WORKSHOP 2 ชั่วโมง · AI OBJECT DETECTION · CLAUDE CODE

AI Detection with YOLO26

สร้างกล้อง AI ตรวจจับผู้บุกรุกด้วย YOLO26 — ตรวจจับคนและรถ แล้วแจ้งเตือนทางอีเมลพร้อมภาพและคลิป 8 วินาที



ออกแบบหลักสูตรโดยสถาบัน ZenityX

Format

Workshop 2 ชั่วโมง · สอนทีละ Prompt

วาง Prompt → AI เขียนโค้ด → รันดูผลทันที

Duration

120 นาที · 8 ช่วงการเรียน

ปูพื้น AI Detection แล้วลงมือสร้างระบบทีละเฟส

Core Tool

Claude Code + YOLO26

Python · OpenCV · Flask · Ultralytics · ffmpeg · Gmail SMTP

Core Output

ระบบกล้องแจ้งเตือนผู้บุกรุก

อีเมลพร้อมภาพ + คลิป 8 วินาที (ก่อนเหตุ 3 วิ หลังเหตุ 5 วิ)

1. ภาพรวมหลักสูตร

Workshop 2 ชั่วโมงที่ผู้เรียนสร้างระบบกล้องวงจรปิดอัจฉริยะได้เองตั้งแต่โฟลเดอร์ว่าง โดยใช้ **Claude Code** เป็น AI Agent เขียนโค้ด Python + OpenCV + Flask + **YOLO26** ให้ทีละเฟส ผู้เรียนเพียงวาง Prompt ที่เตรียมไว้ แล้วตั้งค่าและดูผลทุกอย่างบน **หน้าเว็บของตัวเอง**

เส้นเรื่องของ Workshop

จากกล้องที่ “บันทึก” แต่ไม่มีใครดู → ระบบที่รู้ว่า “อะไร” เข้ามา แล้วส่งหลักฐานถึงมือถือทันที



กล้องวงจรปิดทั่วไป

บันทึก แต่ไม่มีใครดู

รู้ตัวเมื่อเสียหายไปแล้ว

Motion Detection

เตือนทุกอย่างที่ขยับ

ใบไม้ เงา แมว — เตือนหลอกจนคนเลิกสนใจ

AI Detection

รู้ว่า “อะไร” ขยับ

แจ้งเฉพาะคน/รถ ในพื้นที่ที่สนใจ ในเวลาที่ไม่ควรมีใคร

ใช้งานได้กับ บ้านกลางคืน · ร้านค้าหลังปิด · โกดัง · ลานจอดรถ · ห้อง server · แปลงเกษตร

2. วัตถุประสงค์การเรียนรู้

เมื่อจบ Workshop ผู้เรียนจะทำสิ่งเหล่านี้ได้ด้วยตัวเอง บนเครื่องของตัวเอง

01 **อธิบาย AI Object Detection ได้**
แยกจาก Classification และ Segmentation ได้

02 **เข้าใจว่า YOLO26 ทำงานอย่างไร**
มองครั้งเดียว ทายกรอบ คลาส confidence

03 **สร้างหน้าเว็บระบบด้วย Claude Code**
แผงควบคุม + หน้าหลังบ้าน /admin · Python + OpenCV + Flask + YOLO26

04 **อ่าน Prompt ได้เข้าใจทุกคำ**
ทุก Prompt มีคำศัพท์อธิบายคำเฉพาะแบบเข้าใจง่าย

05 **เลือกคลาส วาดพื้นที่ และกำหนดช่วงเวลา**
เฝ้าระวังเฉพาะสิ่งที่สนใจ และลดการแจ้งเตือนหลอก

06 **ส่งอีเมลแจ้งเตือนพร้อมหลักฐาน**
ภาพ + คลิป 8 วินาที ผ่าน Gmail App Password ที่ตั้งในหน้าหลังบ้าน

3. กลุ่มผู้เรียนและพื้นฐาน

กลุ่มผู้เรียน

ผู้เริ่มต้นที่อยากใช้ AI ทำระบบจริง

Claude Code เป็นผู้เขียนโค้ด
ผู้เรียนสั่งงาน รันระบบ และตรวจผลลัพธ์

- ✓ เจ้าของกิจการ ร้านค้า และผู้ดูแลอาคาร
- ✓ นักศึกษาและครูสายคอมพิวเตอร์
- ✓ ไม่ต้องเขียนโค้ดเป็นก็เรียนได้

พื้นฐานที่ต้องมี

ติดตั้ง Claude Code + Python มาก่อนเข้าคลาส

- ✓ โน้ตบุ๊ก Mac หรือ Windows (RAM 8 GB+)
- ✓ บัญชี Gmail ที่เปิด 2-Step Verification

ไม่จำเป็นต้องมี

- ✗ GPU — YOLO26n รันบน CPU ได้
- ✗ กล้องจริง — ในคลาสใช้คลิปวิดีโอ

Work Style — สอนทีละ Prompt · เริ่มจากโฟลเดอร์ว่าง + Starter Kit ที่แจกให้

1. **วาง Prompt ใน Claude Code** — Prompt พร้อมใช้ทุกเฟส →
2. **AI เขียนโค้ดให้** — เขียนและแก้ไฟล์ให้ →
3. **ดูผลบนหน้าเว็บ** — เห็นกรอบบนวิดีโอจริง

4. ผลลัพธ์หลังเรียน

ระบบที่ทำงานได้จริงบนเครื่องของผู้เรียน พร้อมต่อยอดกับกล้องจริงผ่าน webcam หรือ RTSP



01

แผงควบคุมตรวจจับคนและรถ

ภาพสดพร้อมกรอบ + ตัวเลขสด 5 ช่อง
จากคลิป เว็บแคม หรือ RTSP

02

เลือกคลาสที่ต้องการ

80 คลาส ชื่อภาษาไทย ค้นหาได้
เลือกคลาสแนะนำของคลิปให้เอง

03

พื้นที่ + ช่วงเวลาเฝ้าระวัง

คลิกวาดพื้นที่บนภาพ · 22:00–06:00
เจอติดกัน 3 เฟรม + พัก 60 วิ

04

หลักฐานทุกเหตุการณ์

snapshot + คลิป 8 วิ (H.264)
ดูย้อนหลังเต็มจอบนแผงควบคุม

05

หน้าหลังบ้าน /admin

มีรหัสผ่าน · ตั้งผู้ส่ง App Password และผู้รับ เก็บในไฟล์ .env

06

อีเมลแจ้งเตือนจริง

Gmail SMTP · ภาพในตัว + ไฟล์แนบ
ซ้อมเป็นไฟล์ .eml ได้

ไฟล์โปรเจกต์ที่ได้กลับไป

`webapp.py` เว็บ Flask 2 หน้า	`detector.py` ระบบตรวจจับ	`static/` แผงควบคุม + /admin	`.env` รหัสหลังบ้าน + อีเมล	`CLAUDE.md` บันทึก + แบบหน้าเว็บ

5. ตารางเวลา

120 นาที — ปูพื้นความเข้าใจสั้น ๆ แล้วใช้เวลาส่วนใหญ่ลงมือสร้างระบบทีละเฟส จนส่งอีเมลแจ้งเตือนได้จริง

25 นาที	10	75 นาที	10
เปิดคลาส + ทฤษฎี	Setup	สร้างระบบบนหน้าเว็บ 4 เฟส	ต่อยอด

0:00–0:10	INTRO	**เปิดคลาส** แนะนำผู้สอน · ดูผลลัพธ์ปลายทาง (อีเมลแจ้งเตือนจริง) · เป้าหมายวันนี้
0:10–0:25	LECTURE	**AI Detection & YOLO26** Detection คืออะไร · YOLO26 คืออะไร ทำไมต้องใช้ ทำงานอย่างไร · ช่วยกันขโมยอย่างไร
0:25–0:35	SETUP · S0	**เครื่องมือ + Setup** Claude Code · Python · OpenCV · Flask · Ultralytics/YOLO26 · ffmpeg · Gmail · แตก Starter Kit + จด “แบบหน้าเว็บ” ลง CLAUDE.md
0:35–0:55	PHASE 1 · P1	**แผงควบคุม + ตรวจจับคน** 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 · P2	**เลือกคลาส** การ์ดขั้น 2 เลือกจาก 80 คลาส ชื่อภาษาไทย · ค้นหา · ปุ่มเลือกเร็ว
1:05–1:25	PHASE 3 · P3	**พื้นที่ + ช่วงเวลา** การ์ดขั้น 3 คลิกวาดพื้นที่บนภาพ · การ์ดขั้น 4 ความมั่นใจ + ช่วงเวลา 22:00–06:00
1:25–1:50	PHASE 4 · P4A–C	**หลักฐาน + อีเมล** P4A ภาพ + คลิป 8 วิ · P4B หน้าหลังบ้าน /admin · สร้าง App Password · P4C ส่งอีเมลแจ้งเตือน
1:50–2:00	BONUS · B1 + B2	**ต่อยอด + ถาม-ตอบ** B1 กล้อง RTSP · B2 ดูจากมือถือในวง Wi-Fi เดียวกัน · Q&A

