Dissertation on

"Smart Reconnaissance System"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE22CS320B – Capstone Project Phase - 2

Submitted by:

Arnav Satish	**PES1UG22CS107**
Ashrith A Shetty	**PES1UG22CS120**
Sudeep Hosur	**PES1UG22CS619**
Swastika Sharma	**PES1UG22CS639**

Under the guidance of

Prof. Nitin V Pujari
Dean IQAC
PES University

January - May 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

‘Smart Reconnaissance System’

is a bonafide work carried out by

Arnav Satish	**PES1UG22CS107**
Ashrith A Shetty	**PES1UG22CS120**
Sudeep Hosur	**PES1UG22CS619**
Swastika Sharma	**PES1UG22CS639**

in partial fulfilment for the completion of sixth semester Capstone Project Phase - 2 (UE22CS320B) in the Program of Study - **Bachelor of Technology in Computer Science and Engineering** under rules and regulations of PES University, Bengaluru during the period Jan. 2025 – May. 2025. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 6th semester academic requirements in respect of project work.

Signature	Signature	Signature
Nitin V Pujari	Dr. Mamatha H R	Dr. K S Sridhar
Dean IQAC	Chairperson	Registrar

External Viva

Name of the Examiners **Signature with Date**

1. ____________________________ ____________________________

2. ____________________________ ____________________________

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **“Smart Reconnaissance System ”** has been carried out by us under the guidance of **Prof. Nitin V Pujari**, Dean IQAC and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester Jan - May 2025. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1UG22CS107	**Arnav Satish**	________________
PES1UG22CS120	**Ashrith A Shetty**	________________
PES1UG22CS619	**Sudeep Hosur**	________________
PES1UG22CS639	**Swastika Sharma**	________________

ACKNOWLEDGEMENT

I would like to express my gratitude to **Prof. Nitin V Pujari**, Department of Computer Science and Engineering, PES University, for his/her continuous guidance, assistance, and encouragement throughout the development of this UE22CS320B - Capstone Project Phase – 2.

I am grateful to the project coordinator, Dr Priyanka H, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Mamatha H R, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. K.S. Sridhar, Registrar, PES University, for his help.

I am deeply grateful to Prof. Jawahar Doreswamy, Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University, for providing me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

Traditional systems rely on poor human supervision, react extremely slowly to events, and have too many false alarms. Many sensitive applications lack confirmation of the activity of cameras in traditional systems. Generation of reports and checks of cameras are cumbersome manual processes that add to inefficiency, making traditional surveillance not suitable for modern security.

The Smart Reconnaissance System brings a sea change in many sectors by way of enhanced machine learning and real-time data processing; it automatically performs health checks of cameras, detects anomalies, minimizes false alarms, and ensures notifications on time for less human involvement. As a matter of fact, all this will pave the way for operational simplicity and reliability, extending comprehensive reports right down to the level of individual cameras.

SRS offers a broad intelligent surveillance ecosystem that deploys real-time video analytics, object detection, and unusual activity detection. With huge potential for accelerating responses and ensuring the continuity of operations, it forms an ideal system for security applications requiring precision and reliability.

Agile, efficient, and very adaptive, SRS now sets a new benchmark in surveillance by optimizing resource utilization and strengthening security frameworks. This innovation bridges critical gaps existing within traditional systems and makes organizations ready to face the challenge of an evolving security landscape.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

In the modern fast-digitizing, security-conscious world, efficient surveillance has become the pillar of public safety, protection of vital infrastructure, and business. The vast majority of traditional surveillance systems, however, remain static, passive, and much less able to provide actionable intelligence in real-time. The systems rely heavily on human monitoring and after-the-fact analysis, which can result in response delay and loss of critical events. Additionally, with threats becoming increasingly sophisticated—whether it's hacking or physical attacks—the need for more intelligent systems that not only detect but see, act, and react on their own accord without human interaction is more necessary than ever.

Smart Reconnaissance System is an end-to-end system that promises to solve these problems based on the strength of modern technologies such as Artificial Intelligence (AI), edge processing, real-time data analysis, and hybrid surveillance approaches. The system is based on a modular and extensible structure that enables it to seamlessly integrate new components—be it an IP camera, a new AI detection block, or an external alarm system. It is built upon the top of event-driven processing concepts, where the system reacts in real-time to environmental changes, such as loss of connection, anomalous video streams, or suspicious activity. Furthermore, its feedback loop mechanisms guarantee that any detected anomaly triggers alerts and logs, which are presented via a user-friendly dashboard. This renders the system extremely flexible, reliable, and able to support active security operations in areas such as defense, smart cities, critical infrastructure, and enterprise campuses.

CHAPTER-2

PROBLEM DEFINITION

Conventional security systems utilize human-based monitoring, which is extremely inefficient and prone to errors. Operators are likely to overlook real-time events because of fatigue, and the basic motion detection algorithms produce enormous amounts of false alarms, ranging from detecting shadows or the presence of small animals, wasting a great deal of time and diluting the strength of the system. Such security systems are not able to detect complex activities, such as loitering and unauthorized entry, and therefore the systems have deep security weaknesses. Hardware and network faults in security systems seldom reveal themselves until in regular monitoring, resulting in wasted time when, finally, faults are recognized.

The old systems are hence not suited to highly dynamic environments like university campuses. They also do not scale to more cameras well: they become bottlenecks in monitoring and alarm handling. These are hence addressed by the Smart Reconnaissance System which conducts autonomous camera status monitoring, identifies abnormal activity, minimizes false alarms, and creates detailed per-camera reports, which trigger real-time alarms. Employing sophisticated algorithms in machine learning, the SRS provides smooth operations, scalability, and efficiency to requirements of modern surveillance systems.

CHAPTER-3

Data

3.1 Overview

Smart Reconnaissance System gathers and processes various sources of information to facilitate proper monitoring and timely decision. Information is pulled from various sources and utilized for various purposes within the system.

3.1.1. Camera Footage

- The input to the system is predominantly video captured by surveillance cameras.
- This data is used in graphically mapping areas and determining any unusual or suspicious activity, e.g., entering areas that have been restricted.

3.1.2. Connectivity Data

- The system also continuously checks whether the cameras and other devices are switched on and in good working condition.
- As soon as any sensor or camera is disconnected from the system, it is logged and alarms are activated, which assist in detecting technical faults or any attempted tampering.

3.1.3. System Events and Logs

- The system logs all of its actions, including activating an alarm or reporting a variance.
- These logs are invaluable for tuning the system when needed, learning from past events, and holding users accountable.

3.1.4. Alerts and Notifications

- The system will automatically alert users every time it notices anything unusual.
- Each alarm contains vital information such as what happened, when, and where—enabling security officers to react promptly.

3.1.5. User Feedback

- Security officers are able to label alarms as true or false.
- This feedback will help the system learn and become intelligent in the future without producing unnecessary warnings.

Briefly, the system uses a synergistic blend of **visual information**, **device health examination**, **system log events**, and **user input** in order to be highly situation aware and react actively and consistently toward threats.

3.2 Dataset

3.2.1. UCF Crime Dataset

UCF-Crime dataset is amongst the biggest and most complete datasets created for the purpose of surveillance video anomaly detection. It is a product of the University of Central Florida, consisting of more than 1,900 actual surveillance videos that last for around 128 hours. As such, it is best suited for machine learning model training in security contexts. It has 13 categories of anomalous activities like robbery, assault, shoplifting, and vandalism, and regular day-to-day activities like walking or standing within public areas.

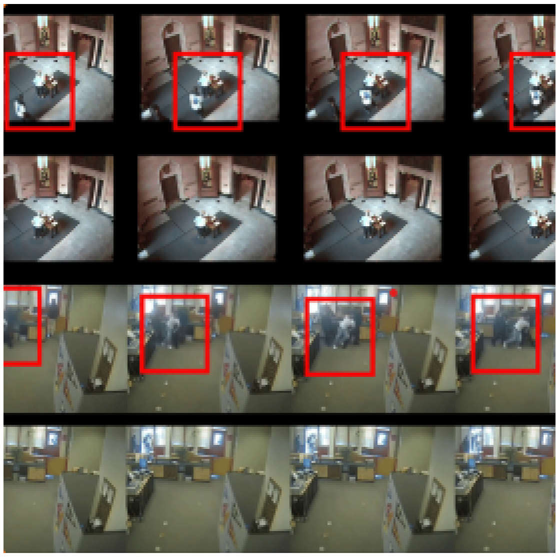


Fig 1:UCF Crime Dataset

These videos are weakly labeled and untrimmed, i.e., labels are only given at the video level but not frame-wise, very similar to how surveillance data is efficiently labeled in real-world settings. This makes the task more difficult and compels models to induce context and temporal patterns. The dataset also includes challenges such as different illumination, viewpoint, dense crowd, and occlusions, which are convenient to check how strong a model can be under real-world deployment situations. UCF-Crime is extensively utilized in the research on constructing and testing video anomaly detection models, particularly for utilization in smart surveillance, public safety, and autonomous security.

