STUDENT HANDBOOK

AI Detection with YOLO26

สร้างกล้อง AI ตรวจจับผู้บุกรุก แจ้งเตือนเข้าอีเมล

สร้างระบบกล้องวงจรปิดอัจฉริยะที่ตรวจจับคนและรถ แล้วแจ้งเตือนทางอีเมลพร้อมภาพและคลิป 8 วินาที ด้วย Claude Code ใน 2 ชั่วโมง



PHASE 1
ตรวจจับ
แผงควบคุมตรวจจับคน ด้วย YOLO26

PHASE 2-3
เฝ้าระวัง
เลือกคลาส พื้นที่ และช่วงเวลา

PHASE 4
เก็บหลักฐาน
ภาพ + คลิป 8 วินาที

PHASE 4
แจ้งเตือน
ตั้งค่าที่หลังบ้าน ส่งถึงมือถือ

OVERVIEW

ภาพรวมเวิร์กช็อป

วันนี้เราจะสร้าง **ระบบกล้องวงจรปิดอัจฉริยะ** ที่มองออกว่าในภาพมี "คน" หรือ "รถ" เข้ามาในพื้นที่ที่ห้ามเข้า ในช่วงเวลาที่ไม่ควรมีใคร แล้วส่งอีเมลแจ้งเตือนพร้อมภาพและคลิปหลักฐานทันที ทุกอย่างตั้งค่าและดูผลบน **หน้าเว็บของเราเอง 2 หน้า** — แผงควบคุม **Intrusion Detection Console** และหน้าหลังบ้าน **/admin** ที่ Claude Code สร้างให้จากคำสั่งภาษาไทย ไม่ต้องเขียนโค้ดเอง

01	02	03	04
วาง Prompt	**AI เขียนโค้ด**	**เปิดหน้าเว็บดูผล**	**ไปเฟสถัดไป**
คัดลอกจากคู่มือไปวางใน Claude Code	Claude Code เขียนและติดตั้งให้	เห็นกรอบตรวจจับบนวิดีโอจริงทันที	เพิ่มความสามารถทีละขั้น จนครบระบบ

วิธีใช้คู่มือเล่มนี้

PROMPT

ตัวอย่างกล่องคำสั่ง

ข้อความในกล่องนี้คือคำสั่งสำเร็จรูป คัดลอกไปวางใน `Claude Code` ได้ทันที
ส่วนที่เป็น **[ตัวสีแดงในวงเล็บ]** ให้เปลี่ยนเป็นข้อมูลของตัวเอง

คำศัพท์ในพรอมต์นี้

Prompt คำสั่งที่เราพิมพ์บอก AI ว่าต้องการอะไร

คำศัพท์ในพรอมต์นี้ กล่องใต้ทุก Prompt อธิบายคำเฉพาะในกล่องนั้นแบบเข้าใจง่าย อ่านก่อนวางได้

ปุ่มคัดลอก	สิ่งที่ควรเห็น	เคล็ดลับ	คำศัพท์
ฉบับเว็บ ทุกกล่องมีปุ่ม "คัดลอก Prompt" มุมขวาบน · แตะภาพเพื่อดูเต็มจอ	ภาพผลลัพธ์จริงจากระบบไว้เทียบกับหน้าจอของเรา	กล่องแดงอ่อน คือจุดที่มักพลาดหรือทางลัดที่ช่วยได้	ใต้ทุก Prompt มีคำอธิบายคำเฉพาะแบบเข้าใจง่าย

LEARNING OUTCOMES

ผลลัพธ์หลังเรียน

- อธิบายได้ว่า **AI Object Detection** คืออะไร และ **YOLO26** ทำงานอย่างไร
- ใช้ **Claude Code** สร้างหน้าเว็บระบบตรวจจับด้วย Python + OpenCV + Flask + YOLO26 ได้เองทีละเฟส โดยเข้าใจคำเฉพาะในทุก Prompt
- อ่าน **ความเร็วตรวจจับ (fps)** ของ YOLO26n บนแผงควบคุม และรู้วิธีช่วยเมื่อเครื่องช้า
- **เลือกคลาส วาดพื้นที่ตรวจจับ และกำหนดช่วงเวลา** ที่ต้องการเฝ้าระวังได้บนแผงควบคุม
- ได้ระบบที่ **ส่งอีเมลแจ้งเตือนพร้อมภาพ + คลิป 8 วินาที** (ก่อนเหตุ 3 วิ + หลังเหตุ 5 วิ) ผ่าน Gmail App Password ที่ตั้งค่าในหน้าหลังบ้าน /admin

TIMETABLE

ตารางสอน 120 นาที

เวลา	หัวข้อ	ทำอะไร
0:00–0:10	**เปิดคลาส**	แนะนำผู้สอน · ดูผลลัพธ์ปลายทาง (อีเมลแจ้งเตือนจริง) · เป้าหมายวันนี้
0:10–0:25	**AI Detection & YOLO26**	Detection คืออะไร · YOLO26 คืออะไร ทำไมต้องใช้ ทำงานอย่างไร · ช่วยกันขโมยอย่างไร
0:25–0:35	**เครื่องมือ + Setup**	Claude Code · Python · OpenCV · Flask · Ultralytics/YOLO26 · ffmpeg · Gmail — Prompt S0 แตก Starter Kit และจด "แบบหน้าเว็บ" ลง CLAUDE.md
0:35–0:55	**Phase 1 แผงควบคุม + ตรวจจับคน**	Prompt P1 · Intrusion Detection Console ที่ `127.0.0.1:8000` · การ์ดขั้น 1 แหล่งภาพ (คลิป / เว็บแคม / RTSP) · ล็อก YOLO26n · ตัวเลขสด 5 ช่อง
0:55–1:05	**Phase 2 เลือกคลาส**	Prompt P2 · การ์ดขั้น 2 เลือกจาก 80 คลาส ชื่อภาษาไทย · ค้นหา · ปุ่มเลือกเร็ว · คลาสแนะนำของคลิป
1:05–1:25	**Phase 3 พื้นที่ + ช่วงเวลา**	Prompt P3 · การ์ดขั้น 3 คลิกวาดพื้นที่บนภาพ · การ์ดขั้น 4 ความมั่นใจ + ช่วงเวลา 22:00–06:00
1:25–1:50	**Phase 4 หลักฐาน + อีเมล**	P4A ภาพ + คลิป 8 วิ · P4B หน้าหลังบ้าน /admin · สร้าง App Password · P4C ส่งอีเมลแจ้งเตือน
1:50–2:00	**ต่อยอด + ถาม-ตอบ**	Bonus B1 กล้อง RTSP · B2 ดูจากมือถือในวง Wi-Fi เดียวกัน · Q&A

> **หมายเหตุถึงผู้เรียน**
> ตารางเวลาเป็นแนวทาง ผู้สอนอาจปรับเวลาแต่ละช่วงให้เหมาะกับความคืบหน้าของผู้เรียน โดยเนื้อหาและผลลัพธ์หลักยังครบถ้วน

BEFORE CLASS

ก่อนเริ่ม — สิ่งที่ต้องเตรียม

เตรียม 6 อย่างนี้ให้พร้อมก่อนเข้าคลาส จะเริ่มลงมือได้ตั้งแต่ Prompt แรก

- ✓ **โน้ตบุ๊ก Mac หรือ Windows**
  RAM 8 GB ขึ้นไป · 1 คน 1 เครื่อง
- ✓ **ติดตั้ง Claude Code และล็อกอินแล้ว**
  แพ็กเกจ Pro ขึ้นไป · ลองพิมพ์สั่งงานได้ก่อนวันเรียน
- ✓ **Python 3.10–3.12**
  ติดตั้งไว้ก่อน ไลบรารีที่เหลือ AI ติดตั้งให้ใน Setup
- ✓ **บัญชี Gmail ที่เปิด 2-Step Verification**
  ใช้สร้าง App Password สำหรับส่งอีเมลแจ้งเตือน (Phase 4)
- ✓ **Starter Kit**
  ดาวน์โหลดจากกล่องด้านล่าง แล้ววางไฟล์ .zip ในโฟลเดอร์โปรเจกต์
- ✓ **อินเทอร์เน็ต**
  ครั้งแรกระบบต้องดาวน์โหลดไลบรารีและไฟล์โมเดล

ไม่ต้องมี GPU — YOLO26n รันบน CPU ได้ · **ไม่ต้องมีกล้องจริง** — ในคลาสใช้คลิปวิดีโอ · **ไม่ต้องเขียนโค้ดเป็น** — Claude Code เขียนให้

สำคัญ ดาวน์โหลด Starter Kit

คลิปกล้องวงจรปิด 7 คลิป (กลางวัน กลางคืน คน รถยนต์ มอเตอร์ไซค์) + ไฟล์ข้อมูลคลิป `videos/clips.yaml` (ชื่อภาษาไทย คลาสและพื้นที่ที่แนะนำ) + โลโก้ใน `brand/` ใช้ทุกเฟส

ดาวน์โหลดก่อนเรียน

yolo26-starter-kit.zip · 21 MB

วางไฟล์ .zip ไว้ในโฟลเดอร์โปรเจกต์ ไม่ต้องแตกเอง — Prompt S0 แตกให้

ดาวน์โหลด .zip

1. **สร้างโฟลเดอร์ว่างสำหรับโปรเจกต์**
   ตั้งชื่อเป็นภาษาอังกฤษ ไม่มีเว้นวรรค เช่น `yolo26-detection`

2. **ดาวน์โหลด Starter Kit แล้ววางไฟล์ .zip ไว้ในโฟลเดอร์นั้น**
   ไม่ต้องแตกไฟล์เอง Prompt S0 จะแตกให้ได้ `videos/` และ `brand/`

3. **เปิด Claude Code ในโฟลเดอร์นี้**
   เปิด Terminal ในโฟลเดอร์ พิมพ์ `claude` แล้วกด Enter · ทุก Prompt ในคู่มือวางที่นี่ เริ่มจาก Prompt S0

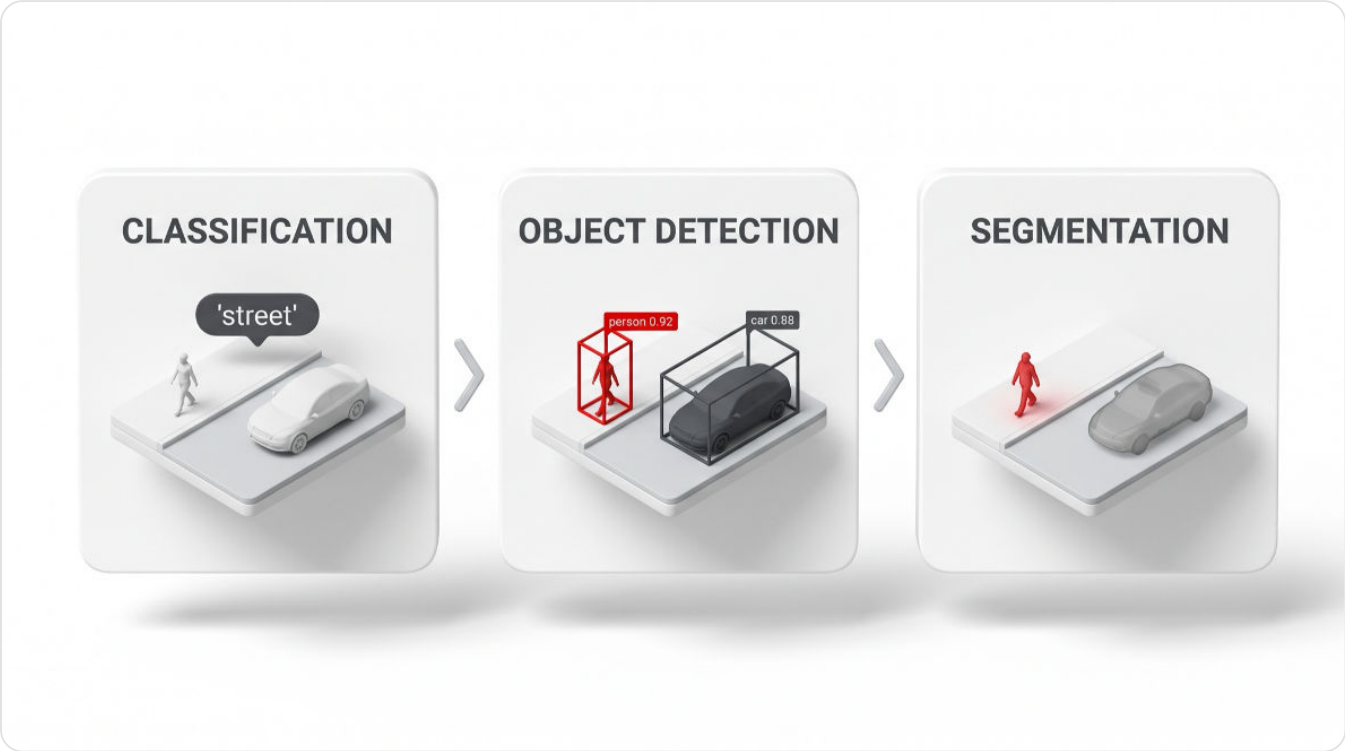
คลิปในชุดนี้ใช้ตอนไหน

ไฟล์ในโฟลเดอร์ videos/	ชื่อบนหน้าเว็บ (จาก clips.yaml)	ใช้ใน
`11_cctv_office_day.mp4`	โจรเปิดประตูเข้าห้องทำงาน	**P1 · P4A**
`14_cctv_parking_car.mp4`	รถยนต์เข้าลานจอด · มีพื้นที่แนะนำ	**P2**
`13_cctv_carport_motorcycle.mp4`	มอเตอร์ไซค์เข้าโรงรถ · มีพื้นที่แนะนำ	**P3**
`12_cctv_office_night_ir.mp4`	ห้องทำงานกลางคืน (ภาพ IR)	**P4C**
`01_door_entry.mp4`	คนเปิดประตูเข้ามา	**ฝึกเพิ่ม**
`02_hallway_door.mp4`	คนเดินมาที่ประตู	**ฝึกเพิ่ม**
`04_office_wide.mp4`	ออฟฟิศมุมกว้าง	**ฝึกเพิ่ม**

CONCEPT · 0:10–0:25

AI Detection คืออะไร

Computer Vision คือการให้คอมพิวเตอร์ "เข้าใจภาพ" แบ่งได้ 3 ระดับ — ระบบกันขโมยของเราใช้ระดับกลาง คือ **Object Detection**



Computer Vision 3 ระดับ — จากบอกว่าภาพนี้คืออะไร ไปจนถึงรูปร่างของวัตถุทีละพิกเซล

ระดับ 1

Classification

บอกว่า "ภาพนี้คืออะไร" ได้คำตอบเดียวทั้งภาพ

ระดับ 2 · ใช้วันนี้

Object Detection

บอกว่า "มีอะไร อยู่ตรงไหน กี่ชิ้น" ตีกรอบให้ทุกวัตถุ

ระดับ 3

Segmentation

บอก "รูปร่าง" ของวัตถุละเอียดทีละพิกเซล

ผลลัพธ์ต่อ 1 วัตถุ

กรอบ `(x, y, w, h)` + คลาส `person` + ความมั่นใจ `confidence 0.87`

วิดีโอ = ภาพนิ่งหลายภาพ

วิดีโอคือภาพนิ่งหลายสิบภาพต่อวินาที AI จึงตรวจได้ทุกเฟรมแบบ real-time

YOLO26 คืออะไร

- **YOLO = You Only Look Once** — มองภาพครั้งเดียว ทายทุกวัตถุพร้อมกัน จึงเร็วพอสำหรับวิดีโอสด
- **YOLO26** = รุ่นล่าสุดของ Ultralytics (ออก มกราคม 2026) ออกแบบมาให้เร็วบนเครื่องทั่วไปและอุปกรณ์ edge
- ฝึกมากับชุดข้อมูล **COCO 80 คลาส**: คน รถยนต์ มอเตอร์ไซค์ รถบรรทุก จักรยาน กระเป๋า สุนัข ฯลฯ — **ใช้ได้ทันทีไม่ต้องเทรน**
- ทำได้หลายงาน: `Detection` `Segmentation` `Pose` `Classification` `OBB (กรอบเอียง)`

YOLO26 มี 5 ขนาด — ใหญ่ขึ้นแม่นขึ้น แต่ช้าลง

รุ่น	ความแม่นยำ mAP50-95	ความเร็ว CPU ONNX (ms/ภาพ)	ขนาด Params (M)
YOLO26n · ใช้ในคลาส	40.9	38.9	2.4
YOLO26s	48.6	87.2	9.5
YOLO26m	53.1	220.0	20.4
YOLO26l	55.0	286.2	24.8
YOLO26x	57.5	525.8	55.7

ทดสอบบน COCO ขนาดภาพ 640 px · ข้อมูลจาก Ultralytics · ms ยิ่งน้อยยิ่งเร็ว · mAP ยิ่งสูงยิ่งแม่น

ทำไมต้อง YOLO26



จุดเด่นของ YOLO26 — ไม่ต้อง NMS · เร็วบน CPU · จับวัตถุเล็ก · นำไปใช้ได้ทุกอุปกรณ์

End-to-end ไม่ต้อง NMS
ทายกรอบสุดท้ายได้เลย ไม่ต้องมีขั้นกรองกรอบซ้ำ ขั้นตอนน้อยลง เร็วขึ้น นำไปใช้ง่าย

เร็วบน CPU
งานวิจัยของผู้พัฒนารายงานว่า YOLO26n เร็วกว่า YOLO11n บน CPU สูงสุดราว 43%

เบา นำไปใช้ได้ทุกที่
ตัด DFL ออก หัวทำนายเรียบง่าย export ไป ONNX / TensorRT / CoreML / OpenVINO ได้ง่าย

