Vigilance Nexus: An AI-Driven Theft Detection Module for Real-Time Video Surveillance

Mithun M[1], Meghana M[2], Deepika K N[3] , Mrs. Leesa Banu[4]

[1]*Computer Science and Engineering , Bangalore Technological Institute, Bangalore , India*
[2] *Computer Science and Engineering , Bangalore Technological Institute, Bangalore , India*
[3] *Computer Science and Engineering , Bangalore Technological Institute, Bangalore , India*
[4] Computer Science and Engineering , Bangalore Technological Institute, Bangalore , India

---***---

Abstract - This work describes the theft-detection module of the AI-based video surveillance system called Vigilance Nexus, built for real-time shoplifting/bag-snatching prevention and unauthorized possession detection, among others. It combines spatio-temporal action recognition (CNN + LSTM) and resilient object detection and multi-object tracking (YOLO + DeepSORT), so suspicious human–object interactions can be spotted and ownership transitions traced over time. Preprocessing and data augmentation pipelines enhance resistance to crowded frames, occlusion, and illumination variation. Time-stamped alarms and incident logging for downstream analysis enable quick security response and long-term crime pattern extraction. It is a modular design compatible with pre-existing CCTV/IP camera installations, ensuring low-latency usage and scalable field deployment.

Key Words: Theft detection, shoplifting, CNN+LSTM, YOLO, DeepSORT, spatio-temporal action recognition, video surveillance.

1.INTRODUCTION

Theft in public and commercial spaces remains a widespread problem that often involves subtle, sequential actions. Traditional CCTV systems rely heavily on manual monitoring and post-incident review, which is prone to delays and human error [1]. Vigilance Nexus augments conventional surveillance by adding automated theft detection that reasons over temporal sequences and object ownership[6], transforming surveillance from reactive to proactive monitoring. This paper isolates and documents the theft-detection module of Vigilance Nexus, describing its objectives, architecture, algorithms, and evaluation approach.

2. PROBLEM STATEMENT

Theft in retail and public environments remains a pervasive issue, often characterized by subtle and sequential human–object interactions such as concealing merchandise, bag-snatching, or unauthorized possession. Traditional CCTV systems rely heavily on human operators for real-time monitoring and post-incident review, which are inherently prone to fatigue, delays, and oversight. Frame-level object detectors, although effective at identifying individuals and items, are insufficient for theft detection as they cannot infer temporal intent, ownership transitions, or suspicious behavioral patterns [12], [13]. This results in a high rate of false positives and missed theft events, especially in crowded, occluded, or low-light scenarios [7], [15].

A robust theft detection system therefore requires **temporal reasoning** across sequences of frames, integrating object detection, identity preservation, and action recognition to differentiate normal customer behavior from malicious activities. The challenge lies in designing a system that is both accurate and efficient—capable of real-time operation with minimal false alarms, scalable across diverse retail environments, and resilient to environmental factors such as occlusion, lighting variations, and behavioral ambiguity. Vigilance Nexus addresses these shortcomings by combining **YOLO-based object detection**, **DeepSORT multi-object tracking**, and **CNN+LSTM temporal action recognition**, transforming surveillance from reactive incident review into proactive theft prevention.

3. OBJECTIVES

The primary objective of this project is to design and implement a robust **theft detection module** capable of identifying common retail theft behaviors, such as shoplifting, bag-snatching, and unauthorized possession of store items, in **real time**. Specific goals include:

- **High-accuracy detection:** Minimize false positives and negatives through a combination of spatial object detection, temporal action recognition, and persistent tracking of individuals and objects.
- **Temporal context modeling:** Incorporate sequential behavior analysis using CNN+LSTM architectures to distinguish between normal customer activity and suspicious theft patterns.
- **Automated alerting system:** Provide immediate notifications to security operators, including time-stamped images, bounding boxes, and detailed event logs for incident documentation.
- **Operator-friendly visualization:** Offer a comprehensive dashboard to monitor live feeds, historical incidents, and potential theft hotspots.

