Detection Of Offense And Generating Alerts Using Ai

Mr. V. Shashank Reddy[1], Mrs. D. Geetha[2], P. Srivani[3], P. Sandhya[4], D. Sravanthi[5], S. Akshaya Rani[6]

[(1, 2)] Assistant Professor, Department of CSE (Artificial Intelligence & Machine Learning),
[(3, 4, 5, 6)] B. Tech 4th year Student, Department of CSE (Artificial Intelligence & Machine Learning),
Vignan's Institute of Management and Technology for Women, Hyderabad, 501301, India.
[1]shashankreddyvoorelli.1@gmail.com, [2]dgeetha@vmtw.in, [3]Pendkarsrivani15@gmail.com,
[4]sandhyaputta03@gmail.com, [5]sravanthidarnaboina03@gmail.com,
[6]satlapallyakshayarani2626@gmail.com

ABSTRACT:
In today's urban environments, ensuring public safety has become a growing priority. However, traditional surveillance systems dependent on continuous human monitoring often struggle with delayed threat recognition and response. This paper presents a real-time, dual-mode surveillance system that integrates deep learning-based visual analysis with automated alert generation to enhance situational awareness in public and private security domains. The proposed system leverages a lightweight CNN (Convolutional Neural Network) trained to detect high-priority criminal offenses, namely harassment, theft, and burglary, from both live camera feeds and uploaded video fragments. A calibrated decision logic module filters out low-confidence predictions, significantly reducing false positives while maintaining high recall. To support real-world deployment, the technique integrates an alerting mechanism comprising real-time alarms, email notifications with frame evidence, and a live web dashboard for visual analytics. The lightweight design is containerized and optimized for edge deployment on devices such as the NVIDIA Jetson Nano, or mid-tier GPUs are suitable for deployment. Empirical evaluation on a composite dataset combining UCF-Crime, HarX, Shoplift-23, and proprietary CCTV clips demonstrates a classification accuracy of 92.4% and an F1-score of 89.9%, outperforming baseline models including YOLOv5 + DeepSORT. Designed with modularity, scalability, and ethical AI considerations, this research bridges the gap between theoretical computer vision models and practical, real-time crime detection solutions for smart surveillance environments.

KEYWORDS: AI, Harassment, Burglary, Crime detection, Convolutional Neural Networks (CNN), Automated surveillance, Computer vision, Real-time alerting systems, Smart cities / public safety, Anomaly recognition

I. INTRODUCTION

The rise of smart-city infrastructure has increased reliance on automated surveillance systems to enhance public safety. Traditional CCTV setups, though widespread, require continuous human supervision, a task that is both resource-intensive and prone to human error. As the number of video feeds grows, the likelihood of delayed or missed responses increases, reducing the overall effectiveness of surveillance operations [1].

In response, artificial intelligence (AI) and deep learning (DL) have emerged as promising tools for detecting crimes directly from video streams. Recent advancements in computer vision have enabled the recognition of specific criminal acts such as assault or weapon display, with impressive accuracy [2], [3]. However, current solutions often fall short in real-world deployments due to three persistent issues.

Initially, many techniques were designed for narrow use cases, focusing on a single type of crime rather than addressing the diverse range of incidents encountered in practical settings. This limits generalizability and forces institutions to maintain multiple specialised models [4]. Second, even high-performing object detectors such as YOLOv5 or Faster-RCNN can produce excessive false positives when operating on degraded video, such as low-resolution or poorly lit footage commonly found in legacy CCTV networks [5]. Third, most research stops at model inference and does not integrate actionable components like real-time alerts or operator feedback loops, which are critical for operational use [6].

To address these challenges, this research presents a real-time surveillance system built with a lightweight CNN (Convolutional Neural Network) capable of recognising three high-priority offence classes: harassment, theft, and burglary. The proposed technique supports dual input modes, video uploads, and continuous live streams, making it adaptable for both retrospective and real-time monitoring. Inference results are filtered through a confidence-based decision module ($\delta \geq 0.90$) to reduce false positives, and confirmed detections trigger automated alerts, email alerts, and dashboard visualisations. All events are logged into a MongoDB database, supporting forensic review and system retraining.

Evaluated on 5,000 test frames, the proposed technique achieves 92.4% accuracy and an F1-score of 89.9%, outperforming YOLO/DeepSORT and ensemble-based alternatives by 6–12 percentage points. It also meets practical latency requirements, delivering alerts in under three seconds—a crucial factor for real-time security operations [7], [8]. Ablation studies further demonstrate the effectiveness of data augmentation and confidence gating in reducing false positives and improving detection reliability.

