
MQTT	1883	選用
PostgreSQL	5432	否
Web UI	3000	是

---

## 7. 資料需求

### 7.1 資料庫 Schema

#### events 表

```
CREATE TABLE events (  
  id VARCHAR(64) PRIMARY KEY,  
  camera_id VARCHAR(64) NOT NULL,  
  class_name VARCHAR(128) NOT NULL,  
  ts TIMESTAMP WITH TIME ZONE NOT NULL,  
  score REAL,  
  fence_id VARCHAR(64),  
  fence_name VARCHAR(128),  
  thumbnail VARCHAR(512),  
  video_path VARCHAR(512),  
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);  
  
CREATE INDEX idx_events_camera_ts ON events(camera_id, ts DESC);  
CREATE INDEX idx_events_class ON events(class_name);
```

#### recordings 表 (未來)

```
CREATE TABLE recordings (  
  id VARCHAR(64) PRIMARY KEY,  
  camera_id VARCHAR(64) NOT NULL,  
  start_ts TIMESTAMP WITH TIME ZONE NOT NULL,  
  end_ts TIMESTAMP WITH TIME ZONE NOT NULL,  
  file_path VARCHAR(512) NOT NULL,  
  file_size BIGINT,  
  duration REAL  
);
```

### 7.2 配置檔案

#### cameras.json

欄位	類型	說明
id	string	攝影機 ID
name	string	顯示名稱
enabled	boolean	是否啟用
stream.resolution	string	解析度
stream.fps	integer	幀率

欄位	類型	說明
ai.modelId	string	AI 模型 ID
virtualFences	array	圍欄配置

**models.json**

欄位	類型	說明
id	string	模型 ID
name	string	模型名稱
weights	string	權重路徑
class_file	string	類別檔路徑
inputSize	[int,int]	輸入尺寸
device	string	cpu/cuda:0

7.3 資料保留政策

資料類型	預設保留	環境變數
事件紀錄	90 天	EVENT_RETENTION_DAYS
事件影片	30 天	EVENT_VIDEO_RETENTION_DAYS
持續錄影	7 天	RECORDING_RETENTION_DAYS
緩衝錄影	10 秒	EVENT_BUFFER_SECONDS

---



## 8. 測試案例

### 8.1 功能測試

#### TC-STREAM-001: USB 攝影機串流

步驟	動作	預期結果
1	啟動服務	正常啟動
2	檢查日誌	攝影機偵測成功
3	VLC 開啟 rtsp://host:8554/cam1_raw	串流播放
4	拔除攝影機	日誌顯示斷線
5	重新接入	串流自動恢復

#### TC-FENCE-001: 虛擬圍欄觸發

步驟	動作	預期結果
1	人物在圍欄外	無事件
2	人物進入圍欄	事件觸發
3	驗證 MQTT	收到事件訊息
4	驗證資料庫	events 有新記錄
5	人物停留 30 秒內	無重複觸發

#### TC-RECORD-001: 事件片段裁剪

步驟	動作	預期結果
1	等待緩衝建立 (> 10s)	緩衝有檔案
2	觸發事件	事件產生
3	等待 POST+5 秒	裁剪完成
4	檢查 share/events/	有新 .mkv
5	播放影片	正常播放，約 20 秒

### 8.2 效能測試

#### TC-PERF-001: 串流延遲

- 方法: 攝影機前顯示毫秒時鐘，逐幀分析
- 目標: HLS < 500ms

#### TC-PERF-002: AI 推論速度

```
start = time.time()
for _ in range(100):
    model.predict(frame)
fps = 100 / (time.time() - start)
assert fps >= 10
```