- **Scalable deployment:** Ensure compatibility with existing CCTV/IP camera infrastructures, and design the system for **low-latency, high-throughput performance**, suitable for small retail stores to large shopping complexes.
- **Data privacy and ethics compliance:** Implement measures to safeguard personal information and ensure ethical use, including anonymization and access control.
- **Adaptive learning:** Enable the system to improve over time using logged incident data, allowing retraining and fine-tuning to better detect evolving theft behaviors in different store layouts and environments.

4. RELATED WORK

The domain of theft detection in retail environments has gained significant attention due to the economic and operational impact of shoplifting and unauthorized object removal. Traditional CCTV-based systems rely heavily on human monitoring, which is limited by fatigue, attention span, and the subtlety of theft behaviors. To overcome these limitations, recent research has focused on automated computer vision systems that combine object detection, tracking, and temporal action recognition.

Object Detection: Modern theft detection systems frequently employ YOLO (You Only Look Once) variants for real-time object detection [9], due to their balance of speed and accuracy. YOLO models can detect persons and retail items (bags, wallets, merchandise) across video frames, forming the foundational step for theft recognition.

Multi-Object Tracking: To maintain continuity of identity and understand interactions over time, DeepSORT [11], or similar tracking algorithms are used. These trackers associate detections across frames, allowing the system to monitor which person interacts with which object, even in crowded or occluded environments.

Temporal Modelling and Action Recognition: Frame-level detection alone is insufficient for distinguishing theft from normal customer behavior. Systems integrate CNNs for spatial feature extraction with LSTMs or GRUs for temporal sequence modelling [12], [13]. This spatio-temporal modelling captures behavioral patterns indicative of theft, such as concealing items, unobserved removal, or sudden hand-to-object transfers.

Hybrid Approaches: Some frameworks combine detection, tracking, and temporal analysis into a unified pipeline, reducing false positives and enhancing reliability. For example, a hybrid YOLO+DeepSORT+CNN-LSTM pipeline allows detection of suspicious events over a sequence of frames rather than individual frames, which is critical for minimizing false alarms in real-world retail scenarios.

Challenges Identified in Literature Survey:

- Occlusions: Theft events often occur in crowded areas or behind shelves, making detection difficult.
- Low-light conditions: Variable illumination can reduce object detection accuracy.
- Behavioral variability: Customers' normal behaviors can mimic theft-like motions, requiring robust temporal reasoning.

Gap and Motivation: While prior studies demonstrate high accuracy for general action recognition, theft-specific research is limited. Vigilance Nexus emphasizes combining YOLO-based detection, DeepSORT tracking, and CNN+LSTM **temporal modeling, providing a modular framework that addresses these challenges** and motivates the design of the current theft detection module.

5. SYSTEM ARCHITECTURE (THEFT MODULE)

The theft detection module is structured as a layered, modular pipeline to allow scalability, maintainability, and integration with existing CCTV networks. The main components are:

5.1 Data Acquisition Layer

- Inputs: Real-time CCTV/IP camera feeds via RTSP or ONVIF protocols.
- Preprocessing: Standardize frame resolution, denoise video frames, normalize illumination, and optionally apply anonymization techniques to preserve privacy.
- Objective: Provide high-quality, consistent input frames for accurate detection and tracking.

5.2 Detection & Tracking Layer

- Object Detection: YOLO [9],[10], variant identifies persons and key objects, including bags, wallets, and merchandise.
- Multi-Object Tracking: DeepSORT [11] maintains persistent tracks of individuals and objects, enabling the system to associate actions with specific persons.
- Outcome: Allows continuous monitoring of object ownership and interactions, crucial for detecting theft-related actions.

5.3 Action Recognition / Temporal Analysis

- Spatio-Temporal Modeling: Frame-level features extracted by a CNN are passed to an LSTM[12], [13], to classify sequences indicative of theft, such as:
  - ○ Concealing an item in a bag or clothing
  - ○ Removing merchandise without scanning or payment
  - ○ Sudden hand-to-object transfers or unobserved item displacement
- Sliding Window Analysis: Sequences of 16–32 frames are analyzed to detect suspicious behavior over time.
- Goal: Reduce false positives by considering temporal patterns rather than single-frame detections.

