_architecture@cref _results@cref
_loss@cref

SecurityVision: Real-Time Video Anomaly Detection using WatchTower and I3D Features

Keshav Chhabra
IIIT Delhi
keshav22247@iiitd.ac.in

Adarsh Jha
IIIT Delhi
adarsh22024@iiitd.ac.in

Kartikeya Malik
IIIT Delhi
kartikeya22243@iiitd.ac.in

Akshat Kothari
IIIT Delhi
askhat22053@iiitd.ac.in

Abstract

Automated surveillance systems are increasingly vital for public safety, requiring robust methods for detecting anomalous events in real-time video feeds. This project, "SecurityVision," presents a system leveraging advanced deep learning techniques for video anomaly detection. We utilize I3D features combined with a modified Self-Supervised Sparse Representation (WatchTower) model, based on S3R. The system processes video input, identifies anomalous segments based on deviations from learned normal patterns, and presents alerts and visualizations through a web-based dashboard. This report details the problem, methodology, implementation, evaluation, and future directions of the SecurityVision system.

1. Problem Statement

Overview: The proliferation of CCTV surveillance necessitates automated systems capable of detecting unusual or threatening activities, such as fights, assaults, or other security breaches, which often go unnoticed in manual monitoring. AI-driven solutions are crucial for analyzing the vast amount of video data generated and providing timely alerts.

Scope of the Problem:

- **Input:** Video feeds from CCTV cameras or uploaded video files.
- **Output:** Real-time or near-real-time identification of anomalous video segments, anomaly scores, and alerts.
- **User:** Security personnel, system administrators monitoring surveillance feeds.
- **User Interface:** A web-based dashboard for monitoring feeds, visualizing detected anomalies, reviewing events, and managing alerts.

2. Related Work

Video anomaly detection (VAD) research has explored various approaches. Early methods often relied on trajectory analysis or handcrafted features. Deep learning methods have shown significant promise.

- **Feature Modeling:** Approaches like SlowFast Networks [6] analyze motion and appearance features at different temporal speeds for action recognition, adaptable to anomaly detection. I3D (Inflated 3D ConvNet) [1] effectively captures spatiotemporal information using 3D convolutions and is a common backbone for VAD.
- **Weakly-supervised VAD:** Methods like the one proposed by Sultani et al. [2] utilize multiple instance learning (MIL) to train models using only video-level labels (normal/anomaly).
- **Dictionary Learning for VAD:** Recent works explore dictionary learning to model normality. S3R (Self-Supervised Sparse Representation) [4], the basis for our WatchTower model, learns a task-specific dictionary for normal events and uses sparsity in reconstruction to identify anomalies. It employs enNormal and deNormal modules to separate and analyze normal/anomalous components. Linformer [3] provides a method for efficient self-attention with linear complexity, applicable for optimizing Transformer-based components.

Our work builds upon the strengths of feature modeling (I3D) and dictionary learning (S3R/WatchTower), potentially incorporating Linformer for computational efficiency.

3. Methodology

Our system employs a pipeline approach combining feature extraction and a specialized anomaly detection model.

3.1. Feature Extraction

We utilize pre-trained Inflated 3D ConvNet (I3D) models [1], specifically variants with ResNet-50 backbones (e.g., 'i3d_r50_kinetics.pth'), to extract robust spatiotemporal features (2048 dimensions) from input video segments. These features capture appearance and motion information crucial for distinguishing normal activities from anomalies.

3.2. Anomaly Detection Model: WatchTower

We adapt the Self-Supervised Sparse Representation (S3R) framework [4], which we refer to as WatchTower in our implementation, for anomaly detection. The core idea is to learn a dictionary representing normal event patterns from training data (UCF-Crime normal videos). Anomalies are detected as deviations that cannot be sparsely reconstructed using this dictionary.

Key WatchTower (based on S3R) components adapted in our system:

- **Task-Specific Dictionary:** A dictionary learned from normal video features (I3D) of the UCF-Crime dataset using Orthogonal Matching Pursuit (OMP). The dictionary file used is 'ucf-crime_dictionaries.taskaware.omp.100iters.50pct.npy'.
- **enNormal Module (potentially with Linformer):** This module reconstructs the normal component of an input feature snippet using the learned dictionary. Efficiency improvements like Linformer-based attention may be incorporated.
- **deNormal Module:** This module aims to filter out the normal components, potentially highlighting the residual anomalous parts.
- **Anomaly Scoring:** Features are passed through embedding layers (Aggregate, Dropout) and classifiers. The final anomaly score for a video segment indicates the likelihood of it being anomalous.