3.2.2 UMN Dataset

The UMN Dataset is among the most widely used benchmark data sets for testing abnormal crowd behavior detection in surveillance. The University of Minnesota specifically created it for collective activity recognition and anomaly detection in crowds research. It contains 11 video sequences of three scenes: lawn scene, plaza scene, and indoor hallway scene. All the videos start with a typical scenario in which people walk around in all directions as usual, and then there is an atypical scenario triggered abruptly—usually an atypical scenario in which people suddenly begin panicking and running around in all directions trying to simulate an evacuation or emergency situation. The switching from typical to atypical is critical to check models' capability to detect and react to unexpected collective motion patterns.

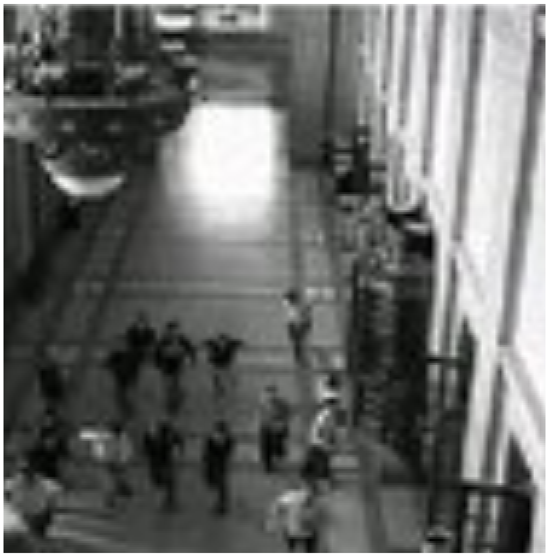


Fig 2: UMN Dataset

The dataset contains fixed-camera viewpoints and controlled settings, thus making it relatively cleaner compared to ground truth surveillance video like UCF-Crime. The dataset is valuable as a baseline dataset for understanding basic crowd behavior dynamics as well as to evaluate anomaly detection algorithms in simple settings. Each video is annotated at the scene level, i.e., the onset of abnormal events is annotated but not individual identities or trajectories. The UMN dataset continues to be used extensively for training and evaluation of unsupervised and semi-supervised learning models, especially motion pattern-based, optical flow-based, or spatiotemporal feature-based models for outlier detection in crowd behavior. Even though the dataset lacks environmental noise and real-world nuances, it plays a significant role in validating the baseline effectiveness of anomaly detection systems before scaling to more challenging datasets like UCF-Crime or Avenue.

3.2.3 ShanghaiTech Dataset

The ShanghaiTech Campus Dataset is a large and densely labeled dataset built for video anomaly detection research, particularly in populated public environments. It was introduced by researchers at ShanghaiTech University and includes 437 video sequences captured over 13 diverse camera scenes at a university campus. These shots include various kinds of settings such as walkways, corridors, public places, and open areas—each with different lighting, population density, and surroundings complexity. The data set is divided into training and test sets where only normal behavior is included in the training set. The test set includes both normal and abnormal behavior such as biking through pedestrian zones, running, fighting, or illegal vehicle entry.

Fig 3: ShanghaiTech Dataset

One of the distinguishing characteristics of the ShanghaiTech dataset is its frame-level ground truth annotations that identify exactly where and when the anomalies occur. This makes it well-suited for pixel-level anomaly localization and fine-grained temporal analysis. The dataset also simulates a few real-world challenges like dynamic backgrounds, dense human activities, and occlusions, employed to test the robustness of a detection model in real-world-like non-ideal conditions. In addition, the videos are captured from diverse views and locations, which prompt models to generalize to many views. ShanghaiTech dataset is a de facto benchmark for testing deep learning-based anomaly detection models, particularly autoencoder-based, convolutional LSTM-based, GAN-based, and frame prediction-based models.

CHAPTER-4

DESIGN DETAILS

4.1 Novelty

This system introduces a novel hybrid surveillance model by integrating network health monitoring (IP pings) with video-based anomaly detection using machine learning. Most existing systems utilize camera video feeds only, but this one also keeps track of the health of cameras, identifies failure types, and raises alerts for network or power failures, hence a proactive rather than a reactive surveillance solution.

4.2 Innovativeness

Key innovations are the event-driven architecture, which enables instantaneous reaction to camera failure or abnormality detected, automated failure classification, SMS notification, and strict adherence to escalation policy if it remains unresolved. Integrated with real-time status and logs dashboard, it constitutes an intelligent and aware surveillance network.

4.3 Interoperability

The system is modular—hardware modules like IP cameras, software modules for alerting, and ML models for video analysis can all interconnect through well-defined APIs or interfaces. It can also integrate with existing security systems (e.g., CCTV NVRs or VMS platforms), making it easy to deploy across varied infrastructures or upgrade legacy systems.

4.4 Performance

Through real-time checks on camera status via ping, automated alerting, and anomaly classification, downtime and response time are minimized. Load testing and ensuring the ML models and alerting systems do not get behind with multiple simultaneous alerts are included in the Testing & Performance Optimization phase.

4.5 Security

Security is multi-tiered. From camera registration at fixed IPs, to status monitoring and access limitation with constraints, all elements are protected. The alert escalation mechanism minimizes blind spots or unmonitored areas of failure or tampering. In addition, compliance with protection of personal data regulations ensures legal processing of video data.

4.6 Reliability

Errors like disconnection, camera malfunction, or suspicious activity are automatically sorted and escalated, so no critical incidents fall between the cracks. The issue resolution loop follows how long it took to close issues, and who closed them, with a good responsibility system of logs and reports.

4.7 Maintainability

The system is susceptible to simple maintenance because it is loosely-coupled and scalable. If a module (e.g., the alert system) must be upgraded, it can be upgraded without affecting the others. Automated health reports allow for anticipation of needs prior to catastrophic failure.

4.8 Portability

It can be deployed across various environments because it has a cloud-ready, network-based architecture. As long as there is connectivity and IP-enabled cameras, the system can be rolled out in universities, public spaces, industrial complexes, or smart cities with minimal re-engineering efforts.

4.9 Legacy to Modernization

The solution is an in-place upgrade bridge for organizations already employing manual or legacy camera systems. Fostering smart detection and connectivity monitoring on the foundation of legacy infrastructure, it is an affordable route to modernization without legacy abandonment.

4.10 Reusability

Essential building blocks such as the video anomaly detection model, the ping monitor module, or the dashboard interface are universal and adaptable for other applications—such as traffic management, plant security, or crowd monitoring—within any sector that needs visual monitoring.

4.11 Application Compatibility

It is compatible with most IP-camera-based systems, network devices, and AI/ML platforms such as TensorFlow or PyTorch. It can also be interfaced with normal data dashboards (such as Grafana or custom web UIs) and hence is application-agnostic.

4.12 Resource Utilization

It is a storage resource and computationally optimized. Pings are event-based optimized, lightweight, and video processing is optimized using triggers, i.e., video analysis occurs only when anomalies will take place. It reduces the use of GPUs and storage to a great extent, particularly in big networks

CHAPTER-5

HIGH LEVEL SYSTEM DESIGN /SYSTEM ARCHITECTURE

5.1. Client-Server Model

The architecture is along the lines of a secure, scalable client-server architecture where the clients are IP-based CCTV cameras and the central processing server is a central monitoring server. Every camera is assigned a unique IP address, which is controlled and monitored individually on the network. Cameras stream video into the central server in real time and respond to sporadic ping requests to verify whether they are operational. The server handles incoming video streams and ping replies, and centralized anomaly detection is enabled. The architecture supports integration with a high number of client cameras, and the system is modularity-enabled, scalable, and flexible for massive deployment in campus or smart city environments.

5.2. Network Communication Layer

The network's communication layer is the support system of the setup, with perfect communication between the server and the cameras. Connectivity of the cameras is regularly tested through ping checks, normally conducted every 180 seconds. Failure to respond within the set time is a sign of potential failure. Processing of the video stream is also done by the layer. Some part of the video captured is processed through machine learning to identify anomalies like unusual movement, tampering, or suspicious activity. With ping response and video data together with analysis, the layer offers dual-mode monitoring, which is more accurate and faster for real-time usage.

5.3. Database Layer

The database layer provides persistent data storage and management of data. It holds multiple types of useful data such as camera metadata (IP addresses, physical location, registration information), connectivity and failure logs, highlighted anomalies in video feed, and authorized staff information for monitoring alert resolution. Data is not lost in centralized storage, enables retrospective analysis, and enables the system to periodically report security and performance. By storing large historical records, the database also simplifies auditing and enables surveillance and data protection law compliance.

5.4. Automated Alerting System

For enabling rapid response in case of system failure and security breaches, the design incorporates an extremely sophisticated automated alert system. Automated alerts are triggered in case of failure of a camera to respond to repeated pings or in case of suspicious activity in the video stream. They are triggered through SMS or system alerts to pre-defined administrators. Escalation facility is provided to handle open cases—if there is no response to an alert or the problem hasn't been resolved within a specified time period, it gets escalated to higher authorities. This prevents any serious problem from being overlooked and increases the effectiveness and accountability of the surveillance system.