จับวัตถุเล็กดีขึ้น (STAL)
เหมาะกับภาพกล้องวงจรปิดมุมสูงที่คนในภาพดูตัวเล็ก

ใช้ง่ายแค่ 3 บรรทัด

```
from ultralytics import YOLO
model = YOLO("yolo26n.pt")
model(frame)
```

ไม่ต้องจำโค้ดนี้ — Claude Code เขียนให้ แค่รู้ว่า "สมอง" ของระบบเรียกใช้ง่ายแค่นี้

YOLO26 ทำงานอย่างไร



ภาพเดียวจบ — ภาพเข้า ผ่าน 3 ส่วนของโมเดล แล้วได้กรอบ คลาส และความมั่นใจกันที

01 **ภาพเข้า** เฟรมจากกล้องหรือวิดีโอ	02 **Backbone** ดึงลักษณะเด่น ขอบ รูปร่าง พื้นผิว	03 **Neck** รวมลักษณะหลายขนาด ของใหญ่และของเล็ก	04 **Head** ทายกรอบ + คลาส + confidence	05 **ผลลัพธ์** ได้ทันที ไม่ต้อง NMS

ช่วยกันขโมยอย่างไร



ปัญหาเดิม — กล้องบันทึกตลอด แต่ไม่มีใครดูทันเวลา



AI Detection — แจ้งเฉพาะคน/รถในพื้นที่ที่กำหนด ในเวลาที่ไม่ควรมีใคร

กล้องวงจรปิดทั่วไป
"บันทึก" แต่ไม่มีใครดู รู้ตัวก็เมื่อเสียหายไปแล้ว

Motion Detection
แจ้งเตือนทุกอย่างที่ขยับ (ใบไม้ เงา แมว) เตือนหลอกจนคนเลิกสนใจ

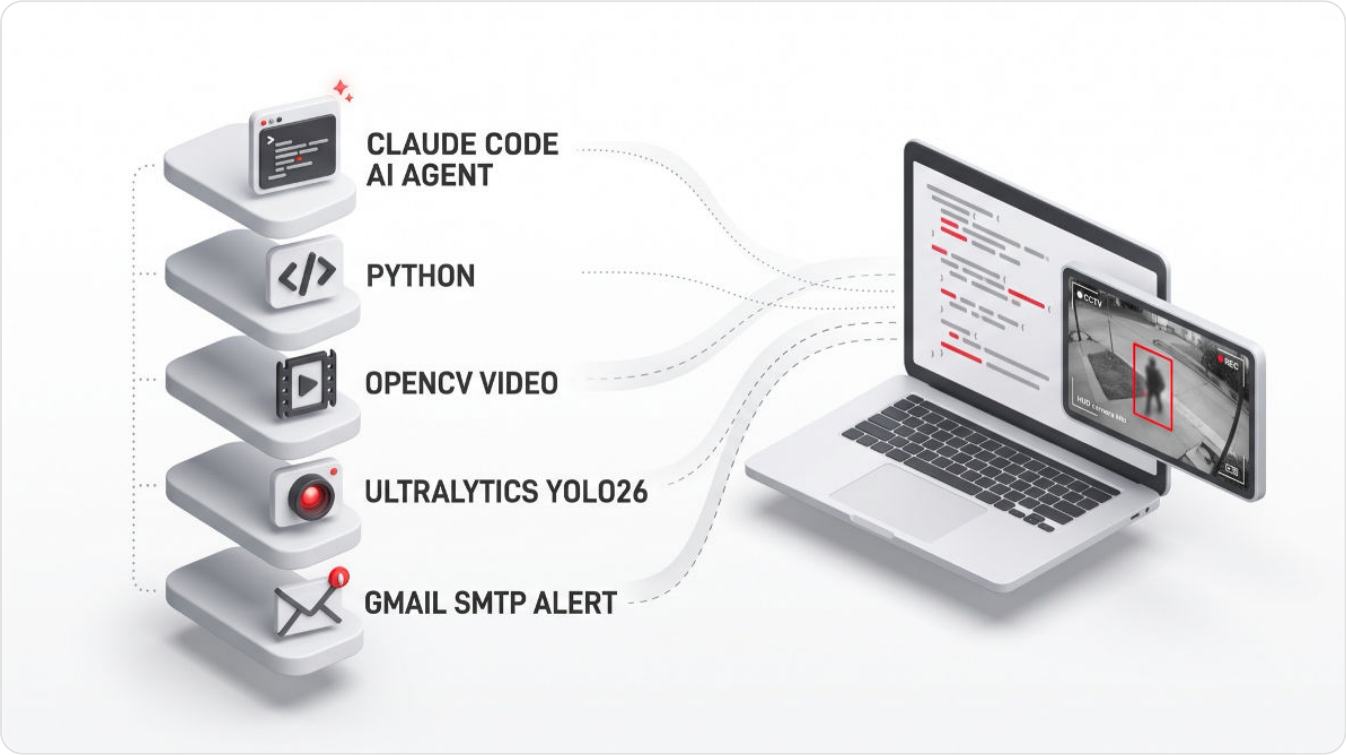
AI Detection
รู้ว่า "อะไร" ขยับ แจ้งเฉพาะคน/รถ ในพื้นที่ที่สนใจ ในเวลาที่ไม่ควรมีใคร

ส่งหลักฐานถึงมือถือทันที
ภาพ + คลิป ก่อนเหตุ 3 วินาที และหลังเหตุ 5 วินาที · ใช้ได้กับ บ้านกลางคืน · ร้านค้าหลังปิด · โกดัง · ลานจอดรถ · ห้อง server · แปลงเกษตร

TOOLS · 0:25–0:35

เครื่องมือวันนี้

ทุกเครื่องมือฟรีหรือใช้บัญชีที่มีอยู่แล้ว — Claude Code เป็นคนติดตั้งและต่อทุกอย่างเข้าด้วยกันให้



เครื่องมือหลักที่ประกอบเป็นระบบตรวจจับผู้บุกรุก

เครื่องมือ	บทบาท	ทำไมใช้
Claude Code	AI Agent เขียนโค้ด รันโค้ด แก้ error ให้ในโฟลเดอร์โปรเจกต์	สั่งเป็นภาษาไทย ไม่ต้องเขียนโค้ดเอง ได้ระบบจริงใน 2 ชม.
Python 3.10+	ภาษาหลักของ AI	ไลบรารี AI ครบที่สุด
OpenCV	อ่านวิดีโอ/กล้อง วาดกรอบ บันทึกคลิป	มาตรฐานงานภาพ ต่อ webcam/RTSP ได้
Ultralytics + YOLO26	"สมอง" ตรวจจับวัตถุ	โมเดลล่าสุด เร็ว ใช้ 3 บรรทัด
Flask	หน้าเว็บ 2 หน้า: แผงควบคุม (ดูภาพสด ตั้งค่าการตรวจจับ) + หน้าหลังบ้าน /admin (ตั้งค่าอีเมล)	เบา ติดตั้งง่าย กดเลือกแทนการแก้ไฟล์
imageio-ffmpeg	แปลงคลิปเป็น H.264	เปิดได้ทุกเครื่องและมือถือ ไม่ต้องลง ffmpeg เอง
Gmail SMTP + App Password	ส่งอีเมลแจ้งเตือน · ตั้งค่าที่หน้าหลังบ้าน /admin	ฟรี ใช้บัญชีที่มีอยู่ ปลอดภัยกว่าใช้รหัสจริง
python-dotenv (.env)	เก็บรหัสหลังบ้านและค่าอีเมลแยกจากโค้ด	รหัสไม่หลุดไปกับโค้ด

> **จำง่าย ๆ**
> Claude Code = วิศวกรที่เราสั่งงาน · Ultralytics = กล่องเครื่องมือ (เหมือนปลั๊กอิน) · YOLO26 = สมองที่มองเห็น · OpenCV = ตาและมือ · Gmail = ผู้ส่งสาร

ระบบทั้งหมดต่อกันอย่างไร



หน้าเว็บ (Flask) เป็นศูนย์กลางควบคุมและดูผล — ข้อมูลไหลจากกล้องจนถึงอีเมลแจ้งเตือน

01	02	03	04	05	06
กล้อง / วิดีโอ	**YOLO26**	**ตัวกรอง**	**เหตุการณ์**	**หลักฐาน**	**อีเมล Gmail**
OpenCV อ่านเฟรม + เก็บ buffer ย้อนหลัง	ตรวจจับวัตถุในทุกเฟรม	คลาส · พื้นที่ · ช่วงเวลา · เจอติดกัน 3 เฟรม	ขึ้นในรายการบนแผงควบคุม · พัก 60 วิ	snapshot + คลิป 8 วิ (H.264)	ภาพในตัว + ไฟล์แนบ · ตั้งค่าที่ /admin

หน้าเว็บ 2 หน้า เปิดด้วยคำสั่งเดียว

- **แผงควบคุม** `http://127.0.0.1:8000` — ดูภาพสด เลือกแหล่งภาพ เลือกคลาส วาดพื้นที่ ตั้งช่วงเวลา และดูหลักฐานย้อนหลัง
- **หน้าหลังบ้าน** `http://127.0.0.1:8000/admin` — ต้องใส่รหัสก่อนเข้า ตั้งบัญชีผู้ส่ง App Password และผู้รับอีเมล (เก็บในไฟล์ .env)
- เปิดทั้งสองหน้าด้วย `.venv/bin/python webapp.py` ไม่ต้องแก้ไฟล์เอง

SETUP · 0:25–0:35

S0 เตรียมโปรเจกต์ + แบบหน้าเว็บ

เป้าหมาย: เครื่องพร้อมรันระบบ — มี .venv ติดตั้งไลบรารีครบ แตก Starter Kit ได้ `videos/` กับ `brand/` และมี `CLAUDE.md` ที่จด **"แบบหน้าเว็บ"** ไว้ ทุกเฟสหลังจากนี้จึงได้หน้าตาเดียวกัน

PROMPT · S0

Setup · เตรียมโปรเจกต์ + แบบหน้าเว็บ

ช่วยเตรียมโปรเจกต์ "AI Detection ระบบตรวจจับผู้บุกรุกด้วย YOLO26" ในโฟลเดอร์นี้ให้หน่อย
1. ตรวจว่าเครื่องมี Python 3.10–3.12 หรือยัง (ถ้า python3 เป็น 3.13 ขึ้นไป ให้หา python3.12 หรือ 3.11 ที่มีในเครื่องแทน)
   ถ้าไม่มีเลยให้บอกวิธีติดตั้งสำหรับเครื่องฉัน (อย่าติดตั้งโปรแกรมระดับเครื่องเองโดยไม่ถาม)
2. สร้าง virtual environment ชื่อ .venv แล้วติดตั้ง ultralytics, opencv-python, flask, pyyaml, python-dotenv, imageio-ffmpeg
3. สร้าง requirements.txt และ .gitignore (ไม่เอา .venv, .env, events/, *.pt)
4. ถ้ามีไฟล์ yolo26-starter-kit.zip ให้แตกไฟล์ในโฟลเดอร์นี้ (จะได้โฟลเดอร์ videos/ และ brand/)
   แล้วสรุปคลิปใน videos/ เป็นตาราง: ชื่อไฟล์ · ชื่อภาษาไทยจาก videos/clips.yaml · ความละเอียด · ความยาว · fps
5. สร้างไฟล์ CLAUDE.md ที่มี
   - โปรเจกต์นี้คืออะไร · ตอบเป็นภาษาไทย · รัน Python ผ่าน .venv ทุกครั้ง
   - โครงไฟล์: detector.py (ระบบตรวจจับ) · webapp.py (เว็บ Flask ที่ http://127.0.0.1:8000)
     · static/index.html (แผงควบคุม) · static/admin.html (หน้าหลังบ้าน) · static/assets/app.css (หน้าตาร่วมของทั้งสองหน้า)
     · static/assets/icons.js (ไอคอน SVG) · config.yaml (ค่าเริ่มต้น) · .env (รหัสผ่านและอีเมล)
   - หัวข้อ "แบบหน้าเว็บ (ใช้ทุกเฟส ห้ามเปลี่ยน)" แล้วคัดลอกข้อความระหว่างเส้นด้านล่างลงไปทั้งหมด

ชื่อเว็บ: Intrusion Detection Console · ข้อความบนหน้าเว็บเป็นภาษาไทยทั้งหมด
ธีม: เปิดมาเป็นธีมสว่างเสมอ (<html data-theme="light">) มีปุ่มสลับธีมมืด แต่ไม่จำค่าไว้
สีธีมสว่าง: พื้นหน้า #F2F3F5 · การ์ด #FFFFFF · พื้นรอง #F6F7F8 · เส้นขอบ #DCE0E5 (เส้นเข้ม #C4CAD1)
  · ตัวหนังสือ #14171B · ตัวหนังสือรอง #5B636D · สีหลักแดง #D80000 (ตอนชี้ #B80000) · พื้นแดงอ่อน rgba(216,0,0,.08)
สีธีมมืด: พื้นหน้า #0A0C0F · การ์ด #12151A · พื้นรอง #181C22 · เส้นขอบ #252A32 · ตัวหนังสือ #E6E9ED · ตัวรอง #8C95A1 · แดง #E00000
ฟอนต์: IBM Plex Sans Thai ขนาด 15px จาก Google Fonts · ตัวเลข ชื่อไฟล์ และค่าทางเทคนิคใช้ JetBrains Mono
สไตล์: เรียบแบบเครื่องมืออาชีพ มุมโค้ง 4–8px เส้นขอบ 1px เงาบางมาก ห้ามไล่สี (gradient) ห้ามอีโมจิ
  ไอคอนเป็น SVG แบบเส้น (สไตล์ Lucide) ขนาด 18px
แถบบนสุด: สูง 60px พื้นสีการ์ด มีเส้นขอบล่าง ติดอยู่ด้านบนเวลาเลื่อนหน้า
  ซ้าย = โลโก้ brand/zenityx-logo.png สูง 22px (ธีมมืดใช้ brand/zenityx-logo-dark.png) · เส้นคั่นแนวตั้ง
  · ชื่อหน้าตัวหนา 15px และบรรทัดรองสีเทา 12px · ขวา = ปุ่มของหน้านั้น และปุ่มไอคอนสลับธีม (พระจันทร์/ดวงอาทิตย์)
  ปุ่มไอคอนเป็นสี่เหลี่ยม 36px กรอบเทา · ป้ายในแถบบนสูง 28–32px · ป้ายสถานะ = พื้นรอง กรอบเทา จุดกลมสีเทา + ข้อความ · ไอคอนแท็บเบราว์เซอร์ใช้ brand/favicon.ico
การ์ด: พื้นสีการ์ด กรอบ 1px มุม 8px · หัวการ์ด = ป้ายเลขขั้น (สี่เหลี่ยมพื้นดำ 22px มุม 4px ตัวเลขขาว)
  + หัวข้อ 15px ตัวหนา + ข้อความเล็กสีเทาชิดขวา · ไม่มีเส้นคั่นใต้หัวการ์ด เนื้อหาต่อจากหัวทันที
ปุ่ม: สูง 36px มุม 6px พื้นสีการ์ด กรอบเทา มีไอคอนนำหน้า · ปุ่มหลัก = พื้นแดงตัวขาว · ปุ่มเล็กสูง 32px
  · ปุ่มใหญ่สูง 46px ตัวหนา · ปุ่มที่กดค้างไว้ = กรอบแดง ตัวแดง พื้นแดงอ่อน · ปุ่มที่กดไม่ได้ให้จางลง
ปุ่มแบบแท็บ: กล่องพื้นรองมีกรอบ ข้างในเป็นปุ่มกว้างเท่ากัน ปุ่มที่เลือกอยู่เป็นพื้นสีการ์ดมีเส้นขอบ
ป้ายเล็ก: สูง 20px มุม 4px ตัวหนังสือ 11.5px พื้นรอง ตัวเทา · ป้ายเน้น = กรอบแดง ตัวแดง พื้นแดงอ่อน
ป้ายที่ลอยบนภาพ: พื้นดำโปร่ง 66% ตัวขาว 12.5px ฟอนต์ IBM Plex Sans Thai มุม 5px สูง 28px
สวิตช์เปิดปิด: ราง 38x22px ปิด = เทา เปิด = แดง · ช่องกรอก: สูง 36px มุม 6px กรอบเทา คลิกแล้วกรอบแดง
ข้อความแจ้งสั้น: กล่องดำมุม 6px ลอยกลางล่างจอ หายเองใน 3 วินาที ถ้าเป็นข้อผิดพลาดใช้พื้นแดง
มือถือ: ทุกหน้าใช้ได้บนมือถือทั้งแนวตั้งและแนวนอน ต้องไม่มีการเลื่อนซ้ายขวา · แนวนอนภาพกล้องสูงไม่เกินความสูงจอ

เสร็จแล้วสรุปว่าพร้อมเริ่มเฟส 1 หรือยัง

คำศัพท์ในพรอมต์นี้

virtual environment (.venv) กล่องแยกเก็บโปรแกรมเสริมของ Python ไว้เฉพาะโปรเจกต์นี้ ไม่ปนกับโปรเจกต์อื่นในเครื่อง

ultralytics / opencv / flask ultralytics = ตัวรันโมเดล YOLO26 · opencv = อ่านวิดีโอและวาดกรอบบนภาพ · flask = ทำหน้าเว็บด้วย Python

CLAUDE.md ไฟล์โน้ตที่ Claude Code อ่านทุกครั้งก่อนทำงาน ใส่แบบหน้าเว็บไว้ที่นี่ ทุกเฟสจึงได้หน้าตาเดียวกัน