The model checkpoint used is 'ucf-crime_s3r_i3d_best.pth' (referred to as WatchTower checkpoint), trained specifically for UCF-Crime using I3D features.

The system processes video, extracts I3D features for temporal segments, feeds features to the WatchTower model, calculates anomaly scores, and presents results via a web interface.

4. Datasets & Evaluation Metrics

Dataset:

- **UCF-Crime** [2]: A large-scale dataset containing long, untrimmed surveillance videos featuring 13 real-world anomalies (e.g., Fighting, Assault, Burglary, Explosion) and normal activities. We use the standard training/testing splits and I3D features provided alongside the S3R methodology for training our WatchTower model.

Evaluation Metrics:

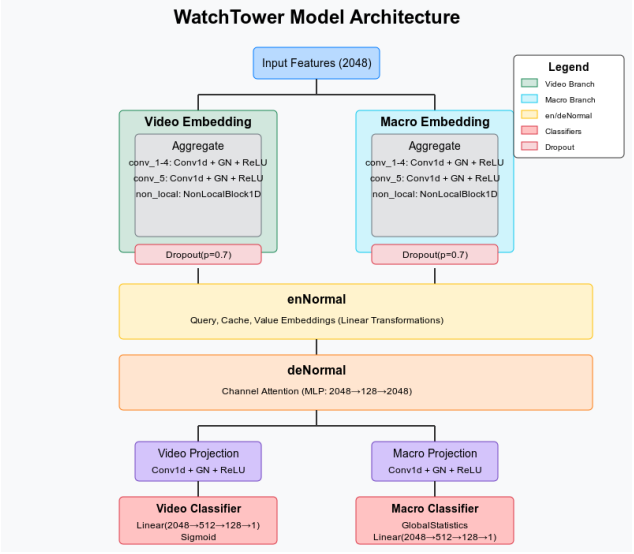


Figure 1. Detailed architecture of the WatchTower anomaly detection model. The layer-by-layer structure is based on the S3R implementation.

- **Anomaly Scoring:** Frame-wise ROC-AUC (Area Under the Receiver Operating Characteristic Curve) score is the primary metric for evaluating anomaly detection performance on datasets like UCF-Crime.

5. Implementation & User Interface

Implementation Details:

- **Backend:** Python with Flask framework.
- **Machine Learning:** PyTorch for model implementation and inference ('torch' 1.6.0, 'torchvision'). OpenCV ('opencv-python') for video processing. Key training parameters include a learning rate of 0.001 and dropout rate of 0.7.
- **Models/Checkpoints:** Uses pre-trained I3D models (e.g.,'i3d_r50_kinetics.pth') and the WatchTower/S3R checkpoint ('ucf-crime_s3r_i3d_best.pth'). A task-aware dictionary ('ucf-crime_dictionaries.taskaware.omp.100iters.50pct.npy') is loaded.
- **Dependencies:** Key libraries include 'numpy' (1.19.2), 'PyYAML', 'einops', 'sqlalchemy'. The training environment utilized CUDA 10.1.

User Interface Design:

- **Web-based Dashboard:** Built using HTML, CSS (Bootstrap), and JavaScript. Displays live feed (if camera access is enabled), system status, uptime, and recent anomalies.
- **Anomaly Visualization:** When an anomaly is detected above a threshold (e.g., 0.6 in 'ml_model.py'), visual and audio alerts are triggered. A replay feature shows the seg-

ment leading up to the alert.

- **Controls:** Users can pause/resume surveillance.

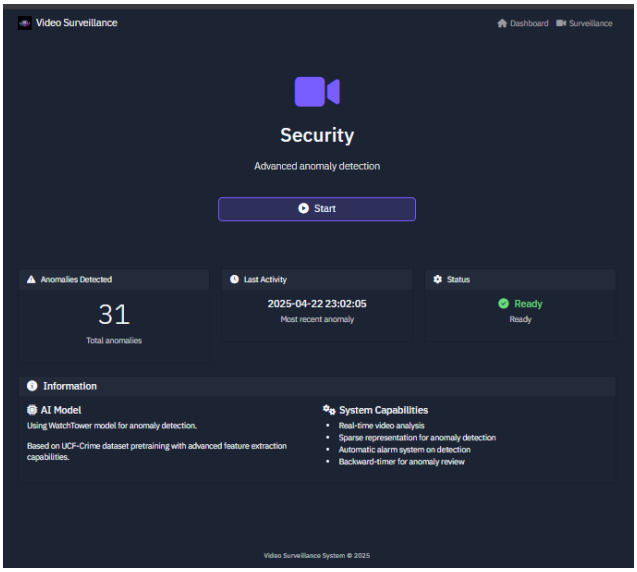


Figure 2. SecurityVision Web Interface (Dashboard View).