5.5. Dashboard & Issue Resolution Module

The dashboard is a central console for system administrators that provides real-time visibility into the operational health of the entire camera network. It shows live camera statuses, event logs, and alerts triggered by ping or video anomalies. Besides, it also includes tools to resolve flagged issues, update camera statuses, and log the resolution timeline. The dashboard can also handle user role management, where access levels vary according to staff responsibilities. This dynamic module not only facilitates easy monitoring but also feeds into the feedback loop for optimally enhancing system responsiveness as well as staff performance.

5.6. Implementation Workflow

The implementation of the system follows a systematic step-by-step workflow for effective deployment as well as functionality. The deployment starts with camera registration, marking the physical location of each camera and documenting the unique IP address for each camera. Then, the system operates with continuous failure detection using network ping tests combined with video monitoring. In the event of a failure, the system automatically sends alerts to administrators via SMS or system alert. The fault process requires human participation for detection of the fault, which then enables the correction process that allows the camera status to be adjusted. The last step of the process is generating reports. These reports note the problem, solution strategy, and response time, which are essential for a long-term basis for system improvement and performance tracking.

5.7. System Development Life Cycle

System development for this surveillance monitoring system follows a disciplined Software Development Life Cycle (SDLC). Phase one is Planning and Requirement Analysis, which involves establishing the goals of the system, the workflows, and the environmental constraints. Phase two is System Design and Development, in this phase the design and architecture are created, including client server designs, layer architectures for the network, and trigger/alert methods. Once development is done, testing and performance optimization is phase three involves Testing and performance optimization includes unit testing, load testing, and optimizing real-time video analytics to limit latency. Finally, once testing and performance optimization is done phase four is Deployment and Maintenance - the system is put in a live system, Under constant scrutiny, its performance examined, and updates available at a possible appropriate stage to address problems identified or technology changes.