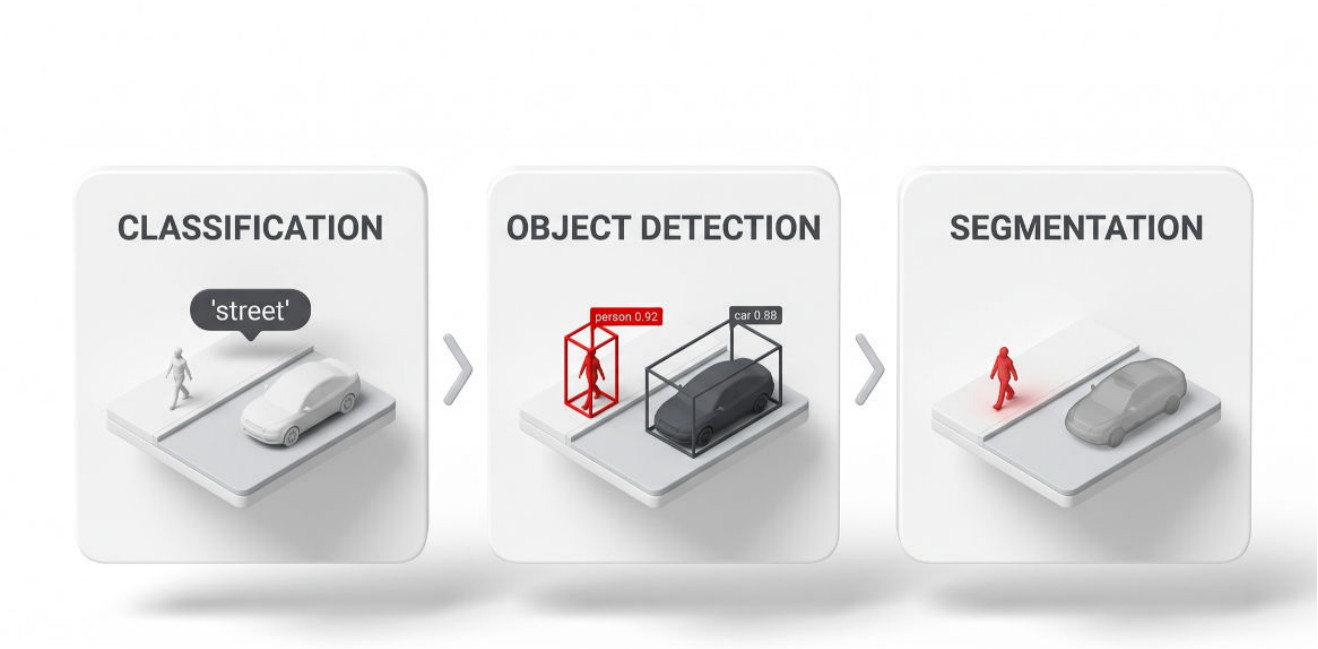
เวลาปรับได้ตามความเหมาะสม — ตารางเวลาเป็นแนวทาง ผู้สอนอาจปรับเวลาแต่ละช่วงให้เหมาะกับความคืบหน้าของผู้เรียน โดยเนื้อหาและผลลัพธ์หลักยังครบถ้วน

S0 · P1 – P4C · B1 · B2 คือรหัส Prompt ของแต่ละช่วง ผู้เรียนคัดลอกไปวางใน Claude Code ได้ทันที · ใต้ทุก Prompt มี “คำศัพท์ในพรอมต์นี้” อธิบายคำเฉพาะแบบเข้าใจง่าย

LECTURE 0:10–0:25 · 15 นาที

AI Detection คืออะไร

Computer Vision มี 3 ระดับ — Workshop นี้ใช้ระดับกลาง คือ **Object Detection** — มีอะไร อยู่ตรงไหน กี่ชิ้น



LEVEL 1
Classification
ภาพนี้คืออะไร

LEVEL 2 · ใช้ในคลาส
Object Detection
มีอะไร อยู่ตรงไหน กี่ชิ้น

LEVEL 3
Segmentation
รูปร่างของวัตถุละเอียดทีละพิกเซล

ผลลัพธ์ของ Detection ต่อ 1 วัตถุ

กรอบ
`(x, y, w, h)` ตำแหน่งในภาพ

คลาส
`person` วัตถุนั้นคืออะไร

ความมั่นใจ
`confidence 0.87`

วิดีโอ = ภาพนิ่งหลายสิบภาพต่อวินาที
ระบบจึงตรวจจับได้ทุกเฟรมแบบ real-time ซึ่งเป็นหัวใจของกล้องที่แจ้งเตือนได้เอง

LECTURE 0:10–0:25 · 15 นาที

YOLO26 คืออะไร ทำงานอย่างไร

YOLO = You Only Look Once — มองภาพครั้งเดียว ทายทุกวัตถุพร้อมกัน จึงเร็วพอสำหรับวิดีโอสด



จากภาพเข้าสู่ผลลัพธ์ ในภาพเดียวจบ

01 · Backbone	→	02 · Neck	→	03 · Head
ดึงลักษณะเด่น ขอบ รูปร่าง พื้นผิว		**รวมลักษณะหลายขนาด** ทั้งของใหญ่และของเล็ก		**ทายผลทันที** กรอบ + คลาส + confidence

รุ่นล่าสุดของ Ultralytics

ออก มกราคม 2026 เร็วบนเครื่องทั่วไปและอุปกรณ์ edge

COCO 80 คลาส

คน รถยนต์ มอเตอร์ไซค์ รถบรรทุก จักรยาน ฯลฯ ใช้ได้ทันทีไม่ต้องเทรน

ทำได้หลายงาน

Detection · Segmentation · Pose · Classification · OBB (กรอบเอียง)

LECTURE 0:10–0:25 · 15 นาที

ทำไมต้อง YOLO26 และใช้รุ่นไหนในคลาส



01 End-to-end ไม่ต้อง NMS
ทายกรอบสุดท้ายได้เลย ไม่ต้องกรองกรอบซ้ำ ขั้นตอนน้อย นำไปใช้ง่าย

02 เร็วบน CPU
ผู้พัฒนารายงานว่า YOLO26n เร็วกว่า YOLO11n บน CPU สูงสุดราว 43%

03 เบา export ง่าย
ตัด DFL ออก ส่งออก ONNX / TensorRT / CoreML / OpenVINO ได้

04 จับวัตถุเล็กดีขึ้น (STAL)
เหมาะกับกล้องวงจรปิดมุมสูงที่คนในภาพดูตัวเล็ก

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

5 ขนาดของ YOLO26 (COCO 640px · ข้อมูล Ultralytics)