รหัสสี (#D80000) รหัสสีที่คอมพิวเตอร์ใช้ #D80000 คือสีแดง ZenityX · rgba(...,.08) คือสีแดงจางเหลือ 8%

px หน่วยวัดขนาดบนจอ 1px คือจุดเล็กที่สุดบนหน้าจอ

favicon ไอคอนเล็กบนแท็บเบราว์เซอร์

สิ่งที่ควรเห็น

- Claude Code แตก `yolo26-starter-kit.zip` แล้วสรุป **ตารางคลิป 7 คลิป** พร้อมชื่อภาษาไทยจาก `videos/clips.yaml` ความละเอียด ความยาว และ fps
- ในโฟลเดอร์มี `.venv/` `videos/` `brand/` `requirements.txt` `.gitignore` และ `CLAUDE.md`
- ใน `CLAUDE.md` มีหัวข้อ **"แบบหน้าเว็บ (ใช้ทุกเฟส ห้ามเปลี่ยน)"** — สี ฟอนต์ ขนาดปุ่ม การ์ด และแถบบนของหน้าเว็บ
- ข้อความสรุปว่า **พร้อมเริ่มเฟส 1**

เคล็ดลับ

- **ทำไม S0 ยาว** — ครึ่งหลังคือแบบหน้าเว็บที่ AI จดลง CLAUDE.md ทุกเฟส AI อ่านไฟล์นี้ก่อนทำงาน หน้าเว็บของทุกคนจึงออกมาหน้าตาเดียวกัน
- ถ้าเครื่องยังไม่มี Python 3.10–3.12 Claude Code จะบอกวิธีติดตั้งและรอให้เราตอบก่อน ไม่ติดตั้งโปรแกรมระดับเครื่องเอง
- ไม่เห็นโฟลเดอร์ `videos/` — เช็คว่าวางไฟล์ `yolo26-starter-kit.zip` ไว้ในโฟลเดอร์โปรเจกต์แล้ว จากนั้นพิมพ์ให้ AI แตกไฟล์อีกครั้ง
- ต่อจากนี้ให้รัน Python ผ่าน `.venv` เสมอ — ถ้าเจอ `ModuleNotFoundError` แปลว่ารันด้วย Python นอก .venv

---

PHASE 1 · 0:35–0:55

01 แผงควบคุม + ตรวจจับคน

เป้าหมาย: ได้หน้าเว็บ **Intrusion Detection Console** ที่ `http://127.0.0.1:8000` — การ์ดขั้น 1 **แหล่งภาพ** (คลิปวิดีโอ · เว็บแคม · กล้อง RTSP) ภาพสดพร้อมกรอบแดงรอบคน และตัวเลขสด 5 ช่อง โมเดลล็อกไว้ที่ YOLO26n

PROMPT · P1

Phase 1 · แผงควบคุม + ตรวจจับคน

สร้างแผงควบคุม "Intrusion Detection Console" ที่ตรวจจับ "คน" จากวิดีโอด้วย YOLO26 ตามแบบหน้าเว็บใน CLAUDE.md
ระบบตรวจจับ (detector.py)
- ใช้โมเดล yolo26n.pt (ล็อกไว้ที่รุ่น n ระบบดาวน์โหลดให้เองครั้งแรก) ตรวจเฉพาะคน (person) ความมั่นใจขั้นต่ำ 0.5
  ตอนกดเริ่ม ให้ลองตรวจ 1 ภาพก่อนเริ่มนับเวลา (วอร์มโมเดล) คลิปจะได้ไม่ข้ามช่วงแรก
- เลือกตัวประมวลผลอัตโนมัติ: การ์ดจอ NVIDIA (cuda) > ชิป Mac (mps) > cpu
- แหล่งภาพ 3 แบบ: ไฟล์ใน videos/ · เว็บแคม (เลขกล้อง 0, 1) · กล้องวงจรปิด rtsp://...
- ไฟล์วิดีโอให้เล่นตามความเร็วจริง 1 รอบจนจบ แล้วสถานะเป็น "เสร็จสิ้น"
- วาดบนภาพ (ตัวหนังสือบนภาพเป็นภาษาอังกฤษ): กรอบแดงหนา 2px มีป้ายพื้นแดงตัวขาว เช่น person 0.87
  และแถบดำสูง 32px ทับขอบบนของภาพ ตัวขาว เช่น 2026-11-12 22:15:08 ARMED hits:1 34 fps
- ให้การตรวจจับทำงานเบื้องหลังแยกจากหน้าเว็บ (thread) แล้วส่งภาพไปหน้าเว็บแบบ MJPEG
หน้าเว็บ (webapp.py ใช้ Flask เปิดที่ http://127.0.0.1:8000 · static/index.html · static/assets/app.css
· static/assets/icons.js)
- แถบบน: ชื่อ "Intrusion Detection Console" บรรทัดรอง "เวิร์กช็อป Python + OpenCV + YOLO26"
  ขวา: ป้าย "device: mps" (ตัว JetBrains Mono 12px กรอบเทา) · ป้ายสถานะ จุดกลมสีเทา + ข้อความ
  (พร้อม · กำลังโหลดโมเดล · กำลังตรวจจับ = กรอบแดง ตัวแดง จุดกะพริบ · เสร็จสิ้น · ผิดพลาด) · ปุ่มสลับธีม
- จอกว้าง: 2 คอลัมน์ ซ้ายเป็นแผงตั้งค่ากว้าง 400px (ติดอยู่ขณะเลื่อนหน้า) ขวาเป็นการ์ดภาพจากกล้อง ระยะห่าง 16px
  จอกว้างไม่ถึง 1080px เรียงเป็นคอลัมน์เดียว ให้ภาพกล้องขึ้นก่อน
- แผงซ้าย การ์ดขั้น 1 "แหล่งภาพ" (หัวการ์ดไม่มีข้อความด้านขวา): ปุ่มแบบแท็บ 3 ปุ่มมีไอคอน คลิปวิดีโอ · เว็บแคม · กล้อง
RTSP
  แท็บคลิปวิดีโอ: รายการคลิปใน videos/ แบ่งกลุ่มตาม videos/clips.yaml (หัวกลุ่มตัวเล็กสีเทา เช่น "กล้องวงจรปิด")
    แต่ละคลิปเป็นแถวกรอบเทา: ซ้าย = ภาพย่อจากเฟรมแรก 16:9 กว้าง 112px มุมขวาล่างมีป้ายดำ "12 วิ"
    ขวา = ชื่อไทยตัวหนา · ชื่อไฟล์ตัว JetBrains Mono สีเทา · ป้ายเล็กคลาสที่แนะนำเป็นภาษาไทย
    และป้ายเน้นสีแดง "มี zone" พร้อมไอคอนห้าเหลี่ยม ถ้าคลิปนั้นมี zone ใน clips.yaml
    คลิปที่เลือกอยู่ กรอบแดง พื้นแดงอ่อน · กดคลิปแล้วภาพเฟรมแรกขึ้นในการ์ดภาพทันที
    กล่องรายการคลิปสูงไม่เกิน 330px (เห็นราว 4 คลิป) เลื่อนดูในกล่อง
  แท็บเว็บแคม: ช่องหมายเลขกล้อง (ค่าเริ่ม 0) + คำแนะนำเรื่องอนุญาตกล้องบน macOS + ปุ่ม "ทดสอบกล้อง / จับภาพตัวอย่าง"
  แท็บกล้อง RTSP: ช่อง URL กล้อง + ปุ่มตัวอย่าง Hikvision · Dahua · Tapo · .env (ใส่ ${CAMERA_URL}) + ปุ่ม
"ทดสอบการเชื่อมต่อ"
- การ์ดภาพจากกล้อง (ขวา): การ์ดไม่มีหัวข้อ ขอบในการ์ด 12px · ภาพเต็มความกว้างตามสัดส่วนวิดีโอ พื้นดำ มุม 6px
  กรอบภาพใช้สัดส่วนจริงของวิดีโอ (เช่น 1280x704) ไม่บังคับ 16:9
  มุมซ้ายบน (ใต้แถบดำบนภาพ) = ป้าย "LIVE" พื้นแดงจุดขาวกะพริบ (ตอนกำลังตรวจจับ) + ป้ายชื่อคลิป
  มุมขวาบน = ปุ่มดูเต็มจอ (ไอคอนลูกศรทแยงออก 2 มุม) พื้นดำโปร่ง
  ยังไม่มีภาพ = ไอคอนกล้องวงจรปิดใหญ่กลางภาพ + "เลือกคลิปวิดีโอเพื่อเริ่ม" · ตอนโหลดโมเดล = ไอคอนหมุน
- ใต้ภาพเป็นแถวเดียวกันทั้งหมด: ซ้ายสุดปุ่ม "เริ่มตรวจจับ" (ปุ่มหลักขนาดใหญ่ ไอคอนเล่น กว้างตามข้อความ ไม่ยืดเต็มแถว)
  · ปุ่ม "หยุด" (ไอคอนสี่เหลี่ยม) · แล้วช่องตัวเลข 5 ช่องกว้างเท่ากันต่อท้ายในแถวเดียวกัน (จอแคบค่อยขึ้นบรรทัดใหม่)
  ช่องตัวเลข = กล่องพื้นรอง ป้ายเล็กสีเทาด้านบน ค่าตัว JetBrains Mono หนา 16px ด้านล่าง:
  เวลา (นาฬิกาตอนนี้) · สถานะระบบ · วัตถุในพื้นที่ (ตัวแดงเมื่อมากกว่า 0)
  · ความเร็วตรวจจับ (จำนวนภาพที่โมเดลตรวจได้ต่อวินาที เช่น 34.3 fps) · ตำแหน่งคลิป เช่น 12.2 / 12.2 วิ
- แถวถัดไป: "วัตถุในเฟรมล่าสุด:" แล้วป้ายกรอบแดงทีละวัตถุ เช่น "คน 0.89" อัปเดตทุกครึ่งวินาที
- ปุ่มเริ่มกดไม่ได้จนกว่าจะเลือกแหล่งภาพ · ระหว่างตรวจจับให้ล็อกแผงตั้งค่าไว้
- เปิดหน้าใหม่ระหว่างตรวจจับ ให้เห็นภาพสดต่อ · ถ้าไม่ได้ตรวจจับอยู่ ป้ายสถานะเป็น "พร้อม" และช่องตัวเลขแสดงค่าของรอบล่าสุดค้างไว้
ทดสอบเอง: เปิดเซิร์ฟเวอร์ เลือก 11_cctv_office_day.mp4 กดเริ่ม เช็คว่าภาพ กรอบ และตัวเลขขึ้นจริงจนคลิปจบ
แล้วบอกคำสั่งเปิดเว็บให้ฉัน และอธิบายโค้ดแต่ละไฟล์สั้น ๆ ให้คนเพิ่งเริ่มเข้าใจ

คำศัพท์ในพรอมต์นี้

yolo26n.pt ไฟล์โมเดล YOLO26 รุ่นเล็กสุด (n = nano) เร็วที่สุด เหมาะกับโน้ตบุ๊ก

cuda / mps / cpu ชิปที่ใช้คำนวณ cuda = การ์ดจอ NVIDIA · mps = ชิป Mac M1–M4 · cpu = ตัวประมวลผลทั่วไป (ช้าสุด)

thread งานที่วิ่งเบื้องหลังพร้อมกันได้ ทำให้ตรวจจับไปพร้อมกับหน้าเว็บตอบสนองได้ไม่ค้าง

Flask / 127.0.0.1:8000 Flask = ตัวทำเว็บด้วย Python · 127.0.0.1 = เครื่องตัวเอง · 8000 = ช่องทาง (พอร์ต) ที่เว็บเปิดอยู่

clips.yaml ไฟล์ข้อมูลคลิปที่มากับ Starter Kit มีชื่อไทย กลุ่ม คลาสที่แนะนำ และพื้นที่ที่แนะนำของแต่ละคลิป

ความมั่นใจ (confidence) คะแนน 0–1 ที่โมเดลบอกว่ามั่นใจแค่ไหน 0.5 = มั่นใจอย่างน้อย 50% จึงนับ

RTSP รูปแบบที่กล้องวงจรปิดใช้ส่งภาพสดผ่านเครือข่าย URL ขึ้นต้นด้วย rtsp://

MJPEG วิธีส่งภาพสดไปหน้าเว็บโดยส่งภาพ JPG ต่อกันเร็ว ๆ จนดูเหมือนวิดีโอ

fps จำนวนภาพที่ตรวจได้ต่อวินาที ยิ่งมากยิ่งลื่น ควรได้ 15 ขึ้นไป

แผงควบคุมจะค่อย ๆ ครบทีละเฟส

คอลัมน์ซ้ายเป็นการ์ดตั้งค่า 4 ขั้น เรียงจากบนลงล่าง คอลัมน์ขวาเป็นภาพสด ตัวเลข และรายการเหตุการณ์

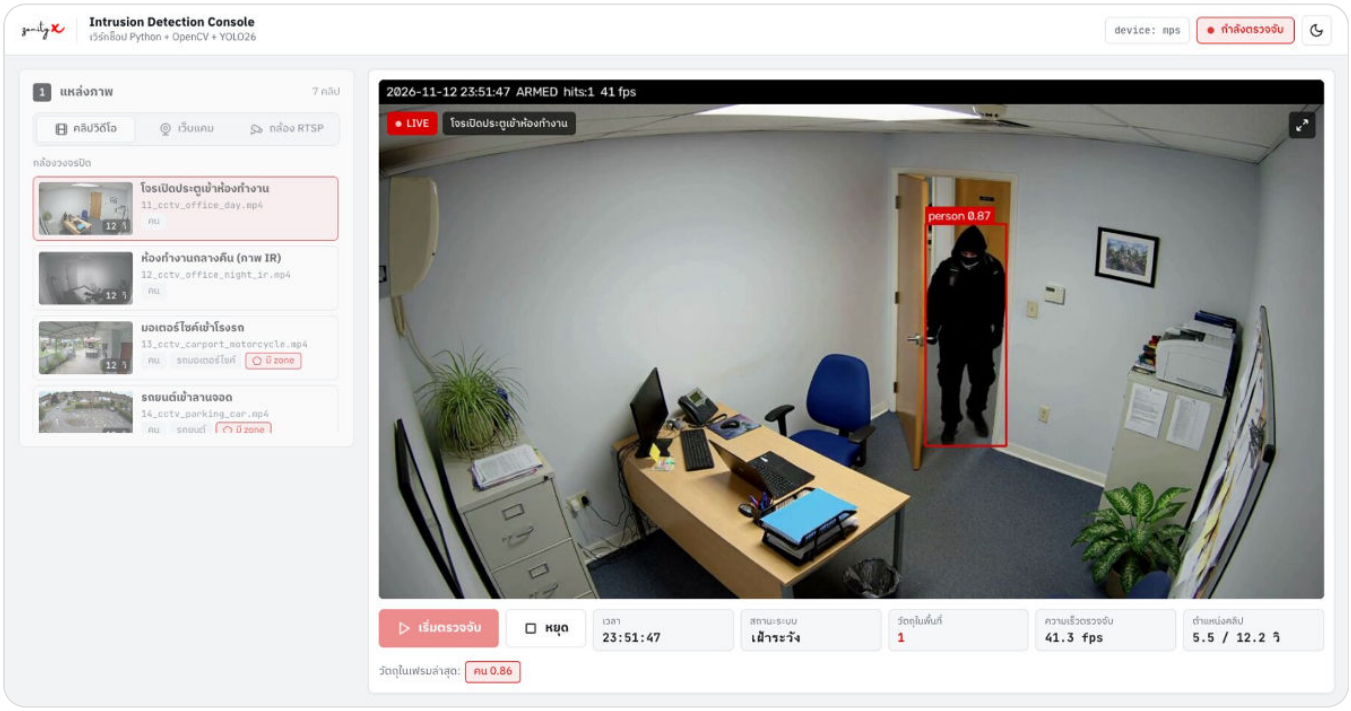
ขั้น 1 · P1
แหล่งภาพ
คลิปวิดีโอ · เว็บแคม · กล้อง RTSP

ขั้น 2 · P2
คลาสที่ตรวจจับ
เลือกจาก 80 คลาส ชื่อภาษาไทย

ขั้น 3 · P3
พื้นที่ตรวจจับ
คลิกวาดพื้นที่บนภาพ

ขั้น 4 · P3 / P4A / P4C
ตั้งค่าการแจ้งเตือน
ความมั่นใจ · ช่วงเวลา · กันเตือนหลอก · อีเมล

✓ สิ่งที่ควรเห็น



แผงควบคุมจริงจาก Phase 1 — เลือกคลิป **11_cctv_office_day** กดเริ่มตรวจจับ ภาพสดมีป้าย **LIVE** กรอบแดงรอบคน และตัวเลข 5 ช่อง: เวลา · สถานะระบบ · วัตถุในพื้นที่ · ความเร็วตรวจจับ · ตำแหน่งคลิป

เคล็ดลับ

- เปิดเว็บด้วยคำสั่ง `.venv/bin/python webapp.py` (Windows `.venv\Scripts\python webapp.py`) แล้วเปิด `http://127.0.0.1:8000` ในเบราว์เซอร์
- ครั้งแรกระบบดาวน์โหลด `yolo26n.pt` ให้เอง ถ้าเน็ตช้าหรือไฟร์วอลล์บล็อก ผู้สอนแจกไฟล์ .pt ให้วางในโฟลเดอร์โปรเจกต์
- ไฟล์วิดีโอเล่นตามความเร็วจริง ช่อง **ความเร็วตรวจจับ** จึงไม่เกิน fps ของคลิป (ราว 24–30) เป็นเรื่องปกติ · ต่ำกว่า 15 fps ให้สั่ง AI ลดขนาดภาพที่ส่งให้ YOLO เป็น 480
- หน้าเว็บไม่เหมือนภาพนี้ ให้พิมพ์ว่า **"ทำหน้าเว็บให้ตรงตามแบบหน้าเว็บใน CLAUDE.md"**

PHASE 2 · 0:55–1:05

02 เลือกคลาสที่ต้องการตรวจจับ

เป้าหมาย: การ์ดขั้น 2 **คลาสที่ตรวจจับ** — เลือกจาก 80 คลาสของ YOLO26 ด้วยชื่อภาษาไทย ค้นหาได้ทั้งไทยและอังกฤษ มีปุ่มเลือกเร็ว และเลือกคลาสที่แนะนำของคลิปให้อัตโนมัติ