CHAPTER-6

DESIGN DESCRIPTION

6.1 Master-Class Diagram

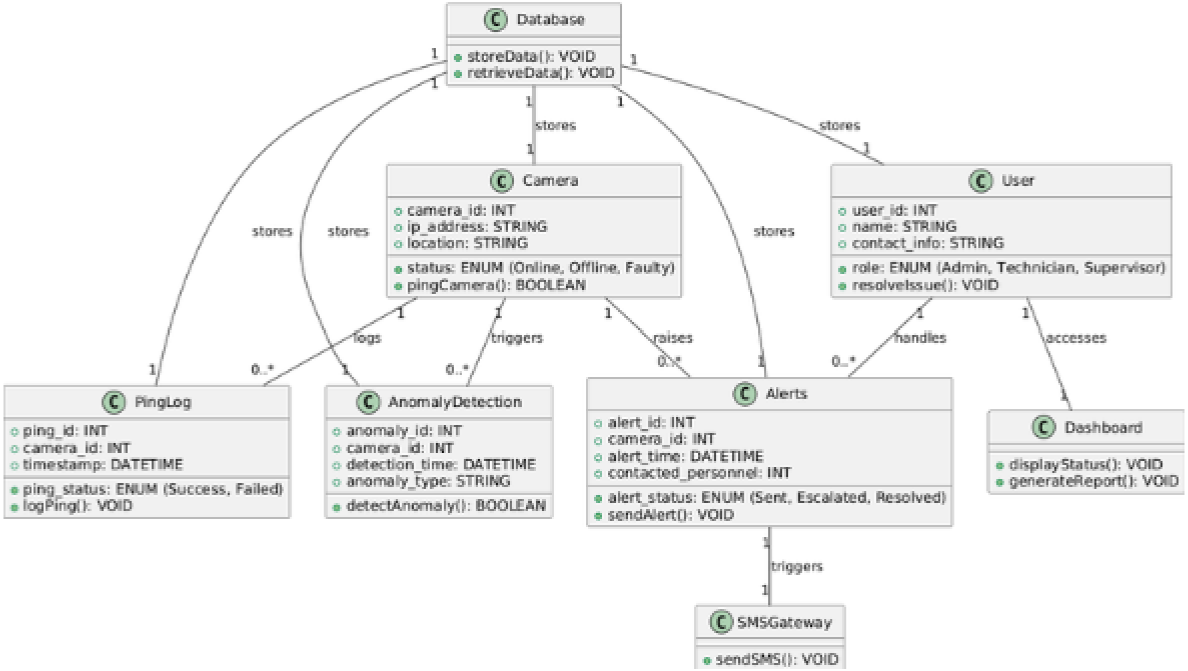


Fig 4: Master-Class Diagram

6.2 Swimlane Diagram

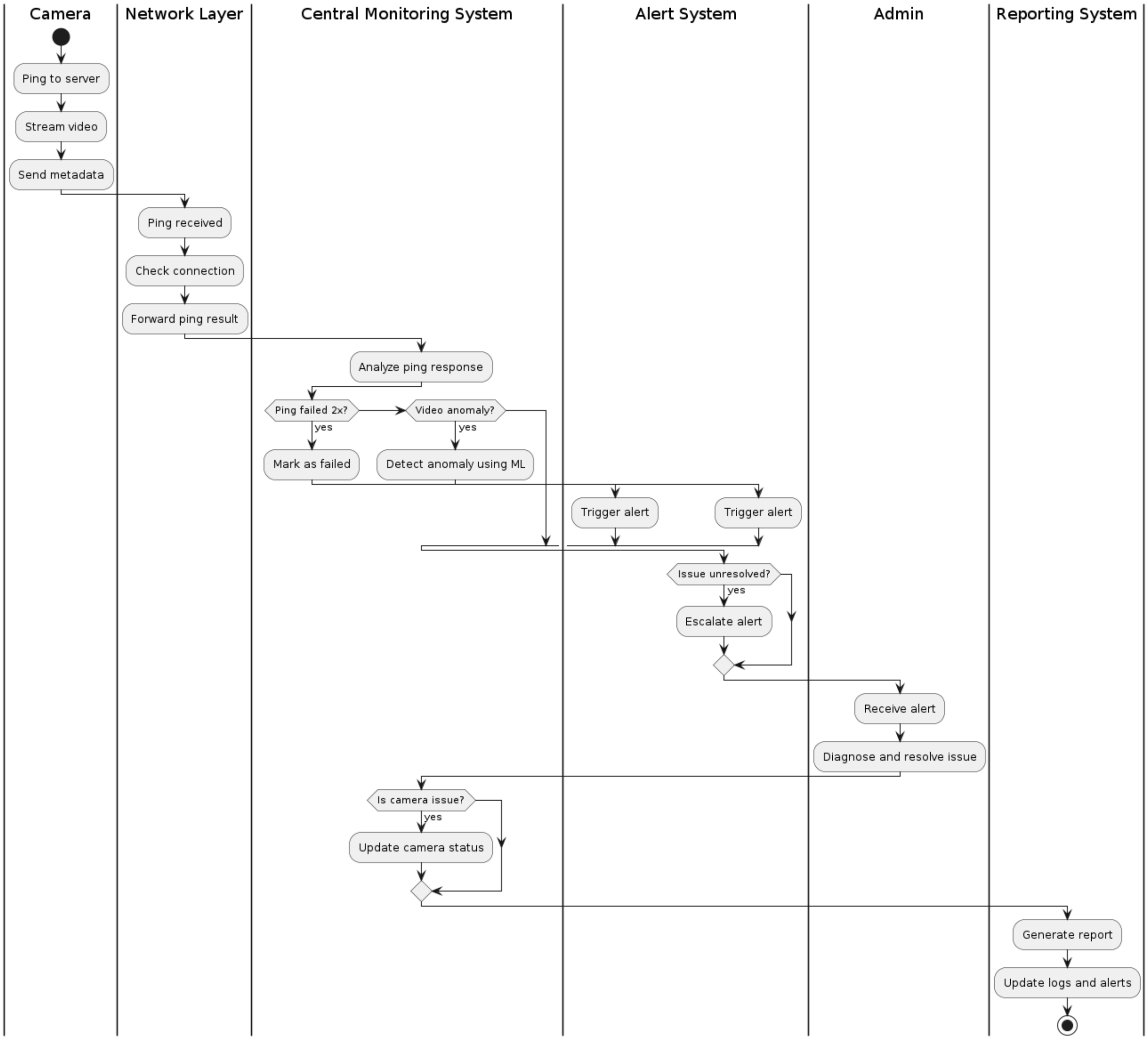


Fig 5: Swimlane Diagram

6.3 User Interface Diagram

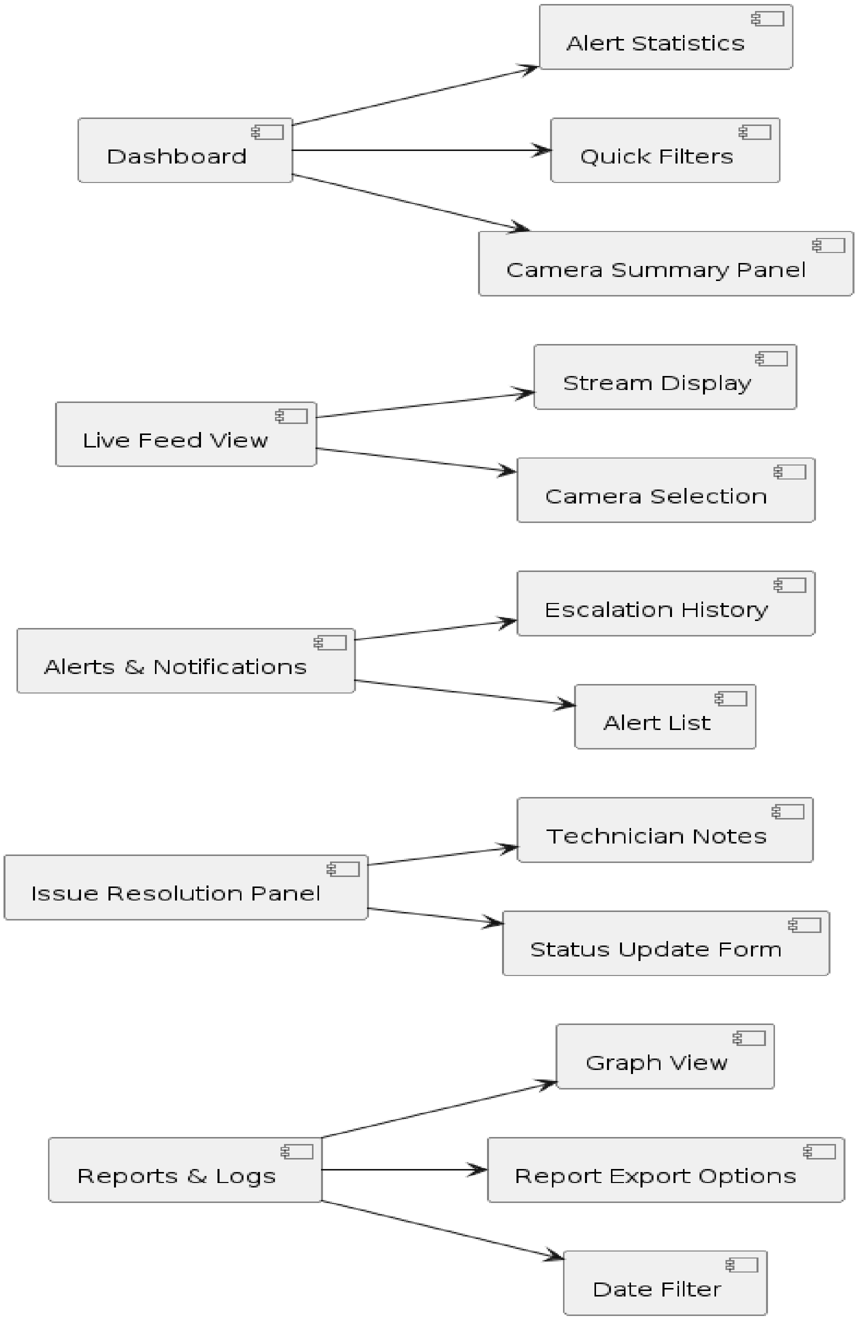
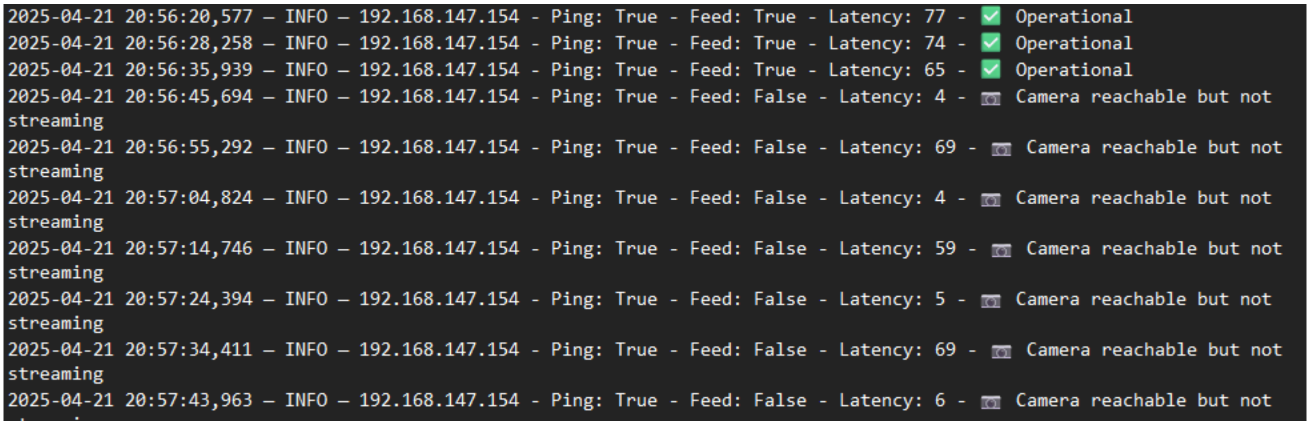


Fig 6: User Interface Diagram

6.4. Report Layouts



```
2025-04-21 20:56:20,577 – INFO – 192.168.147.154 - Ping: True - Feed: True - Latency: 77 - ✅ Operational
2025-04-21 20:56:28,258 – INFO – 192.168.147.154 - Ping: True - Feed: True - Latency: 74 - ✅ Operational
2025-04-21 20:56:35,939 – INFO – 192.168.147.154 - Ping: True - Feed: True - Latency: 65 - ✅ Operational
2025-04-21 20:56:45,694 – INFO – 192.168.147.154 - Ping: True - Feed: False - Latency: 4 - 📷 Camera reachable but not
streaming
2025-04-21 20:56:55,292 – INFO – 192.168.147.154 - Ping: True - Feed: False - Latency: 69 - 📷 Camera reachable but not
streaming
2025-04-21 20:57:04,824 – INFO – 192.168.147.154 - Ping: True - Feed: False - Latency: 4 - 📷 Camera reachable but not
streaming
2025-04-21 20:57:14,746 – INFO – 192.168.147.154 - Ping: True - Feed: False - Latency: 59 - 📷 Camera reachable but not
streaming
2025-04-21 20:57:24,394 – INFO – 192.168.147.154 - Ping: True - Feed: False - Latency: 5 - 📷 Camera reachable but not
streaming
2025-04-21 20:57:34,411 – INFO – 192.168.147.154 - Ping: True - Feed: False - Latency: 69 - 📷 Camera reachable but not
streaming
2025-04-21 20:57:43,963 – INFO – 192.168.147.154 - Ping: True - Feed: False - Latency: 6 - 📷 Camera reachable but not
```