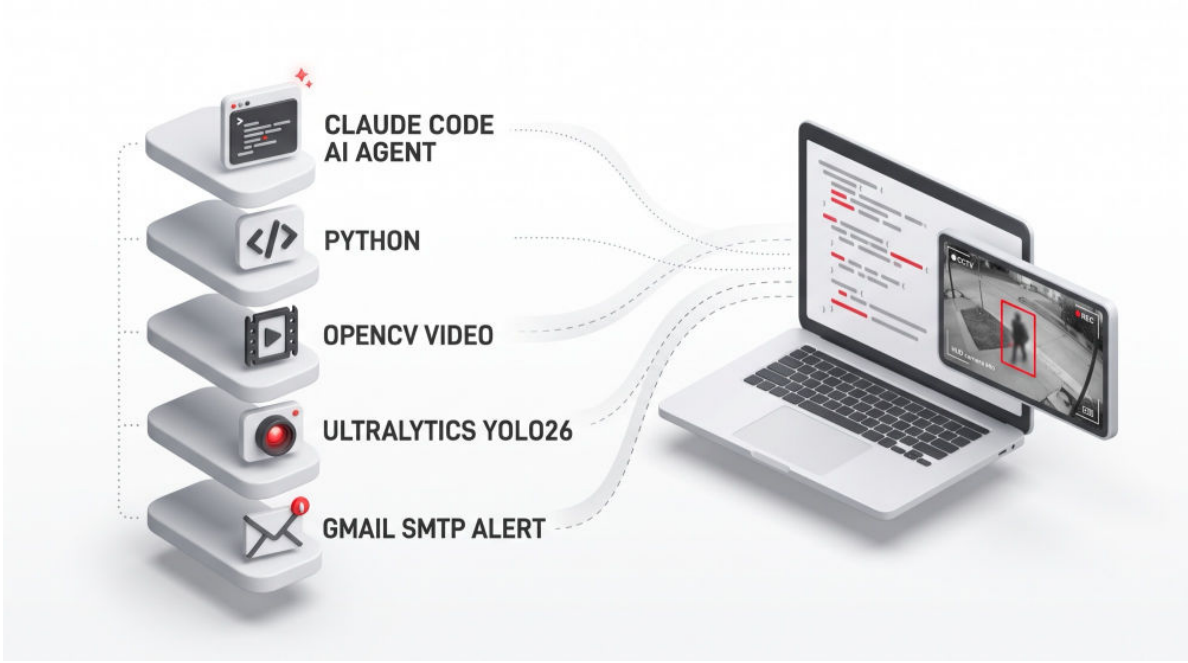
รุ่น	mAP50-95	CPU ONNX (ms)	Params (M)	ในคลาส
YOLO26n	40.9	38.9	2.4	ใช้ในคลาส
YOLO26s	48.6	87.2	9.5	—
YOLO26m	53.1	220.0	20.4	—
YOLO26l	55.0	286.2	24.8	—
YOLO26x	57.5	525.8	55.7	—

SETUP 0:25–0:35 · 10 นาที

>_ Prompt S0

เครื่องมือวันนี้ และทำไมต้องใช้

เตรียมโปรเจกต์: สร้าง `.venv` ติดตั้งไลบรารี (รวม Flask) แตก Starter Kit และสร้าง `CLAUDE.md` ที่จด “แบบหน้าเว็บ” ไว้ ทุกเฟสจึงได้หน้าตาเดียวกัน



เครื่องมือ	บทบาท	ทำไมใช้
Claude Code	AI Agent เขียนโค้ด รันโค้ด แก้ error ให้ในโฟลเดอร์โปรเจกต์	สั่งเป็นภาษาไทย ไม่ต้องเขียนโค้ดเอง
Python 3.10–3.12	ภาษาหลักของ AI	ไลบรารี AI ครบที่สุด
OpenCV	อ่านวิดีโอ/กล้อง วาดกรอบ บันทึกคลิป	ต่อ webcam / RTSP ได้
Ultralytics + YOLO26	“สมอง” ตรวจจับวัตถุ	โมเดลล่าสุด เร็ว ใช้ 3 บรรทัด
Flask	หน้าเว็บ 2 หน้า: แผงควบคุม + หน้าหลังบ้าน /admin	เบา ติดตั้งง่าย กดเลือกแทนการแก้ไฟล์
imageio-ffmpeg	แปลงคลิปเป็น H.264	เปิดได้ทุกเครื่องและมือถือ
Gmail SMTP + App Password	ส่งอีเมลแจ้งเตือน · ตั้งค่าที่หน้าหลังบ้าน /admin	ฟรี ปลอดภัยกว่าใช้รหัสจริง
python-dotenv (.env)	เก็บรหัสหลังบ้านและค่าอีเมลแยกจากโค้ด	รหัสไม่หลุดไปกับโค้ด

Claude Code
วิศวกรที่เราสั่งงาน

Ultralytics
กล่องเครื่องมือ

YOLO26
สมองที่มองเห็น

OpenCV
ตาและมือ

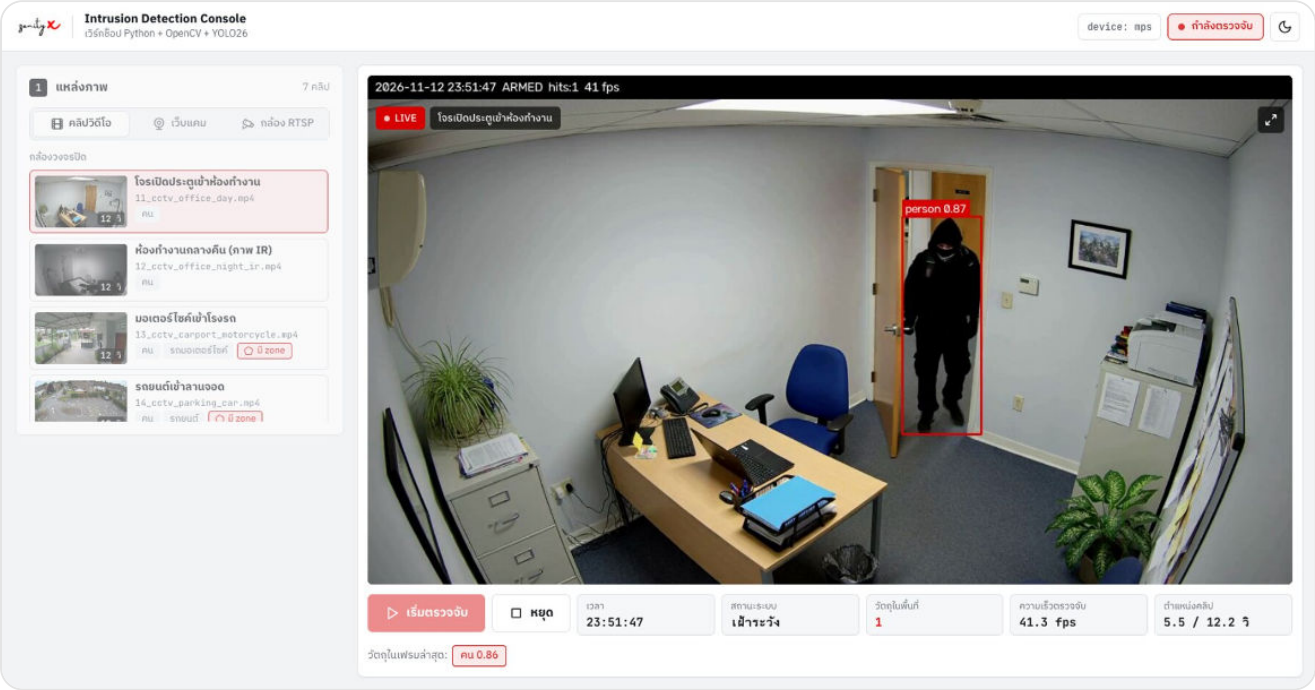
Gmail
ผู้ส่งสาร

PHASE 1 0:35–0:55 · 20 นาที

>_ Prompt P1

แผงควบคุม + ตรวจจับคน

ให้ Claude Code เขียน `detector.py` + `webapp.py` (Flask) เปิดแผงควบคุม **Intrusion Detection Console** ที่ `127.0.0.1:8000` แสดงภาพสดพร้อมกรอบตรวจจับคนทุกเฟรม



แผงควบคุมจริงจาก Phase 1 — คลิป 11_cctv_office_day ภาพสดป้าย LIVE กรอบแดงรอบคน และตัวเลขสด 5 ช่อง

01 การ์ดขั้น 1 แหล่งภาพ
คลิปวิดีโอ (จัดกลุ่มตาม clips.yaml) · เว็บแคม · กล้อง RTSP

02 ภาพสด MJPEG + กรอบแดง
ป้าย LIVE · person 0.87 · โมเดลล็อกไว้ที่ YOLO26n

03 ตัวเลขสด 5 ช่อง
เวลา · สถานะระบบ · วัตถุในพื้นที่ · ความเร็วตรวจจับ · ตำแหน่งคลิป

04 หน้าตาตาม “แบบหน้าเว็บ”
S0 จดสี ฟอนต์ ปุ่ม การ์ดไว้ใน CLAUDE.md ทุกเครื่องได้หน้าตาเดียวกัน

ช่องความเร็วตรวจจับ (fps)
ไฟล์วิดีโอเล่นตามความเร็วจริง ค่าจึงไม่เกิน fps ของคลิป (ราว 24–30) · ต่ำกว่า 15 fps ให้ลดขนาดภาพที่ส่งให้ YOLO เป็น 480 และปิดโปรแกรมอื่น

