ITAI-1378-Computer Vision-AI

THEFT PREVENTION using a Computer Vision Solution

Tier: 3 - Motion Analysis for Real-Time Theft Detection

AI-Powered Security for Museums Using Real-Time Behavior Analysis

Presented By

Khalida Bestani

Malcolm Richardson

R. Brandon Thompson

Problem

- Museums lost over $5 billion last year due to thefts and damage.

Traditional security systems are:

- Too expensive
- Cannot monitor every exhibit
- React too slowly to prevent loss
- Human monitoring leads to fatigue, blind spots, and delayed response.

For that:

- Museums need an affordable, scalable, real-time security system.
- Goal: Detect suspicious behavior BEFORE damage or theft occurs.
- Protect artifacts that are priceless and irreplaceable.

Our Solution

A real-time Computer Vision Theft Prevention System:

- AI-powered computer vision system for museum security
- Detects suspicious behavior in real-time: approaching artifacts, reaching, and bag interactions
- Alerts authorities immediately when a potential theft is detected
- Uses YOLOv8 object detection on people, hands, bags, and artifacts
- Multi-source dataset training ensures robust detection

Key System Capabilities

- Track suspects across rooms
- Understand motion patterns, intent, and behavior
- Handle 20–30 FPS live video on 4K cameras

Technical Approach

Perfect for museum theft prevention, where detection speed and accuracy matter more than complex multi-modal models.

Model

YOLOV8 (ULTRALYTICS)