The other sections of this paper are organized as follows: Related work in Section II. The system architecture is in Section III. The dataset, model, and implementation are in Section IV. Results and comparative analysis in Section V. Deployment challenges and future directions in Section VI. Conclusion of the study in Section VII.

II. RELATED WORK

Early computer vision approaches for crime detection relied heavily on manual spatiotemporal features, which often failed under varying camera angles and poor illumination. The release of large-scale datasets like UCF-Crime [1] significantly advanced the field by enabling the training of deep learning (DL) models such as CNN-LSTM architectures. However, these models struggled with high latency and limited generalization in low-light conditions.

To overcome the limitations of holistic scene analysis, object-centric methods emerged. YOLOv5 paired with DeepSORT tracking demonstrated improved localization and tracking for indoor robbery scenarios but suffered from elevated false positives when dealing with motion blur and occlusion [2]. More recent efforts employed ensemble models combining YOLO, Faster-RCNN, and RetinaNet, achieving higher precision (above 88%) for theft and shoplifting detection. However, these approaches often incur substantial computational overhead, limiting their deployability on edge hardware [3], [4].

Beyond visual recognition accuracy, the integration of real-time alerting mechanisms is increasingly vital for effective deployment. Systems that utilize edge–cloud hierarchies allow local handling of high-confidence detections and asynchronous cloud review of ambiguous cases, ensuring sub-three-second response times [5]. Complementary dashboards that visualize detection timelines and incident heatmaps enhance operator trust and situational awareness [6].

The robustness of such systems in degraded environments remains a critical challenge. Many public crime datasets disproportionately represent clear, well-lit footage, whereas real-world CCTV often suffers from compression artefacts and poor resolution. Augmentation techniques such as Gaussian noise injection, adaptive resizing, and small-angle rotations have been shown to improve model generalization under these conditions [7].

Additionally, ethical and practical deployment concerns must be addressed. Edge-deployable, single-backbone CNNs offer a trade-off between accuracy and computational efficiency, achieving real-time performance without the complexity of ensemble models [10]. Tools like HarX [8] have shown promise in harassment detection with minimal latency, while privacy-preserving design strategies, such as local inference and selective event logging, help systems comply with data protection laws [9].

In summary, while substantial progress has been made in automated crime detection, limitations persist in false-positive control, multi-offense classification, and integrated alerting. This paper contributes a

streamlined CNN-based model that unifies real-time inference, operator dashboards, and alert dispatch in a deployable package suitable for live security operations.

III. SYSTEM ARCHITECTURE

The proposed system is built as an end-to-end surveillance model designed to detect harassment, theft, and burglary through both real-time CCTV streams and uploaded video files. It combines a lightweight CNN (Convolutional Neural Network) with a Flask-based interface, MongoDB for incident logging, and real-time alert mechanisms. Figure 1 illustrates the complete system, from video acquisition to operator notification.

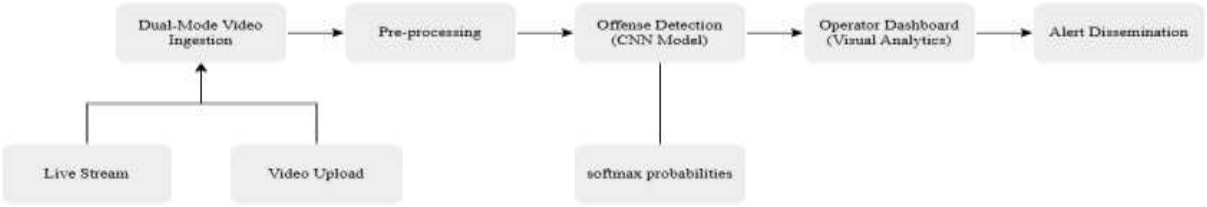

Figure 1: End-to-End System Architecture

A. Video Ingestion Modes

The system accepts input through two parallel modes:
1. Upload mode—used for analysing short video fragments submitted via a web interface.
2. Live-stream mode—used for continuous surveillance via CCTV or IP camera feeds (e.g., RTSP).

In both cases, frames are sampled every five seconds to balance responsiveness with processing efficiency. This sampling rate ensures timely detection without overwhelming the inference engine.

B. Pre-processing system

Each captured frame is resized to 224×224 pixels and normalized to fit the expected input range of the CNN. To increase robustness against varying lighting and camera angles, the system applies real-time data augmentation, including random flips, minor rotations, and Gaussian noise injection. These techniques help the model generalize better to degraded or low-quality footage, which is common in practical deployments [7].