ชิ้นงานที่ได้

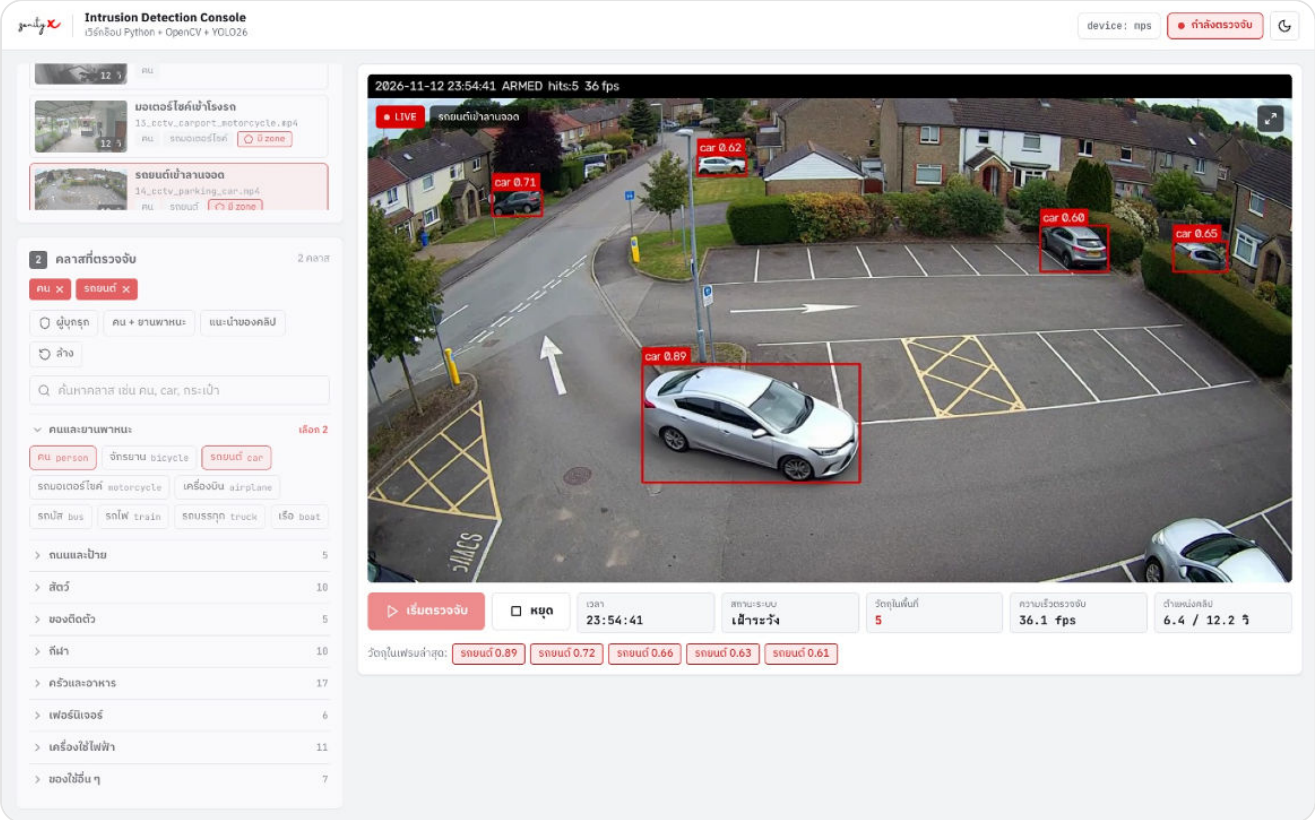
แผงควบคุมตรวจจับคนได้จริง
เลือกคลิป กดเริ่มตรวจจับ เห็นภาพสด กรอบตรวจจับ และตัวเลขสดจนคลิปจบ

PHASE 2 0:55–1:05 · 10 นาที

>_ Prompt P2

เลือกคลาสที่ต้องการตรวจจับ

YOLO26 รู้จัก 80 คลาส — การ์ดขั้น 2 ให้เลือกเฉพาะสิ่งที่ต้องเฝ้าระวังด้วยชื่อภาษาไทย ระบบตรวจเฉพาะคลาสที่เลือก



ผลจริงจากคลิป 14_cctv_parking_car — เลือก คน + รถยนต์ แล้วกรอบรถยนต์ขึ้นทันที

01 80 คลาส ชื่อภาษาไทย
แบ่ง 9 กลุ่ม พับเก็บได้ · ค้นหาได้ทั้งไทยและอังกฤษ

02 ปุ่มเลือกเร็ว
ผู้บุกรุก · คน + ยานพาหนะ · แนะนำของคลิป · ล้าง

03 เลือกตามคลิปให้อัตโนมัติ
กดคลิปแล้วเลือกคลาสที่แนะนำใน clips.yaml ให้เอง

04 ตรวจเฉพาะคลาสที่เลือก
แถว “วัตถุในเฟรมล่าสุด” เป็นชื่อไทย เช่น รถยนต์ 0.90

รถที่จอดอยู่แล้วก็ถูกนับ
เป็นเรื่องปกติของเฟสนี้ — Phase 3 แก้ด้วยพื้นที่ตรวจจับ ให้นับเฉพาะสิ่งที่อยู่ในพื้นที่ที่สนใจ

ชิ้นงานที่ได้

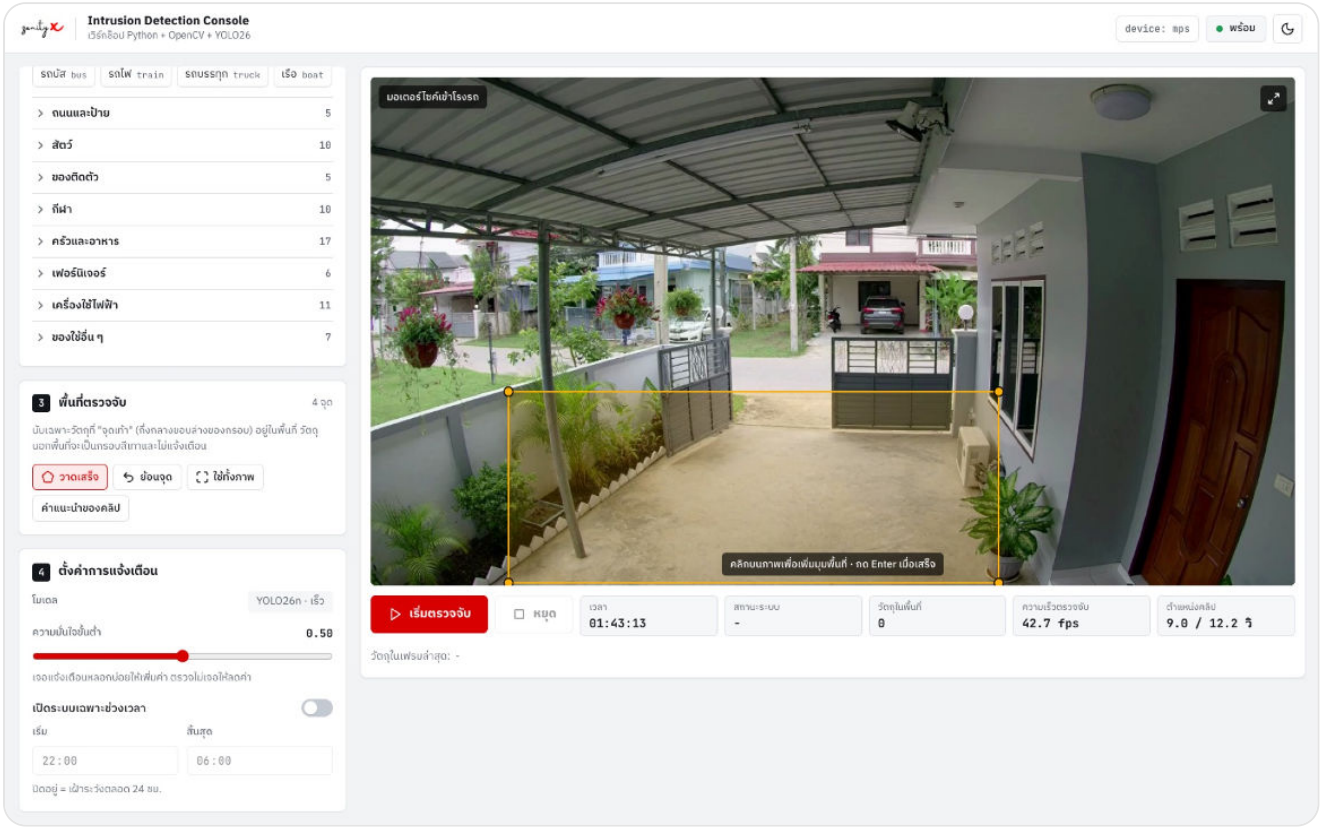
ตรวจจับเฉพาะสิ่งที่เลือก
การ์ดขั้น 2 คลาสที่ตรวจจับ — ปุ่มเริ่มกดไม่ได้จนกว่าจะเลือกอย่างน้อย 1 คลาส