Fig 7: Report Layouts

6.5 External Interfaces

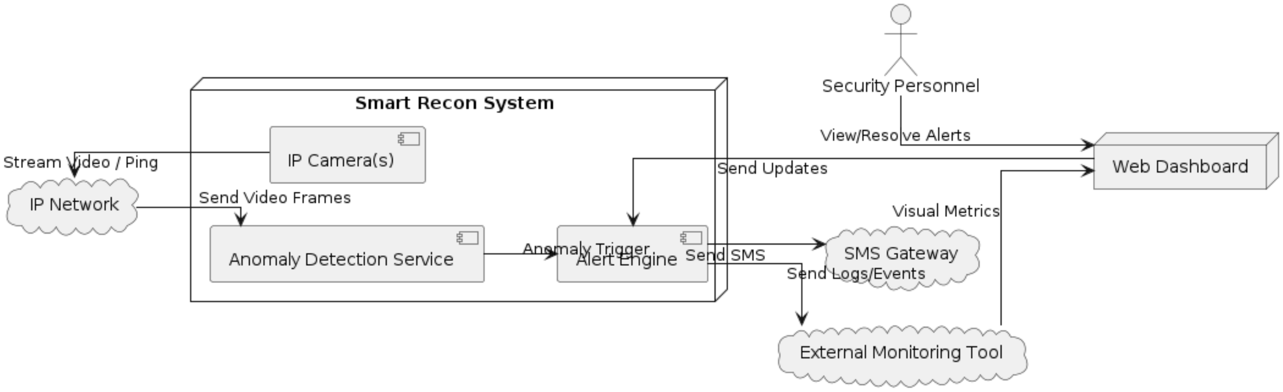


Fig 8: External Interfaces

6.6 Packaging and Deployment Diagram

6.6.1 Packaging Diagram

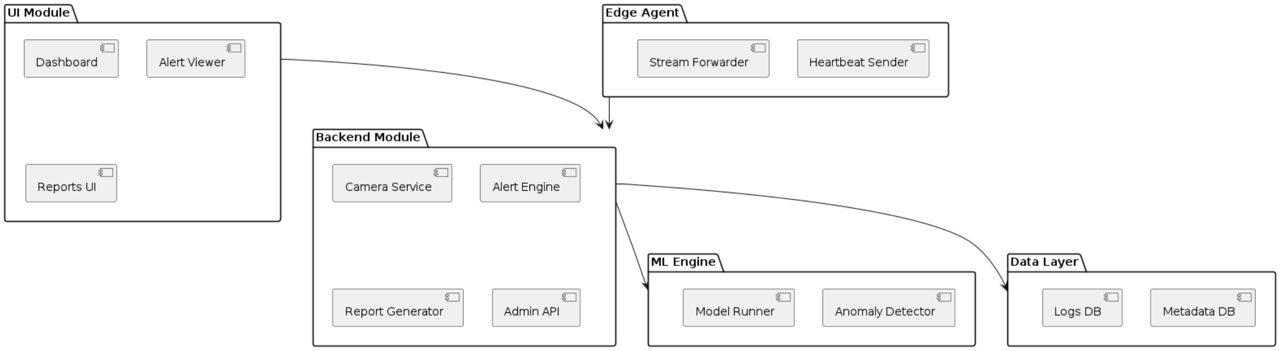


Fig 9: Packaging Diagram

6.6.2 Deployment Diagram

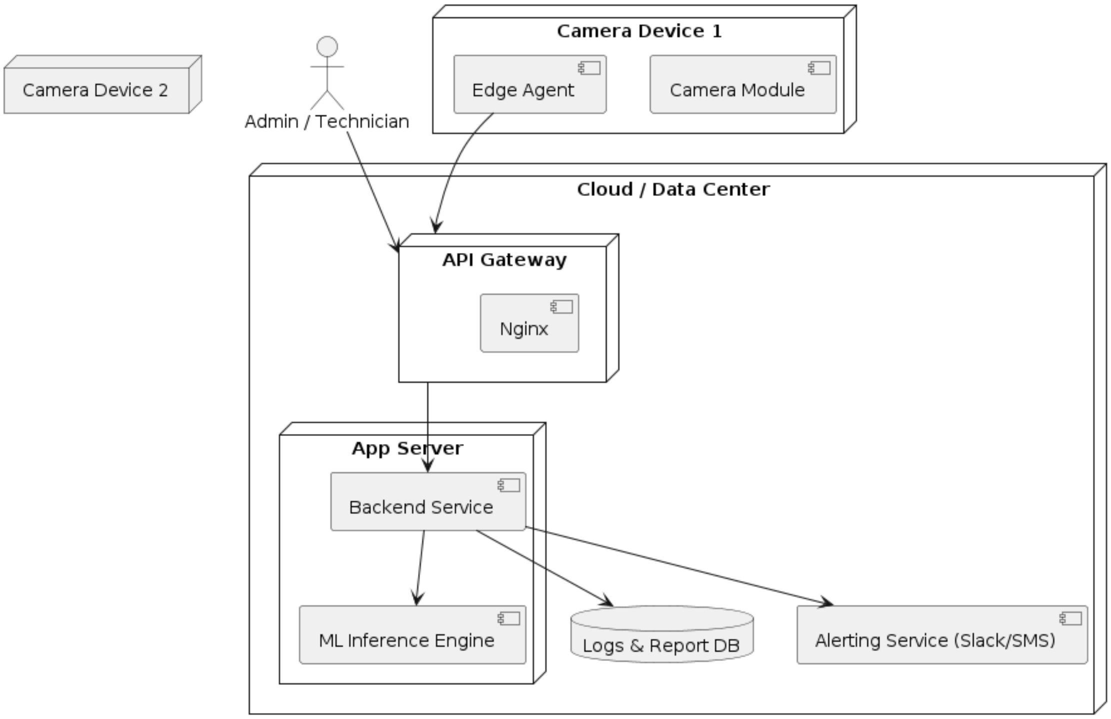


Fig 10: Deployment Diagram

CHAPTER-7

TECHNOLOGIES USED

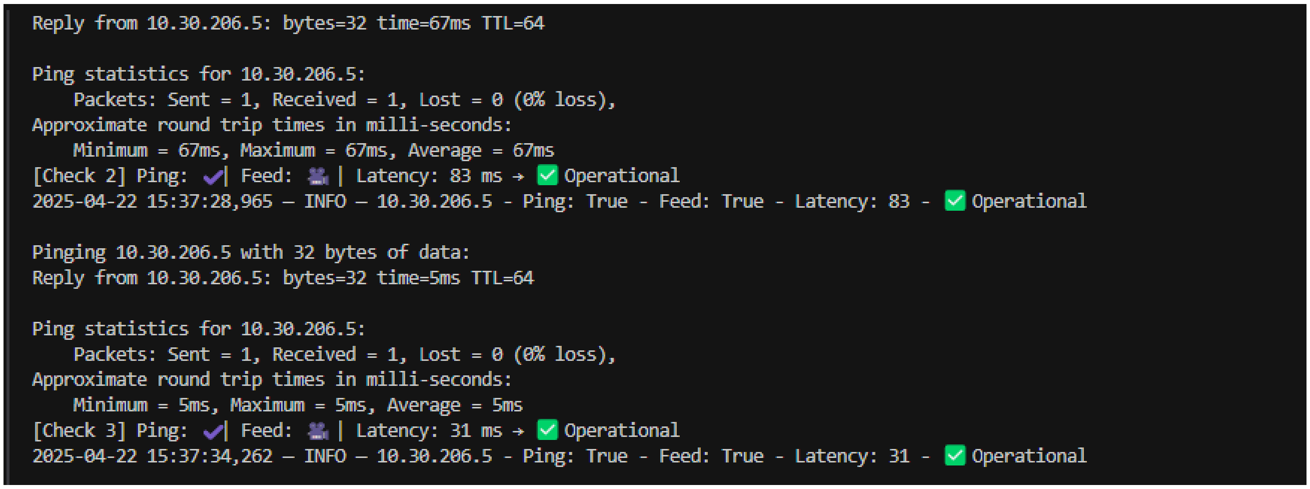
Our system leverages various deep learning frameworks, video processing toolkits, system utilities, and real-time communication protocols in order to establish an intelligent adaptive surveillance system. To detect anomaly in videos, we employ various deep learning algorithms like CNNs, LSTMs, and Transformers that were trained on various benchmark datasets including UCF-Crime, ShanghaiTech, and UMN.Training and testing is done on cloud platforms like Google Colab and Kaggle for the use of GPU acceleration.Preprocessing operations on video such as frame pulling, resizing, normalization, and data augmentation are performed using libraries such as OpenCV and Roboflow to provide normalized inputs to the model during training. Ping and HTTP protocols are standardized for the network monitoring module to check connections of cameras, latency, and availability of live feed. Rule-based classification mechanism is used for identifying the status of the camera operations and triggering alarms in the SMTP protocol for recurrent failures. Structured logging is used for monitoring events over time and trends, while SMTP protocol is utilized for alerting by email. The system is modular and scalable with web-based dashboard support optionally and lightweight datastores like SQLite or Firebase as support for storage and data visualization. This deep technology set makes it possible to have the solution elegantly scale from research through deployment environments.

CHAPTER-8

DATA PREPROCESSING AND IMPLEMENTATION

8.1 Camera Health Monitoring – Rule-Based Implementation

One of our default model setups was for a Python-rule-based system to monitor the functional status of IP-enabled CCTV cameras. The system makes three important checks at intervals: (1) it pings the camera IP address to verify network connectivity, (2) it tries to grab a snapshot off the live feed to test whether the camera is indeed broadcasting, and (3) it measures network latency from ping responses. All ping response, feed status, and latency readings are tracked and stored in a fixed-size temporal history queue (deque) for temporal analysis. They are subsequently analyzed by a rule-based classifier to classify the camera status as "Operational," "Reachable but not streaming," "High latency," "Power failure or disconnected," or "Intermittent issue."



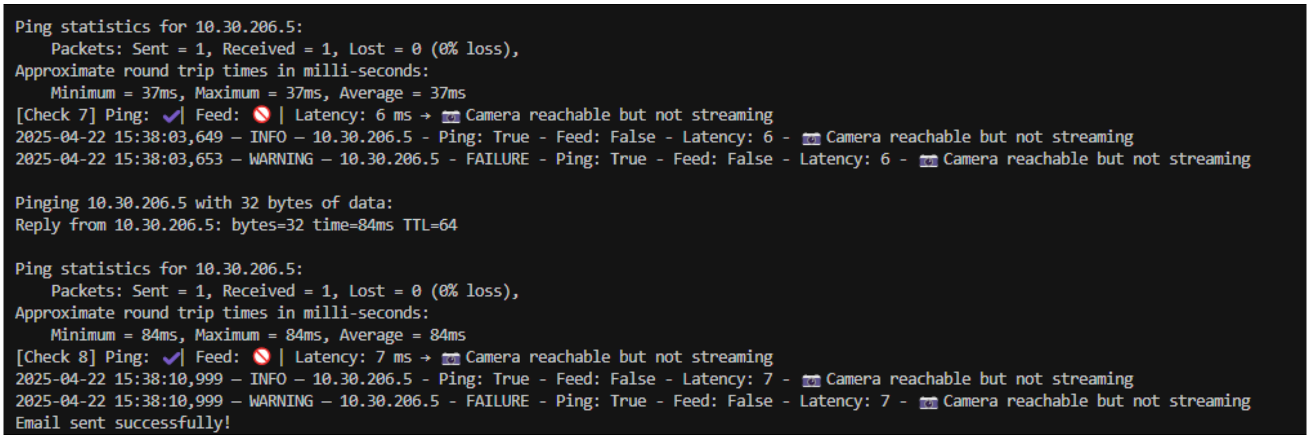
```
Reply from 10.30.206.5: bytes=32 time=67ms TTL=64

Ping statistics for 10.30.206.5:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 67ms, Maximum = 67ms, Average = 67ms
[Check 2] Ping: ✔| Feed: 🎥 | Latency: 83 ms → ✅ Operational
2025-04-22 15:37:28,965 — INFO — 10.30.206.5 - Ping: True - Feed: True - Latency: 83 - ✅ Operational

Pinging 10.30.206.5 with 32 bytes of data:
Reply from 10.30.206.5: bytes=32 time=5ms TTL=64

Ping statistics for 10.30.206.5:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 5ms, Average = 5ms
[Check 3] Ping: ✔| Feed: 🎥 | Latency: 31 ms → ✅ Operational
2025-04-22 15:37:34,262 — INFO — 10.30.206.5 - Ping: True - Feed: True - Latency: 31 - ✅ Operational
```