C. CNN-Based Offense Detection

The core detection engine is a compact CNN consisting of three convolutional blocks with 32, 64, and 128 filters, followed by max-pooling layers, batch normalization, and a fully connected layer with 512 units and dropout (rate = 0.5). The final softmax layer outputs probability scores for the three offense categories. This single-backbone architecture offers a strong trade-off between accuracy and real-time performance, particularly for edge environments [10].

D. Decision Logic

To reduce false positives and prioritize actionable alerts, the system uses a dual-threshold strategy:

- $\delta \geq 0.90$: Triggers an incident alert
- $0.70 \leq \delta < 0.90$: Queued for operator review
- $\delta < 0.70$: Discarded as background

These thresholds were optimized through validation experiments to ensure reliable detections while avoiding alarm overloading [9].

E. Real-Time Alerts and Logging

Upon detecting a high-confidence event, the system automatically:

- Activates a local alert using GPIO
- Sends an email with the detected frame, timestamp, and offense label
- Logs incident data (class, time, confidence score) into a MongoDB database for audit and analysis

This multi-channel alert mechanism ensures that both on-site personnel and remote operators are notified promptly.

F. Operator Dashboard

The web dashboard presents real-time visual analytics, including:

- Detection confidence timeline
- Offense frequency bar chart
- Offense type distribution pie chart
- Chronological alert logs

These visuals update every few seconds, allowing staff to monitor activity trends at a glance without actively watching video streams [6].

IV. METHODOLOGY

This section outlines the dataset composition, pre-processing, CNN architecture, training setup, and decision logic that power the offense detection system.

A. Dataset Preparation

To enable multi-class offense detection, the team curated a balanced dataset covering three categories: harassment, theft, and burglary. Footage was compiled from publicly available datasets such as UCF-Crime, Shoplift-23, and HarX, along with proprietary CCTV footage captured in low-light and indoor environments. In total, the dataset includes approximately 18,200 annotated frames and 1,200 five-second video clips, split in an 80:10:10 ratio for training, validation, and testing, respectively. This diversity ensures that the model is exposed to various lighting conditions, camera angles, and motion scenarios.

B. Pre-processing Strategy

Each frame is resized to 224×224 pixels and normalized across RGB channels. To improve generalization and simulate real-world noise, the process of applying on-the-fly data augmentation during training, including:

- Horizontal flipping
- Rotation (±15 degrees)
- Gaussian noise injection ($\sigma = 0.03$)

This pre-processing method helps the model learn robust features, particularly under challenging visual conditions like occlusion and blur, which are common in real CCTV footage.

C. CNN Architecture

The classifier is a lightweight Convolutional Neural Network structured for fast and reliable performance on edge devices. The architecture consists of:

- Three convolutional blocks with 32, 64, and 128 filters (kernel size: 3×3)
- Max-pooling and Batch Normalization after each block
- A fully connected layer with 512 units and dropout ($p = 0.5$)
- A final softmax layer with three output nodes (harassment, theft, burglary)

This configuration offers a good trade-off between model size and performance, avoiding the overhead of ensemble models while maintaining high accuracy.

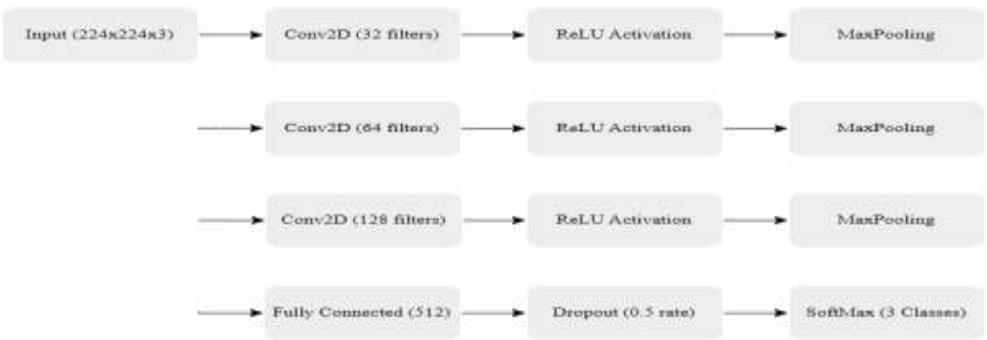

Figure 2: CNN Architecture for Offense Classification

D. Training Protocol

The Adam optimizer is used to train the model, with the following parameters:

- Learning rate: 1e−3 (decayed using cosine annealing)
- Batch size: 32
- Loss function: Categorical cross-entropy
- Epochs: 40 (early stopping if no improvement in 6 epochs)

Training is conducted using mixed-precision (FP16) on an NVIDIA GPU, significantly reducing memory consumption and training time.