PRETRAINED + FINE-TUNED

Framework

PYTORCH
OPENCV

ROBOFLOW API

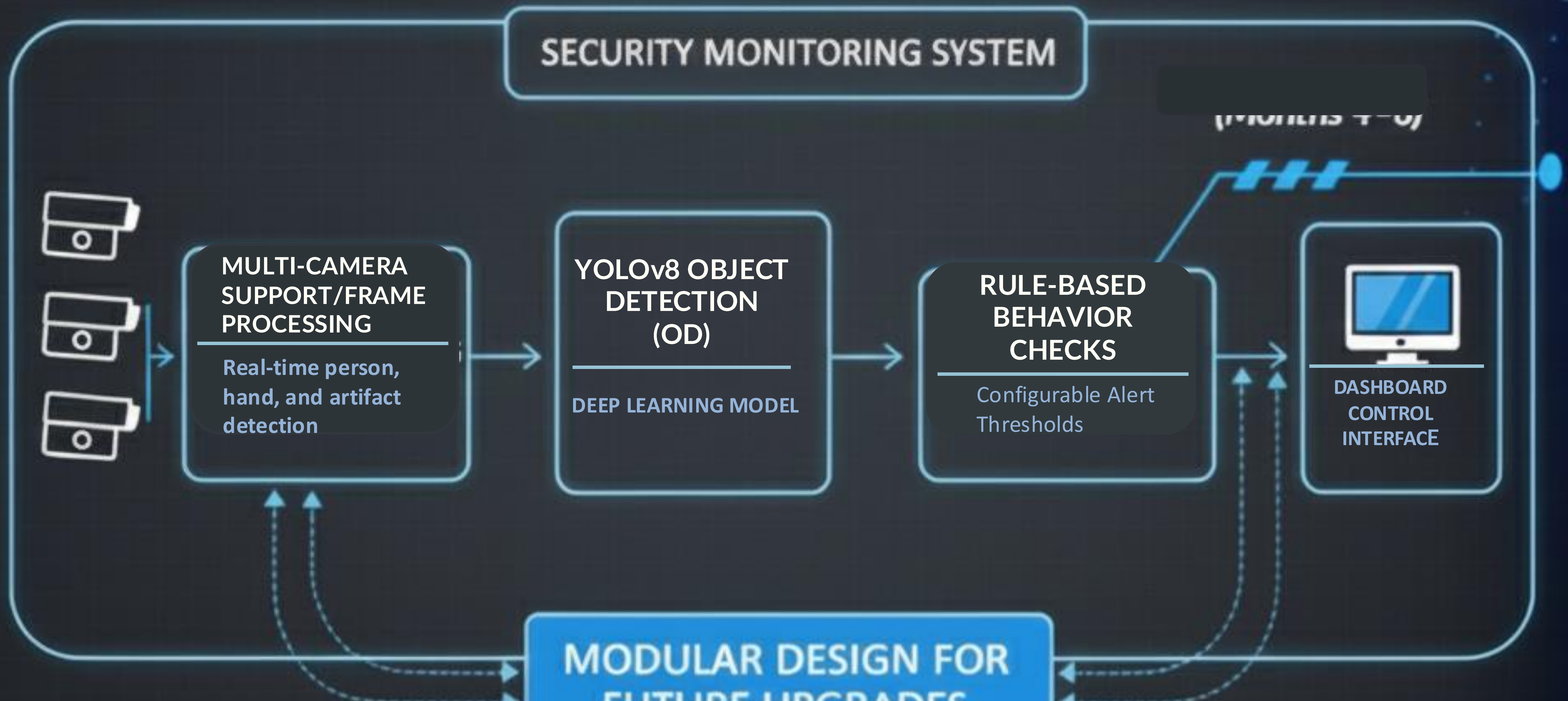
Technical Approach

Pipeline:

1. Video capture (live or recorded)
2. Frame extraction (OpenCV, 20–30 FPS)
3. YOLOv8 object detection
4. Suspicious behavior rule-check (distance, hand/object interactions)
5. Alert generation (console/dashboard)

ARCHIITECTURE DIAGRAM

- Multi-Camera Support
- Configurable Alert Thresholds
- Real-time person, hand, and artifact detection
- Modular Design for Future Upgrades



Testing, pilot, deployment & support

Dataset & Preprocessing

DATASETS

- UCF-Crime — real suspicious behavior
- Shoplifting Video Dataset — annotated shoplifting videos
- DCSASS — theft, robbery, vandalism
- LeapGestRecog — hand gestures/poses
- MoMA Collection — artifacts images + metadata
- Hajj & Umrah Crowd — normal crowd movements
- Roboflow Face Detection — people/faces

PREPROCESSING

- Automatic downloads via Kaggle + Roboflow
- Unzipping & organizing YOLOv8-ready format
- Merge datasets for comprehensive detection
- Train/validation/test split
- Frame extraction at 30 FPS

LIVE DEMO

- Run your system live in front of class! Most important part!

Results

Primary Metric:

- Recall = 76.39% → important to avoid missing real thefts

Secondary Metrics:

- Precision = 88.85% → when the model detects suspicious activity, it is usually correct
- mAP50 (overall) = 82.54% → strong detection accuracy
- Real-time FPS = 20–30 on T4 GPU → suitable for live CCTV monitoring
- mAP50-95 = 53% → moderate performance at stricter detection thresholds

Notes:

- The model is reliable in detecting humans
- Suspicious behavior detection is harder (some misses)

Results

Confusion Matrix Observations:

- Human class: High recall (0.809) & high precision (0.905) → very reliable

- Suspicious behavior: Lower recall (0.719) → some theft events are missed

- Errors mostly occur when:
  - Actions are subtle or partially blocked
  - Low-light or crowded scenes
  - Small-sized humans in the frame

Overall:
- Model is consistent
- Model is real-time capable and accurate for most scenarios

Key Learnings

- Combining multiple datasets increases robustness.
- AI's success depends on having clean, high-volume and unbiased training data.
- The formal process required for quickly gathering real-world threat intelligence and use to retrain and update the AI models was made obvious throughout the creation of this model.
- The necessary historical data (e.g., access to updated and trainable datasets) for training was unexpectedly slow due to issues using API keys.

Future Work

- Add object/person tracking for better temporal analysis
- Add artifact proximity alerts using boundaries/zones
- Improve low-light performance via augmentation
- Build a dashboard for alert monitoring
- Deploy on edge devices (Jetson Nano/Orin)
- Expand with temporal action recognition in the future

Acknowledgments

Special Thanks

- Professor Patricia McManus
- Teammates Bestani Khalida
  Malcolm Richardson
  R. Brandon Thompson