PHASE 3 1:05–1:25 · 20 นาที

Prompt P3

วาดพื้นที่ตรวจจับบนภาพ

ระบบที่ดีเฝ้าเฉพาะ **พื้นที่ที่สนใจ** — การ์ดขั้น 3 ให้คลิกวาดพื้นที่บนภาพเฟรมแรกของคลิป แล้วนับเฉพาะวัตถุที่ **จุดเท้า** อยู่ในพื้นที่



วาดพื้นที่ — คลิกทีละมุมบนภาพเฟรมแรกของคลิป 13_cctv_carport_motorcycle แล้วกด Enter

01 การ์ดขั้น 3 พื้นที่ตรวจจับ
วาดพื้นที่ · ย้อนจุด · ใช้ทั้งภาพ · ค่าแนะนำของคลิป

02 คลิกทีละมุมบนภาพ
อย่างน้อย 3 จุด กด Enter เมื่อเสร็จ · เส้นสีเหลืองอำพัน

03 นับเฉพาะ “จุดเท้า” ในพื้นที่
กึ่งกลางขอบล่างของกรอบ = ตำแหน่งที่ยืนจริงบนพื้น

04 เก็บเป็นสัดส่วน 0–1
ใช้ได้กับวิดีโอทุกความละเอียด · คลิปที่มี zone ใส่ค่าแนะนำให้เอง

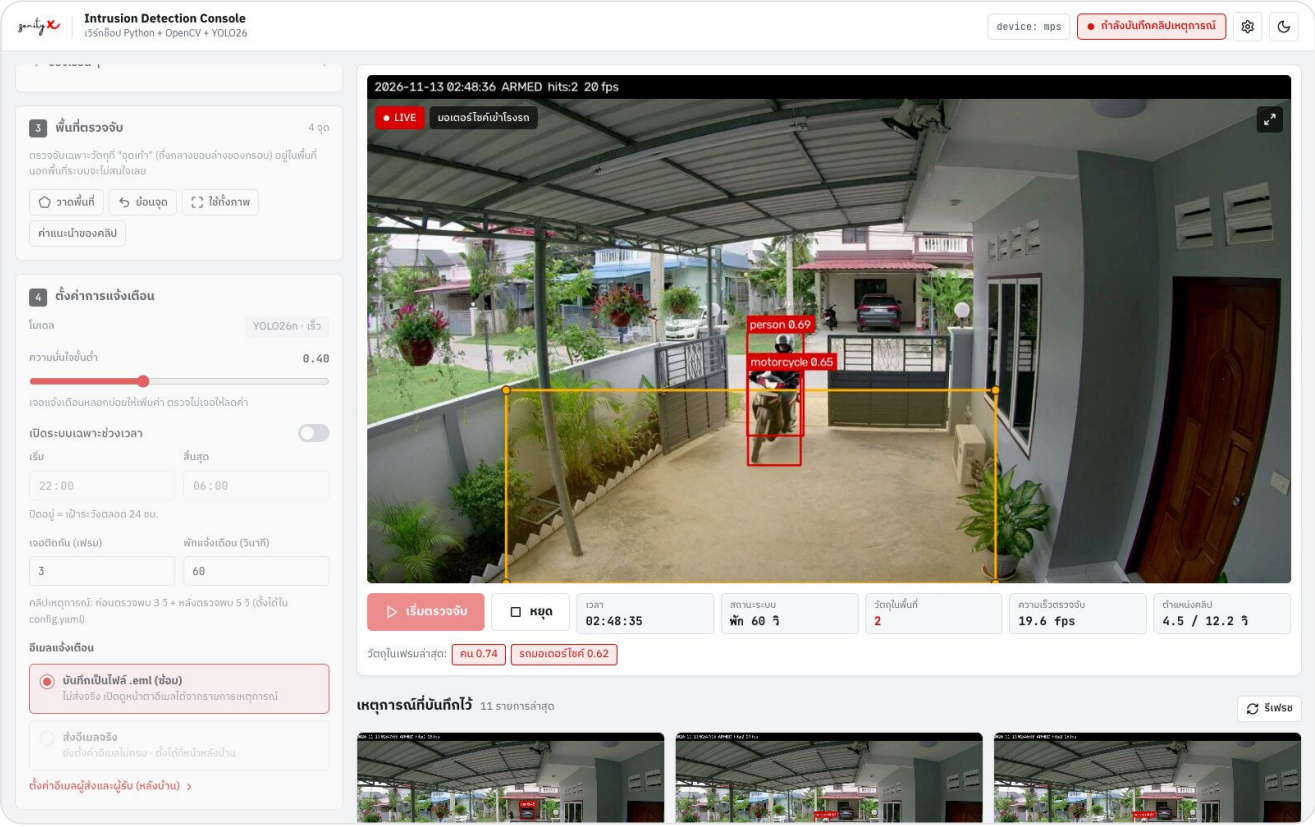
ผู้เรียนลองวาดพื้นที่เอง
ฝึกกับคลิป `13_cctv_carport_motorcycle` · วาดบนพื้นโรงรถ เพราะระบบนับจากจุดเท้า

PHASE 3 1:05–1:25 · 20 นาที

>_ Prompt P3

ช่วงเวลาเฝ้าระวัง + ตั้งค่าการแจ้งเตือน

ระบบตรวจจับ **เฉพาะในกรอบที่วาด** นอกกรอบไม่สนใจเลย และแจ้งเตือนเฉพาะ **ในช่วงเวลาเฝ้าระวัง** — ตั้งค่าในการ์ดขั้น 4 บนแผงควบคุม ไม่ต้องแก้ไฟล์เอง



ผลจริงจากคลิป 13 — ตรวจเฉพาะในกรอบ มอเตอร์ไซค์ในโรงรถกรอบแดง รถฝั่งตรงข้ามนอกกรอบไม่มีกรอบเลย

01 ตรวจเฉพาะในกรอบที่วาด
นอกกรอบไม่แสดง ไม่นับ ไม่เตือน · ยังไม่วาด = ทั้งภาพ

02 การ์ดขั้น 4 ตั้งค่าการแจ้งเตือน
โมเดล YOLO26n (ล็อก) · ความมั่นใจขั้นต่ำ 0.10–0.90

03 ช่วงเวลา 22:00–06:00
ข้ามเที่ยงคืนได้ · ปิดสวิตช์ = เฝ้าตลอด 24 ชม.

04 สถานะ เฝ้าระวัง / นอกเวลา
นอกเวลายังเห็นภาพและกรอบในพื้นที่ แต่ไม่นับ

ทดสอบตอนกลางวัน
ตั้งช่วงเวลาให้ครอบเวลาตอนนี้ หรือปิดสวิตช์ “เปิดระบบเฉพาะช่วงเวลา” · ค่าเริ่มต้นทั้งหมดเก็บใน `config.yaml`

ชิ้นงานที่ได้

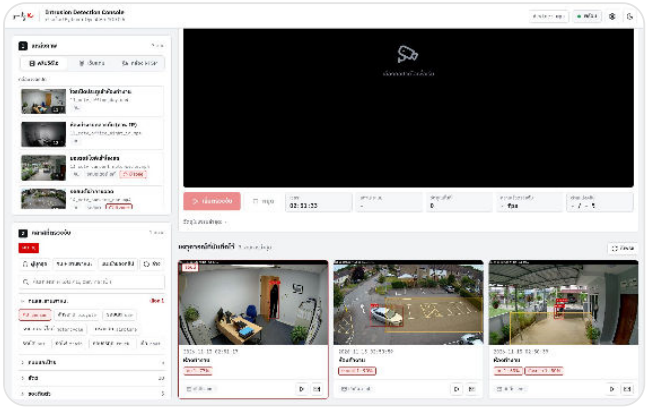
ระบบเฝ้าระวังตามพื้นที่และเวลา
แจ้งเฉพาะคน/รถในพื้นที่ที่สนใจ ในเวลาที่ไม่ควรมีใคร