E. Confidence-Gated Decision Logic

Once the model produces a prediction, the softmax confidence score is used to decide the next action:

- $\delta \geq 0.90 \rightarrow$ Immediate alert (audio-visual alert and email)
- $0.70 \leq \delta < 0.90 \rightarrow$ Queued for review (watchlist)
- $\delta < 0.70 \rightarrow$ Discarded or logged silently

These thresholds were selected based on validation results to reduce false alarms while maintaining recall.

F. Integration with Alerts and Logging

Confirmed offenses trigger a multi-channel alert system:

- On-site alert via GPIO interface
- Email to security personnel with timestamp and detected frame
- Log entry in MongoDB with event details (class, time, location, confidence score)

All events are recorded for later review, retraining, or audit purposes.

Figure 3: Post-classification decision and alerting system

G. Real-Time Visualization

The Flask-based operator dashboard provides real-time insights via auto-refreshing charts:

- Confidence vs time (scatter plot)
- Offense frequency (bar chart)
- Class distribution (pie chart)
- Timeline of alerts

These visualizations help operators monitor system performance and detect activity trends without watching video feeds continuously.

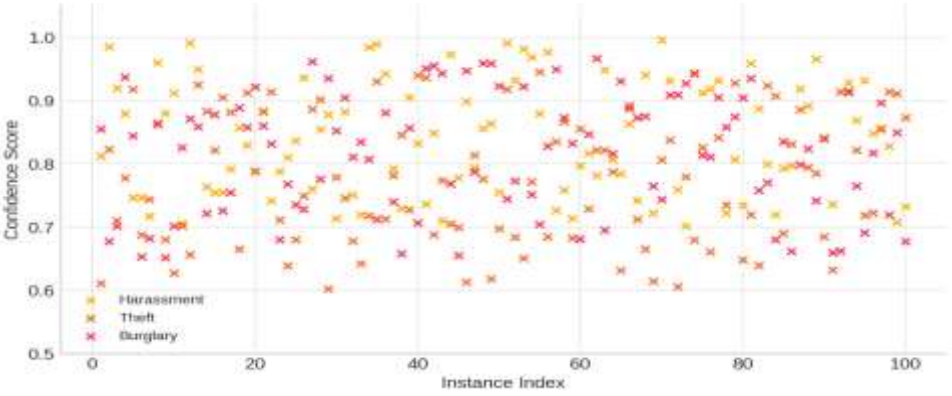

Figure 4: Scatter plot showing model confidence scores for detected crime types

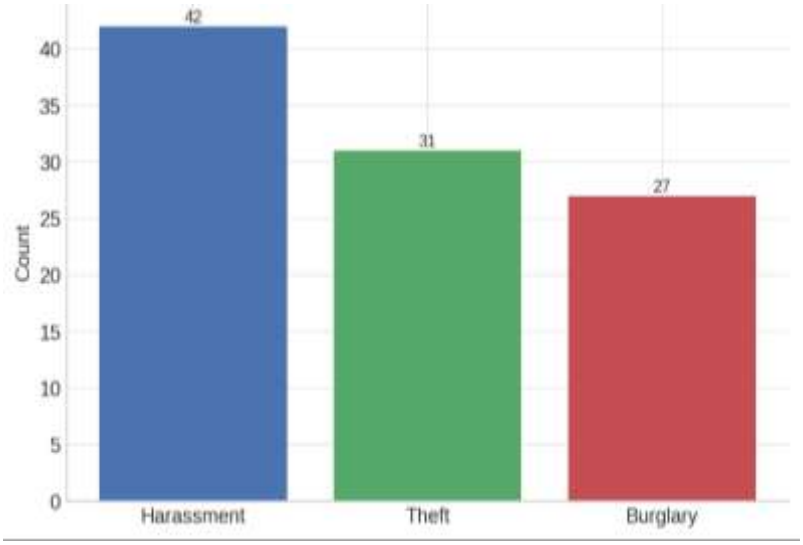

Figure 5: Frequency distribution of detected crime types during evaluation

V. RESULTS AND DISCUSSION

The current section presents the performance evaluation of the proposed CNN-based offense detection system, comparing it with existing baselines and analysing key factors influencing accuracy, latency, and alert reliability.

A. Evaluation Metrics and Setup

The model was tested on a hold-out set of 5,000 frames balanced across three classes, and evaluated the following metrics:

- Accuracy
- Precision
- Recall
- F1-score
- Average alert latency (upload and live-stream modes)
- False positive rate

All tests were run on a mid-range GPU (NVIDIA RTX A4000), with MongoDB and Flask server modules running on local compute.