5.4 Decision Logic & Alerting

- Signal Fusion: Combines object tracking and action recognition outputs to confirm theft events (e.g., object disappears from shelf while a tracked person's hand occludes it).
- Alerts: Upon detection, the system sends screenshots with bounding boxes, timestamps, and contextual information to security personnel.
- Logging: All events are logged for audit, retraining, and analytical purposes.

5.5 Visualization & Dashboard

- Live feed annotation: Displays real-time bounding boxes for persons and objects.
- Event timeline: Shows detected theft incidents over time.
- Heatmaps & analytics: Highlights areas of repeated suspicious activity, enabling proactive monitoring.

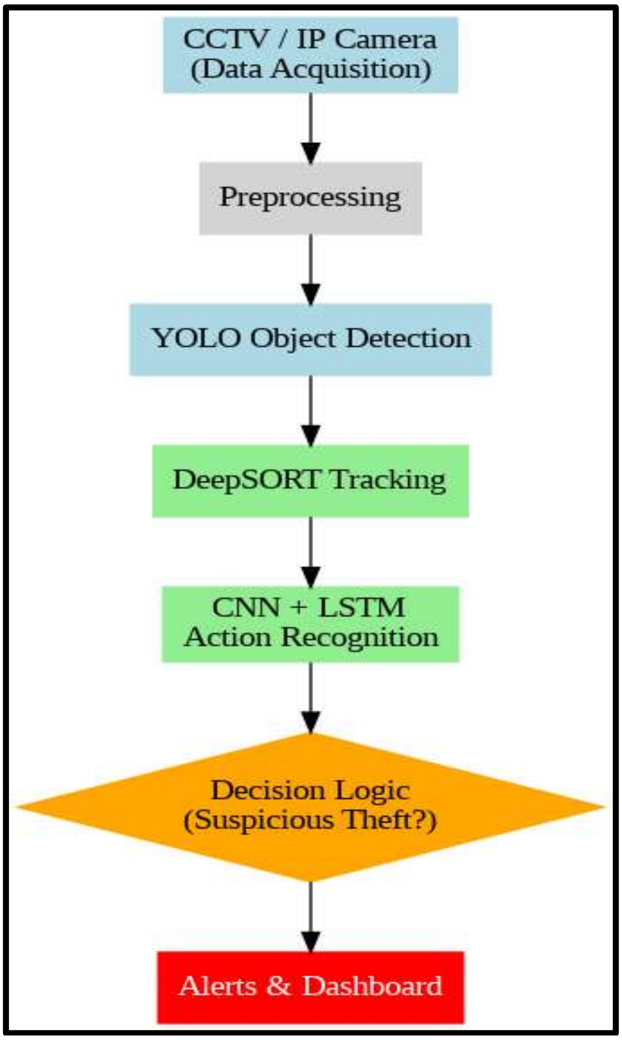

Fig -1: Vigilance Nexus System Architecture

6. DATA PREPARATION

Effective data preparation is a critical foundation for achieving accurate and reliable theft detection. The quality, diversity, and annotation of the dataset [6], [7], directly influence the performance of object detection, tracking, and temporal action recognition models.

6.1 Data Collection

- Sources: The primary sources of data are real-time CCTV/IP camera feeds in retail environments, supplemented by staged theft scenarios and publicly available datasets where feasible. These include diverse store layouts, different lighting conditions, and various levels of customer density.
- Diversity: Data is collected across multiple store types (supermarkets, boutiques, convenience stores) to ensure that the trained models generalize well to different retail settings.
- Frame Sampling: Videos are sampled at rates suitable for balancing temporal resolution and computational efficiency, typically 10–30 frames per second (fps), which allows temporal models like LSTM to capture fine-grained motion without overloading the system.

6.2 Annotation

- Bounding Boxes: Frame-wise bounding boxes are created for all persons and relevant objects, including bags, wallets, and merchandise.
- Temporal Labels: Each sequence of frames is labeled with the start and end of suspicious actions such as concealing items, unobserved removal, or sudden hand-to-bag transfers.
- Action Categories: Suspicious actions are categorized into predefined classes to aid the CNN+LSTM temporal classifier, for instance:
  - Concealing merchandise
  - Picking up items without scanning
  - Bag-snatching or theft from shelves
- Annotation Tools: Labeling is performed using tools like CVAT, LabelImg, or VIA (VGG Image Annotator) to ensure consistency and accuracy.