PROMPT · P2

Phase 2 · เลือกคลาสที่ต้องการตรวจจับ

```
ตอนนี้ตรวจแค่คน ช่วยเพิ่มการ์ดขั้น 2 "คลาสที่ตรวจจับ" ใต้การ์ดแหล่งภาพ ให้เลือกได้ทั้ง 80 คลาสของ YOLO26 ตามแบบหน้าเว็บใน CLAUDE.md
- ชื่อคลาสภาษาไทยครบ 80 คลาส (กลุ่มแรกใช้ชื่อ: คน · จักรยาน · รถยนต์ · รถมอเตอร์ไซค์ · เครื่องบิน · รถบัส · รถไฟ · รถบรรทุก · เรือ)
  แบ่ง 9 กลุ่มตามลำดับเลขคลาส: คนและยานพาหนะ (0-8) · ถนนและป้าย (9-13) · สัตว์ (14-23)
  · ของติดตัว (24-28) · กีฬา (29-38) · ครัวและอาหาร (39-55) · เฟอร์นิเจอร์ (56-61) · เครื่องใช้ไฟฟ้า (62-72) · ของใช้อื่น ๆ (73-79)
- หัวการ์ดด้านขวาแสดงจำนวนที่เลือก เช่น "3 คลาส"
- บนสุด: คลาสที่เลือกแสดงเป็นป้ายพื้นแดงตัวขาว มีปุ่ม x เอาออก · ยังไม่เลือก = "ยังไม่ได้เลือกคลาส · เลือกอย่างน้อย 1 คลาส"
- ปุ่มเล็กเลือกเร็ว: "ผู้บุกรุก" (ไอคอนโล่ = คน) · "คน + ยานพาหนะ" (คน รถยนต์ รถมอเตอร์ไซค์ จักรยาน รถบรรทุก รถบัส ไม่มีไอคอน)
  · "แนะนำของคลิป" (ตาม clips.yaml ไม่มีไอคอน) · "ล้าง" (ไอคอนลูกศรวน)
- ช่องค้นหามีไอคอนแว่นขยาย ข้อความในช่อง "ค้นหาคลาส เช่น คน, car, กระเป๋า" ค้นได้ทั้งไทยและอังกฤษ
- รายการกลุ่มแบบพับเก็บได้: มีเส้นคั่นด้านบนแต่ละกลุ่ม ลูกศรอยู่ซ้ายชื่อกลุ่มและหมุนลงเมื่อเปิด กลุ่มแรกเปิดไว้
  ด้านขวาหัวกลุ่มเป็นตัวเลขจำนวนคลาสอย่างเดียว (เช่น 5) หรือ "เลือก 3" ตัวแดงเมื่อมีที่เลือก
  ในกลุ่มเป็นปุ่มคลาสเรียงต่อกัน: ชื่อไทย + ชื่ออังกฤษตัว JetBrains Mono เล็กสีเทา เช่น "คน person" · ที่เลือกแล้ว กรอบแดง ตัวแดง พื้นแดงอ่อน
- กดคลิปในการ์ดแหล่งภาพแล้ว เลือกคลาสที่แนะนำของคลิปนั้นให้อัตโนมัติ พร้อมข้อความแจ้งสั้นบอก
- ปุ่มเริ่มตรวจจับกดไม่ได้ถ้ายังไม่เลือกคลาส · detector.py ตรวจเฉพาะคลาสที่เลือก
- แถว "วัตถุในเฟรมล่าสุด" ใช้ชื่อภาษาไทย เช่น "รถยนต์ 0.90"
ทดสอบกับ videos/14_cctv_parking_car.mp4 (คน + รถยนต์) ให้ดูว่ามีกรอบรถยนต์ขึ้นจริง
```

คำศัพท์ในพรอมต์นี้

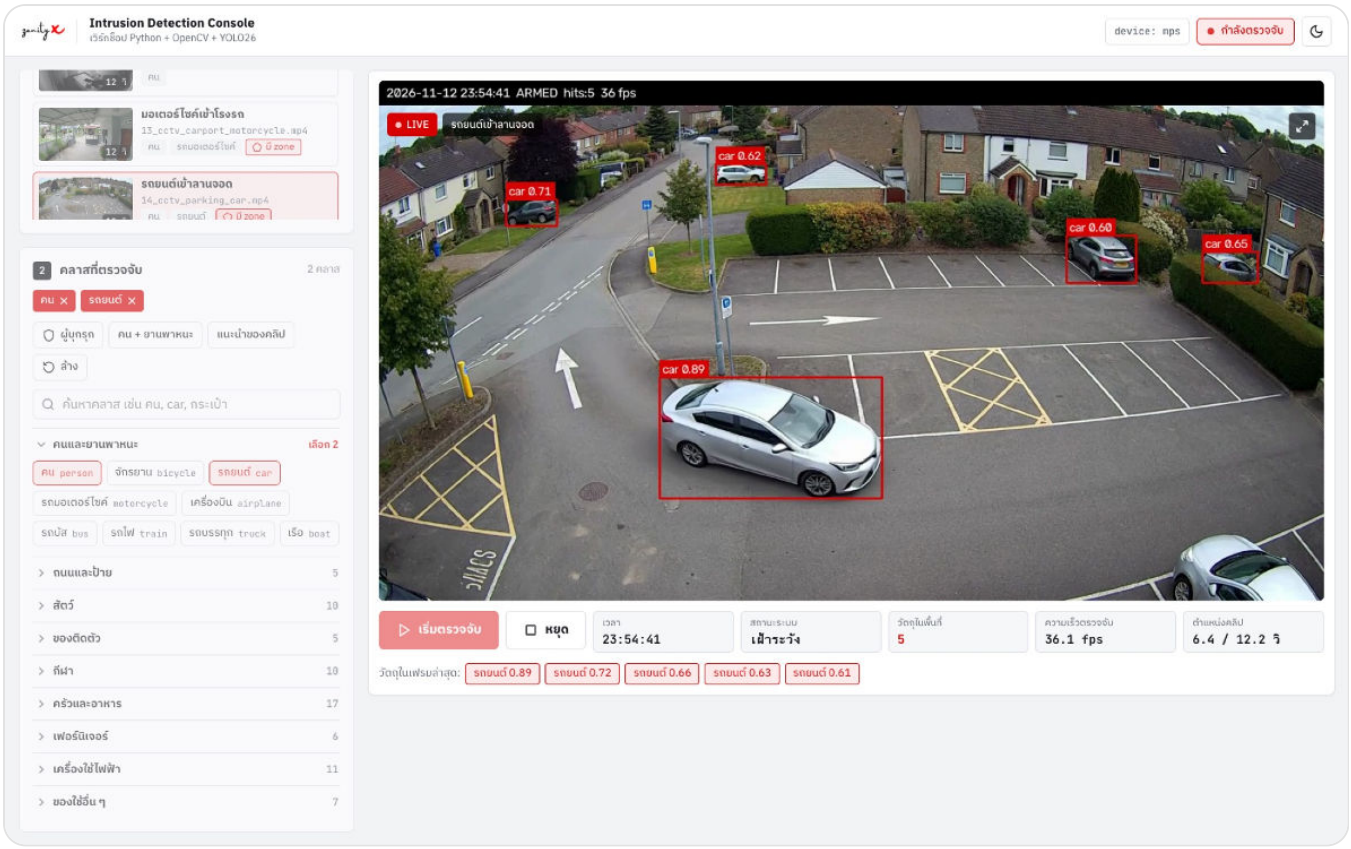
คลาส (class) ชนิดของสิ่งที่โมเดลรู้จัก เช่น คน รถยนต์ สุนัข YOLO26 รุ่นนี้รู้จัก 80 คลาส

เลขคลาส แต่ละคลาสมีเลขประจำ 0–79 เช่น 0 = person (คน) 2 = car (รถยนต์) ใช้จัดกลุ่ม

80 คลาส (COCO) COCO คือชุดภาพตัวอย่างที่ใช้สอนโมเดล มี 80 ชนิดของในชีวิตประจำวัน

พับเก็บได้ กดหัวกลุ่มแล้วเปิดดูรายการข้างใน กดอีกครั้งพับเก็บ ช่วยให้รายการ 80 คลาสไม่ยาวจนรก

✓ สิ่งที่ควรเห็น



ผลจริงจากคลิป **14_cctv_parking_car** — การ์ดขั้น 2 เลือก **คน + รถยนต์** กรอบรถยนต์ขึ้นทันที และแถว "วัตถุในเฟรมล่าสุด" แสดงชื่อไทย เช่น **รถยนต์ 0.90**

เคล็ดลับ

- กดคลิปในการ์ดแหล่งภาพแล้ว ระบบเลือกคลาสที่แนะนำของคลิปนั้นให้เอง เพิ่มหรือเอาออกเองได้
- ปุ่ม "เริ่มตรวจจับ" กดไม่ได้จนกว่าจะเลือกอย่างน้อย 1 คลาส
- รถที่จอดอยู่แล้วก็ถูกนับ เป็นเรื่องปกติของเฟสนี้ — Phase 3 แก้ด้วย **พื้นที่ตรวจจับ**

PHASE 3 · 1:05–1:25

03 พื้นที่ตรวจจับ + ช่วงเวลาเฝ้าระวัง

เป้าหมาย: การ์ดขั้น 3 **พื้นที่ตรวจจับ** คลิกวาดพื้นที่สีเหลืองอำพันบนภาพ นับเฉพาะวัตถุที่ **จุดเท้า** อยู่ในพื้นที่ และการ์ดขั้น 4 **ตั้งค่าการแจ้งเตือน** — ความมั่นใจขั้นต่ำ กับช่วงเวลา **22:00–06:00** สถานะขึ้น "เฝ้าระวัง" หรือ "นอกเวลา"

PROMPT · P3

Phase 3 · พื้นที่ตรวจจับ + ช่วงเวลาเฝ้าระวัง

เพิ่ม "พื้นที่" และ "ช่วงเวลา" ที่ระบบเฝ้าระวัง ตามแบบหน้าเว็บใน CLAUDE.md
การ์ดขั้น 3 "พื้นที่ตรวจจับ" (ใต้การ์ดคลาส) หัวการ์ดด้านขวาแสดง "ทั้งภาพ" / "4 จุด" / "วาดอีก 2 จุด"
- คำอธิบายสีเทา: ตรวจจับเฉพาะวัตถุที่ "จุดเท้า" (กึ่งกลางขอบล่างของกรอบ) อยู่ในพื้นที่ นอกพื้นที่ระบบจะไม่สนใจเลย
- ปุ่มเล็ก: "วาดพื้นที่" (ไอคอนห้าเหลี่ยม ระหว่างวาดเปลี่ยนเป็น "วาดเสร็จ" แบบกดค้าง) · "ย้อนจุด" · "ใช้ทั้งภาพ" · "ค่าแนะนำของคลิป" (แสดงเฉพาะคลิปที่มี zone ใน clips.yaml)
- วาดบนภาพเฟรมแรกในการ์ดภาพ (ภาพที่ยังไม่มีกรอบ): คลิกทีละมุม เคอร์เซอร์เป็นกากบาท กลางล่างของภาพมีป้าย "คลิกบนภาพเพื่อเพิ่มมุมพื้นที่ · กด Enter เมื่อเสร็จ" · ต้องมีอย่างน้อย 3 จุด พื้นที่เป็นเส้นสีเหลืองอำพัน #FFB000 พื้นข้างในโปร่ง 16% จุดมุมเป็นวงกลมเหลืองขอบดำ (วาดด้วย SVG ทับบนภาพ)
- เก็บพิกัดเป็นสัดส่วน 0–1 ของภาพ · กดคลิปที่มี zone ใน clips.yaml แล้วใส่พื้นที่แนะนำให้อัตโนมัติ
- ตอนตรวจจับ วาดพื้นที่บนภาพสดด้วยสีเดียวกัน และตรวจจับเฉพาะในพื้นที่เท่านั้น: วัตถุในพื้นที่ = กรอบแดง · วัตถุที่จุดเท้าอยู่นอกพื้นที่ = ตัดทิ้ง ไม่วาดกรอบ ไม่นับ ไม่แสดงในแถว "วัตถุในเฟรมล่าสุด" (เหมือนไม่มีอยู่) ยังไม่ได้วาดพื้นที่ = ตรวจทั้งภาพ

การ์ดขั้น 4 "ตั้งค่าการแจ้งเตือน" (ใต้การ์ดพื้นที่)
- "โมเดล" แสดงเป็นป้ายคงที่ "YOLO26n · เร็ว" (ล็อกไว้ ไม่มีให้เลือกรุ่นอื่น)
- แถบเลื่อน "ความมั่นใจขั้นต่ำ" 0.10–0.90 ทีละ 0.05 ค่าเริ่ม 0.50 แสดงตัวเลขด้านขวา ใต้แถบเขียนว่า "เจอแจ้งเตือนหลอกบ่อยให้เพิ่มค่า ตรวจไม่เจอให้ลดค่า"
- สวิตช์ "เปิดระบบเฉพาะช่วงเวลา" + ช่องเวลา "เริ่ม" 22:00 และ "สิ้นสุด" 06:00 วางคู่กัน ปิดสวิตช์ = ช่องเวลากดไม่ได้และเฝ้าระวังตลอด 24 ชม. · ต้องรองรับช่วงที่ข้ามเที่ยงคืน
- ช่อง "สถานะระบบ" แสดง "เฝ้าระวัง" หรือ "นอกเวลา" · แถบบนภาพเขียน ARMED หรือ DISARMED · นอกเวลายังแสดงกรอบในพื้นที่แต่ไม่นับ

สร้าง config.yaml เก็บค่าเริ่มต้นทั้งหมด (มีคำอธิบายภาษาไทยทุกบรรทัด) ค่าที่ตั้งบนหน้าเว็บส่งไปใช้ตอนกดเริ่ม
ทดสอบ: videos/13_cctv_carport_motorcycle.mp4 ใช้พื้นที่แนะนำ กดปุ่ม "คน + ยานพาหนะ" ตั้งความมั่นใจ 0.40 แล้วเช็คว่ามอเตอร์ไซค์และคนในโรงรถเป็นกรอบแดง ส่วนรถยนต์ที่จอดฝั่งตรงข้ามถนน (นอกพื้นที่) ต้องไม่มีกรอบเลย (ตอนมอเตอร์ไซค์เพิ่งเข้าภาพ โมเดลรุ่น n อาจเรียกเป็น "จักรยานยนต์" แวบหนึ่ง เป็นเรื่องปกติของความมั่นใจ 0.40 ไม่ต้องแก้) และทดสอบช่วงเวลา 2 แบบ: ตั้งให้ครอบเวลาตอนนี้ (ต้องเป็นเฝ้าระวัง) และไม่ครอบ (ต้องเป็นนอกเวลา)

คำศัพท์ในพรอมต์นี้

พื้นที่ตรวจจับ (zone) รูปหลายเหลี่ยมที่วาดบนภาพ ระบบตรวจจับเฉพาะสิ่งที่อยู่ข้างใน นอกกรอบไม่สนใจเลย เช่น พื้นโรงรถ

จุดเท้า จุดกึ่งกลางขอบล่างของกรอบ คือตำแหน่งที่คนหรือรถแตะพื้นจริง ใช้ตัดสินว่าอยู่ในพื้นที่ไหม

สัดส่วน 0–1 เก็บตำแหน่งเป็นสัดส่วนของภาพ เช่น 0.5 = กึ่งกลาง ใช้ได้กับวิดีโอทุกความละเอียด

SVG ภาพแบบเส้นที่วาดทับบนหน้าเว็บได้คมทุกขนาด ใช้วาดพื้นที่ทับภาพ

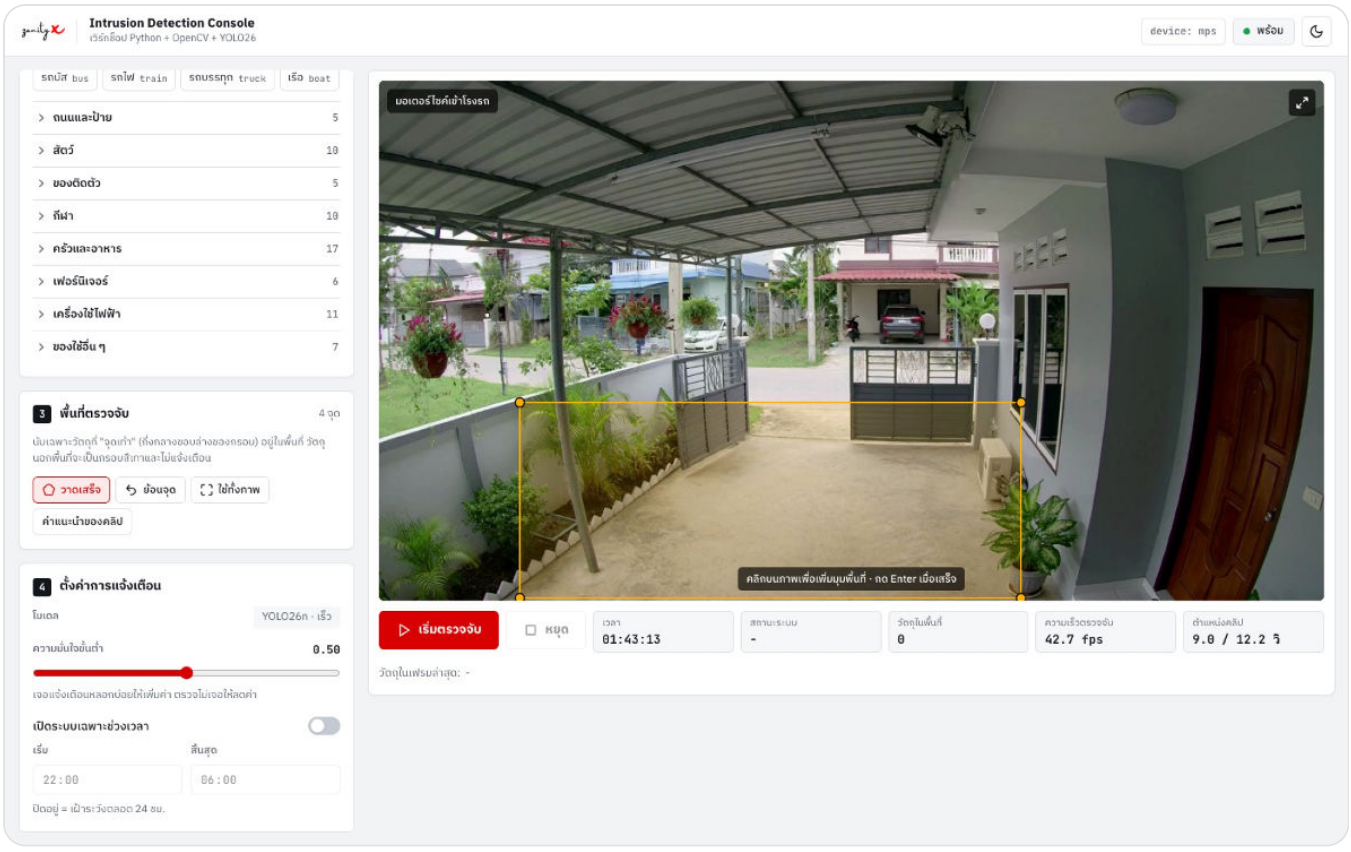
config.yaml ไฟล์ข้อความเก็บค่าตั้งต้นของระบบ เปิดแก้ด้วยโปรแกรมแก้ข้อความได้