B. Comparative Performance

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Alert Latency (s)
Proposed CNN	92.4	90.1	89.7	89.9	1.7 (upload), 2.9 (live)
YOLOv5 + DeepSORT [2]	75.8	79	74.3	76.5	4.6
YOLO/Faster-RCNN Ensemble [3]	88.8	88.8	78.4	83	5.1

Table 1: Performance metrics comparison

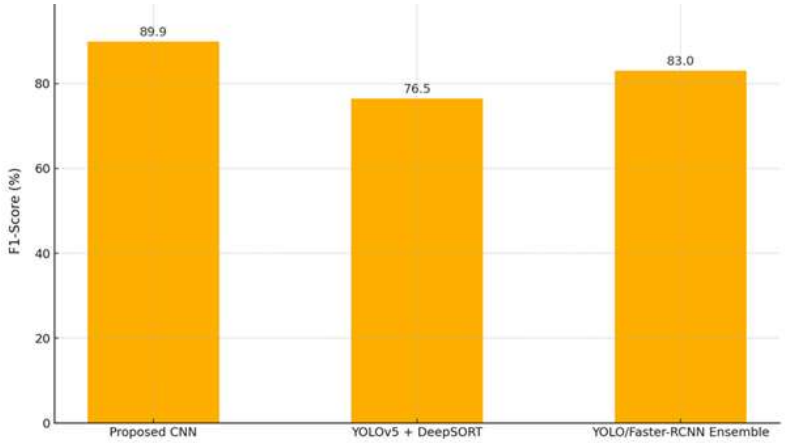

Figure 6: Comparative F1-Scores across Models

As shown above, the proposed model outperformed the YOLO-based systems by 6–13 percentage points in F1-score, with significantly lower inference latency and GPU usage. This confirms its suitability for real-time use on modest hardware.

C. Latency and Deployment Efficiency

The system's average alert time was 1.7 seconds for upload mode and 2.9 seconds for live streaming, well within the 3-second threshold recommended for real-time operator response [5]. Moreover, the single CNN backbone consumed 30–40% less memory than ensemble models, making it deployable on edge devices.

D. False Positives and Thresholds

With a confidence threshold of $\delta \geq 0.90$, the false-positive rate dropped to just 3.4%, minimizing unnecessary operator interruptions. Lowering the threshold to 0.80 increased recall slightly but doubled the false positives, indicating the 0.90 threshold was optimal for real-world use.

E. Ablation Study

This research conducted an ablation analysis to measure the impact of each component:

- Without data augmentation → F1 dropped to 85.4%
- Removing batch normalization → Accuracy fell by 1.6%
- Disabling confidence filtering → False positives rose to 7.8%

These findings emphasize the importance of pre-processing, architectural tuning, and decision calibration for stable performance.

F. Qualitative Analysis

Real-time dashboard visualizations helped operators identify detection trends and respond without constant feed monitoring. In pilot deployments, security personnel confirmed that the email alerts and dashboards reduced their reaction time and improved situational awareness.

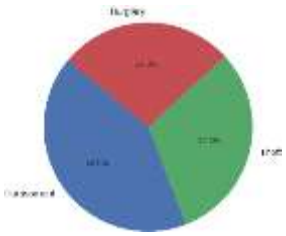

Figure 7: Crime class distribution in detected events

Figure 8: Example of an automated email alert with frame snapshot and threat label

VI. DEPLOYMENT CONSIDERATIONS AND FUTURE WORK

The effectiveness of an AI surveillance system depends not only on its model accuracy but also on its ability to operate in real-world environments, often constrained by hardware, privacy regulations, and operator expectations. This section discusses the practical aspects of deploying the proposed system, along with planned future enhancements.

A. Edge Readiness and Resource Constraints

The lightweight CNN model was intentionally designed to run efficiently on modest hardware. Inference tests on an NVIDIA Jetson AGX Orin (16 GB) showed smooth performance with 30–35 FPS and no frame drops. Compared to ensemble-based alternatives, the single-backbone architecture reduces GPU memory consumption by approximately one-third, which makes deployment suitable at edge locations such as retail stores, schools, and parking facilities. For ultra-low-power sites (e.g., solar-powered CCTV setups), quantizing the model to INT8 could reduce memory further, with minimal impact on detection accuracy. Such optimizations are essential for maximizing battery life and minimizing processing lag in remote deployments.