Fig 11: When camera is working

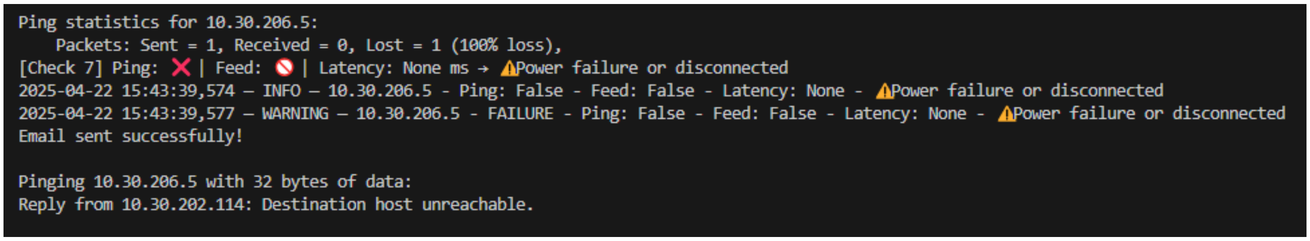


```
Ping statistics for 10.30.206.5:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 37ms, Maximum = 37ms, Average = 37ms
[Check 7] Ping: ✔| Feed: 🚫 | Latency: 6 ms → 📷 Camera reachable but not streaming
2025-04-22 15:38:03,649 – INFO – 10.30.206.5 - Ping: True - Feed: False - Latency: 6 - 📷 Camera reachable but not streaming
2025-04-22 15:38:03,653 – WARNING – 10.30.206.5 - FAILURE - Ping: True - Feed: False - Latency: 6 - 📷 Camera reachable but not streaming

Pinging 10.30.206.5 with 32 bytes of data:
Reply from 10.30.206.5: bytes=32 time=84ms TTL=64

Ping statistics for 10.30.206.5:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 84ms, Maximum = 84ms, Average = 84ms
[Check 8] Ping: ✔| Feed: 🚫 | Latency: 7 ms → 📷 Camera reachable but not streaming
2025-04-22 15:38:10,999 – INFO – 10.30.206.5 - Ping: True - Feed: False - Latency: 7 - 📷 Camera reachable but not streaming
2025-04-22 15:38:10,999 – WARNING – 10.30.206.5 - FAILURE - Ping: True - Feed: False - Latency: 7 - 📷 Camera reachable but not streaming
Email sent successfully!
```

Fig 12: When Camera is not Streaming

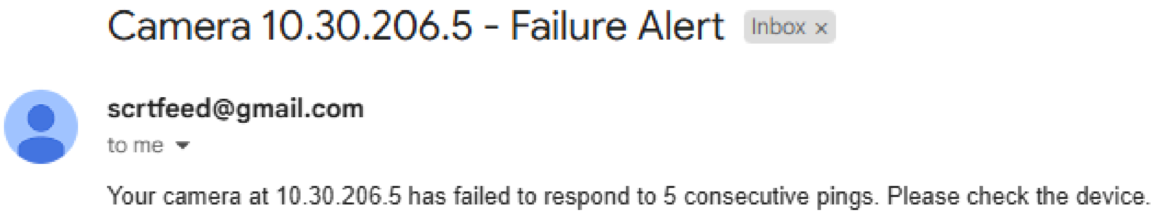


```
Ping statistics for 10.30.206.5:
    Packets: Sent = 1, Received = 0, Lost = 1 (100% loss),
[Check 7] Ping: ❌ | Feed: 🚫 | Latency: None ms → ⚠Power failure or disconnected
2025-04-22 15:43:39,574 – INFO – 10.30.206.5 - Ping: False - Feed: False - Latency: None - ⚠Power failure or disconnected
2025-04-22 15:43:39,577 – WARNING – 10.30.206.5 - FAILURE - Ping: False - Feed: False - Latency: None - ⚠Power failure or disconnected
Email sent successfully!

Pinging 10.30.206.5 with 32 bytes of data:
Reply from 10.30.202.114: Destination host unreachable.
```

Fig 13: When camera is disconnected

For the real-time diagnosis, the script stores all the health check details into a universal log file (as in Fig 7) as well as to a failure log file. Besides this, it also stores multiple failures and in case the camera is not responding to repeated numbers of checks (say, 5 consecutive pings), then automatically the system sends an email notification to the admin through an SMTP-based notification module. This configuration allows us to go out and actively look for potential hardware, network, or power problems in advance and become knowledgeable about them prior to when they are at a serious point.

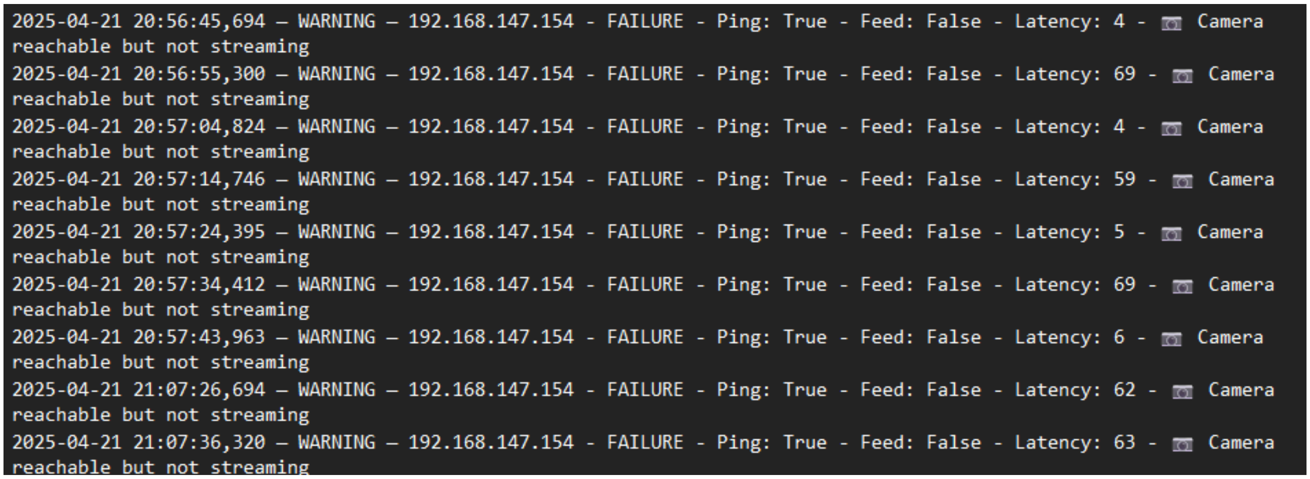


Camera 10.30.206.5 - Failure Alert

scrtfeed@gmail.com
to me ▾

Your camera at 10.30.206.5 has failed to respond to 5 consecutive pings. Please check the device.

Fig 14: Alert when camera is not working



```
2025-04-21 20:56:45,694 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 4 - 📷 Camera
reachable but not streaming
2025-04-21 20:56:55,300 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 69 - 📷 Camera
reachable but not streaming
2025-04-21 20:57:04,824 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 4 - 📷 Camera
reachable but not streaming
2025-04-21 20:57:14,746 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 59 - 📷 Camera
reachable but not streaming
2025-04-21 20:57:24,395 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 5 - 📷 Camera
reachable but not streaming
2025-04-21 20:57:34,412 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 69 - 📷 Camera
reachable but not streaming
2025-04-21 20:57:43,963 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 6 - 📷 Camera
reachable but not streaming
2025-04-21 21:07:26,694 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 62 - 📷 Camera
reachable but not streaming
2025-04-21 21:07:36,320 – WARNING – 192.168.147.154 - FAILURE - Ping: True - Feed: False - Latency: 63 - 📷 Camera
reachable but not streaming
```

Fig 15: Log file for failure

8.2 Model Architecture: CNN-LSTM for Action Classification

The deep model that we built takes the form of a two stage pipeline: Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for action recognition in video frames. The aim of our project was to detect and classify actions in video frames for Fighting, Vandalism, Stealing and Normal activity using the UCF crime dataset.