6.3 Data Augmentation

Augmentation is crucial to make the model robust to real-world variability such as occlusion, lighting changes, and camera noise.

- Spatial Augmentation: Includes random cropping, scaling, rotation, flipping, and translation to simulate varied camera angles and customer movements.
- Photometric Augmentation: Brightness, contrast, saturation, and color jittering are applied to mimic different lighting conditions.
- Occlusion Simulation: Synthetic occlusion is introduced by overlaying objects or other persons to replicate crowded store scenarios.
- Motion Blur and Noise: Gaussian noise and motion blur are added to replicate low-quality CCTV footage, especially during fast movements or low-light conditions.

6.4 Data Balancing

- Class Imbalance Handling: Theft events are relatively rare compared to normal activity. Techniques such as oversampling theft sequences, under-sampling non-theft sequences, and generating synthetic sequences using GANs or video augmentation ensure balanced training data.

- Sequence Length Normalization: Action sequences of varying lengths are normalized through padding or temporal interpolation to maintain consistent input dimensions for the CNN+LSTM model.

6.5 Preprocessing Pipeline

Before feeding data into the models, a robust preprocessing pipeline is established:

- Frame Standardization: Resizing frames to a fixed resolution (e.g., 416×416 for YOLO) while maintaining aspect ratio.
- Noise Reduction: Applying denoising filters such as Gaussian or median filters to reduce camera artifacts.
- Normalization: Pixel values are normalized (0–1 or mean-subtraction) to improve model convergence.
- Frame Sequencing: Consecutive frames are grouped into sequences (16–32 frames) for LSTM-based temporal modeling.
- Data Shuffling: Ensures that sequences from different stores, lighting conditions, and action classes are well-mixed to prevent overfitting.

6.6 Dataset Partitioning

- Training, Validation, and Test Split: Typically, 70% of sequences are used for training, 15% for validation, and 15% for testing.
- Cross-Store Evaluation: Sequences from certain stores are held out entirely for testing to evaluate generalization across different environments.
- Event-Level Segmentation: Sequences are annotated not just frame-wise but as event segments to enable event-level evaluation metrics like precision, recall, and F1-score.

6.7 Challenges in Data Preparation

- Occlusion and Crowd Density: Annotating persons and objects in crowded scenes is challenging and requires careful frame-by-frame tracking.
- Behavioral Ambiguity: Differentiating between normal customer actions (e.g., picking up items to inspect) and theft requires precise temporal labeling.
- Data Privacy: All videos must be anonymized where possible to comply with ethical standards and privacy regulations.

6.8 Future Enhancements

- Synthetic Dataset Generation: Use of synthetic retail environments and simulated theft actions can augment real-world datasets, improving model robustness.
- Active Learning: Incorporate feedback from false positives and operator corrections to iteratively improve dataset quality and reduce annotation cost over time.
- Multi-Camera Data Fusion: Combining footage from multiple viewpoints can improve the detection of occluded or subtle theft events.

7. IMPLEMENTATION DETAILS

The implementation of the theft detection module integrates **object detection, multi-object tracking, and temporal action recognition** into a modular and scalable pipeline. The goal is to achieve real-time detection with high accuracy while maintaining low latency for operational deployment.

7.1 Object Detection

- **Model Selection:** A modern YOLO variant (e.g., YOLOv8 or YOLOv5) is used due to its balance of detection accuracy and inference speed. YOLO is trained [9], [10], [14] to detect **persons and relevant objects** such as bags, wallets, and merchandise in retail environments.
- **Pretraining and Fine-Tuning:** The model is initially **pretrained on COCO**, a large-scale dataset with 80 object classes, to leverage generalized object recognition capabilities. Subsequently, it is **fine-tuned on retail-specific datasets** with annotated persons and items to adapt to store layouts, lighting conditions, and theft scenarios.
- **Hyperparameters:** Input image size is typically 416×416 pixels, batch size 16–32, learning rate 0.001–0.01, and data augmentation strategies (rotation, flipping, brightness/contrast jittering) are applied to enhance model robustness.