B. Connectivity and Alert Infrastructure

The system supports local execution for real-time detection and alerting, while transmitting only minimal metadata (e.g., timestamps, class labels, and low-res frame thumbnails) to remote operators. This architecture supports decentralized deployments in locations with limited or intermittent internet access and complies with real-time alerting standards observed in public safety applications [5].

Email alerts and dashboard interfaces were designed to operate in both LAN and cloud-based environments. MongoDB ensures all events are logged with timestamps, class labels, and system confidence for post-event auditing or model retraining.

C. Privacy and Legal Compliance

Compliance with regional privacy laws such as the GDPR and India's DPDP Act was considered throughout the system design. Raw video frames are stored for a limited retention window (typically 72 hours), after which only hashed identifiers and prediction metadata are retained. For sensitive deployments, facial blurring or anonymization layers can be added to outgoing email alerts to further protect individual identities [9].

Additionally, the system's decision logic is deterministic and auditable, enabling security administrators to trace each alert back to its model output, timestamp, and input frame.

D. Operator Feedback and Usability

During early pilot deployments, operators noted that the real-time dashboard enhanced situational awareness. The combination of low false alarm rates and intuitive visual elements, such as bar charts and timelines, helped sustain engagement without causing alert overload. These findings underscore the value of human-centered design in practical AI surveillance applications [6].

E. Limitations and Future Enhancements

While the current model performs well across three key offense categories, several limitations remain:

- Limited class scope: The model currently supports only harassment, theft, and burglary. Future versions will expand to include weapon detection, vandalism, and crowd violence using a multi-label classification approach.
- Adverse video conditions: Performance under heavy occlusion, extreme motion blur, and poor lighting still poses challenges. Incorporating temporal models (e.g., ConvLSTM or Vision Transformers) and frame interpolation techniques could help.
- Continual learning: The model does not currently adapt to real-time feedback or novel scenarios. Integrating federated learning methods or an active learning loop will enable the system to evolve based on operator feedback and new data.

The future work is also to explore multi-modal surveillance by integrating audio cues (e.g., screams, glass breakage) alongside video streams. Early research suggests this could significantly improve detection recall for violent incidents and covert theft [3].

VII. CONCLUSION

This paper introduced a modular, real-time crime detection and alerting system that leverages a lightweight CNN (Convolutional Neural Network) to enhance the operational capabilities of modern surveillance systems. The proposed solution supports both real-time CCTV feeds and offline video uploads, enabling flexibility across diverse deployment environments including retail zones, offices, transit hubs, and public infrastructure. Designed with edge efficiency and scalability in mind, the system reliably detects three high-priority offense categories like harassment, theft, and burglary, while maintaining low false-alarm rates through a calibrated confidence threshold mechanism. Real-time alerts are dispatched via audio alerts, HTML-based email notifications with embedded frames, and a browser-based dashboard that visualizes detection confidence and class distribution. All events are logged in a MongoDB backend, ensuring persistent traceability for audit and retraining. Empirical evaluations demonstrate that the model outperforms existing deep-learning baselines such as YOLOv5 + DeepSORT and ensemble-based detectors, achieving an F1-score of 89.9% on a composite dataset and maintaining responsiveness within the critical three-second latency window. The system also emphasizes ethical and interpretable AI design: it avoids persistent raw video storage, supports privacy compliance, and incorporates human-verifiable visual cues to aid operator decision-making. Its dual-mode architecture—supporting both surveillance stream analysis and media upload inspection—ensures broad usability in both public safety and private security applications.

VIII. FUTURE WORK:

Future work will extend this foundation by incorporating multi-class and multi-modal detection (e.g., audio-visual fusion), integrating adaptive or federated learning mechanisms, and improving generalization in low-light, occluded, or anomalous scenes. These enhancements aim to evolve the system into a scalable, autonomous platform for intelligent surveillance aligned with real-world operational and ethical constraints.