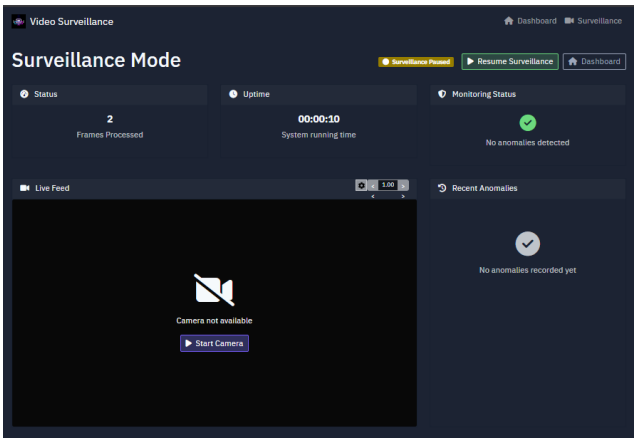


Figure 3. SecurityVision Web Interface (Anomaly Alert).

6. Results and Analysis

The S3R model, upon which our WatchTower implementation is based, achieves state-of-the-art or competitive performance on standard video anomaly detection benchmarks.

Performance on UCF-Crime: The S3R methodology using I3D features reports a frame-level AUC of 85.99% on the UCF-Crime dataset [4]. Our WatchTower model aims to replicate or build upon this performance.

Observations:

- The reported AUC of ∼86% indicates a strong capability of the S3R/WatchTower model with I3D features to distinguish anomalous frames from normal frames in the challenging UCF-Crime dataset.
- The use of dictionary learning allows the model to effectively capture the manifold of normal events.
- Real-time performance depends on feature extraction speed, model inference time, and the processing strategy (e.g., processing every Nth frame). Further optimization might be needed for true real-time deployment on constrained hardware.

Table 1. Reported S3R Performance on UCF-Crime

Dataset	Method	Feature	AUC (%)
UCF-Crime	S3R	I3D	85.99

*Result based on the WatchTower implementation details [4].

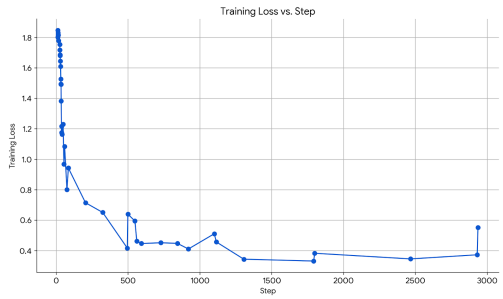


Figure 4. Training loss curve for the WatchTower model on the UCF-Crime dataset.

7. Compute Requirements

Training and deploying deep learning models for video analysis requires significant computational resources.

- **Training:** Training the WatchTower model and potentially the I3D feature extractor requires GPUs (e.g., NVIDIA RTX series) with sufficient memory (e.g., RTX 2080 Ti used in experiments), especially for large datasets like UCF-Crime. Dictionary learning itself can also be computationally intensive.
- **Inference:** Real-time inference benefits greatly from GPU acceleration. Our current system checks for CUDA availability (tested with CUDA 10.1) and uses it if possible. CPU inference is possible but significantly slower.
- **Edge Computing:** For deployment on edge devices, model optimization techniques like quantization and hardware acceleration are crucial but need careful evaluation.
- **Memory/Storage:** Handling video streams and features requires adequate RAM and potentially fast storage (SSDs).
- **Software Environment:** The system relies on Python (3.6 tested), PyTorch (1.6.0 tested), Flask, and associated

libraries.

While initial development might be feasible on moderate hardware, scaling up for robust real-time deployment across multiple streams likely necessitates more powerful resources.

8. Individual Tasks

Table 2. Team Member Responsibilities

Team Member	Assigned Tasks
Keshav Chhabra	Data preprocessing, model training (WatchTower/S3R adaptation), evaluation.
Kartikeya Malik	User interface design (Flask/HTML/JS), API integration, deployment setup.
Adarsh Jha	Model training (WatchTower/S3R adaptation), Linformer integration analysis, evaluation.
Akshat Kothari	Dataset handling (UCF-Crime features), documentation, performance analysis, results reporting.

9. Future Work

Building upon the current SecurityVision system utilizing WatchTower with I3D features, future work can focus on several areas:

1. **Explore more efficient feature extractors:** I3D happens to be a computationally heavy model. Exploring alternatives like MobileViT [7] could offer better performance/cost trade-offs, potentially enabling edge-device inference.
2. **Online learning:** Since the dictionary representing normality is built offline, implementing online or continual learning approaches [8] could allow the model to adapt to specific deployment environments and evolving normal patterns over time.
3. **Multiple Cameras:** Utilizing synchronized feeds from multiple cameras via multi-view learning techniques [9] could provide richer context, handle occlusions better, and potentially increase detection robustness.