ARMED / DISARMED ARMED = อยู่ในช่วงเวลาเฝ้าระวัง พร้อมแจ้งเตือน · DISARMED = นอกเวลา ไม่แจ้งเตือน

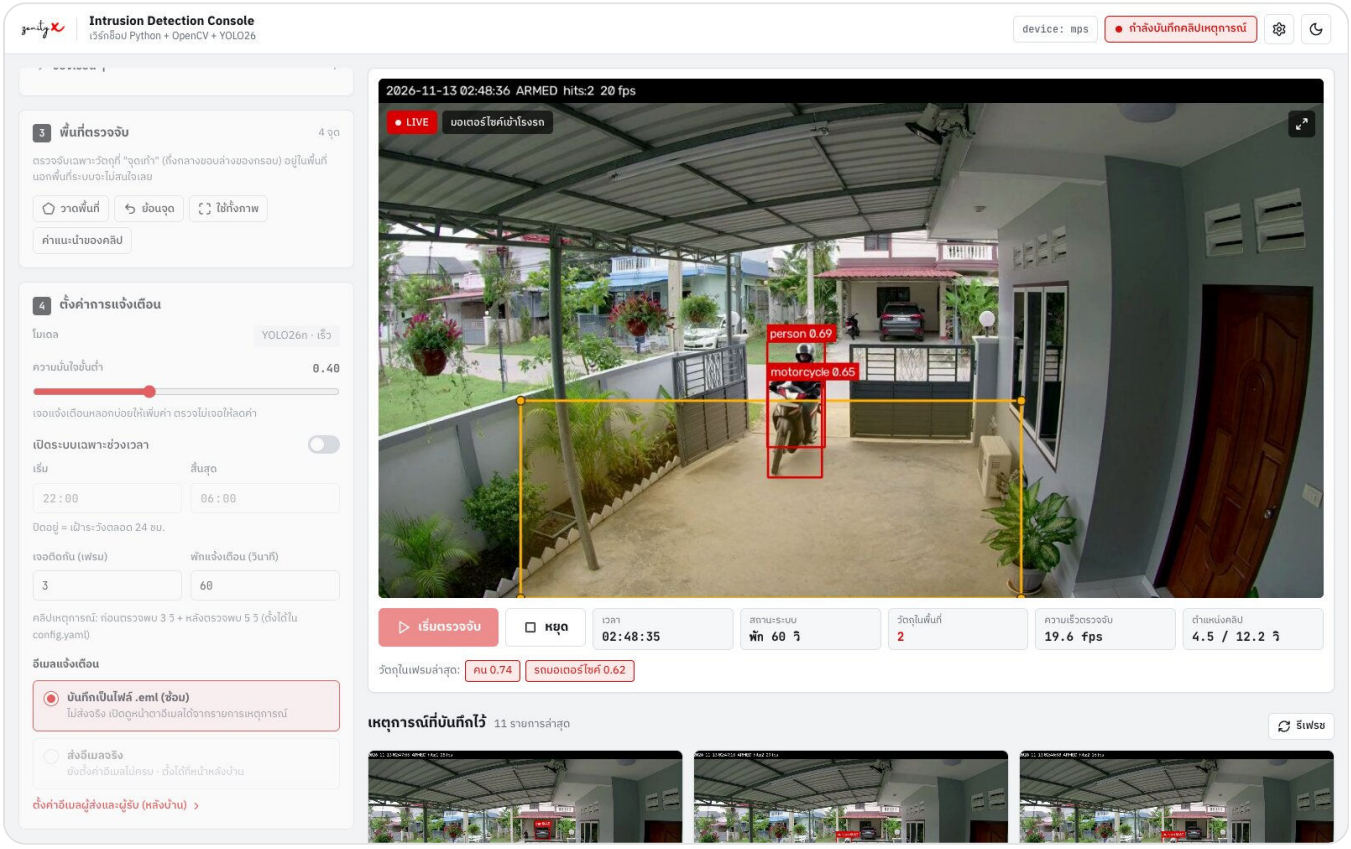
เคล็ดลับ

- **เรียนตอนกลางวัน** สถานะจะขึ้น "นอกเวลา" — ตั้งช่วงเวลาให้ครอบเวลาตอนนี้ หรือปิดสวิตช์ "เปิดระบบเฉพาะช่วงเวลา" (= เฝ้าระวังตลอด 24 ชม.)
- ระบบนับ "จุดเท้า" (กึ่งกลางขอบล่างของกรอบ) จึงวาดพื้นที่ **บนพื้น** · คลิปที่มีป้าย "มี zone" กด "ค่าแนะนำของคลิป" ได้เลย
- นอกเวลายังเห็นภาพและกรอบ แต่ไม่นับและไม่แจ้งเตือน
- เตือนหลอกบ่อย ให้วาดพื้นที่ให้แคบลง หรือเพิ่มความมั่นใจขั้นต่ำเป็น 0.6

✓ สิ่งที่ควรเห็น



วาดพื้นที่ — กด "วาดพื้นที่" แล้วคลิกทีละมุมบนภาพเฟรมแรกของคลิป **13_cctv_carport_motorcycle** พื้นที่สีเหลืองอำพันคลุมพื้นโรงรถ กด Enter เมื่อเสร็จ



ระหว่างตรวจจับ — ระบบ**ตรวจเฉพาะในกรอบที่วาด** มอเตอร์ไซค์และคนในโรงรถเป็นกรอบแดง ส่วนรถที่จอดฝั่งตรงข้ามถนน (นอกพื้นที่) **ไม่มีกรอบเลย** ไม่นับ ไม่เตือน

PHASE 4 · 1:25–1:50

04 เก็บหลักฐาน + ส่งอีเมลแจ้งเตือน

เป้าหมาย: เมื่อพบวัตถุในพื้นที่ ระบบเก็บ **snapshot + คลิป 8 วินาที** (ก่อนเหตุ 3 วิ + หลังเหตุ 5 วิ) ให้ดูย้อนหลังบนแผงควบคุม ตั้งค่าอีเมลที่ **หน้าหลังบ้าน /admin** แล้วส่ง **อีเมลภาษาไทยพร้อมภาพและไฟล์แนบ** — ทำตามลำดับ: P4A → P4B (+ ตั้ง ADMIN_PASSWORD เอง) → สร้าง App Password → P4C

4.1 เก็บหลักฐาน ภาพ + คลิป 8 วินาที

PROMPT · P4A

Phase 4 (1/3) · เก็บหลักฐาน ภาพ + คลิป 8 วินาที

เมื่อพบวัตถุในพื้นที่ ให้ระบบบันทึก "เหตุการณ์" พร้อมภาพและคลิป และดูย้อนหลังได้บนหน้าเว็บ ตามแบบหน้าเว็บใน CLAUDE.md
กันแจ้งเตือนหลอก (เพิ่มในการ์ดขั้น 4 เป็นช่องตัวเลขวางคู่กัน): "เจอติดกัน (เฟรม)" ค่าเริ่ม 3 · "พักแจ้งเตือน (วินาที)" ค่าเริ่ม 60
- นับเป็นเหตุการณ์เมื่อเจอในพื้นที่ติดกันครบจำนวนเฟรม และอยู่ในช่วงเวลา แล้วพักตามวินาทีที่ตั้ง
  ระหว่างพัก ช่อง "สถานะระบบ" แสดง เช่น "พัก 53 วิ"
เก็บหลักฐาน
- เก็บภาพย้อนหลังไว้ในหน่วยความจำตลอดเวลา (buffer) แล้วบันทึก snapshot.jpg และ clip.mp4 (มีกรอบ)
  คลิปยาว 8 วินาที = ก่อนเกิดเหตุ 3 วินาที + หลังเกิดเหตุ 5 วินาที (ตั้งค่าได้ใน config.yaml)
- บนคลิป มุมซ้ายล่างประทับชื่อกล้องและวันเวลา (camera_name: CAM-01 ใน config.yaml)
  ใต้แถบดำด้านบนมีจุดแดง + คำว่า "ALERT" ตั้งแต่วินาทีที่ตรวจพบ
- แปลงคลิปเป็น H.264 ด้วย ffmpeg จากแพ็กเกจ imageio-ffmpeg ให้เปิดดูในมือถือได้
  ไฟล์วิดีโอให้ใช้เวลาของวิดีโอ คลิปจะยาวถูกต้อง · ถ้าภาพย้อนหลังไม่ครบ 3 วินาที ให้ใช้เท่าที่มี
- เก็บที่ events/[วันที่_เวลา]/ พร้อม event.json (เวลา · ชื่อกล้อง · สถานที่ camera_location: ห้องทำงาน · สิ่งที่เจอ จำนวน
  และความมั่นใจสูงสุด · ความยาวคลิป)
- บันทึกเบื้องหลังแยกจากการตรวจจับ (thread) ภาพสดต้องไม่สะดุด ระหว่างบันทึกป้ายสถานะแถบบนเป็น "กำลังบันทึกคลิปเหตุการณ์"
หน้าเว็บ ส่วน "เหตุการณ์ที่บันทึกไว้" ใต้การ์ดภาพ (คอลัมน์ขวา)
- หัวข้อ 17px ตัวหนา + จำนวน เช่น "32 รายการล่าสุด" สีเทา (แสดงทุกเหตุการณ์ ใหม่สุดก่อน) + ปุ่มเล็ก "รีเฟรช" (ไอคอนลูกศรวน) ชิดขวา
- การ์ดเหตุการณ์เรียงเป็นตาราง กว้างอย่างน้อย 280px ต่อใบ ห่างกัน 12px (หน้ากว้าง 1600px ต้องได้ 3 ใบต่อแถว)
  บนสุดเป็นภาพ snapshot 16:9 เต็มความกว้างการ์ด (กดแล้วดูเต็มจอ) · ใต้ภาพ: เวลา (ตัว JetBrains Mono สีเทา)
  · ชื่อสถานที่ตัวหนา · ป้ายเน้นสีแดง เช่น "คน 1 · 77%" · แถวล่างสุดมีเส้นคั่น และปุ่มไอคอนเล่นคลิปชิดขวา
  เหตุการณ์ของรอบนี้ การ์ดกรอบแดง และมีป้ายเน้น (กรอบแดง ไม่ใช่พื้นแดงทึบ) "รอบนี้" มุมซ้ายบนของภาพ
  ยังไม่มีเหตุการณ์ = กล่องเส้นประ ไอคอนถาดจดหมาย "ยังไม่มีเหตุการณ์"
- หน้าต่างดูเต็มจอ: พื้นดำทึบเกือบสุด แถบหัวแสดงชื่อสถานที่ + "เวลา · สิ่งที่เจอ · 3/12" · ปุ่มแบบแท็บ "ภาพ" / "คลิป" · ปุ่มปิด x
  ภาพหรือคลิปใหญ่กลางจอ · ปุ่มลูกศรวงกลมซ้ายขวาเปลี่ยนเหตุการณ์ · ใช้ปุ่มลูกศรบนคีย์บอร์ดหรือปัดบนมือถือได้ · Esc หรือกดพื้นหลังเพื่อปิด
- เหตุการณ์ใหม่ขึ้นเองระหว่างตรวจจับ · คลิปจบแล้วแจ้ง "ตรวจจับเสร็จ · บันทึก 1 เหตุการณ์"
- ท้ายการ์ดขั้น 4 เขียนสีเทา "คลิปเหตุการณ์: ก่อนตรวจพบ 3 วิ + หลังตรวจพบ 5 วิ (ตั้งได้ใน config.yaml)"
ทดสอบกับ videos/11_cctv_office_day.mp4 แล้วเช็คว่าคลิปยาวประมาณ 8 วินาทีจริง
(ใช้คำสั่ง ffmpeg -i จาก imageio-ffmpeg เพราะแพ็กเกจนี้ไม่มี ffprobe)

คำศัพท์ในพรอมต์นี้

เฟรม ภาพนิ่ง 1 ภาพในวิดีโอ วิดีโอ 1 วินาทีมีราว 24–30 เฟรม

เจอติดกัน 3 เฟรม ต้องเห็นต่อเนื่องหลายภาพจึงเชื่อ กันเงาหรือแสงไฟรถที่โผล่แวบเดียวทำให้แจ้งเตือนหลอก

พักแจ้งเตือน (cooldown) หลังแจ้งเตือนแล้วรอช่วงหนึ่งก่อนแจ้งใหม่ กันอีเมลรัวเป็นสิบฉบับจากเหตุการณ์เดียว

buffer ที่พักภาพย้อนหลังในหน่วยความจำ ทำให้คลิปมีภาพ "ก่อน" เกิดเหตุได้

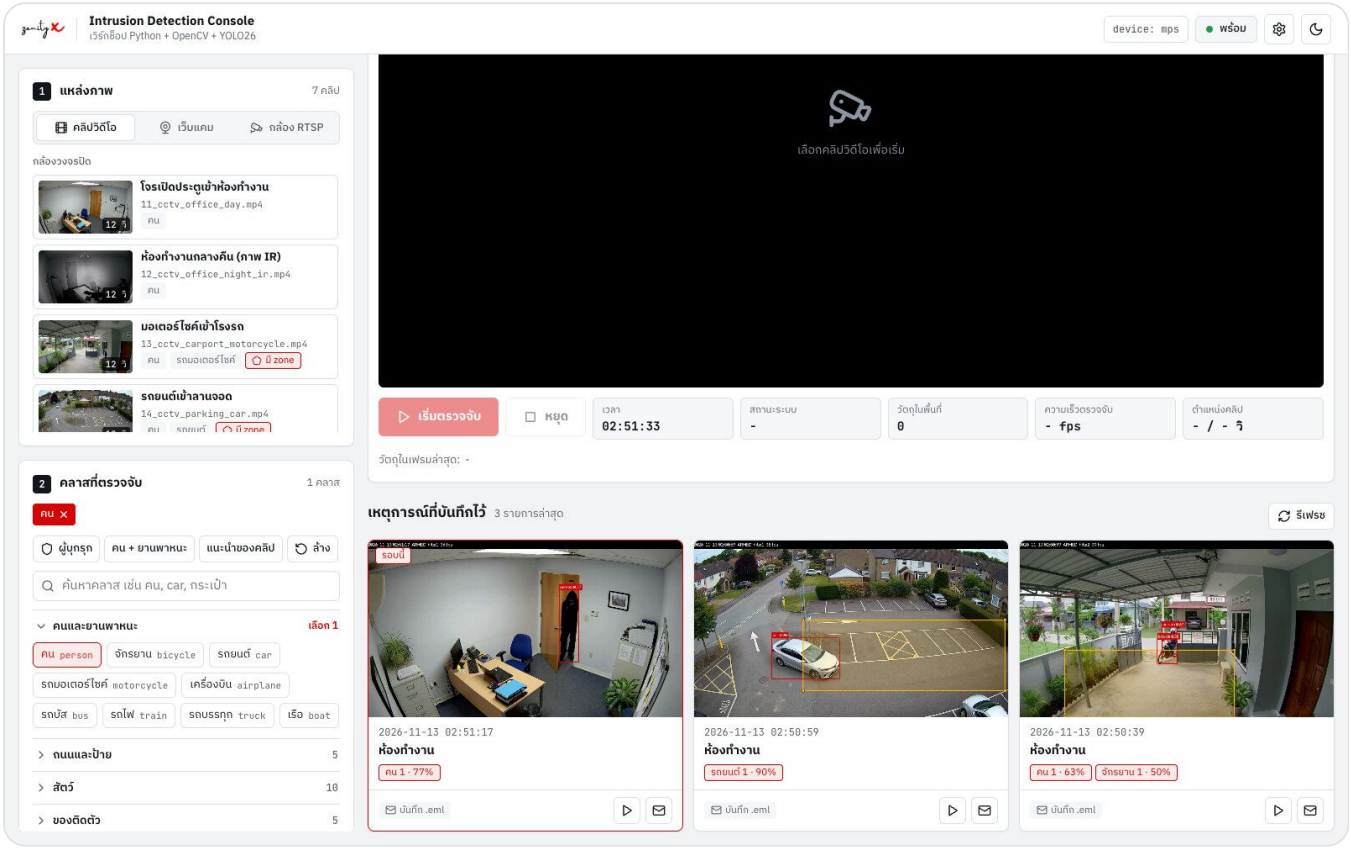
snapshot ภาพนิ่งตอนที่ตรวจพบ พร้อมกรอบ

H.264 / ffmpeg H.264 = รูปแบบวิดีโอที่มือถือและอีเมลเปิดได้ทุกเครื่อง · ffmpeg = โปรแกรมแปลงวิดีโอ