PHASE 4 · 1:25–1:50 · 25 นาที

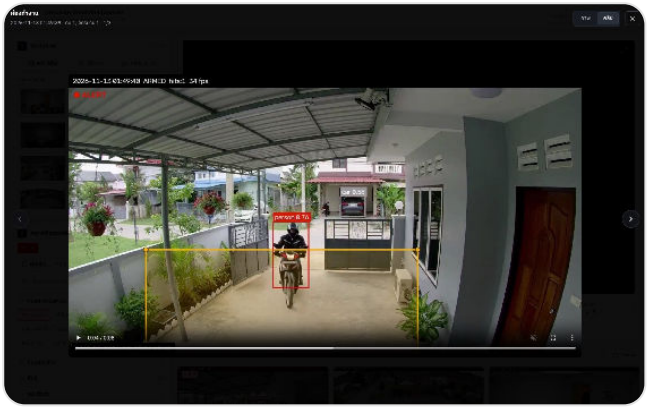
`>_ Prompt P4A`

เก็บหลักฐาน ภาพ + คลิป 8 วินาที

Phase 4 มี 3 Prompt: **P4A** เก็บหลักฐาน → **P4B** หน้าหลังบ้าน /admin → **P4C** ส่งอีเมล · เมื่อพบวัตถุในพื้นที่ ระบบบันทึกภาพและคลิปที่มีช่วง **ก่อนเหตุ** ด้วย แล้วดูย้อนหลังบนแผงควบคุม



เหตุการณ์ที่บันทึกไว้ — การ์ดของรอบนี้กรอบแดง



ดูเต็มจอ แท็บ ภาพ / คลิป · Esc ปิด

ก่อนเหตุ 3 วิ
จาก buffer ย้อนหลัง

⚠ เหตุการณ์

หลังเหตุ 5 วิ
บันทึกต่อจนครบ

ANTI FALSE ALARM

กันแจ้งเตือนหลอก

- เจอติดกัน 3 เฟรม จึงนับเป็นเหตุการณ์
- พักแจ้งเตือน 60 วิ · สถานะขึ้น “พัก N วิ”
- ปรับ 2 ค่านี้ได้ในการ์ดขั้น 4

EVIDENCE

ไฟล์ต่อเหตุการณ์

- snapshot.jpg + clip.mp4 (H.264 ด้วย imageio-ffmpeg)
- บันทึก `events/…/event.json` ทุกเหตุการณ์
- คลิปประทับชื่อกล้อง วันเวลา และป้าย ALERT

ชิ้นงานที่ได้

หลักฐานทุกเหตุการณ์บนแผงควบคุม

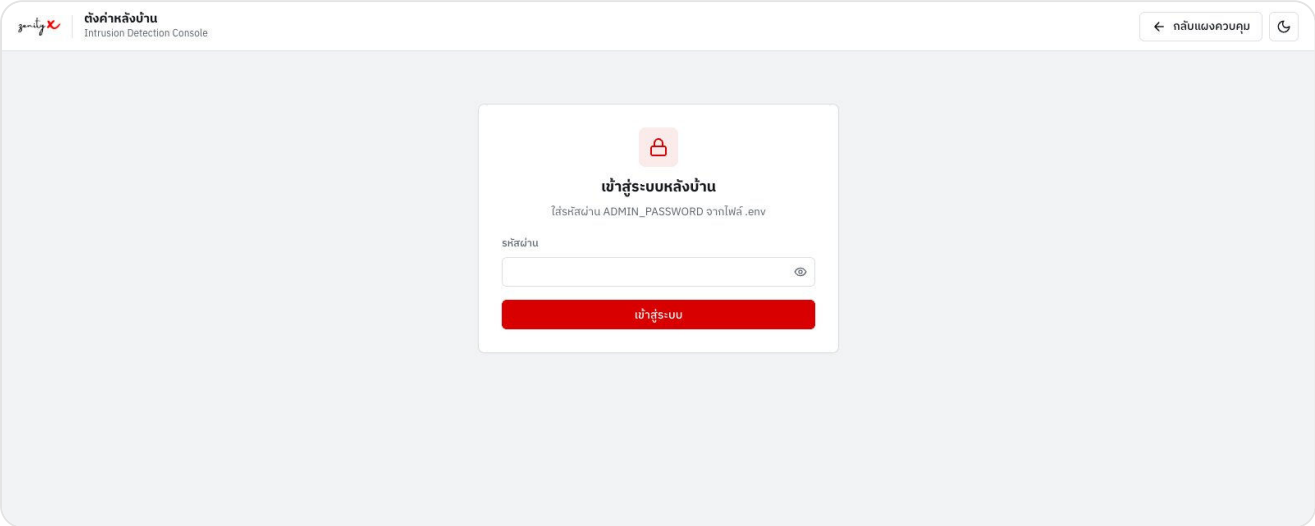
ส่วน “เหตุการณ์ที่บันทึกไว้” ใต้ภาพสด กดภาพเพื่อดูเต็มจอ เล่นคลิป 8 วินาที เปลี่ยนเหตุการณ์ด้วยลูกศร

PHASE 4 1:25–1:50 · 25 นาที

>_ Prompt P4B

หน้าหลังบ้าน /admin ตั้งค่าอีเมล

แยกค่าที่เป็นความลับออกจากแผงควบคุม — หน้าหลังบ้านต้องใส่รหัสก่อนเข้า ไว้ตั้ง **บัญชีผู้ส่ง** App Password และ **ผู้รับการแจ้งเตือน** แล้วบันทึกลงไฟล์ `.env`



หน้าเข้าสู่ระบบหลังบ้านที่ 127.0.0.1:8000/admin — ใช้รหัส ADMIN_PASSWORD จากไฟล์ .env

ผู้เรียนทำเอง — ตั้งรหัสเข้าหน้าหลังบ้าน

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 กดไอคอนเฟืองมุมขวาบนของแผงควบคุมได้

4. **ใส่รหัสแล้วกด “เข้าสู่ระบบ”**
   ยังไม่ตั้งรหัส จะเห็นการ์ดบอกขั้นตอน + ปุ่ม **ตรวจอีกครั้ง**

ห้ามวางรหัสในแชท AI
พิมพ์รหัสในไฟล์ .env และหน้าหลังบ้านด้วยตัวเองเท่านั้น

ปลอดภัยตั้งแต่ต้น
session cookie แบบ HttpOnly · ผิด 5 ครั้งรอ 60 วิ · ไม่แสดงรหัสกลับบนหน้าเว็บ

PHASE 4 1:25–1:50 · 25 นาที

สร้าง Google App Password ทีละขั้น

ทำเองบนบัญชี Gmail ของตัวเอง ระหว่าง P4B กับ P4C — ระบบใช้รหัสนี้แทนรหัส Gmail จริง

1 เปิดการยืนยันแบบ 2 ขั้นตอน
myaccount.google.com/signinoptions/twosv
เปิด **2-Step Verification** ถ้ายังไม่เปิด

2 เปิดหน้า App Passwords
myaccount.google.com/apppasswords
(ล็อกอินอีกครั้งถ้าถูกถาม)

3 ตั้งชื่อแอป แล้วกดสร้าง
เช่น `YOLO26 Alert` → กด **สร้าง**

4 คัดลอกรหัส 16 ตัว
รหัสแสดงครั้งเดียว · เว้นวรรคหรือไม่ก็ได้

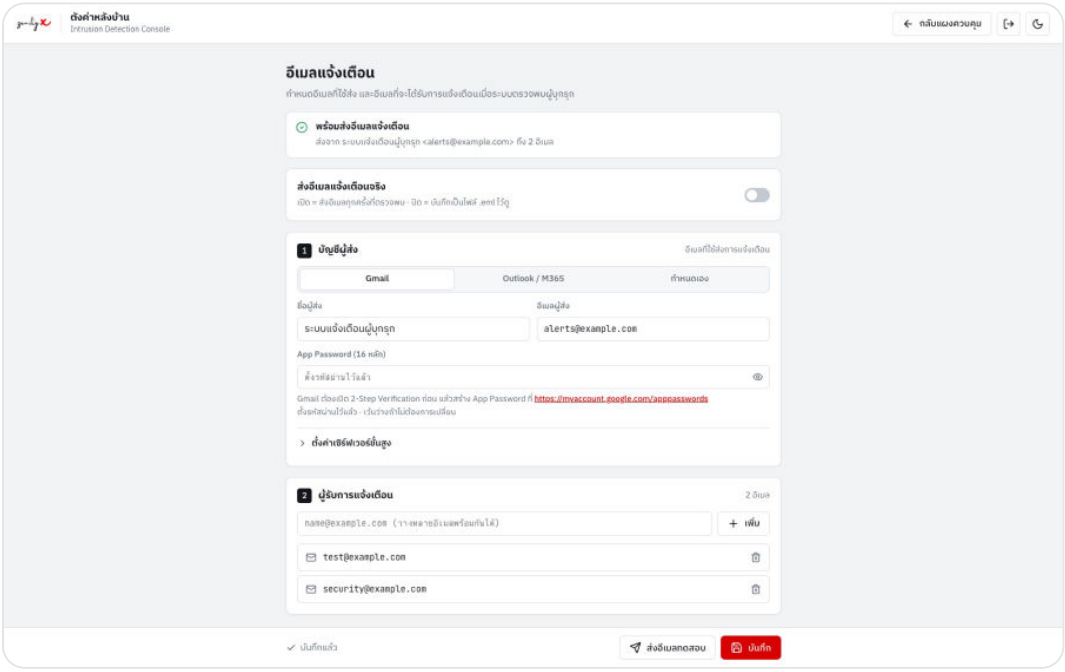
5 กรอกที่หน้าหลังบ้าน /admin (การ์ดบัญชีผู้ส่ง)
อีเมลผู้ส่ง · App Password → เพิ่มผู้รับ → กด **บันทึก**