#### TC-PERF-003: 並發連線

- 啟動 10 個 VLC 實例連線
- 驗證全部正常播放

### 8.3 安全測試

#### TC-SEC-001: RTSP 認證

步驟	動作	預期結果
1	無認證推流	拒絕
2	錯誤密碼推流	拒絕
3	正確密碼推流	成功

## 9. 附錄

### 9.1 需求追蹤矩陣

需求 ID	PRD 來源	UC 關聯	TC 關聯	優先級
FR-STREAM-001	3.1.1	UC-01	TC-STREAM-001	P1
FR-STREAM-002	3.1.2	UC-01	TC-STREAM-002	P1
FR-STREAM-003	3.1.3	UC-01	TC-STREAM-003	P1
FR-AI-001	3.2.1	UC-05	TC-AI-001	P1
FR-AI-002	3.2.2	UC-04	TC-AI-001	P1
FR-AI-003	3.2.3	UC-19	TC-AI-001	P1
FR-FENCE-001	3.3.1	UC-08	TC-FENCE-001	P1
FR-FENCE-002	3.3.2	UC-08	TC-FENCE-001	P1
FR-FENCE-003	3.3.2	UC-03	TC-FENCE-002	P1
FR-FENCE-004	3.3.3	UC-11	TC-FENCE-001	P1
FR-RECORD-001	3.4.1	UC-15	TC-RECORD-001	P2
FR-RECORD-002	3.4.2	UC-13	TC-RECORD-001	P1
FR-RECORD-003	3.4.3	UC-13	TC-RECORD-001	P1
FR-WEB-001	3.5.1	UC-01	TC-WEB-001	P1
FR-WEB-002	3.5.2	UC-04	TC-WEB-002	P1
FR-WEB-003	3.5.3	UC-11	TC-WEB-001	P1

### 9.2 環境變數清單

變數	預設值	說明
VIDEO_DEVICE	/dev/video0	攝影機裝置
VIDEO_RESOLUTION	1280x720	解析度
VIDEO_FPS	15	幀率
ACTION_HWACCEL	auto	硬體加速
MODEL_CONFIDENCE	0.5	信心度閾值
FENCE_COOLDOWN_SEC	30	冷卻期
FENCE_LEAVE_SEC	5	離開重置
SEGMENT_SECONDS	300	錄影分段
EVENT_BUFFER_SECONDS	10	緩衝長度

變數	預設值	說明
EVENT_PRE_SECONDS	10	事件前秒數
EVENT_POST_SECONDS	10	事件後秒數
POSTGRES_HOST	postgres	DB 主機
POSTGRES_PORT	5432	DB 埠口
POSTGRES_DB	aihub	資料庫名
MQTT_HOST	mqtt	MQTT 主機
MQTT_PORT	1883	MQTT 埠口

9.3 專案結構

```
action_recognition_server/
├── app/                # 行為辨識服務
├── models_classify/    # 模型推論服務
├── fence/              # 虛擬圍欄服務
├── record/             # 錄影服務
├── html116/            # Next.js Web UI
├── go2rtc/             # 串流伺服器配置
├── MQTT/               # MQTT Broker 配置
├── postgres/           # 資料庫初始化
├── share/              # 共享資料目錄
│   ├── cameras.json
│   ├── models.json
│   ├── events/
│   └── recordings/
├── doc/                # 文件資源
├── docker-compose.yml
├── prd_customer.md
├── prd_internal.md
└── srs.md              # 本文件
```

9.4 修訂紀錄

版本	日期	說明
1.0	2025-12-29	初版建立

## 10. 功能實作細節

### 10.1 串流處理模組細節

#### 10.1.1 raw\_restreamer - USB 攝影機擷取

細項功能	規格說明
裝置支援	/dev/video0, /dev/video1, ... (UVC 1.1+ 相容)
輸入格式	YUYV422, MJPEG (自動偵測)
輸出解析度	640×480, 1280×720, 1920×1080
輸出幀率	10, 15, 30 FPS
輸出編碼	H.264 (libx264 / h264_vaapi)
輸出協議	RTSP over TCP
位元率	CBR 2000-2500 kbps
編碼優化	ultrafast preset, zerolatency tune
斷線處理	自動偵測 · 2 秒後重連 · 最多 10 次
GOP 設定	fps × 2 (例: 15fps → GOP=30)

#### FFmpeg 命令範例:

```
ffmpeg -f v4l2 -input_format yuyv422 \  
-video_size 1280x720 -framerate 15 \  
-i /dev/video0 \  
-c:v libx264 -preset ultrafast -tune zerolatency \  
-b:v 2000k -maxrate 2500k -bufsize 5000k \  
-g 30 -bf 0 \  
-f rtsp -rtsp_transport tcp \  
rtsp://publisher:secret@go2rtc:8554/cam1_raw
```

#### 10.1.2 action\_recognition\_server - AI 疊加

細項功能	規格說明
輸入串流	rtsp://go2rtc:8554/cam1_raw
輸出串流	rtsp://go2rtc:8554/cam1_overlay
偵測資料來源	MQTT: vision/cam1/detections
邊界框顏色	綠色 (0, 255, 0) · 可依類別配置
邊界框線寬	2 像素

細項功能	規格說明
標籤格式	<code>{class_name} {score:.0%}</code>
標籤字體	HERSHEY_SIMPLEX, size=0.8
標籤位置	邊界框左上角上方 10 像素
解析度調整	自動縮放 (cv2.INTER_LINEAR)
處理延遲	< 100ms