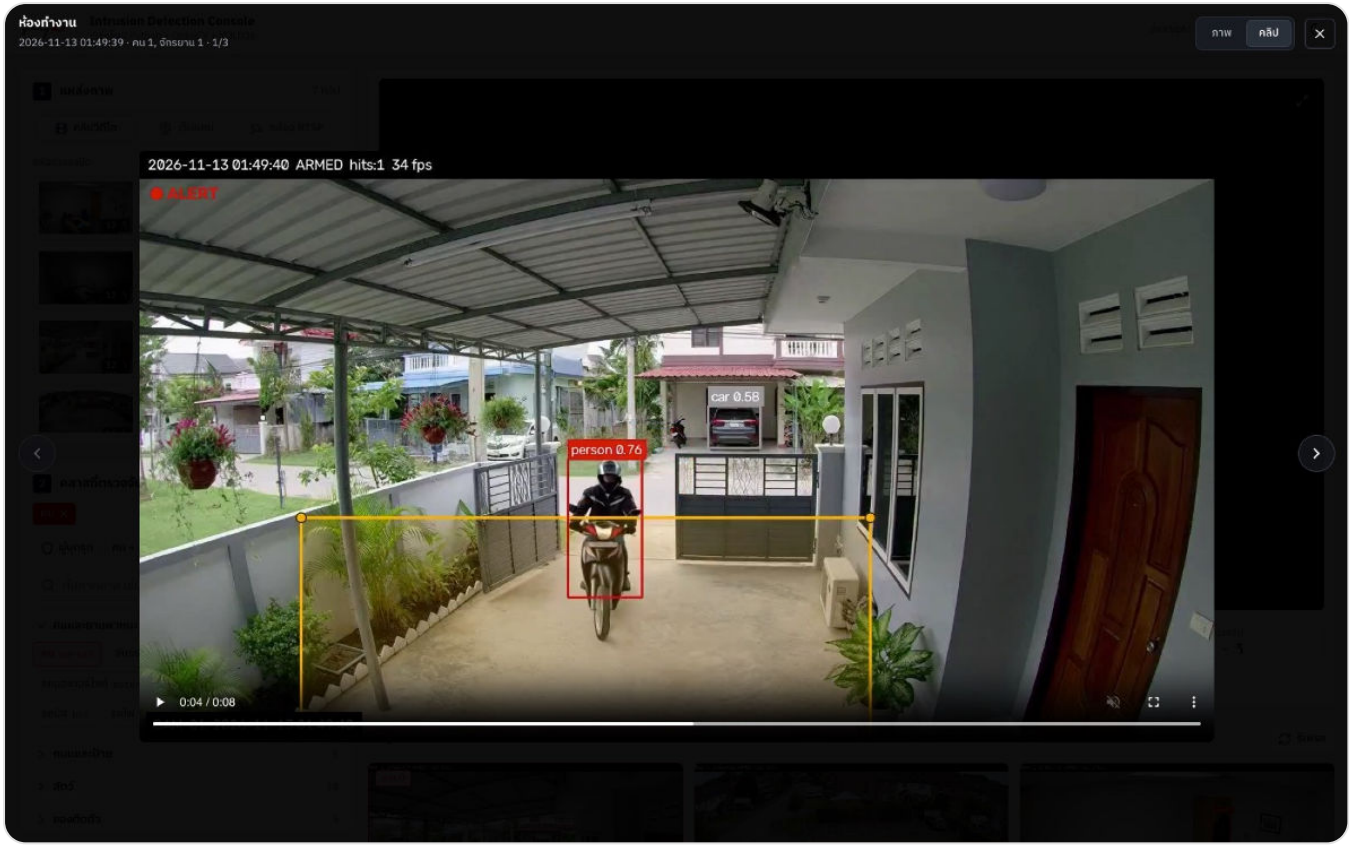
event.json ไฟล์บันทึกรายละเอียดเหตุการณ์แบบที่โปรแกรมอ่านต่อได้

ffprobe โปรแกรมดูข้อมูลวิดีโอ ไม่ได้มากับ imageio-ffmpeg จึงใช้ ffmpeg -i แทน

สิ่งที่ควรเห็น



ส่วน **เหตุการณ์ที่บันทึกไว้** ใต้ภาพสด — การ์ดของรอบนี้กรอบแดงพร้อมป้าย "รอบนี้" มีเวลา ชื่อสถานที่ และป้ายสิ่งที่เจอ เช่น **คน 1 · 77%** กดภาพเพื่อดูเต็มจอ



หน้าต่างดูเต็มจอ แท็บ **คลิป** — คลิป 8 วินาทีมีชื่อกล้อง วันเวลา และป้าย **ALERT** · ปุ่มลูกศรเปลี่ยนเหตุการณ์ · Esc หรือกดพื้นหลังเพื่อปิด

ในโฟลเดอร์ `events/[วันที่_เวลา]/` จะมี `snapshot.jpg` `clip.mp4` และ `event.json` — คลิปเป็น H.264 เปิดได้ทั้งบนคอมและมือถือ

เคล็ดลับ

- หลังเกิดเหตุการณ์ ช่อง "สถานะระบบ" ขึ้น **พัก N วิ** ตามค่า "พักแจ้งเตือน" (ค่าเริ่ม 60 วินาที) กันอีเมลรัวจากเหตุการณ์เดียว
- ไม่มีเหตุการณ์เลย ส่วนใหญ่เพราะสถานะ "นอกเวลา" — ปิดสวิตช์ช่วงเวลา หรือตั้งให้ครอบเวลาตอนนี้
- ถ้าเกิดเหตุก่อนมีภาพย้อนหลังครบ 3 วินาที ระบบใช้เท่าที่มี คลิปจึงอาจสั้นกว่า 8 วินาทีเล็กน้อย

4.2 หน้าหลังบ้าน /admin ตั้งค่าอีเมล

PROMPT · P4B

Phase 4 (2/3) · หน้าหลังบ้าน /admin ตั้งค่าอีเมล

สร้าง "หน้าหลังบ้าน" ที่ http://127.0.0.1:8000/admin (static/admin.html) มีรหัสผ่าน ไว้ตั้งค่าอีเมลแจ้งเตือน แยกจากแผงควบคุม ใช้ app.css ร่วมกัน ตามแบบหน้าเว็บใน CLAUDE.md
แถบบน: ชื่อ "ตั้งค่าหลังบ้าน" บรรทัดรอง "Intrusion Detection Console" · ขวา: ปุ่ม "กลับแผงควบคุม" (ไอคอนลูกศรซ้าย)
· ปุ่มไอคอนออกจากระบบ (แสดงหลังเข้าสู่ระบบแล้ว) · ปุ่มสลับธีม
รหัสผ่านหลังบ้าน
- อ่านจาก ADMIN_PASSWORD ในไฟล์ .env ทุกครั้งที่ล็อกอิน (แก้แล้วไม่ต้องรีสตาร์ต) อย่างน้อย 6 ตัวอักษร
- ยังไม่ได้ตั้ง = การ์ด "ยังไม่ได้ตั้งรหัสผ่านหลังบ้าน" บอก 3 ขั้น (เปิดไฟล์ .env ถ้าไม่มีให้คัดลอกจาก .env.example · ใส่รหัสที่บรรทัด ADMIN_PASSWORD=
· กดปุ่มด้านล่าง) + ปุ่มหลักเต็มความกว้าง "ตรวจอีกครั้ง"
- หน้าเข้าสู่ระบบ: การ์ดกว้าง 440px อยู่กลางแนวนอน ห่างแถบบน 64px (ไม่จัดกลางแนวตั้ง) ระยะขอบในการ์ด 28px
ไอคอน หัวข้อ และคำอธิบาย จัดกึ่งกลาง · ด้านบนเป็นไอคอนแม่กุญแจสีแดงในสี่เหลี่ยมพื้นแดงอ่อน 48px
หัวข้อ "เข้าสู่ระบบหลังบ้าน" · คำอธิบายสีเทา "ใส่รหัสผ่าน ADMIN_PASSWORD จากไฟล์ .env"
ช่อง "รหัสผ่าน" มีปุ่มรูปตาแสดง/ซ่อนรหัส · ปุ่มหลักขนาดปกติ (สูง 36px) เต็มความกว้าง "เข้าสู่ระบบ" · รหัสผิดขึ้นข้อความแดงใต้ช่อง
ผิด 5 ครั้งให้รอ 60 วินาที · จำการล็อกอินด้วย session cookie (HttpOnly, SameSite=Strict) และเทียบรหัสด้วย hmac.compare_digest
หลังเข้าสู่ระบบ (เนื้อหากว้างสูงสุด 780px กลางจอ)
- หัวข้อ "อีเมลแจ้งเตือน" + คำอธิบายสีเทา "กำหนดอีเมลที่ใช้ส่ง และอีเมลที่จะได้รับการแจ้งเตือนเมื่อระบบตรวจพบผู้บุกรุก"
- แถบสรุป: ยังไม่ครบ = กรอบแดงพื้นแดงอ่อน ไอคอนเตือน "ยังตั้งค่าไม่ครบ" และบอกว่าขาดอะไร
ครบแล้ว = กรอบเทา ไอคอนถูก "พร้อมส่งอีเมลแจ้งเตือน" และ "ส่งจาก [ผู้ส่ง] ถึง [N] อีเมล"
- การ์ดสวิตช์ "ส่งอีเมลแจ้งเตือนจริง" คำอธิบาย "เปิด = ส่งอีเมลทุกครั้งที่ตรวจพบ · ปิด = บันทึกเป็นไฟล์ .eml ไว้ดู"
- การ์ดขั้น 1 "บัญชีผู้ส่ง" (หัวการ์ดขวา "อีเมลที่ใช้ส่งการแจ้งเตือน"):
ปุ่มแบบแท็บ Gmail · Outlook / M365 · กำหนดเอง (เลือกแล้วใส่ server และพอร์ตให้อัตโนมัติ)
ช่อง "ชื่อผู้ส่ง" กับ "อีเมลผู้ส่ง" วางคู่กัน · ช่อง "App Password (16 หลัก)" มีปุ่มรูปตา
ใต้ช่องมีคำแนะนำ + ลิงก์สีแดง https://myaccount.google.com/apppasswords (Gmail ต้องเปิด 2-Step Verification ก่อน)
ส่วนพับเก็บ "ตั้งค่าเซิร์ฟเวอร์ขั้นสูง": SMTP server · พอร์ต · การเข้ารหัส (SSL / STARTTLS / ไม่เข้ารหัส) · อีเมล From
- การ์ดขั้น 2 "ผู้รับการแจ้งเตือน" (หัวการ์ดขวา เช่น "2 อีเมล"): ช่องใส่อีเมล + ปุ่ม "เพิ่ม" (กด Enter ได้ วางหลายอีเมลพร้อมกันได้ สูงสุด 20)
ผู้รับแต่ละคนเป็นแถวกรอบเทา มีไอคอนจดหมาย อีเมลตัว JetBrains Mono และปุ่มลบ · ยังไม่มีผู้รับ = กล่องเส้นประ
- แถบปุ่มติดขอบล่างจอ: ซ้าย "บันทึกแล้ว" หรือ "มีการแก้ไขที่ยังไม่บันทึก" (ตัวแดง) · ปุ่ม "ส่งอีเมลทดสอบ" · ปุ่มหลัก "บันทึกการเก็บค่า"
- บันทึกลงไฟล์ .env: SMTP_HOST · SMTP_PORT · SMTP_SECURITY · SMTP_USER · SMTP_PASSWORD · EMAIL_FROM · EMAIL_FROM_NAME
· EMAIL_TO (คั่นด้วยจุลภาค) · EMAIL_SEND โดยไม่ลบบรรทัดอื่นในไฟล์ และสร้าง .env.example ที่ไม่มีค่าจริงให้ด้วย
- ห้ามส่งรหัสกลับมาแสดงที่หน้าเว็บ ให้เขียนแค่ "ตั้งรหัสผ่านไว้แล้ว · เว้นว่างถ้าไม่ต้องการเปลี่ยน"
App Password ของ Gmail ที่มีช่องว่างให้ตัดช่องว่างออกเอง · ห้ามใส่รหัสจริงในโค้ด และห้ามแสดงรหัสในแชท
- ปุ่ม "ส่งอีเมลทดสอบ" ส่งอีเมลสั้น ๆ หัวเรื่อง "[ทดสอบ] ระบบแจ้งเตือนผู้บุกรุก" ถ้าไม่สำเร็จให้บอกสาเหตุเป็นภาษาไทย
เช่น ล็อกอินไม่ผ่าน ตรวจ App Password หรือยังไม่เปิด 2-Step · เชื่อมต่อเซิร์ฟเวอร์ไม่ได้
ทดสอบเอง: ยังไม่ตั้งรหัส → เห็นการ์ดบอกขั้นตอน · ตั้งรหัสทดสอบชั่วคราวแล้วล็อกอิน บันทึกผู้รับ 1 คน เช็คว่า .env ถูกต้อง แล้วลบรหัสทดสอบออก ฉันจะตั้ง ADMIN_PASSWORD และกรอก App Password เอง

คำศัพท์ในพรอมต์นี้

หน้าหลังบ้าน (/admin) หน้าตั้งค่าที่ต้องใส่รหัสก่อนเข้า แยกจากหน้าใช้งาน คนที่แค่ดูกล้องจึงแก้ค่าอีเมลไม่ได้

ADMIN_PASSWORD ชื่อบรรทัดในไฟล์ .env ที่เก็บรหัสเข้าหน้าหลังบ้าน ตั้งเองอย่างน้อย 6 ตัวอักษร

hmac.compare_digest วิธีเทียบรหัสแบบปลอดภัยของ Python กันการเดารหัสจากเวลาที่ใช้ตรวจ

SSL / STARTTLS วิธีเข้ารหัสการส่งอีเมลให้คนกลางอ่านไม่ได้ Gmail ใช้ SSL พอร์ต 465

2-Step Verification การยืนยันตัวตน 2 ขั้นของบัญชี Google ต้องเปิดก่อนจึงสร้าง App Password ได้

.env ไฟล์เก็บค่าลับ เช่น รหัสผ่าน ไม่อยู่ในโค้ด และไม่ส่งต่อให้ใคร · .env.example = ไฟล์ตัวอย่างที่ไม่มีค่าจริง

session cookie ข้อมูลเล็ก ๆ ที่เบราว์เซอร์จำไว้ว่าเราล็อกอินแล้ว · HttpOnly / SameSite=Strict = ตั้งค่าให้สคริปต์และเว็บอื่นแอบใช้ไม่ได้

SMTP ระบบส่งอีเมลออกจากโปรแกรม ต้องมีชื่อเซิร์ฟเวอร์ (เช่น smtp.gmail.com) พอร์ต และการเข้ารหัส

App Password รหัส 16 ตัวที่ Google สร้างให้โปรแกรมใช้ส่งอีเมลแทนรหัส Gmail จริง ยกเลิกทิ้งได้ทุกเมื่อ

ทำเอง ตั้งรหัสเข้าหน้าหลังบ้าน ADMIN_PASSWORD ในไฟล์ .env

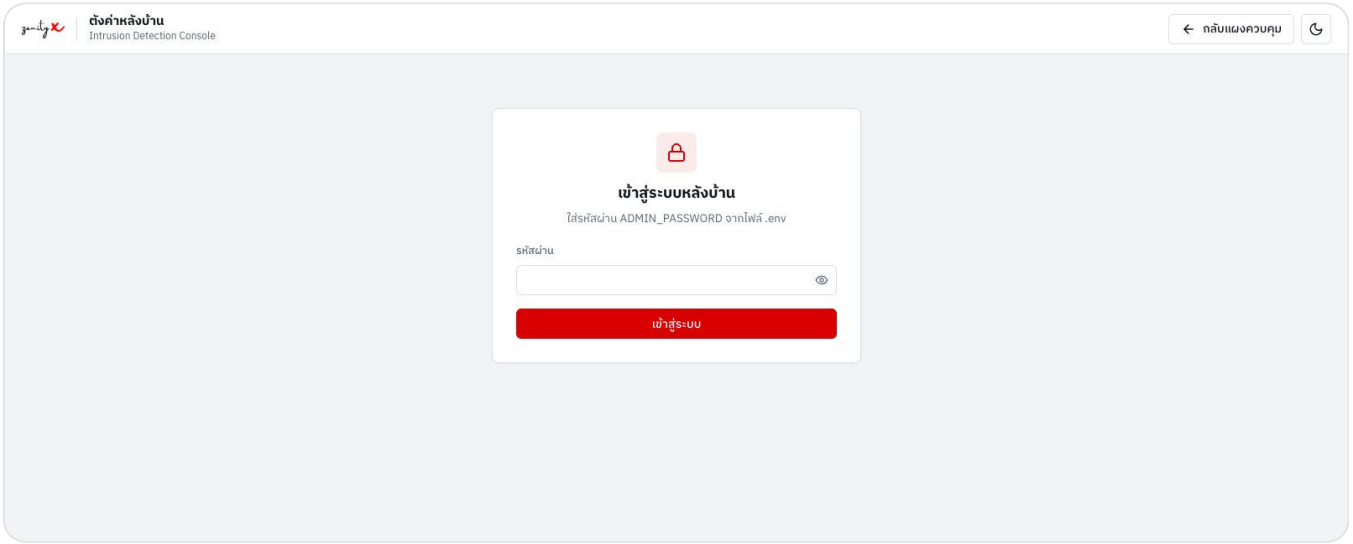
หน้าหลังบ้านต้องใส่รหัสก่อนเข้า รหัสนี้เราตั้งเองในไฟล์ .env — AI ไม่ได้ตั้งให้