6 กดส่งอีเมลทดสอบ
กดปุ่ม **ส่งอีเมลทดสอบ** ที่แถบล่างของหน้าหลังบ้าน แล้วเช็คกล่องจดหมาย

ห้ามวางรหัสในแชท AI
กรอกที่หน้าหลังบ้าน /admin เท่านั้น ระบบเก็บใน .env ไม่แสดงรหัสกลับ

ไม่เห็นเมนู App Password
ยังไม่เปิด 2-Step หรือเป็นบัญชีองค์กร (Google Workspace) ที่ปิดเมนูนี้ไว้

ใช้รหัส Gmail ปกติไม่ได้
ต้องเป็น App Password — เจอ `535` ให้สร้างรหัสใหม่แล้วกรอกที่หน้าหลังบ้าน



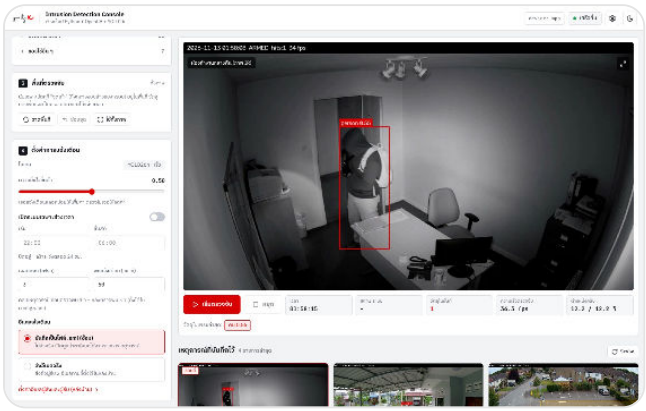
หน้าหลังบ้าน (ขั้น 5–6) — การ์ดบัญชีผู้ส่ง + ผู้รับการแจ้งเตือน ค่าที่บันทึกอยู่ในไฟล์ .env

PHASE 4 1:25–1:50 · 25 นาที

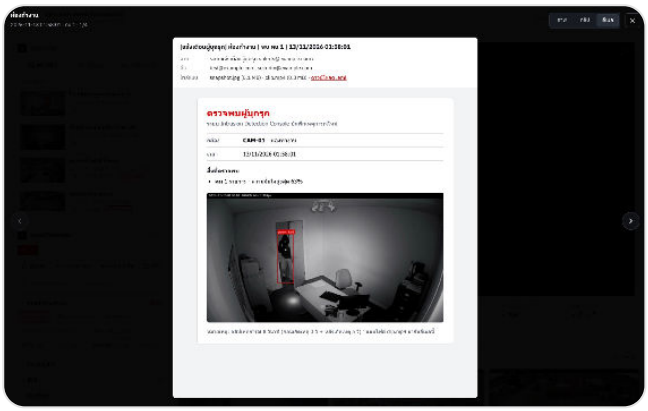
>_ Prompt P4C

ส่งอีเมลแจ้งเตือนจากเหตุการณ์

ต่อระบบเหตุการณ์เข้ากับอีเมลที่ตั้งไว้ในหน้าหลังบ้าน — ส่งเมื่อบันทึกคลิปเสร็จ พร้อมภาพในตัวและไฟล์แนบ หรือซ้อมเป็นไฟล์ `.eml` ก่อน



การ์ดขั้น 4 — บันทึก .eml (ซ้อม) / ส่งอีเมลจริง



แท็บ อีเมล ในหน้าต่างดูเหตุการณ์

CONSOLE

แผงควบคุม

- การ์ดขั้น 4 เลือก “บันทึกเป็นไฟล์ .eml (ซ้อม)” หรือ “ส่งอีเมลจริง”
- “ส่งอีเมลจริง” กดได้เมื่อหลังบ้านตั้งค่าครบ
- ไอคอนเฟืองมุมขวาบนพาไปหน้าหลังบ้าน
- การ์ดเหตุการณ์แสดงสถานะอีเมล + แท็บ “อีเมล”

EMAIL

อีเมลแจ้งเตือน

- หัวเรื่อง [แจ้งเตือนผู้บุกรุก] สถานที่ · สิ่งที่พบ · เวลา
- เนื้อหา HTML ภาษาไทย ภาพ snapshot ในตัว
- แนบ snapshot.jpg + clip.mp4 (ไม่เกิน 20 MB)
- เก็บสำเนา email.eml ทุกเหตุการณ์

ชิ้นงานที่ได้

อีเมลแจ้งเตือนถึงกล่องจดหมายจริง

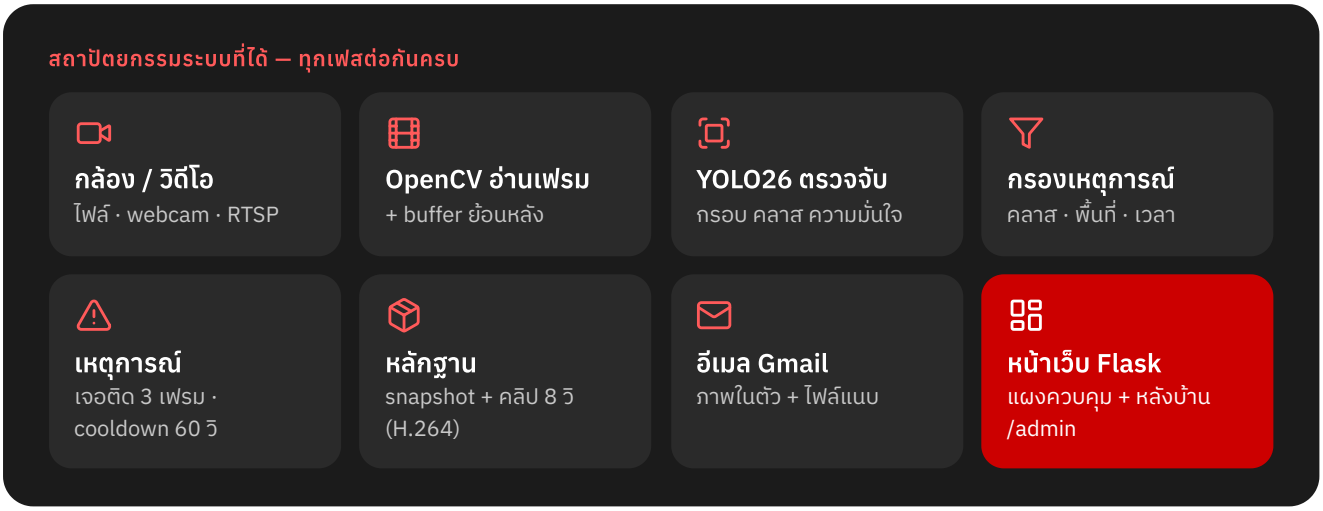
ภาพเหตุการณ์ในตัวอีเมล + คลิป 8 วินาทีเป็นไฟล์แนบ ส่งไม่สำเร็จระบบยังตรวจจับต่อ และบอกสาเหตุเป็นภาษาไทย

BONUS + Q&A · 1:50–2:00 · 10 นาที

>_ Prompt B1 + B2

ต่อยอด และถาม-ตอบ

ทบทวนระบบที่ได้ — แผงควบคุมเป็นศูนย์กลางควบคุมและดูผล หน้าหลังบ้าน /admin เก็บค่าอีเมล แล้วผู้สอนสาธิตการต่อยอดกับกล้องจริงและมือถือในบ้าน