The Model Architecture consisted of the following approaches:

CNN for spatial feature extraction:

We have used a pre-trained ResNet18 model to extract the spatial features from each frame in the video. By removing the last fully-connected layer in ResNet18, and using the intermediate feature maps, we were able to extract the visual patterns from each frame without truncating the possibly relevant feature sets.

LSTM for temporal modeling:

The frame-level features from the CNN were passed, one by one, into the LSTM. LSTMs are able to learn long term dependencies from the ordered and complex sequence of frames. This enables learning of motion and dynamic temporal action dependencies.

Fully connected classifier:

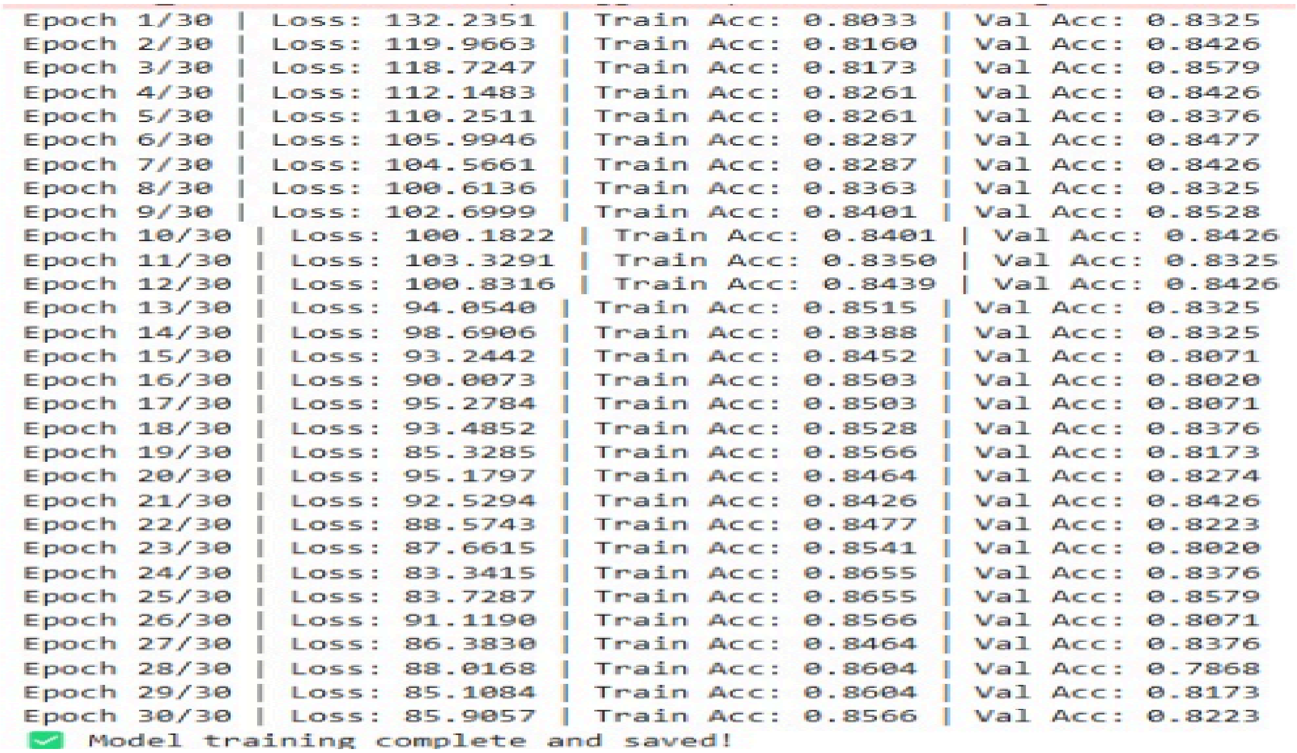
The last hidden state of the LSTM was a fully-connected layer that produces a score for each of the four classes. The model employs CrossEntropyLoss to perform multi-class classification.

Performance:

The model acquired the following values during training:

Training Accuracy: 87.2%

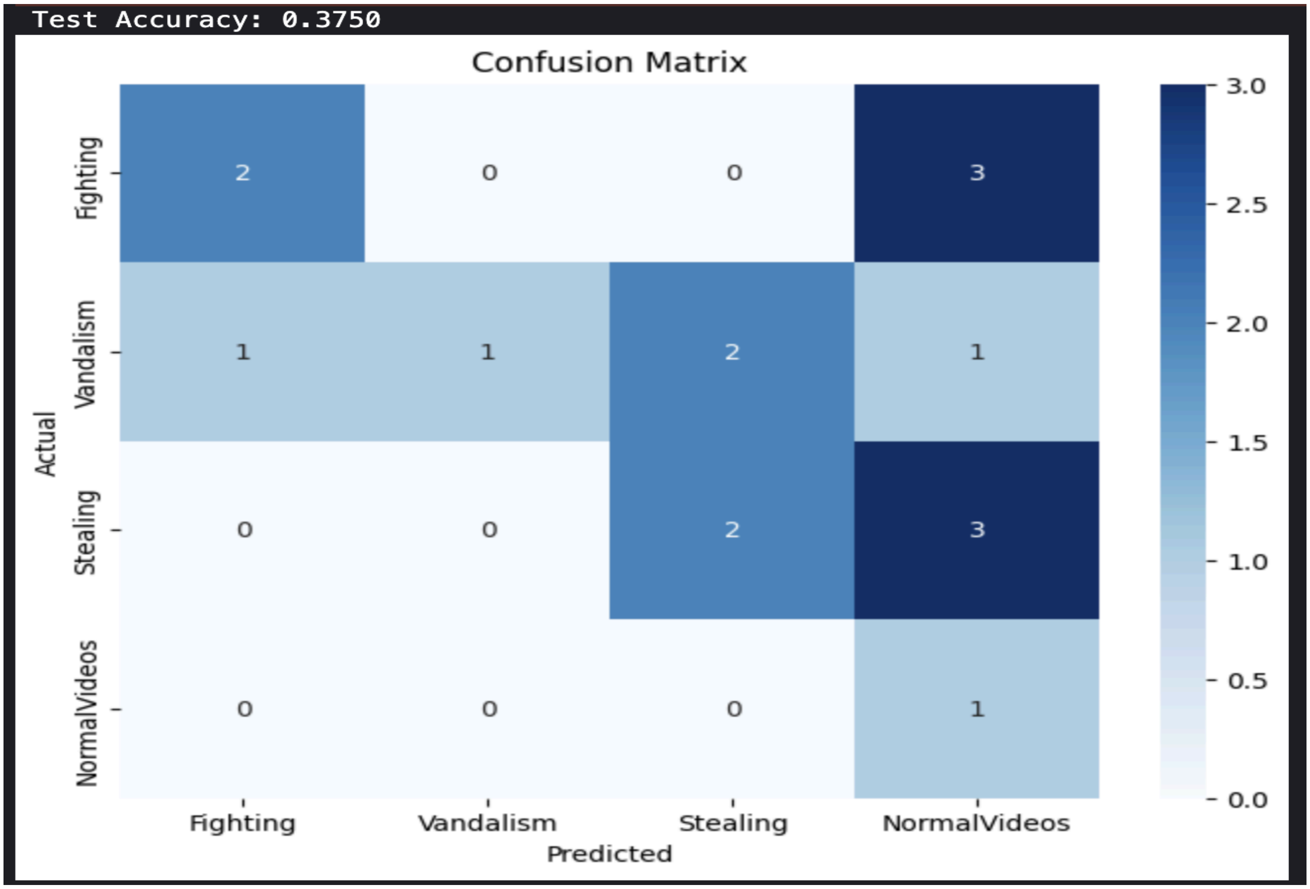
Validation Accuracy: 85.2%



```
Epoch 1/30 | Loss: 132.2351 | Train Acc: 0.8033 | Val Acc: 0.8325
Epoch 2/30 | Loss: 119.9663 | Train Acc: 0.8160 | Val Acc: 0.8426
Epoch 3/30 | Loss: 118.7247 | Train Acc: 0.8173 | Val Acc: 0.8579
Epoch 4/30 | Loss: 112.1483 | Train Acc: 0.8261 | Val Acc: 0.8426
Epoch 5/30 | Loss: 110.2511 | Train Acc: 0.8261 | Val Acc: 0.8376
Epoch 6/30 | Loss: 105.9946 | Train Acc: 0.8287 | Val Acc: 0.8477
Epoch 7/30 | Loss: 104.5661 | Train Acc: 0.8287 | Val Acc: 0.8426
Epoch 8/30 | Loss: 100.6136 | Train Acc: 0.8363 | Val Acc: 0.8325
Epoch 9/30 | Loss: 102.6999 | Train Acc: 0.8401 | Val Acc: 0.8528
Epoch 10/30 | Loss: 100.1822 | Train Acc: 0.8401 | Val Acc: 0.8426
Epoch 11/30 | Loss: 103.3291 | Train Acc: 0.8350 | Val Acc: 0.8325
Epoch 12/30 | Loss: 100.8316 | Train Acc: 0.8439 | Val Acc: 0.8426
Epoch 13/30 | Loss: 94.0540 | Train Acc: 0.8515 | Val Acc: 0.8325
Epoch 14/30 | Loss: 98.6906 | Train Acc: 0.8388 | Val Acc: 0.8325
Epoch 15/30 | Loss: 93.2442 | Train Acc: 0.8452 | Val Acc: 0.8071
Epoch 16/30 | Loss: 90.0073 | Train Acc: 0.8503 | Val Acc: 0.8020
Epoch 17/30 | Loss: 95.2784 | Train Acc: 0.8503 | Val Acc: 0.8071
Epoch 18/30 | Loss: 93.4852 | Train Acc: 0.8528 | Val Acc: 0.8376
Epoch 19/30 | Loss: 85.3285 | Train Acc: 0.8566 | Val Acc: 0.8173
Epoch 20/30 | Loss: 95.1797 | Train Acc: 0.8464 | Val Acc: 0.8274
Epoch 21/30 | Loss: 92.5294 | Train Acc: 0.8426 | Val Acc: 0.8426
Epoch 22/30 | Loss: 88.5743 | Train Acc: 0.8477 | Val Acc: 0.8223
Epoch 23/30 | Loss: 87.6615 | Train Acc: 0.8541 | Val Acc: 0.8020
Epoch 24/30 | Loss: 83.3415 | Train Acc: 0.8655 | Val Acc: 0.8376
Epoch 25/30 | Loss: 83.7287 | Train Acc: 0.8655 | Val Acc: 0.8579
Epoch 26/30 | Loss: 91.1190 | Train Acc: 0.8566 | Val Acc: 0.8071
Epoch 27/30 | Loss: 86.3830 | Train Acc: 0.8464 | Val Acc: 0.8376
Epoch 28/30 | Loss: 88.0168 | Train Acc: 0.8604 | Val Acc: 0.7868
Epoch 29/30 | Loss: 85.1084 | Train Acc: 0.8604 | Val Acc: 0.8173
Epoch 30/30 | Loss: 85.9057 | Train Acc: 0.8566 | Val Acc: 0.8223
✅ Model training complete and saved!
```