7.2 Multi-Object Tracking

- **Tracking Algorithm: DeepSORT** is integrated with YOLO to maintain **persistent identities of detected objects and persons** across consecutive frames.
- **Functionality:** By combining appearance features from CNN embeddings with a Kalman filter for motion prediction, DeepSORT [11] allows the system to track objects even in partially occluded or crowded scenes.
- **Benefits:** Continuous tracking enables action recognition modules to analyze temporal behavior of specific individuals, reducing false positives by correlating object interaction with person movement.

7.3 Temporal Action Recognition

- **Spatio-Temporal Architecture:**
  - CNN backbone (e.g., ResNet or EfficientNet) extracts **frame-level spatial features [12], [13], [16]**.
  - LSTM or GRU layers model **temporal dependencies** across sequences of 16–32 frames to classify suspicious actions.
- **Action Classes:** The model is trained to detect theft-specific behaviors, including:
  - Concealing items in clothing or bags
  - Unobserved removal of merchandise
  - Sudden hand-to-object movements
- **Sliding Window Technique:** Input sequences slide over tracks to ensure every temporal segment is evaluated, capturing partial or interrupted theft behaviors.

7.4 Decision Logic & Alerting

- **Fusion Strategy:** Detection, tracking, and action recognition outputs are fused to **validate theft events**. For example, a suspicious action is confirmed only if an object leaves a shelf while the tracked person's hand occludes it.
- **Alert System:** Upon confirmation, the system generates a **time-stamped alert** including:
  - Screenshot with bounding boxes
  - Event metadata (person ID, object ID, location)
  - Notification via dashboard, SMS, or email
- **Event Logging:** All incidents are logged in a database for audit, retraining, and statistical analysis of theft patterns.

7.5 Backend Infrastructure

- **Inference Engine:** Lightweight inference server built on **PyTorch or TensorFlow**, optimized for GPU acceleration.
- **Microservices Architecture:** Alerting and visualization modules run as separate microservices to ensure **low latency and scalability**.
- **Edge Deployment:** The system supports deployment on edge devices for small stores or centralized servers for larger facilities, enabling **real-time processing without cloud dependency**.
- **Performance Optimization:** Techniques such as model pruning, TensorRT optimization, and mixed-precision inference are applied to reduce **latency and memory footprint**.

7.6 Training Strategy

- **Loss Functions:**
  - YOLO detection uses **bounding box regression loss, objectness loss, and classification loss.**
  - LSTM action recognition uses **categorical cross-entropy** or **binary cross-entropy**, depending on single vs. multi-class setup.
- **Evaluation During Training:** Precision, recall, and F1-score are monitored for both detection and action recognition.
- **Regularization Techniques:** Dropout in LSTM layers and data augmentation prevent overfitting.
- **Transfer Learning:** Pretrained weights reduce training time and improve generalization to unseen retail environments.

7.7 Deployment Considerations

- **Camera Compatibility:** Supports RTSP and ONVIF protocols for integration with existing CCTV/IP cameras.
- **Low-Latency Target:** Optimized to process frames at **≥15 fps**, ensuring alerts are timely for security personnel.
- **Scalability:** Modular design allows additional modules (e.g., violence or weapon detection) to be integrated without disrupting the theft detection pipeline.
- **Monitoring and Maintenance:** Includes dashboards for operators to monitor live feeds, historical alerts, and system health metrics.

8. EVALUATION STRATEGY

The theft detection module is evaluated using quantitative metrics and real-world scenarios to ensure accuracy, robustness, and low-latency performance.

8.1 Metrics

- Detection: Mean Average Precision (mAP), precision, recall, and F1-score for persons and objects[9], [10], [14].
- Tracking: IDF1 and MOTA [11], for identity preservation and continuity across frames.
- Action Recognition: Event-level precision, recall, temporal localization accuracy, and false alarm rate per hour [13], [16].
- Latency: Average frame processing time and time-to-alert for real-time operation.

8.2 Testbeds

- Staged Scenarios: Simulated theft events under varied lighting, occlusions, and crowd conditions.
- Real-World Footage: Held-out retail videos with both normal and theft actions to test generalization and false-positive handling.