硬體加速選項 (ACTION\_HWACCEL):

選項	說明
auto	自動偵測 /dev/dri/renderD128
vaapi	強制使用 Intel VA-API
none	純軟體編碼 (libx264)

### 10.1.3 go2rtc - 多協議串流伺服器

細項功能	規格說明
RTSP 輸入	接收 publisher 推流，認證可選
RTSP 輸出	Port 8554, TCP/UDP
HLS 輸出	Port 1984, /api/stream.m3u8?src={id}
MSE 輸出	Port 1984, /stream.html?src={id}
WebRTC	Port 8555 (保留)
API 端點	/api/streams, /api/stream
並發連線	≥ 10 同時連線
認證	publisher:secret (可配置)

串流 ID 命名規則:

ID	用途
cam1_raw	原始串流 (無 AI)
cam1_overlay	疊加串流 (含 AI 結果)

## 10.2 AI 物件偵測模組細節

### 10.2.1 model\_launcher - 模型管理

細項功能	規格說明
配置檔位置	share/models.json
支援模型類型	YOLOv8 (n/s/m/l/x)
模型匹配	精確匹配 + slugify 模糊匹配
多攝影機	同一模型可服務多個攝影機
URL 重寫	127.0.0.1 自動轉為容器內 host
程序監控	異常時 5 秒內自動重啟
配置熱載入	支援 (需觸發 reload)

#### models.json 完整配置:

```
{
  "models": [{
    "id": "yolov8n",
    "name": "YOLOv8 Nano",
    "type": "yolov8",
    "weights": "/models/yolov8n.pt",
    "class_file": "/class/coco.txt",
    "inputSize": [640, 640],
    "device": "cpu",
    "confidence": 0.5,
    "iou": 0.45,
    "runner": "/app/runner/yolov8.py"
  }]
}
```

#### 10.2.2 推論引擎

細項功能	規格說明
推論框架	Ultralytics YOLOv8
推論速度	YOLOv8n: $\geq 10$ FPS (CPU 4核)
信心度閾值	預設 0.5 (MODEL_CONFIDENCE)
IOU 閾值 (NMS)	預設 0.45 (MODEL_IOU)
輸入尺寸	640×640 (可配置)
輸出格式	正規化座標 [x1, y1, x2, y2]
類別檔案	/class/coco.txt (80 類)
自訂類別	支援自訂 class_file

支援的偵測類別:

- COCO 80 類: person, car, truck, bus, bicycle, dog, cat, ...
- 自訂類別: jumping, sitting\_down, standing\_up, falling\_down, picking\_up\_object

10.2.3 MQTT 偵測結果發布

細項功能	規格說明
Topic	vision/{cameraId}/detections
QoS	0 (At most once)
發布頻率	每幀一次 (同步於推論速度)
Retain	false

訊息欄位說明:

欄位	類型	說明
cameraId	string	攝影機 ID
timestamp	ISO8601	偵測時間戳
frameId	int	幀序號 (遞增)
inferenceTime	float	推論耗時 (ms)
detections[].class_id	int	類別編號
detections[].class_name	string	類別名稱
detections[].bbox	[x1,y1,x2,y2]	正規化邊界框
detections[].score	float	信心度 (0-1)
detections[].center	[x,y]	中心點座標

10.3 虛擬圍欄模組細節

10.3.1 圍欄區域定義

細項功能	規格說明
配置位置	cameras.json → virtualFences
多邊形頂點	3-10 個
座標格式	正規化座標 (0-1)
座標轉換	像素座標自動正規化
圍欄數量	每攝影機無限制 (建議 ≤5)



virtualFences 配置欄位:

欄位	類型	必填	說明
name	string	✓	圍欄名稱
points	array	✓	多邊形頂點 [{x, y}, ...]
enabled	boolean	✓	是否啟用
detectObjects	array	✓	觸發類別清單
alertLevel	string	○	告警等級 (low/medium/high)

10.3.2 Ray Casting 入侵偵測

細項功能	規格說明
演算法	Ray Casting (光線投射法)
判斷點	偵測框中心點 (center)
支援形狀	凸多邊形、凹多邊形
處理時間	< 1ms (單次判斷)
準確率	≥ 99.9%

觸發條件 (全部滿足):

- 1. `point_in_polygon(detection.center, fence.points) == true`
- 2. `detection.class_name` IN `fence.detectObjects`
- 3. `fence.enabled == true`
- 4. `NOT in_cooldown(fence.name, detection.class_name)`