BONUS B1

ต่อกล้องจริง

RTSP · URL กล้องเก็บใน .env (CAMERA_URL) · ต่อใหม่เองเมื่อหลุด

BONUS B2

ดูจากมือถือในบ้าน

เปิดแผงควบคุมจากมือถือใน Wi-Fi เดียวกัน ล็อกอินด้วย ADMIN_PASSWORD

Q&A

ถาม-ตอบ

ทบทวนระบบที่ได้ และแนวทางต่อยอดกับสถานที่จริงของผู้เรียน

แก้ปัญหาที่พบบ่อย

อาการ	สาเหตุ	วิธีแก้
`ModuleNotFoundError: ultralytics`	รันด้วย Python นอก .venv	ใช้ `.venv/bin/python` (Windows `.venv\Scripts\python`)
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”
ไม่มีเหตุการณ์เลย	อยู่นอกช่วงเวลา (สถานะ “นอกเวลา”)	ปิดสวิตช์ช่วงเวลา หรือตั้งให้ครอบเวลาตอนนี้
เข้าหน้าหลังบ้านไม่ได้	ยังไม่ตั้ง ADMIN_PASSWORD	ใส่รหัสในไฟล์ .env แล้วกด “ตรวจอีกครั้ง”
`535 Username and Password not accepted`	ใช้รหัส Gmail ปกติ / พิมพ์ผิด	สร้าง App Password ใหม่ กรอกที่หน้าหลังบ้าน /admin

7. ขอบเขตการสอน

กำหนดชัดตั้งแต่ต้นว่า 2 ชั่วโมงนี้สอนอะไร และอะไรอยู่นอกขอบเขต

✓ อยู่ในขอบเขต

- พื้นฐาน AI Object Detection และ YOLO26
- ใช้ Claude Code สร้างแผงควบคุม + หน้าหลังบ้านทีละเฟส
- ตรวจจับจากไฟล์วิดีโอ webcam และ RTSP
- เลือกคลาส วาดพื้นที่ และกำหนดช่วงเวลา
- หลักฐานภาพ + คลิป 8 วินาที
- อีเมลแจ้งเตือนผ่าน Gmail App Password ตั้งที่หน้าหลังบ้าน

✕ นอกขอบเขต

- การเทรนโมเดลด้วยข้อมูลของตัวเอง (ใช้โมเดล COCO 80 คลาสที่พร้อมใช้)
- การติดตั้งกล้องและเดินระบบเครือข่ายจริง
- การนำระบบขึ้น server หรือ cloud
- การเขียนโค้ดเชิงลึกด้วยตัวเอง

8. การประเมินผลในคลาส

ประเมินจากระบบที่รันได้จริงบนเครื่องของผู้เรียน — ไม่มีข้อสอบ

- ✓ แผงควบคุมแสดงภาพสด กรอบ และตัวเลขสด 5 ช่อง บนคลิปทดสอบได้
- ✓ เลือกคลาสได้ และระบบตรวจเฉพาะคลาสที่เลือก
- ✓ วาดพื้นที่บนภาพ และเฝ้าเฉพาะช่วงเวลาที่ตั้ง (เฝ้าระวัง / นอกเวลา)
- ✓ เหตุการณ์มีภาพ + คลิป 8 วินาที ดูย้อนหลังเต็มจอบนแผงควบคุมได้
- ✓ เข้าหน้าหลังบ้าน /admin ได้ และปุ่ม “ส่งอีเมลทดสอบ” ส่งถึงกล่องจดหมายจริง

หลักความปลอดภัยของรหัส

รหัสอยู่ใน .env
ADMIN_PASSWORD + App Password แยกจากโค้ด

ไม่วางในแชท AI
ผู้เรียนพิมพ์รหัสในไฟล์ .env และหน้าหลังบ้านเอง

ยกเลิกได้ทุกเมื่อ
ลบ App Password ได้ที่หน้าเดิม

9. สิ่งที่ผู้เรียนต้องเตรียม

เตรียมให้พร้อมก่อนวันเรียน เพื่อให้ทุกช่วงของ 2 ชั่วโมงได้ลงมือสร้างระบบจริง

1 **โน้ตบุ๊ก Mac หรือ Windows (RAM 8 GB+)**
ไม่ต้องมี GPU — YOLO26n รันบน CPU ได้

2 **ติดตั้ง Claude Code และล็อกอินแล้ว**
แพ็กเกจ Pro ขึ้นไป

3 **Python 3.10–3.12**
ติดตั้งให้เรียบร้อยก่อนเข้าคลาส

4 **บัญชี Gmail ที่เปิด 2-Step Verification**
ใช้สร้าง App Password ในช่วง Phase 4

5 **ดาวน์โหลด Starter Kit จาก Hub**
`yolo26-starter-kit.zip` (21 MB) วางในโฟลเดอร์โปรเจกต์

6 **อินเทอร์เน็ต**
สำหรับติดตั้งไลบรารีและดาวน์โหลดโมเดล

ไม่ต้องมีกล้องจริง
ใช้คลิปทดสอบ แล้วต่อกล้องจริงได้ภายหลัง

ไม่ต้องเขียนโค้ดเป็น
Claude Code เขียนโค้ดให้ ผู้เรียนสั่งงาน

Starter Kit — คลิปทดสอบ 7 คลิป + videos/clips.yaml (ชื่อไทย คลาสและพื้นที่แนะนำ) + โลโก้ใน brand/

`11_cctv_office_day` โจรเปิดประตูเข้าห้องทำงาน	`12_cctv_office_night_ir` ห้องทำงานกลางคืน (ภาพ IR)
`13_cctv_carport_motorcycle` มอเตอร์ไซค์เข้าโรงรถ	`14_cctv_parking_car` รถยนต์เข้าลานจอด
`01_door_entry` คนเปิดประตูเข้ามา	`02_hallway_door` คนเดินมาที่ประตู
`04_office_wide` ออฟฟิศมุมกว้าง	

10. ผู้สอน



INSTRUCTOR

เอกรัตน์ ใจชนะ (นิว)

AI Instructor, ZenityX

จุดเชื่อมกับ Workshop นี้

รางวัลชนะเลิศด้านนวัตกรรม คือผลงาน **“ระบบกล้องวงจรปิดตรวจจับอัจฉริยะ”** — Workshop นี้จึงสอนจากประสบการณ์ทำระบบนี้จริง

รางวัลและผลงาน

รางวัลชนะเลิศการนำเสนอบทความวิจัย แบบบรรยาย ระดับดีเด่น อันดับที่ 1
AUCC 2026 · Track Information Technology
การประชุมวิชาการระดับปริญญาตรีด้านคอมพิวเตอร์ภูมิภาคเอเชีย ครั้งที่ 14

รางวัลชนะเลิศ ประเภท Oral Presentation (ด้านนวัตกรรม)
ผลงาน “ระบบกล้องวงจรปิดตรวจจับอัจฉริยะ” — การประกวดผลงานสหกิจศึกษาและการศึกษาเชิงบูรณาการกับการทำงาน (CWIE) ดีเด่น เครือข่ายพัฒนา CWIE ภาคเหนือตอนล่าง

รางวัลระดับดีเด่น Best of Track
Track Information Technology — การนำเสนอผลงานวิจัย

PUBLICATION

Chaiwannakoop, K., **Jaichana, A.**, & Thunjai, S. (2026). *Home Visit Information System for Student Behavior Management.* Rattanakosin Journal of Science and Technology (RJST), 8(2), 152–166.

11. ปิดท้าย

หลังจบ 2 ชั่วโมง

ผู้เรียนกลับไปพร้อมระบบกล้องวงจรปิดอัจฉริยะ ที่ตรวจจับคนและรถ แล้วส่งหลักฐานถึงอีเมลได้เอง

YOLO26
ตรวจจับคนและรถ
แบบ real-time

คลาส · พื้นที่ · เวลา
เฉ้าเฉพาะสิ่งที่สนใจ

ภาพ + คลิป 8 วิ
หลักฐานทุกเหตุการณ์

อีเมลแจ้งเตือน
ผ่าน Gmail
App Password



Workshop 2 ชั่วโมง · 120 นาที
สอนทีละ Prompt · เริ่มจากโฟลเดอร์ว่าง
รันดูผลได้ทันทีทุกเฟส

Claude Code + YOLO26
Python · OpenCV · Flask · Ultralytics
imageio-ffmpeg · Gmail SMTP

ออกแบบหลักสูตรโดยสถาบัน ZenityX

ช่องทางติดต่อ

โทร
062-262-9644

อีเมล
admin@zenityxai.com