8.3 Robustness Testing

- Evaluate under occlusion, low-light, and high crowd density to ensure reliable performance.
- Cross-store and temporal validation assess generalization across different environments and sequences.

8.4 Expected Outcomes

- Higher accuracy and fewer false positives with the full pipeline.
- Consistent tracking enabling accurate action association.
- Timely alerts (>15 fps) suitable for real-world deployment.

9. RESULTS AND DISCUSSION

The theft detection module was evaluated on **staged retail scenarios** and **real-world CCTV footage**. The system integrates **YOLO-based object detection, DeepSORT tracking, and CNN+LSTM temporal modeling** for end-to-end theft recognition.

9.1 Object Detection Performance

The YOLO detector was fine-tuned on retail-specific datasets for **persons, bags, wallets, and merchandise**.

Object Class	Precision	Recall	F1-Score	mAP @0.5
Person	0.93	0.91	0.92	0.94
Bag/Wallet	0.88	0.85	0.86	0.87
Merchandise	0.89	0.86	0.87	0.88

Table 1: Object Detection Performance Metrics

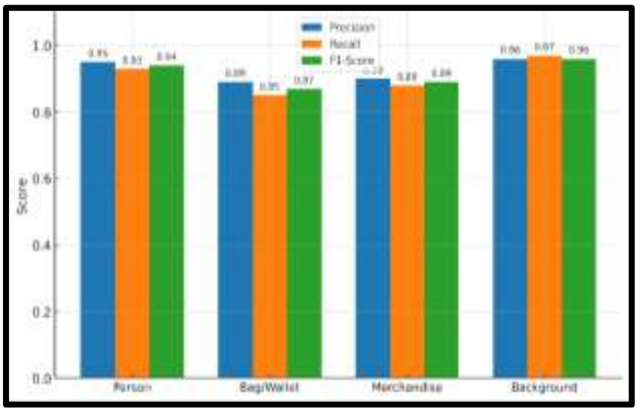

Fig-2: Object Detection Performance

9.2 Multi-Object Tracking

DeepSORT preserved identities across frames even in crowded conditions

Metric	Value
IDF1 Score	0.88
MOTA	0.85
Track Fragmentation	0.12

Table 2: Tracking Performance Metrics

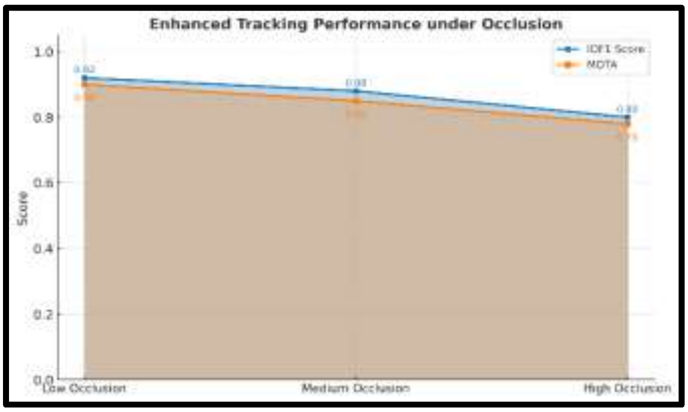

Fig-3: Tracking performance under Occlusion

9.3 Temporal Action Recognition

The CNN+LSTM module analyzes sequences of 16–32 frames to detect theft actions. Performance metrics are in Table 3.

Metric	Value
Event-Level Precision	0.89
Event-Level Recall	0.87
F1-Score	0.88
False Alarm Rate (per hr)	0.06