1. **เปิดไฟล์ .env ในโฟลเดอร์โปรเจกต์**
   ถ้ายังไม่มี ให้คัดลอก `.env.example` แล้วตั้งชื่อเป็น `.env`
2. **พิมพ์รหัสต่อท้าย ADMIN_PASSWORD=**
   ตั้งเองอย่างน้อย 6 ตัวอักษร แล้วบันทึกไฟล์
3. **เปิด 127.0.0.1:8000/admin**
   หลังทำ P4C กดไอคอนเฟืองมุมขวาบนของแผงควบคุมได้เลย
4. **ใส่รหัสแล้วกด "เข้าสู่ระบบ"**
   แก้รหัสในไฟล์ได้ตลอด ไม่ต้องรีสตาร์ต

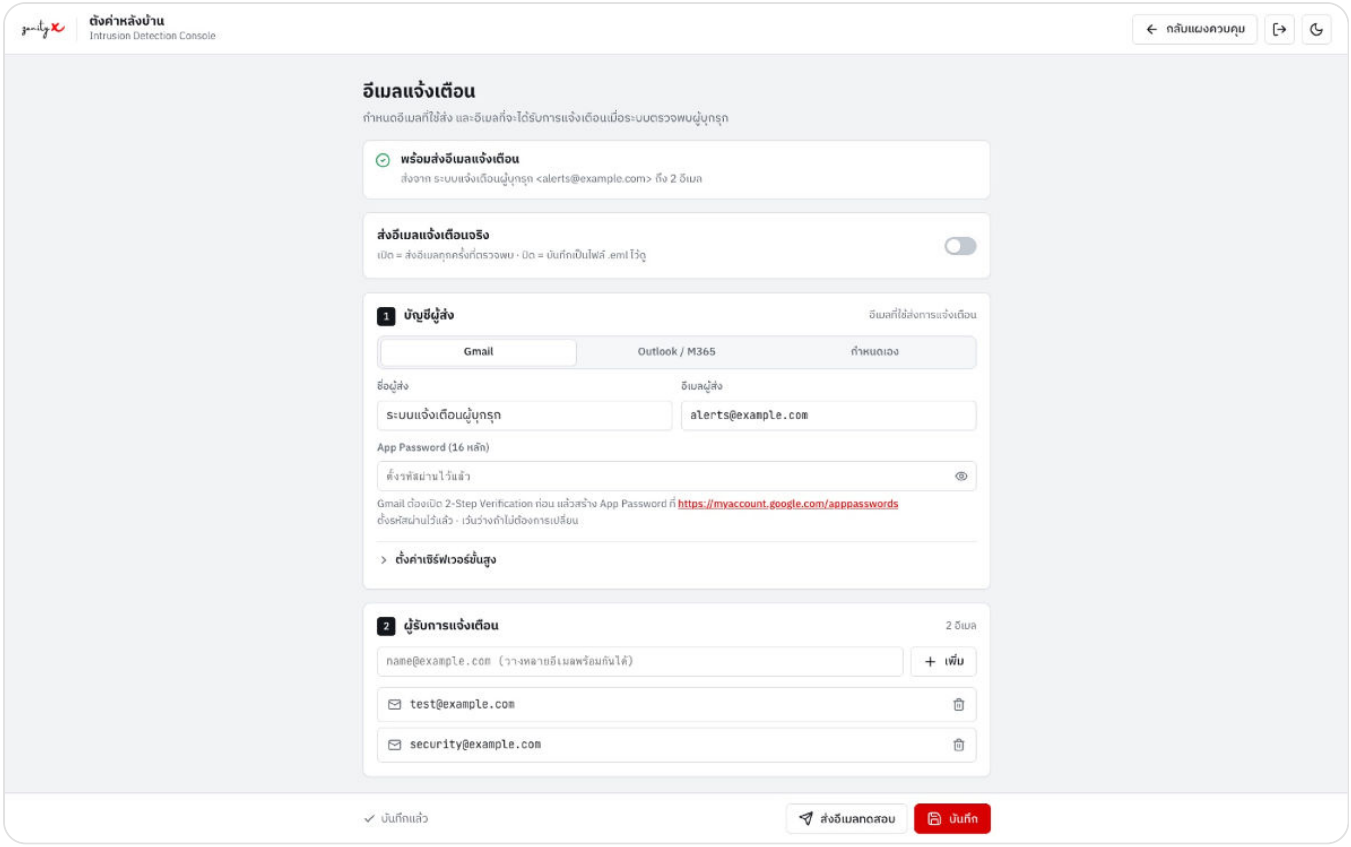
```
# ไฟล์ .env
ADMIN_PASSWORD=[รหัสที่ตั้งเอง]
# ส่วนอีเมลด้านล่าง หน้าหลังบ้านเขียนให้เอง
SMTP_USER=
EMAIL_TO=
```

ห้ามวางรหัสในแชท AI — พิมพ์รหัสในไฟล์ .env และหน้าหลังบ้านด้วยตัวเองเท่านั้น

สิ่งที่ควรเห็น



หน้า **เข้าสู่ระบบหลังบ้าน** ที่ `127.0.0.1:8000/admin` — ใส่รหัส ADMIN_PASSWORD จากไฟล์ .env · ถ้ายังไม่ตั้ง จะเห็นการ์ดบอก 3 ขั้นตอนพร้อมปุ่ม **ตรวจอีกครั้ง** แทน



หลังเข้าสู่ระบบ — การ์ดขั้น 1 **บัญชีผู้ส่ง** (Gmail · Outlook / M365 · กำหนดเอง) และการ์ดขั้น 2 **ผู้รับการแจ้งเตือน** ค่าที่บันทึกไปอยู่ในไฟล์ .env และไม่แสดงรหัสกลับมาบนหน้าเว็บ

4.3 สร้าง Google App Password (ทำเองทีละขั้น)

ห้ามวางรหัสในแชท AI — App Password คือกุญแจเข้าอีเมลของเรา ให้ **กรอกที่หน้าหลังบ้าน /admin เท่านั้น** (การ์ดบัญชีผู้ส่ง) ระบบเก็บไว้ในไฟล์ .env และไม่แสดงรหัสกลับมา ไม่ต้องบอกใคร รวมถึง Claude Code

1. **เปิดการยืนยันแบบ 2 ขั้นตอน (2-Step Verification)**
   ไปที่ https://myaccount.google.com/signinoptions/twosv แล้วเปิด ถ้ายังไม่เปิด

2. **เปิดหน้า App Password**
   ไปที่ https://myaccount.google.com/apppasswords (ล็อกอินอีกครั้งถ้าถูกถาม)

3. **ตั้งชื่อแอป แล้วกด "สร้าง"**
   เช่น `YOLO26 Alert`

4. **คัดลอกรหัส 16 ตัวที่แสดง**
   รหัสแสดงครั้งเดียว · จะเว้นวรรคหรือไม่ก็ได้ ระบบตัดช่องว่างออกให้เอง

5. **กรอกที่หน้าหลังบ้าน /admin (การ์ดบัญชีผู้ส่ง)**
   อีเมลผู้ส่ง · App Password → เพิ่มอีเมลผู้รับในการ์ดผู้รับการแจ้งเตือน → กด "บันทึก" (ระบบเก็บไว้ใน .env)

6. **กดส่งอีเมลทดสอบ**
   ปุ่ม "ส่งอีเมลทดสอบ" ที่แถบล่างของหน้าหลังบ้าน แล้วเช็คกล่องจดหมาย

⚠ ถ้าทำไม่ได้

- **ไม่เห็นเมนู App Password** — ยังไม่เปิด 2-Step หรือเป็นบัญชีองค์กร (Google Workspace) ที่แอดมินปิดไว้
- **ใช้รหัส Gmail ปกติไม่ได้** ต้องเป็น App Password เท่านั้น · ยกเลิกรหัสได้ทุกเมื่อที่หน้าเดิม
- ยังไม่มีรหัส ก็ทำ P4C ต่อได้ — เลือก "บันทึกเป็นไฟล์ .eml (ซ้อม)" ไว้ก่อน แล้วกลับมากรอกทีหลัง

4.4 ส่งอีเมลแจ้งเตือนจากเหตุการณ์

PROMPT · P4C

Phase 4 (3/3) · ส่งอีเมลแจ้งเตือนจากเหตุการณ์

ต่อระบบเหตุการณ์เข้ากับอีเมลที่ตั้งไว้ในหน้าหลังบ้าน ตามแบบหน้าเว็บใน CLAUDE.md
อีเมลแจ้งเตือน
- ส่งเมื่อบันทึกคลิปเหตุการณ์เสร็จ โดยใช้ค่าจาก .env ที่หน้าหลังบ้านบันทึกไว้
- หัวเรื่อง เช่น "[แจ้งเตือนผู้บุกรุก] ห้องทำงาน | พบ คน 1 | 12/11/2026 22:15:08"
- เนื้อหาแบบ HTML ภาษาไทย: หัวข้อสีแดง "ตรวจพบผู้บุกรุก" · ตาราง กล้อง / เวลา · รายการสิ่งที่ตรวจพบ พร้อมจำนวนและความมั่นใจ
  · ภาพ snapshot แสดงในตัวอีเมล · หมายเหตุ "คลิปเหตุการณ์ 8 วินาที (ก่อนเกิดเหตุ 3 วิ + หลังเกิดเหตุ 5 วิ)"
- แนบ snapshot.jpg และ clip.mp4 (คลิปใหญ่เกิน 20 MB ไม่ต้องแนบ ให้บอกที่เก็บไฟล์แทน)
- เก็บสำเนา email.eml ในโฟลเดอร์เหตุการณ์ทุกครั้ง · ถ้าเลือกแบบซ้อมหรือหลังบ้านยังตั้งค่าไม่ครบ ให้บันทึกแค่ไฟล์ ไม่ส่ง
- เขียนส่วนอีเมลไว้ใน detector.py · ส่งเบื้องหลังแยกจากการตรวจจับ (thread) ส่งไม่ผ่านต้องไม่ทำให้ระบบหยุด และบันทึกสาเหตุเป็นภาษาไทย
- ถ้าเลือก "ส่งอีเมลจริง" แต่หลังบ้านตั้งค่าไม่ครบ ให้บันทึกเป็นไฟล์แทน และขึ้นข้อความแจ้งบนหน้าเว็บ
แผงควบคุม (index.html)
- แถบบน: เพิ่มปุ่มไอคอนเฟืองก่อนปุ่มสลับธีม ลิงก์ไป /admin (คำอธิบายเมื่อชี้ "ตั้งค่าหลังบ้าน")
- การ์ดขั้น 4 เพิ่มหัวข้อย่อย "อีเมลแจ้งเตือน" เป็นตัวเลือกแบบการ์ด 2 อัน (ปุ่มวงกลมด้านซ้าย หัวข้อหนา คำอธิบายสีเทา):
  "บันทึกเป็นไฟล์ .eml (ซ้อม)" — ไม่ส่งจริง เปิดดูหน้าตาอีเมลได้จากรายการเหตุการณ์
  "ส่งอีเมลจริง" — "ส่งถึงผู้รับ 2 อีเมลตามที่ตั้งไว้ในหลังบ้าน" (หลังบ้านยังตั้งไม่ครบ = กดไม่ได้ และเขียน "ยังตั้งค่าอีเมลไม่ครบ · ตั้งได้ที่หน้าหลังบ้าน")
  ตัวเลือกที่เลือก กรอบแดงพื้นแดงอ่อน · ค่าเริ่มต้นตาม EMAIL_SEND · ใต้ตัวเลือกมีลิงก์สีแดง "ตั้งค่าอีเมลผู้ส่งและผู้รับ (หลังบ้าน)"
- การ์ดเหตุการณ์ แถวล่างซ้ายแสดงป้ายสถานะอีเมลมีไอคอนจดหมาย: "บันทึก .eml" · "ส่งอีเมลแล้ว" · "ส่งไม่สำเร็จ"
  และเพิ่มปุ่มไอคอนจดหมายดูอีเมล ต่อจากปุ่มเล่นคลิป
- หน้าต่างดูเต็มจอ เพิ่มแท็บที่ 3 "อีเมล": การ์ดขาวกว้างไม่เกิน 760px ด้านบนเป็น หัวเรื่อง (ตัวหนา) · จาก · ถึง · ไฟล์แนบ
  พร้อมลิงก์สีแดง "ดาวน์โหลด .eml" แล้วแสดงเนื้อหาอีเมลจริงด้านล่างในกรอบแยก (iframe แบบ sandbox)
  อ่านหัวอีเมลภาษาไทยด้วย email.header.make_header(decode_header(...)) ชื่อไทยจะไม่แตกเป็นช่องว่าง
ทดสอบแบบ "บันทึกเป็นไฟล์ .eml" กับ videos/12_cctv_office_night_ir.mp4 ให้ดูก่อน แล้วเปิดแท็บอีเมลให้ดูหน้าตา
สุดท้ายอัปเดต CLAUDE.md ว่าระบบทำอะไรได้บ้าง และบอกขั้นตอนที่ฉันต้องทำต่อเพื่อส่งอีเมลจริง

คำศัพท์ในพรอมต์นี้

HTML (อีเมล) อีเมลแบบมีรูปแบบ ตัวหนา ตาราง และรูปภาพในตัว เหมือนหน้าเว็บเล็ก ๆ

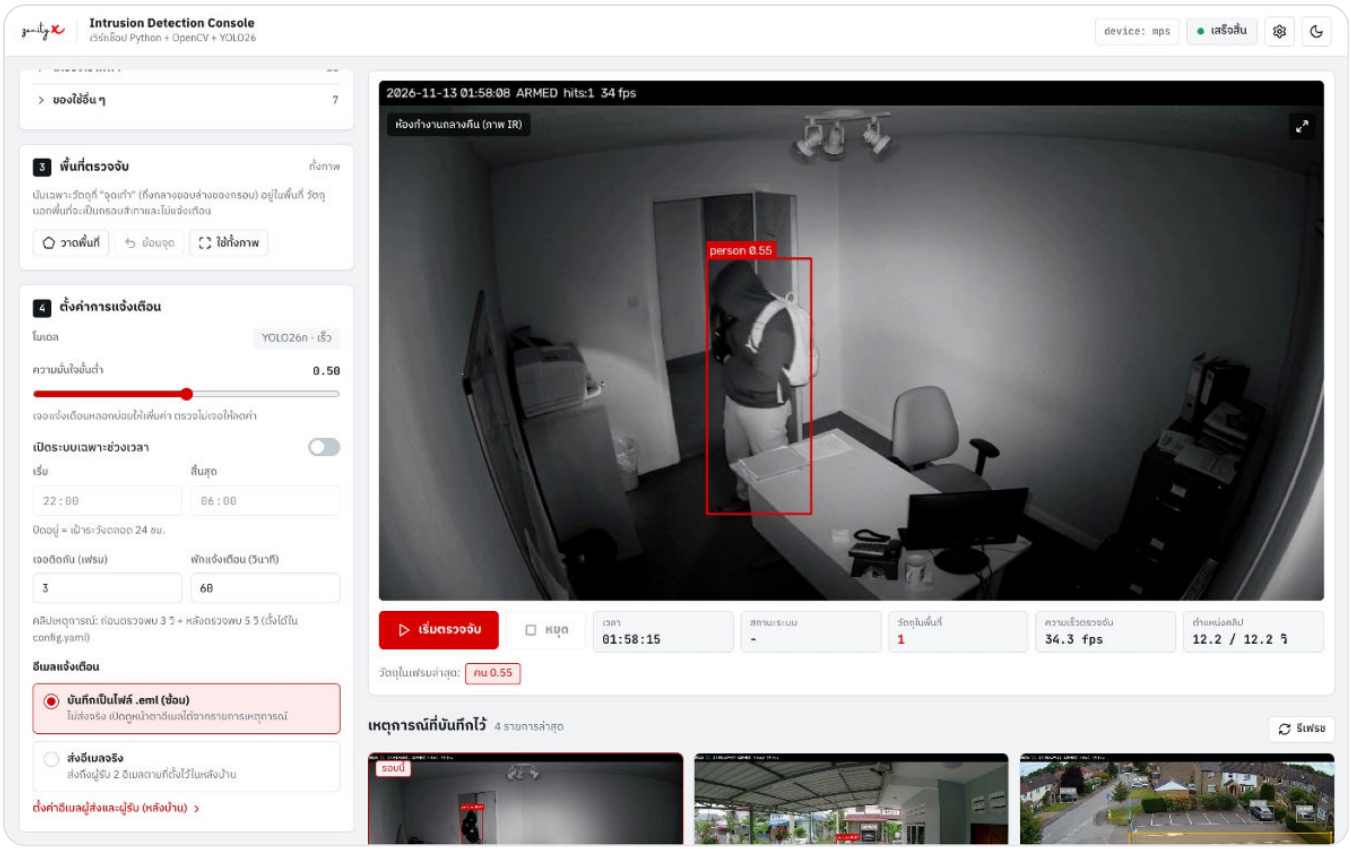
.eml ไฟล์อีเมลทั้งฉบับ เปิดด้วยโปรแกรมอีเมลได้ ใช้ซ้อมดูหน้าตาโดยไม่ต้องส่งจริง

iframe แบบ sandbox กรอบแสดงหน้าเว็บย่อยในหน้าเว็บ แบบที่ห้ามรันสคริปต์ ใช้แสดงเนื้อหาอีเมลอย่างปลอดภัย