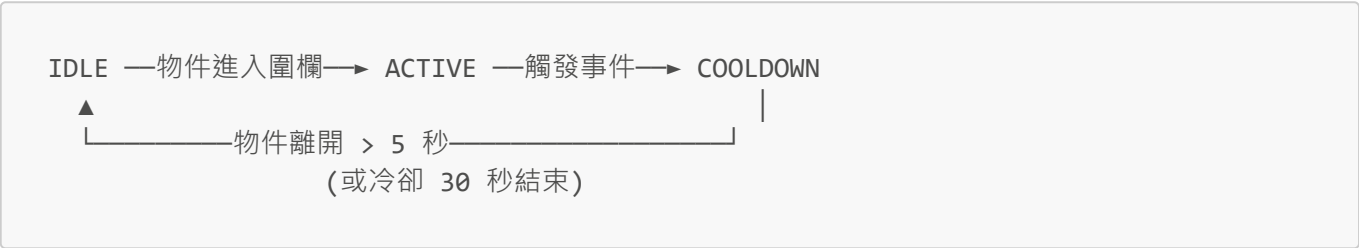
演算法實作:

```
def point_in_polygon(x, y, polygon):
    """Ray Casting: 從點向右發射射線，計算交點數"""
    inside = False
    n = len(polygon)
    j = n - 1
    for i in range(n):
        xi, yi = polygon[i]['x'], polygon[i]['y']
        xj, yj = polygon[j]['x'], polygon[j]['y']
        if ((yi > y) != (yj > y)) and \
            (x < (xj - xi) * (y - yi) / (yj - yi) + xi):
            inside = not inside
        j = i
    return inside # 奇數交點 = 內部
```

10.3.3 冷卻機制

細項功能	規格說明
冷卻期	FENCE_COOLDOWN_SEC = 30 秒
離開重置	FENCE_LEAVE_SEC = 5 秒
冷卻鍵值	{camera}:{fence}:{class}
獨立冷卻	不同圍欄/類別各自獨立

狀態機流程:



10.3.4 事件產生

細項功能	規格說明
事件 ID	evt_{timestamp}_{uuid6位}
資料庫	PostgreSQL events 表
MQTT Topic	vision/{cameraId}/events
MQTT QoS	1 (At least once)

事件記錄欄位:

欄位	來源	說明
id	自動生成	全域唯一 ID
camera_id	偵測訊息	攝影機 ID
class_name	偵測訊息	觸發類別
ts	偵測訊息	事件時間
score	偵測訊息	信心度
fence_name	圍欄配置	觸發圍欄名稱

10.4 錄影管理模組細節

10.4.1 持續錄影 (Recorder)

細項功能	規格說明
錄影模式	24×7 全時錄影
分段長度	SEGMENT_SECONDS = 300 秒
時鐘對齊	segment_atclocktime (整點)
輸入格式	RTSP (copy, 不重編碼)
中間格式	.ts (MPEG-TS)
輸出格式	.mkv (Matroska)
後處理	ts → mkv 自動轉檔
穩定等候	POSTPROCESS_STABLE_SECONDS = 2 秒

目錄結構:

```

recordings/
├── {cameraId}/
│   ├── {YYYY-MM}/
│   │   ├── {DD}/
│   │   │   ├── 00-00-00.mkv
│   │   │   ├── 00-05-00.mkv
│   │   │   └── ...

```

#### 10.4.2 緩衝錄影 (BufferRecorder)

細項功能	規格說明
啟用	EVENT_BUFFER_ENABLED = 1
緩衝長度	EVENT_BUFFER_SECONDS = 10 秒
分段長度	EVENT_BUFFER_SEGMENT_SECONDS = 1 秒
重新編碼	EVENT_BUFFER_REENCODE = 1
GOP 大小	EVENT_BUFFER_GOP = 10
自動清理	超過保留時長自動刪除

目錄結構:

```

recordings_buffer/
├── {cameraId}/
│   ├── {YYYY-MM}/{DD}/
│   │   ├── 14-32-10.ts
│   │   ├── 14-32-11.ts
│   │   └── ... (循環覆蓋)

```

### 10.4.3 事件裁剪 (EventClipper)

細項功能	規格說明
觸發方式	MQTT: vision/+/events
事前秒數	EVENT_PRE_SECONDS = 10 秒
事後秒數	EVENT_POST_SECONDS = 10 秒
穩定等候	SEGMENT_READY_GRACE = 2 秒
最大等候	SEGMENT_MAX_WAIT = 15 秒
輸出路徑	share/events/{eventId}.mkv
輸出長度	PRE + POST 秒 (約 20 秒)
DB 更新	events.thumbnail, events.video_path