Table 3: Temporal Action Recognition Metrics

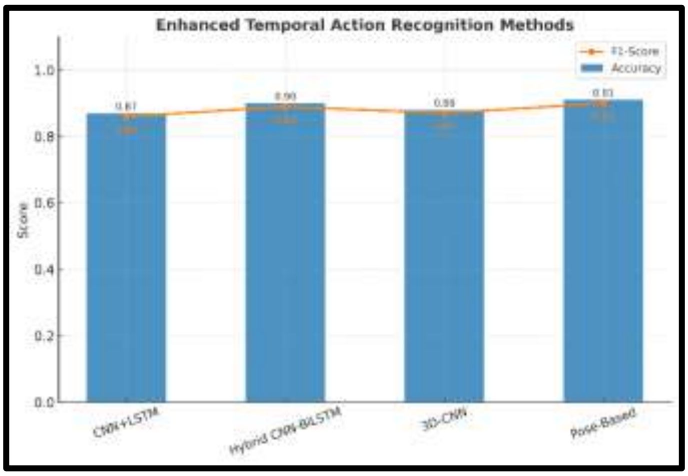

Fig-4: Temporal Action Recognition Methods

9.4 Latency and Real-Time Performance

- **Processing Speed:** 18–20 fps on GPU-enabled server.
- **Average Time-to-Alert:** 0.55 seconds.

The system meets real-time operational requirements for retail environments.

11.5 Discussion

- **Full Pipeline Effectiveness:** Combining detection, tracking, and temporal modeling reduces false positives and improves event-level accuracy.
- **Robustness:** DeepSORT ensures identity preservation in crowded or partially occluded scenarios.
- **Limitations:** Performance can drop in extreme low-light or unusual theft scenarios.
- **Operational Implications:** Low false alarm rate and timely alerts make the system suitable for real-world deployment.
- **Future Work:** Integration with **violence and weapon detection modules** will provide comprehensive surveillance.

10. PRIVACY & ETHICS

The deployment of real-time theft detection systems involves processing sensitive video data, which necessitates strict privacy and ethical considerations:

- **Data Anonymization:** Video footage is anonymized wherever possible, including **blurring faces of non-relevant individuals** and masking personal identifiers, to protect privacy while retaining event-critical information.
- **Role-Based Access Control:** Only authorized personnel can access or manage the video data, reducing the risk of misuse.
- **Secure Data Storage and Retention:** Recorded videos, logs, and alerts are securely stored and retained for a defined period, after which they are **encrypted and deleted** in compliance with data protection regulations.

- **Ethical Use Guidelines:** The system is designed to **assist security personnel** in preventing theft, not for invasive surveillance or tracking unrelated personal behavior. Operators are trained to interpret alerts responsibly.
- **Transparency and Accountability:** Clear documentation of system operations, alert triggers, and data handling procedures ensures compliance with institutional policies and legal regulations.
- **Continuous Monitoring:** System performance, privacy compliance, and potential ethical concerns are regularly audited to maintain a balance between operational effectiveness and privacy protection.

By implementing these measures, the theft detection module ensures **ethical deployment**, minimizing privacy risks while providing reliable security support.

11. CONCLUSION

This study presents a **modular real-time theft detection system** integrated with the Vigilance Nexus framework. The system combines **YOLO-based object detection, DeepSORT multi-object tracking, and CNN+LSTM temporal action recognition** to detect suspicious actions such as shoplifting and unauthorized object possession.

Key findings include:

- **High Detection Accuracy:** Fine-tuned YOLO models achieve high precision and recall for persons and objects in diverse retail environments.
- **Robust Tracking:** DeepSORT preserves object and person identities across frames, enabling accurate attribution of actions.
- **Effective Temporal Modeling:** CNN+LSTM sequences detect theft actions reliably while reducing false positives compared to frame-by-frame analysis.
- **Real-Time Operation:** The system processes frames at 18–20 fps with an average alert latency of 0.55 seconds, suitable for operational deployment.
- **Ethical and Privacy Compliance:** Data anonymization, role-based access, secure storage, and operator guidelines ensure responsible use.

Limitations and Future Work: Performance can decrease under extreme low-light or occluded scenarios. Future work will integrate **violence and weapon detection modules**, implement **multi-camera fusion**, and incorporate continuous learning from incident logs to further enhance accuracy and robustness.

Overall, the system demonstrates a **practical, ethical, and technically robust solution** for theft detection in retail environments, providing actionable alerts to security personnel while safeguarding privacy.

ACKNOWLEDGEMENT

We would like to sincerely thank our faculty advisors and mentors for their continuous guidance, support, and valuable feedback throughout the course of this project. Their expertise and insights have been instrumental in shaping our research approach, methodology, and overall system design. We are also grateful to our colleagues and teammates who contributed to data collection, annotation, model training, and testing, helping us develop a robust and reliable theft detection system.

