AI-Powered Surveillance: Step-by-Step Build, Run & Test Guide

Generated: 2025-08-23 11:17:21

PREREQUISITES

1) OS: Windows 10/11, Ubuntu 22.04+, or macOS 12+
2) Install: Python 3.10+, pip, Git, and (optional) CUDA-capable GPU + CUDA Toolkit.
3) Create venv and install deps:

```
   python -m venv .venv && source .venv/bin/activate  (Windows: .venv\Scripts\activate)
   pip install -r requirements.txt
```

PROJECT LAYOUT

```
ai_surveillance/
  requirements.txt
  src/ ... (pipeline code)
  streamlit_app.py (dashboard)
  make_synth_video.py (synthetic demo clip)
  train_autoencoder.py (optional small CAE)
  data/samples/demo_synth_abandon.mp4 (auto-generated)
```

STEP 1 — (Optional) Get public datasets

- Avenue / UCSD Pedestrians. Place videos under data/.
- For CAE training, export normal frames to data/normal_frames/*.jpg

STEP 2 — Verify OpenCV

```
python - <<'PY'
import cv2; print('OpenCV', cv2.__version__)
PY
```

STEP 3 — (Optional) YOLOv5

- Auto-tries torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)
- If fails, fallback HOG+color detector works on the demo clip.

STEP 4 — Generate demo video (already done)

```
python make_synth_video.py
```

STEP 5 — Run pipeline

```
python src/main.py --video data/samples/demo_synth_abandon.mp4 --save
```

Outputs:

```
  outputs/annotated.mp4
  outputs/alerts.csv
```

STEP 6 — Dashboard

```
streamlit run streamlit_app.py
```

- Enter video path and run. Review alerts & playback.

STEP 7 — Tune thresholds (src/config.yaml)

- loitering.seconds_threshold
- abandonment.seconds_threshold, radius_pixels
- speed_anomaly.speed_threshold_mps

STEP 8 — Train toy Conv Autoencoder (optional)

- Put normal frames into data/normal_frames/
- python train_autoencoder.py -> models/cae_small.pt

STEP 9 — Deliverables mapping

A) Detection models: YOLOv5 (pretrained) + rules; optional CAE weight.
B) Code: all scripts in src/ + apps.
C) Dashboard: streamlit_app.py; export alerts CSV + annotated video.
D) Bonus Synthetic: make_synth_video.py clip with loitering, drop & sprint.

TESTING CHECKLIST

- Expect loitering, abandonment, speed alerts on demo video.
- Try a real clip; adjust thresholds as needed.
- Compare alerts.csv across versions for regressions.

TROUBLESHOOTING

- Torch install: match CUDA version from pytorch.org.
- No YOLO detections: first run needs internet; otherwise fallback works on demo.
- FPS=0 in some videos: code uses 30 FPS default; adjust baseline in config.