裁剪流程:

1. 接收事件 MQTT (vision/cam1/events)
2. 解析事件時間 ts
3. 計算裁剪範圍 [ts - PRE, ts + POST]
4. 等待 POST\_SECONDS + GRACE 時間
5. 從緩衝區收集「事前」片段列表
6. 錄製「事後」片段 (即時串流)
7. 建立 FFmpeg concat 清單
8. 執行串接與精確裁剪
9. 輸出至 share/events/{eventId}.mkv
10. 更新資料庫 thumbnail 欄位

## 10.5 Web 管理介面細節

### 10.5.1 技術堆疊

項目	技術
框架	Next.js 16 (App Router)
UI 函式庫	React 18
樣式	Tailwind CSS
狀態管理	React Context + SWR
串流播放	HLS.js
即時通訊	MQTT over WebSocket

### 10.5.2 頁面功能

頁面	路徑	功能
儀表板	/	多畫面串流監控、即時告警
攝影機管理	/cameras	CRUD、圍欄編輯
事件瀏覽	/events	列表、篩選、播放
錄影管理	/recordings	目錄瀏覽、播放、下載
系統設定	/settings	參數配置
儲存空間	/storage	使用量顯示

### 10.5.3 元件清單

元件	檔案	功能
VideoPlayer	camera-feed.tsx	HLS 串流播放器
CameraGrid	camera-grid.tsx	多畫面網格佈局
EventsList	events-list.tsx	事件卡片列表
FenceEditor	(內建)	Canvas 圍欄繪製
Sidebar	sidebar.tsx	側邊導航選單
TopBar	top-bar.tsx	頂部狀態列

### 10.5.4 API 端點

端點	方法	功能
/api/cameras	GET	取得攝影機列表
/api/cameras/[id]	GET/PUT	取得/更新攝影機
/api/camera-config	POST	儲存攝影機配置
/api/events	GET	查詢事件 (支援篩選)
/api/events/[id]	DELETE	刪除事件
/api/recordings	GET	取得錄影列表
/api/go2rtc/*	Proxy	go2rtc API 代理

### 10.6 環境變數完整清單

類別	變數	預設值	說明
----	----	-----	----

類別	變數	預設值	說明
通用	TZ	Asia/Taipei	時區
	LOG_LEVEL	INFO	日誌等級
串流	VIDEO_DEVICE	/dev/video0	USB 裝置
	VIDEO_RESOLUTION	1280x720	解析度
	VIDEO_FPS	15	幀率
	ACTION_HWACCEL	auto	硬體加速
	VAAPI_DEVICE	/dev/dri/renderD128	VAAPI 裝置
AI	MODEL_CONFIDENCE	0.5	信心度閾值
	MODEL_IOU	0.45	NMS IOU
	MODEL_DEVICE	cpu	推論裝置
圍欄	FENCE_COOLDOWN_SEC	30	冷卻期 (秒)
	FENCE_LEAVE_SEC	5	離開重置 (秒)
錄影	SEGMENT_SECONDS	300	持續錄影分段
	EVENT_BUFFER_ENABLED	1	緩衝啟用
	EVENT_BUFFER_SECONDS	10	緩衝長度
	EVENT_BUFFER_SEGMENT_SECONDS	1	緩衝分段
	EVENT_BUFFER_REENCODE	1	緩衝重編碼
	EVENT_BUFFER_GOP	10	緩衝 GOP
	EVENT_PRE_SECONDS	10	事件前秒數
	EVENT_POST_SECONDS	10	事件後秒數
	SEGMENT_READY_GRACE	2	穩定等候
	SEGMENT_MAX_WAIT	15	最大等候
資料庫	POSTGRES_HOST	postgres	DB 主機
	POSTGRES_PORT	5432	DB 埠口
	POSTGRES_DB	aihub	資料庫名
	POSTGRES_USER	aihub	使用者
	POSTGRES_PASSWORD	(必填)	密碼
MQTT	MQTT_HOST	mqtt	Broker 主機
	MQTT_PORT	1883	Broker 埠口
	MQTT_USERNAME	(選用)	認證帳號

類別	變數	預設值	說明
	MQTT_PASSWORD	(選用)	認證密碼

文件版本: v1.0

最後更新: 2025-12-29