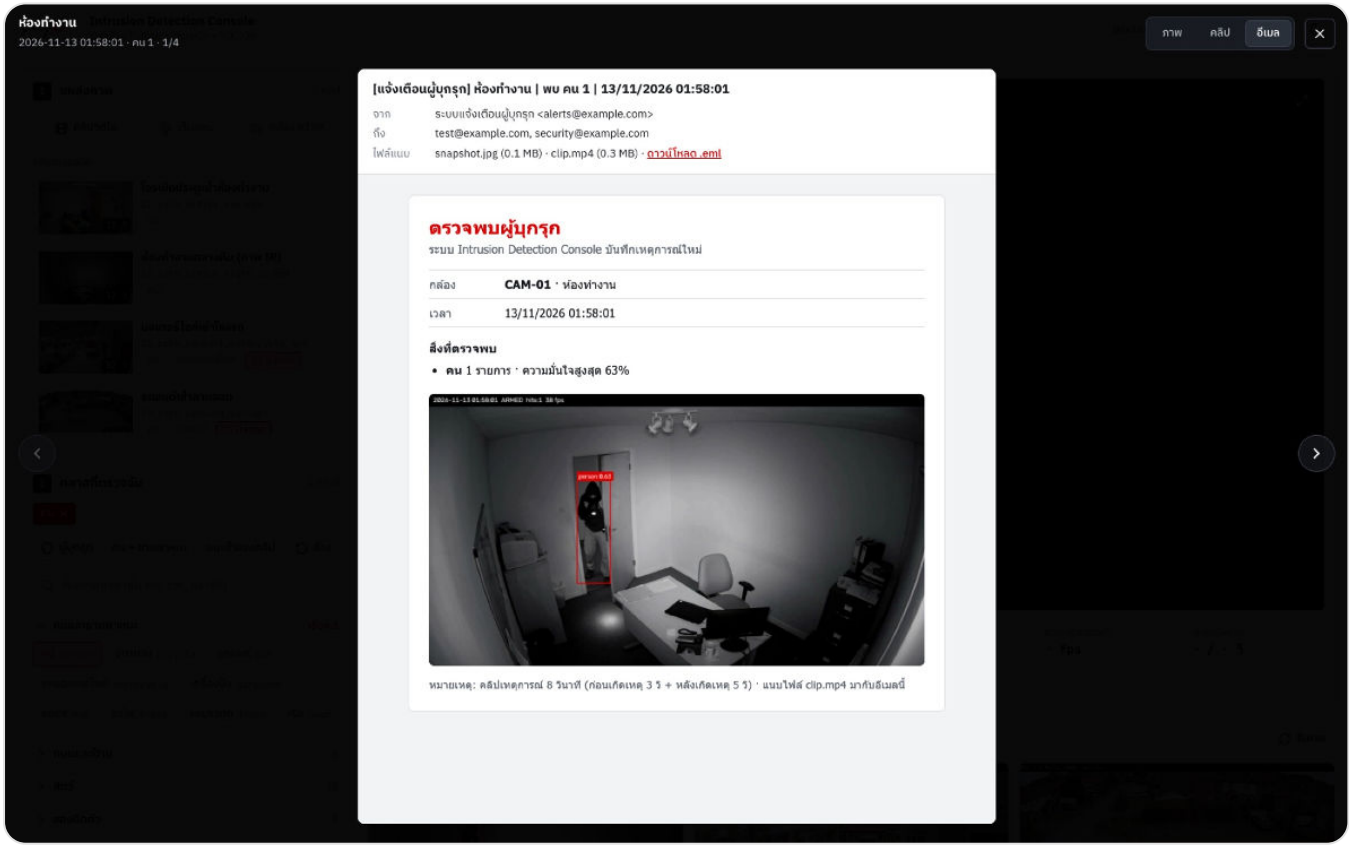
ไฟล์แนบ 20 MB Gmail รับไฟล์แนบได้ไม่เกิน 25 MB จึงตั้งเพดาน 20 MB เผื่อไว้

EMAIL_SEND บรรทัดใน .env ที่บอกว่าเปิดส่งจริงไหม (true = ส่ง · false = บันทึกเป็นไฟล์) ตั้งจากสวิตช์ในหน้าหลังบ้าน

✓ สิ่งที่ควรเห็น



การ์ดขั้น 4 ของแผงควบคุม — เลือก **บันทึกเป็นไฟล์ .eml (ซ้อม)** หรือ **ส่งอีเมลจริง** (กดได้เมื่อหลังบ้านตั้งค่าครบ) · ไอคอนเฟืองมุมขวาบนพาไปหน้าหลังบ้าน



แท็บ **อีเมล** ในหน้าต่างดูเหตุการณ์ — หัวเรื่อง · จาก · ถึง · ไฟล์แนบ · ลิงก์ดาวน์โหลด .eml และเนื้อหาอีเมลจริงพร้อมภาพ snapshot

- เลือก "บันทึกเป็นไฟล์ .eml (ซ้อม)" ระบบบันทึก `email.eml` ในโฟลเดอร์เหตุการณ์ เปิดดูได้จากแท็บอีเมล หรือดับเบิลคลิกเปิดในโปรแกรมอีเมล

- เลือก "ส่งอีเมลจริง" อีเมลเข้ากล่องจดหมาย หัวเรื่องเช่น `[แจ้งเตือนผู้บุกรุก] ห้องทำงาน | พบ คน 1 | 12/11/2026 22:15:08`
- การ์ดเหตุการณ์แสดงสถานะ **บันทึก .eml / ส่งอีเมลแล้ว / ส่งไม่สำเร็จ** พร้อมไอคอนจดหมายดูอีเมล

> **เคล็ดลับ**
> - "ส่งอีเมลจริง" กดไม่ได้ แปลว่าหลังบ้านยังตั้งค่าไม่ครบ — กดลิงก์ "ตั้งค่าอีเมลผู้ส่งและผู้รับ (หลังบ้าน)" ใต้ตัวเลือก
> - ขึ้น `535 Username and Password not accepted` แปลว่าใช้รหัส Gmail ปกติหรือพิมพ์ผิด — สร้าง App Password ใหม่แล้วกรอกที่หน้าหลังบ้าน /admin

✓ หน้าตาโฟลเดอร์เมื่อจบคลาส

```
yolo26-detection/
├── videos/              คลิปทดสอบ 7 คลิป + clips.yaml (ชื่อไทย คลาส พื้นที่แนะนำ)
├── brand/               โลโก้และ favicon ของหน้าเว็บ
├── detector.py          ระบบตรวจจับด้วย YOLO26
├── webapp.py            เว็บ Flask ที่ 127.0.0.1:8000 (แผงควบคุม + /admin)
├── static/
│   ├── index.html       แผงควบคุม Intrusion Detection Console
│   ├── admin.html       หน้าหลังบ้าน ตั้งค่าอีเมล
│   └── assets/          app.css (หน้าตาร่วม) · icons.js (ไอคอน SVG)
├── config.yaml          ค่าเริ่มต้น: ความมั่นใจ · ช่วงเวลา · คลิปเหตุการณ์ · ชื่อกล้อง
├── .env                 ADMIN_PASSWORD + ค่าอีเมลจากหน้าหลังบ้าน (ไม่แชร์)
├── .env.example         ตัวอย่างไฟล์ .env ไม่มีค่าจริง
├── events/              snapshot.jpg · clip.mp4 · event.json · email.eml
├── requirements.txt · .gitignore
└── CLAUDE.md            บันทึกโปรเจกต์ + แบบหน้าเว็บ ให้ AI จำได้
```

> **เคล็ดลับ**
> - เปิด session ใหม่วันหลัง Claude Code จะอ่าน CLAUDE.md แล้วรู้ทันทีว่าระบบทำอะไรได้ หน้าตาเป็นแบบไหน และเปิดเว็บด้วยคำสั่งอะไร
> - แก้รหัสหลังบ้านได้ที่บรรทัด `ADMIN_PASSWORD=` ในไฟล์ .env ส่วนค่าอีเมลแก้ที่หน้าหลังบ้าน ไม่ต้องแก้ไฟล์เอง

BONUS · 1:50–2:00

ต่อยอด: กล้องจริงและดูจากมือถือ

สองขั้นนี้ไม่บังคับในคลาส ผู้สอนสาธิตให้ดู แล้วผู้เรียนนำกลับไปทำต่อกับกล้องที่บ้านหรือร้านของตัวเองได้

B1 ต่อกล้องวงจรปิดจริง (RTSP)

PROMPT · B1

Bonus · ต่อกล้องวงจรปิดจริง (RTSP)

ฉันจะต่อระบบกับกล้องวงจรปิดจริง ช่วยเตรียมให้หน่อย
- อธิบายรูปแบบ URL RTSP ของกล้องยี่ห้อยอดนิยม (Hikvision, Dahua, TP-Link Tapo) ทั้งภาพหลักและภาพรอง และแนะนำว่าควรใช้ภาพรอง (sub stream) เพราะอะไร
- เก็บ URL กล้องที่มีรหัสผ่านไว้ใน .env (CAMERA_URL) แล้วใช้ปุ่ม ".env" ในแท็บกล้อง RTSP หน้าเว็บต้องไม่แสดงรหัสกล้อง
- ถ้ากล้องหลุดระหว่างทำงาน ให้ต่อใหม่อัตโนมัติและแสดงในป้ายสถานะ ถ้าต่อไม่ได้ตั้งแต่แรกให้แจ้งข้อผิดพลาดทันที
- ทดสอบกับเว็บแคมของเครื่องนี้ให้ดู

กล้องของฉัน: [ยี่ห้อ] · IP [192.168.1.xx] · ชื่อผู้ใช้ [admin] (รหัสผ่านฉันจะใส่ใน .env เอง)

คำศัพท์ในพรอมต์นี้

main stream / sub stream กล้องส่งภาพได้ 2 ชุด ภาพหลักละเอียดสูงแต่หนัก ภาพรองเล็กกว่าแต่ลื่น เหมาะกับ AI

IP เลขที่อยู่ของกล้องในเครือข่ายบ้านหรือออฟฟิศ เช่น 192.168.1.64 ดูได้จากแอปกล้องหรือเราเตอร์

CAMERA_URL บรรทัดใน .env ที่เก็บ URL กล้องพร้อมรหัส แยกออกจากโค้ด

เคล็ดลับ

- ทดสอบ URL กล้องในโปรแกรม VLC ก่อน ถ้า VLC เปิดไม่ได้ ระบบก็เปิดไม่ได้ · คอมกับกล้องต้องอยู่วง LAN เดียวกัน · ใช้ sub stream
- URL กล้องที่มีรหัสผ่านเก็บใน `.env` (`CAMERA_URL`) แล้วกดปุ่ม **.env** ในแท็บกล้อง RTSP — หน้าเว็บไม่แสดงรหัสกล้อง
- เว็บแคมบน Mac เปิดไม่ได้ → System Settings → Privacy → Camera → อนุญาตแอปที่ใช้รัน

B2 เปิดดูจากมือถือในวง Wi-Fi เดียวกัน

PROMPT · B2

Bonus · เปิดดูจากมือถือในวง Wi-Fi เดียวกัน

อยากเปิดแผงควบคุมนี้จากมือถือที่ต่อ Wi-Fi เดียวกัน อย่างปลอดภัย
- เพิ่มตัวเลือก python webapp.py --host 0.0.0.0 ให้เครื่องอื่นในวงแลนเข้าได้ (ค่าเริ่มต้นยังเป็น 127.0.0.1) และบอก IP ที่ต้องพิมพ์ในมือถือ
- เมื่อเปิดให้เครื่องอื่นเข้า ให้แผงควบคุมและภาพสดต้องเข้าสู่ระบบก่อน ใช้รหัส ADMIN_PASSWORD และหน้าเข้าสู่ระบบแบบเดียวกับหลังบ้าน
- เช็คว่าหน้าเว็บใช้บนมือถือแนวตั้งและแนวนอนได้สะดวก

อธิบายความเสี่ยงถ้าเปิดออกอินเทอร์เน็ตตรง ๆ และทางที่ปลอดภัยกว่า

คำศัพท์ในพรอมต์นี้

วงแลน (LAN) เครือข่ายในบ้านหรือออฟฟิศ อุปกรณ์ที่ต่อ Wi-Fi เดียวกันมองเห็นกันได้

0.0.0.0 ตั้งให้เว็บรับการเชื่อมต่อจากเครื่องอื่นในวงแลน (127.0.0.1 = เฉพาะเครื่องตัวเอง)

✓ รัน `.venv/bin/python webapp.py --host 0.0.0.0` แล้วมือถือที่ต่อ Wi-Fi เดียวกันเปิดแผงควบคุมได้ด้วย IP ที่ Claude Code บอก ต้องใส่รหัส ADMIN_PASSWORD ก่อนดูภาพสด — ค่าเริ่มต้นยังเปิดเฉพาะ `127.0.0.1` ในเครื่องตัวเอง

เคล็ดลับ

- อย่าเปิดหน้าเว็บออกอินเทอร์เน็ตตรง ๆ — ให้ Claude Code อธิบายความเสี่ยงและทางที่ปลอดภัยกว่าตามที่ Prompt B2 ขอไว้

TROUBLESHOOTING

แก้ปัญหาที่พบบ่อย

อาการ	สาเหตุ	วิธีแก้
`ModuleNotFoundError: ultralytics`	รันด้วย Python นอก .venv	ใช้ `.venv/bin/python` (Windows `.venv\Scripts\python`)
ดาวน์โหลด yolo26n.pt ไม่ได้	อินเทอร์เน็ต/ไฟร์วอลล์	ผู้สอนแจกไฟล์ .pt วางในโฟลเดอร์โปรเจกต์
เปิด 127.0.0.1:8000 ไม่ขึ้น	ยังไม่ได้รัน webapp.py / พอร์ตถูกใช้	รัน `.venv/bin/python webapp.py` · ให้ Claude Code เปลี่ยนพอร์ต
หน้าเว็บไม่เหมือนในสไลด์	AI ไม่ได้อ่านแบบหน้าเว็บ	พิมพ์ "ทำหน้าเว็บให้ตรงตามแบบหน้าเว็บใน CLAUDE.md"
ความเร็วตรวจจับ (fps) ต่ำมาก	เครื่องช้า / ภาพใหญ่เกิน	ลดขนาดภาพที่ส่งให้ YOLO เป็น 480 · ปิดโปรแกรมอื่น · ไฟล์วิดีโอเล่นตามความเร็วจริง fps จึงไม่เกิน fps ของคลิป
เตือนหลอกบ่อย	ไม่มีพื้นที่ / ความมั่นใจต่ำไป	วาดพื้นที่ · เพิ่มความมั่นใจเป็น 0.6 · เจอติดกัน 5 เฟรม
ไม่มีเหตุการณ์เลย	อยู่นอกช่วงเวลา (สถานะ "นอกเวลา")	ปิดสวิตช์ช่วงเวลา หรือตั้งให้ครอบเวลาตอนนี้
แจ้งเตือนตั้งแต่เฟรมแรก	มีรถจอดอยู่ในพื้นที่แล้ว	วาดพื้นที่ไม่ให้ครอบรถที่จอดอยู่ หรือกด "ค่าแนะนำของคลิป"
เข้าหน้าหลังบ้านไม่ได้	ยังไม่ตั้ง ADMIN_PASSWORD	ใส่รหัสในไฟล์ .env แล้วกด "ตรวจอีกครั้ง"
"ส่งอีเมลจริง" กดไม่ได้	หลังบ้านยังตั้งค่าไม่ครบ	กรอกผู้ส่ง App Password และผู้รับที่ /admin แล้วกด "บันทึก"
คลิปเปิดในมือถือไม่ได้	ยังเป็น mp4v	แปลง H.264 ด้วย imageio-ffmpeg (P4A)
`535 Username and Password not accepted`	ใช้รหัส Gmail ปกติ / พิมพ์ผิด	สร้าง App Password ใหม่ กรอกที่หน้าหลังบ้าน /admin (การ์ดบัญชีผู้ส่ง)
เว็บแคมเปิดไม่ได้บน Mac	ยังไม่ให้สิทธิ์กล้อง	System Settings → Privacy → Camera → อนุญาต Terminal/แอปที่รัน
RTSP ต่อไม่ได้	URL/รหัสผิด หรือคนละวง LAN	ทดสอบ URL ใน VLC ก่อน · ใช้ sub stream

QUICK REFERENCE

Prompt Cheat Sheet

Prompt ทั้งหมดของวันนี้เรียงตามลำดับ — ฉบับเว็บกดคัดลอกได้ที่บทของแต่ละ Prompt และอ่านคำศัพท์ได้ใต้กล่อง

ID	Prompt	ช่วงเวลา
S0	Setup · เตรียมโปรเจกต์ + แบบหน้าเว็บ	0:25–0:35
P1	Phase 1 · แผงควบคุม + ตรวจจับคน	0:35–0:55
P2	Phase 2 · เลือกคลาสที่ต้องการตรวจจับ	0:55–1:05
P3	Phase 3 · พื้นที่ตรวจจับ + ช่วงเวลาเฝ้าระวัง	1:05–1:25
P4A	Phase 4 (1/3) · เก็บหลักฐาน ภาพ + คลิป 8 วินาที	1:25–1:50
P4B	Phase 4 (2/3) · หน้าหลังบ้าน /admin ตั้งค่าอีเมล	1:25–1:50
P4C	Phase 4 (3/3) · ส่งอีเมลแจ้งเตือนจากเหตุการณ์	1:25–1:50
B1	Bonus · ต่อกล้องวงจรปิดจริง (RTSP)	1:50–2:00
B2	Bonus · เปิดดูจากมือถือในวง Wi-Fi เดียวกัน	1:50–2:00

ผู้สอน



เอกรัตน์ ใจชนะ (นิว) · AI Instructor, ZenityX

สอนจากประสบการณ์ทำระบบนี้จริง — ผลงาน "ระบบกล้องวงจรปิดตรวจจับอัจฉริยะ" ได้รางวัลชนะเลิศ Oral Presentation (ด้านนวัตกรรม) การประกวดผลงานสหกิจศึกษา CWIE ดีเด่น เครือข่ายพัฒนา CWIE ภาคเหนือตอนล่าง

ติดต่อสถาบัน ZenityX

โทร
062-262-9644

อีเมล
admin@zenityxai.com

ออกแบบหลักสูตรโดยสถาบัน ZenityX · AI Detection with YOLO26 · Workshop 2 ชั่วโมง