By addressing these points, we aim to enhance SecurityVision into a more robust, efficient, and versatile real-time anomaly detection system suitable for practical surveillance applications.

10. Code Repository

The source code for this project is available on GitHub: https://github.com/kv-248/SecurityVision

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017. 2, 3

[2] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6479–6488, 2018. 2, 3

[3] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2

[4] Jhih-Ciang Wu, He-Yen Hsieh, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Self-Supervised Sparse Representation for Video Anomaly Detection. In *European Conference on Computer Vision (ECCV)*, pages 105–121, 2022. 2, 3, 4

[5] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K. Roy-Chowdhury, and Larry S. Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 733–742, 2016.

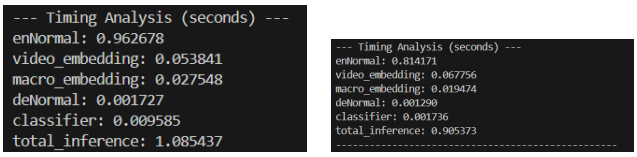
[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 6202–6211, 2019. 2

[7] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 5

[8] Wenju Cai, Michael B. Zaremba, Ashton B. Thickstun, Jason Kuen, Richard T. Chen, Kuan-Hui Lee, Gábor Bartók, Duncan T. Howcroft, Ashok Ravichandran, and Stephan Mandt. Lifelong anomaly detection via rehearsal-aided pseudo-residual learning. *arXiv preprint arXiv:2306.04195*, 2023. 5

[9] Zilong Zhang, Zhongdao Liu, Chen Change Loy, and Dahua Lin. Cross-view action recognition via viewpoint decomposition and recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7084–7093, 2019. 5

A. Timing Analysis Screenshots



(a) Before optimization (b) After optimization

Figure 5. Component-wise timing analysis of our inference pipeline.

B. System architecture

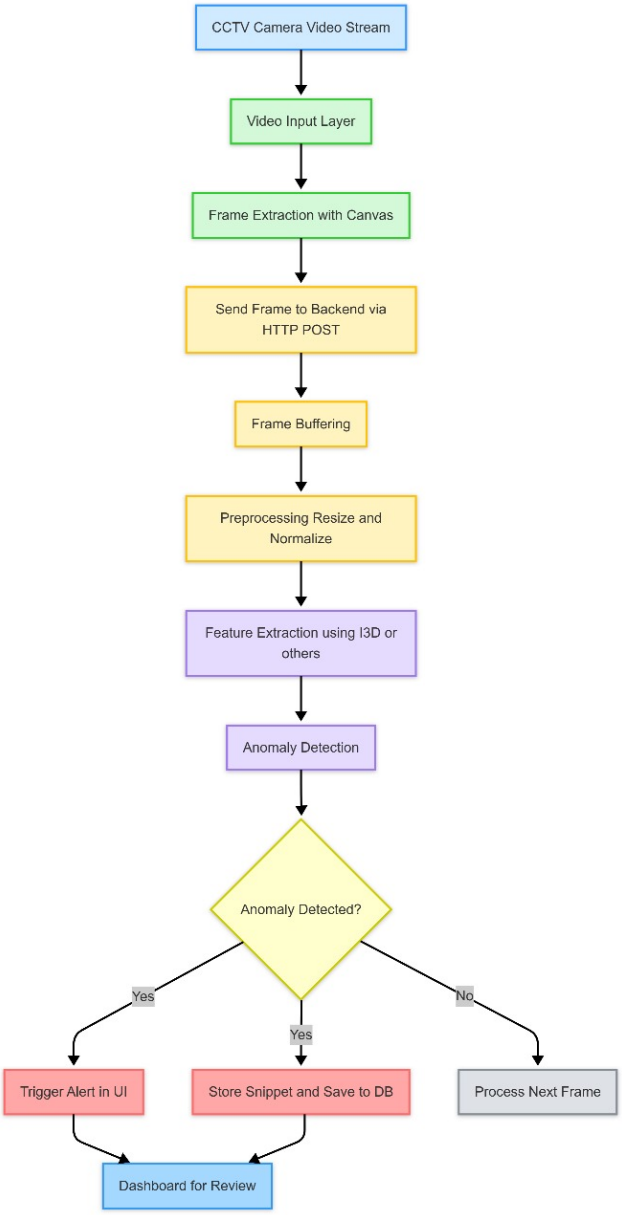


Figure 6. System architecture