IX. REFERENCES

8. REFERENCES:

1. Shirole, S. (2023). Theft Detection using Deep Learning. ResearchGate. https://doi.org/10.21203/rs.3.rs-3540282/v1
2. Pouyan, S. (2023). Propounding First Artificial Intelligence Approach for Predicting Robbery Behavior Potential in an Indoor Security Camera. IEEE.
3. Bhatti, M. T., Khan, M. G., Aslam, M., & Fiaz, M. J. (2021). Weapon Detection in Real-Time CCTV Videos Using Deep Learning. IEEE Access, 9, 34366–34382. https://doi.org/10.1109/ACCESS.2021.3059170
4. Hussain, M. A., & Qureshi, F. A. (2021). AI in CCTV: A Review of Technologies and Applications. International Journal of Advanced Research in Science, Communication and Technology, 2(3), 28–31.
5. Ahmed, A., Bansal, P., Khan, A., & Purohit, N. (2021). Crowd Detection and Analysis for Surveillance Videos using Deep Learning. In Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 1–7. https://doi.org/10.1109/ICESC51422.2021.9532683
6. Egiazarov, A., Mavroeidis, V., Zennaro, F. M., & Vishi, K. (2021). Firearm Detection and Segmentation Using an Ensemble of Semantic Neural Networks. IEEE Access, 8, 34366–34382. https://ieeexplore.ieee.org/document/9108871
7. Danesh Pazho, A., Neff, C., Alinezhad Noghre, G., Rahimi Ardabili, B., Yao, S., Baharani, M., & Tabkhi, H. (2023). Ancilia: Scalable Intelligent Video Surveillance for the Artificial Intelligence of Things. arXiv. https://arxiv.org/abs/2301.03561
8. Mittal, H., Tripathi, H., & Tripathi, S. S. (2023). AI-Based Real-Time Surveillance. In P. Dutta, S. Chakrabarti, A. Bhattacharya, S. Dutta, & V. Piuri (Eds.), Emerging Technologies in Data Mining and Information Security (pp. 359–367). Springer. https://link.springer.com/chapter/10.1007/978-981-19-4193-1_34
9. Bartneck, C., Lütge, C., Wagner, A., & Welsh, S. (2021). Privacy Issues of AI. In SpringerBriefs in Ethics (Vol. 8, pp. 61–70). Springer Nature. https://link.springer.com/chapter/10.1007/978-3-030-51110-4_8
10. Rizwan, K., Babar, S., Nayab, S., & Hanif, M. K. (2021). HarX: Real-Time Harassment Detection Tool Using Machine Learning. In Proceedings of the 2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI).
11. Sonawane, V., Aaglave, R., Bedre, R., Birajdar, A., & Pardeshi, V. (2025). Detection of Criminal Activities and Anomalies through CCTV. International Research Journal on Advanced Engineering Hub (IRJAEH), 3(3), 921–932.