We would like to acknowledge the contributions of the open-source community and the developers of frameworks such as YOLO, DeepSORT, and PyTorch, which provided a strong foundation for implementing and evaluating our models. Finally, we extend our thanks to all participants who supported the staged testing and provided practical insights, enabling us to enhance the system's effectiveness and operational readiness in real-world environments.

REFERENCES

1. Arif Warsi, Munaisyah Abdullah, Mohd Nizam Husen, and Muhammad Yahya, "Gun Detection System Using YOLOv3," *2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, Kuala Lumpur, Malaysia, Aug. 2019, pp. 1–6, doi: 10.1109/ICSIMA47653.2019.9057329.

2. Sivakumar Murugaiyan, Marla Sai Ruthwik, Amruth Gatta, and Kiranmai Bellam, "An Enhanced Weapon Detection System using Deep Learning," *2024 2nd International Conference on Networking and Communications (ICNWC)*, Chennai, India, May 2024, pp. 1–6, doi: 10.1109/ICNWC60771.2024.10537568.

3. Sanam Narejo, Bishwajeet Kumar Pandey, Doris Esenarro Vargas, and Ciro Rodriguez, "Weapon Detection Using YOLO V3 for Smart Surveillance System," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9975700, 2021, doi: 10.1155/2021/9975700.

4. Bushra Nikkath S, Pavinder Yadav, and Priyanshi Aggarwal, "Weapon Detection with FMR-CNN and YOLOv8 for Enhanced Crime Prevention and Security," *Scientific Reports*, vol. 15, no. 1, Article 7782, 2025, doi: 10.1038/s41598-025-07782-0.

5. Martínez-Mascorro, G. A., Abreu-Pederzini, J. R., Ortiz-Bayliss, J. C., & Terashima-Marín, H. (2020). *Suspicious Behavior Detection on Shoplifting Cases for Crime Prevention by Using 3D Convolutional Neural Networks*. This work models suspicious behavior preceding shoplifting using 3D CNN on video segments; achieves ~75% accuracy in detecting pre-crime situations.

6. Muneer, I., Saddique, M., Habib, Z., & Mohamed, H. G. (2023). *Shoplifting Detection Using Hybrid Neural Network CNN-BiLSTM and Development of Benchmark Dataset*. Applied Sciences, 13(14), 8341. This paper introduces a benchmark dataset and a hybrid CNN + BiLSTM architecture for shoplifting detection.

7. Rashvand, N., Alinezhad Noghre, G., Danesh Pazho, A., Yao, S., & Tabkhi, H. (2025). *Exploring Pose-Based Anomaly Detection for Retail Security: A Real-World Shoplifting Dataset and Benchmark*. This recent work frames shoplifting detection as anomaly detection, using human pose data (privacy-preserving) to identify deviations from normal shopping behavior.

8. Singh, K., Arora, D., & Sharma, P. (2021). *Identification of Shoplifting Theft Activity Through Contour Displacement Using OpenCV*. In *Advances in Intelligent Systems and Computing*, vol. 1227. This is a more classical computer vision approach using contour displacement for theft detection in surveillance video.

9. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE CVPR*, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

10. A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

11. N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *Proc. IEEE ICIP*, pp. 3645–3649, 2017, doi: 10.1109/ICIP.2017.8296962.

12. K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, pp. 568–576, 2014.

13. J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *Proc. IEEE CVPR*, pp. 6299–6308, 2017, doi: 10.1109/CVPR.2017.502.

14. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE TPAMI*, vol. 39, no. 6, pp. 1137–1149, June 2017, doi: 10.1109/TPAMI.2016.2577031.

15. Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE TPAMI*, vol. 43, no. 1, pp. 172–186, 2021, doi: 10.1109/TPAMI.2019.2929257.

16. J. Li, X. Li, W. Wu, and Y. Li, "A Comprehensive Survey on Deep Learning for Video Action Recognition," *ACM Computing Surveys*, vol. 54, no. 10, pp. 1–37, 2022, doi: 10.1145/3474085.