Fig 16: Training and Validation Accuracy

The model acquired a testing accuracy of 37.5% providing a massive scope for improvement as we proceed further.



```
Test Accuracy: 0.3750
```

Actual \ Predicted	Fighting	Vandalism	Stealing	NormalVideos
Fighting	2	0	0	3
Vandalism	1	1	2	1
Stealing	0	0	2	3
NormalVideos	0	0	0	1

Fig 17: Testing Accuracy

CHAPTER-9

CONCLUSION OF CAPSTONE PROJECT PHASE-2

Phase 2 of the Smart Reconnaissance System capstone project entailed architecting, designing, and integrating an intelligent watch system. Here, the system was architected on a modular and scalable architecture with network-level video anomaly detection and camera surveillance driven by AI. The most critical design artifacts such as the class diagram, state diagram, swimlane diagram, and external interface diagram were created to assist in envisioning and refining system interaction, workflows, and accountability among pieces.

System design was constructed to exhibit real-time responsiveness, automatic alerting, and escalation process and thus be useful in high-security applications. Construction of feedback-driven and event-driven mechanisms reduces downtime and detects faults or anomalies in real time. Design problems regarding future-proofing were also retained in terms of such variables as portability, reusability, maintainability, and compliance with external interfaces (e.g., SMS gateways, monitoring packages). As Phase 2 has already been achieved, the foundation is now set in place for full implementation, deployment, and testing for the following phase.

CHAPTER-10

PLAN OF WORK FOR CAPSTONE PROJECT PHASE - 3

The next phase of the capstone project will begin with the deployment of the entire Smart Reconnaissance System from the design and specification developed during Phase 2. This includes coding the initial set of modules such as the camera ping monitor system, video anomaly detection engine, auto-alert system, and the user dashboard. Each module will be deployed separately before becoming part of the system. External interfaces such as the SMS gateway and alarm tools will be made available in order to support real-time communication and alarm escalation.

When utilized as an operating system, considerable testing effort will be utilized in an effort to confirm operation and stability. Standalone units of the equipment will be tested in isolation in order to ensure that they function as needed, whereas integration testing will be utilized in order to confirm coordination between the units. It will try to check the system response for different loads of operation, i.e., concurrent streams of videos or trigger notifications. Machine learning-based anomaly detection models will learn how to adjust with real-world conditions such as fluctuating lights and crowd movement.

After technical validation, the system will be deployed to a trial environment—a campus location or simulated surveillance environment—and its quasi-real-world scenario performance measured. In this phase, one of the key performance measures, there will be accuracy monitoring of watch list hits, system availability, response time, and escalation effectiveness. The user interaction feedback will also be recorded later in the form of the dashboard to enable usability and reliability.

Lastly, the project team will be handed over all the project deliverables like system documentation, user manual, aThe last stage of the capstone project shall start with the deployment of the whole Smart Reconnaissance System from the design and specification worked out in Phase 2. This includes coding the initial modules like the camera ping monitor system, video anomaly detection engine, auto-alert system, and the user dashboard. The modules will be released individually prior to their incorporation into the system. External interfaces like the SMS gateway and monitoring tools will be installed to provide real-time exchange and alarm escalation.

For use as an operating system, the optimal amount of testing effort shall be utilized with a perspective towards demonstrating operation and stability. Individual units of equipment will be tested individually to verify that they work when required, while integration testing will be used in an effort to verify coordination among units. It will try to test the response of the system for various operation loads, i.e., streams of videos at the same time or quantities of trigger notifications. The anomaly detection models based on machine learning will learn to respond based on real-world situations like changing light and crowd directions.

Upon technically validating the system, it will be deployed to a test environment—a campus location or simulated surveillance setting—and its quasi-real-world scenario performance evaluated. During this phase, one of the main performance metrics, there will be watch list hit accuracy monitoring, system availability, response time, and escalation effectiveness. The user interaction feedback will also be captured later in the form of the dashboard to enable usability and reliability.

Finally, the capstone project team will be handed over all the project deliverables such as system documentation, user manual, and final report in final form addressing the design, implementation, test results, and lessons learned. A live system demonstration will be offered to showcase its capability.performance, and real-time capabilities. This last phase will confirm the real value of the solution and conclude the capstone project cycle. A final report in full form summarizing the design, implementation, test results, and lessons learned. A demonstration of the live system will be provided to emphasize its capability. This final phase will verify the real-world applicability of the solution and will finalize the capstone project cycle.

REFERENCES/BIBLIOGRAPHY

[1] W. Sultani, C. Chen, and M. Shah, “Real-world anomaly detection in surveillance videos,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 6479–6488. [Online]. Available: https://www.crcv.ucf.edu/projects/real-world/

[2] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, “Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes,” *Comput. Vis. Image Underst.*, vol. 172, pp. 88–97, 2018. [Online]. Available: https://github.com/StevenLiuWen/ano_pred_cvpr2018

[3] University of Minnesota, “UMN unusual crowd activity dataset,” [Online]. Available: http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi

APPENDIX A DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Acronyms

Acronym	Full Form
AI	Artificial Intelligence
CCTV	Closed-Circuit Television
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
IP	Internet Protocol
SMS	Short Message Service
SDLC	Software Development Life Cycle
UI	User Interface
API	Application Programming Interface
GPU	Graphics Processing Unit
SMTP	Simple Mail Transfer Protocol
NVR	Network Video Recorder
VMS	Video Management System
UCF	University of Central Florida
UMN	University of Minnesota

SQL	Structured Query language
GAN	Generative Adversarial Network
HTTP	HyperText Transfer Protocol

Definitions

Term	Definition
Anomaly Detection	Identifying unusual patterns or behaviors in video footage that deviate from normal activity.
Event-Driven Architecture	A design where the system reacts in real-time to events like device disconnection or detected threats.
Hybrid Surveillance	Combining video analytics and network/device monitoring for robust surveillance.
Ping	A signal sent to test connectivity and response time of networked devices like IP cameras.
Camera Health Monitoring	The process of tracking camera functionality, availability, and performance metrics.
Escalation Policy	Protocol for alerting higher authorities when an issue persists beyond a set time limit.

Rule-Based Classifier	A logic-based system to categorize issues based on rules instead of learned patterns.
Real-Time Alerts	Instant notifications triggered by faults or anomalies.

Abbreviations

Abbreviation	Expanded Form
ResNet	Residual Network (type of deep CNN)
Colab	Google Colaboratory
OpenCV	Open Source Computer Vision Library
Roboflow	CV tool for image labeling and augmentation
SQLite	Lightweight embedded relational database

1% Matches

1	Internet	ijaem.net	<1%
2	Internet	arxiv.org	<1%
3	Internet	ebin.pub	<1%
4	Internet	pure.tue.nl	<1%