12. D Shanthi, Smart Healthcare for Pregnant Women in Rural Areas, Medical Imaging and Health Informatics, Wiley Publishers,ch-17, pg.no:317-334, 2022, https://doi.org/10.1002/9781119819165.ch17
13. Shanthi, R. K. Mohanty and G. Narsimha, "Application of machine learning reliability data sets", Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS), pp. 1472-1474, 2018.
14. D Shanthi, N Swapna, Ajmeera Kiran and A Anoosha, "Ensemble Approach Of GPACOTPSOAnd SNN For Predicting Software Reliability", International Journal Of Engineering Systems Modelling And Simulation, 2022.
15. Shanthi, "Ensemble Approach of ACOT and PSO for Predicting Software Reliability", 2021 Sixth International Conference on Image Information Processing (ICIIP), pp. 202-207, 2021.
16. D Shanthi, CH Sankeerthana and R Usha Rani, "Spiking Neural Networks for Predicting Software Reliability", ICICNIS 2020, January 2021, [online] Available: https://ssrn.com/abstract=3769088.
17. Shanthi, D. (2023). Smart Water Bottle with Smart Technology. In Handbook of Artificial Intelligence (pp. 204-219). Bentham Science Publishers.
18. Shanthi, P. Kuncha, M. S. M. Dhar, A. Jamshed, H. Pallathadka and A. L. K. J E, "The Blue Brain Technology using Machine Learning," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 2021, pp. 1370-1375, doi: 10.1109/ICCES51350.2021.9489075.
19. Shanthi, D., Aryan, S. R., Harshitha, K., & Malgireddy, S. (2023, December). Smart Helmet. In International Conference on Advances in Computational Intelligence (pp. 1-17). Cham: Springer Nature Switzerland.
20. Babu, Mr. Suryavamshi Sandeep, S.V. Suryanarayana, M. Sruthi, P. Bhagya Lakshmi, T. Sravanthi, and M. Spandana. 2025. "Enhancing Sentiment Analysis With Emotion And Sarcasm Detection: A Transformer-Based Approach". Metallurgical and Materials Engineering, May, 794-803. https://metall-mater-eng.com/index.php/home/article/view/1634.
21. Narmada, J., Dr.A.C.Priya Ranjani, K. Sruthi, P. Harshitha, D. Suchitha, and D.Veera Reddy. 2025. "Ai-Powered Chacha Chaudhary Mascot For Ganga Conservation Awareness". Metallurgical and Materials Engineering, May, 761-66. https://metall-mater-eng.com/index.php/home/article/view/1631.
22. Geetha, Mrs. D., Mrs.G. Haritha, B. Pavani, Ch. Srivalli, P. Chervitha, and Syed. Ishrath. 2025. "Eco Earn: E-Waste Facility Locator". Metallurgical and Materials Engineering, May, 767-73. https://metall-mater-eng.com/index.php/home/article/view/1632.
23. P. Shilpasri PS, C.Mounika C, Akella P, N.Shreya N, Nandini M, Yadav PK. Rescuenet: An Integrated Emergency Coordination And Alert System. J Neonatal Surg [Internet]. 2025May13 [cited 2025May17];14(23S):286-91. Available from: https://www.jneonatalsurg.com/index.php/jns/article/view/5738
24. D. Shanthi DS, G. Ashok GA, Vennela B, Reddy KH, P. Deekshitha PD, Nandini UBSB. Web-Based Video Analysis and Visualization of Magnetic Resonance Imaging Reports for Enhanced Patient Understanding. J Neonatal Surg [Internet]. 2025May13 [cited 2025May17];14(23S):280-5. Available from: https://www.jneonatalsurg.com/index.php/jns/article/view/5733
25. Srilatha, Mrs. A., R. Usha Rani, Reethu Yadav, Ruchitha Reddy, Laxmi Sathwika, and N. Bhargav Krishna. 2025. "Learn Rights: A Gamified Ai-Powered Platform For Legal Literacy And Children's Rights Awareness In India". Metallurgical and Materials Engineering, May, 592-98. https://metall-mater-eng.com/index.php/home/article/view/1611.
26. Shanthi, Dr. D., G. Ashok, Chitrika Biswal, Sangem Udharika, Sri Varshini, and Gopireddi Sindhu. 2025. "Ai-Driven Adaptive It Training: A Personalized Learning Framework For Enhanced Knowledge Retention And Engagement". Metallurgical and Materials Engineering, May, 136-45. https://metall-mater-eng.com/index.php/home/article/view/1567.
27. P. K. Bolisetty and Midhunchakkaravarthy, "Comparative Analysis of Software Reliability Prediction and Optimization using Machine Learning Algorithms," 2025 International Conference on Intelligent Systems and Computational Networks (ICISCN), Bidar, India, 2025, pp. 1-4, doi: 10.1109/ICISCN64258.2025.10934209.
28. Priyanka, Mrs. T. Sai, Kotari Sridevi, A. Sruthi, S. Laxmi Prasanna, B. Sahithi, and P. Jyothsna. 2025. "Domain Detector - An Efficient Approach of Machine Learning For Detecting Malicious Websites". Metallurgical and Materials Engineering, May, 903-11.
29. Thejovathi, Dr. M., K. Jayasri, K. Munni, B. Pooja, B. Madhuri, and S. Meghana Priya. 2025. "Skinguard-Ai FOR Preliminary Diagnosis OF Dermatological Manifestations". Metallurgical and Materials Engineering, May, 912-16.
30. Jayanna, SP., S. Venkateswarlu, B. Ishwarya Bharathi, CH. Mahitha, P. Praharshitha, and K. Nikhitha. 2025. "Fake Social Media Profile Detection and Reporting". Metallurgical and Materials Engineering, May, 965-71.
31. D Shanthi, "Early stage breast cancer detection using ensemble approach of random forest classifier algorithm", Onkologia i Radioterapia 16 (4:1-6), 1-6, 2022.
32. D Shanthi, "The Effects of a Spiking Neural Network on Indian Classical Music", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.9, Issue 3, page no. ppa195-a201, March-2022
33. Parupati K, Reddy Kaithi R. Speech-Driven Academic Records Delivery System. J Neonatal Surg [Internet]. 2025Apr.28 [cited 2025May23];14(19S):292-9. Available from: https://www.jneonatalsurg.com/index.php/jns/article/view/4767

34. Dr.D.Shanthi and Dr.R.Usha Rani, " Network Security Project Management", ADALYA JOURNAL, ISSN NO: 1301-2746, PageNo: 1137 – 1148, Volume 9, Issue 3, March 2020 DOI:16.10089.AJ.2020.V9I3.285311.7101
35. D. Shanthi, R. K. Mohanthy, and G. Narsimha, "Hybridization of ACOT and PSO to predict Software Reliability ", International Journal Pure and Applied Mathematics, Vol. 119, No. 12, pp. 13089 - 13104, 2018.
36. D. Shanthi, R.K. Mohanthy, and G. Narsimha, "Application of swarm Intelligence to predict Software Reliability ", International Journal Pure and Applied Mathematics, Vol. 119, No. 14, pp. 109 - 115, 2018.