Real Time Security Monitoring with ML And Sending Alerts

***Mr. Karthiban R*[1]*, Ms. Monika P*[2]*, Ms. B M Sahana*[3]*, Ms. Sandhiya C*[4]*, Ms. Sandhiya S*[5]**

[1]*Assistant Professor, Computer Science Engineering (Cyber Security), Sri Shakthi Institute and Engineering and Technology, Coimbatore, India*

[2]*IInd Year Student, Computer Science Engineering (Cyber Security), Sri Shakthi Institute and Engineering and Technology, Coimbatore, India*

[3]*IInd Year Student, Computer Science Engineering (Cyber Security), Sri Shakthi Institute and Engineering and Technology, Coimbatore, India*

[4]*IInd Year Student, Computer Science Engineering (Cyber Security), Sri Shakthi Institute and Engineering and Technology, Coimbatore, India*

[5]*IInd Year Student, Computer Science Engineering (Cyber Security), Sri Shakthi Institute and Engineering and Technology, Coimbatore, India*

***Mr. Karthiban R**
E-Mail Id: rkarthiban@siet.ac.in*

ABSTRACT

In the era of increasing security challenges, traditional surveillance systems are often limited by manual monitoring and delayed response. This project, AlertCam, presents a smart security solution that leverages machine learning to enable real-time monitoring and automated threat detection. By using live video feeds and intelligent algorithms such as facial recognition and motion analysis, the system can accurately identify unauthorized access or suspicious behavior. Upon detection, it immediately sends alerts through SMS or cloud-based notifications to ensure swift action. AlertCam is a scalable and efficient security system suitable for campuses, homes, and workplaces, aiming to enhance safety through automation and intelligent decision-making.

Keywords: *Real-time Monitoring, AlertCam, Machine Learning, Facial Recognition, Automated Alert, Smart campus security.*

1. INTRODUCTION

In today's rapidly evolving technological landscape, ensuring the safety and security of institutional and organizational environments has become increasingly critical. Conventional surveillance systems often rely on manual monitoring or rudimentary access control mechanisms, which are not only labour-intensive but also prone to human error and delayed responses. To address these limitations, the adoption of intelligent systems that leverage machine learning and computer vision technologies has gained significant traction. Among these, facial recognition stands out as a powerful tool for non-intrusive and real-time identification of individuals, offering a seamless and automated layer of security.

This paper presents the design and implementation of a real-time facial recognition system powered by machine learning algorithms, specifically tailored to enhance the surveillance capabilities of educational campuses and secure facilities. The system employs a structured workflow that begins with capturing facial data through a system-integrated camera, followed by preprocessing and dataset generation. A custom machine learning model is trained to accurately recognize known individuals, and the system is deployed to monitor live video feeds continuously. Upon detecting a face that does not match the trained dataset, it instantly issues alerts via SMS using Twilio and emails the intruder's snapshot to the designated authority, enabling timely and effective responses.

By moving away from reliance on prebuilt libraries and focusing on a customized ML-driven approach, the project achieves greater control, interpretability, and academic relevance. The system demonstrates high accuracy, precision, and recall under real-time conditions, even with varying facial expressions and environmental settings. Furthermore, its modular design allows for future expansion, including integration with CCTV streams and cloud-based analytics. This paper underscores the potential of machine learning in building proactive, intelligent security solutions that are both scalable and adaptable to diverse operational environments.

2. LITERATURE SURVEY

Recent advancements in machine learning (ML) and deep learning (DL) have significantly transformed the landscape of surveillance and intrusion detection systems. Saranya et al. [1] introduced a secure TinyML-powered framework for the Internet of Medical Things (MIoT), highlighting the viability of lightweight ML for real-time emergency alerts and intrusion detection in resource-constrained environments. Similarly, Mukto et al. [2] proposed a crime monitoring system utilizing DL, demonstrating strong real-time accuracy in detecting criminal activities through visual data analysis. Multiple studies have focused on optimizing video surveillance systems for timely response. Kumar et al. [3] and Sreejith et al. [4] developed real-time video alert architectures that effectively balance sensitivity and performance using ML-based motion and anomaly detection. Thomas et al. [5] advanced this further by incorporating crowd behavior and anomaly detection using DL algorithms, proving its utility in public surveillance and emergency management. Additionally, Selvi et al. [6] employed enhanced convolutional neural networks (CNNs) to identify suspicious actions in surveillance video with high precision. End-to-end surveillance pipelines such as E2E- VSDL by Gandapur [7] demonstrated the benefits of integrated DL models for autonomous criminal activity detection. Rahimi et al. [8] addressed both technical and ethical dimensions of AI-based surveillance, emphasizing policy frameworks for public safety. Use cases beyond public safety—like emotion-aware baby monitoring systems [9] and automotive safety via facial detection [10]— underscore the versatility of ML in real-time monitoring applications. On the cybersecurity front, real-time intrusion detection systems (IDS) have made progress through explainable AI and efficient architectures. Chen et al. [11] and Zhou et al. [12] presented hardware-accelerated and transformer- based IDS solutions that offer interpretability and speed. Vishwakarma and Kesswani [13] demonstrated the application of deep neural networks (DNNs) for detecting novel threats in IoT networks, while Kale et al. [14] and Laghrissi et al. [15] showed that hybrid models and LSTM-based frameworks can achieve high anomaly detection rates across varied attack scenarios. Collectively, the reviewed literature confirms that ML and DL methods are not only effective in detecting anomalies and intrusions across domains but also capable of real-time deployment in constrained environments. These studies provide a strong foundation for developing lightweight, accurate, and responsive ML- based facial recognition systems for surveillance applications.

3. PROPOSED SYSTEM

The Real-Time Security Monitoring System is an advanced, machine learning-based facial recognition solution designed to enhance institutional security. It integrates with existing CCTV infrastructure (RTSP/HTTP streams) to capture and process live video, performing face detection and preprocessing. A core innovation is its custom-trained ML model, built from scratch for transparency, customizability, and educational value, distinguishing it from conventional pre-built solutions. This model extracts and compares facial embeddings against a secure database of authorized individuals. Upon detecting an unfamiliar or unauthorized face, the system immediately triggers automated SMS alerts via Twilio API and emails a snapshot of the intruder to administrators. Designed for scalability and validated by strong performance metrics, this system provides a robust, adaptable, and proactive solution to modern surveillance challenges.

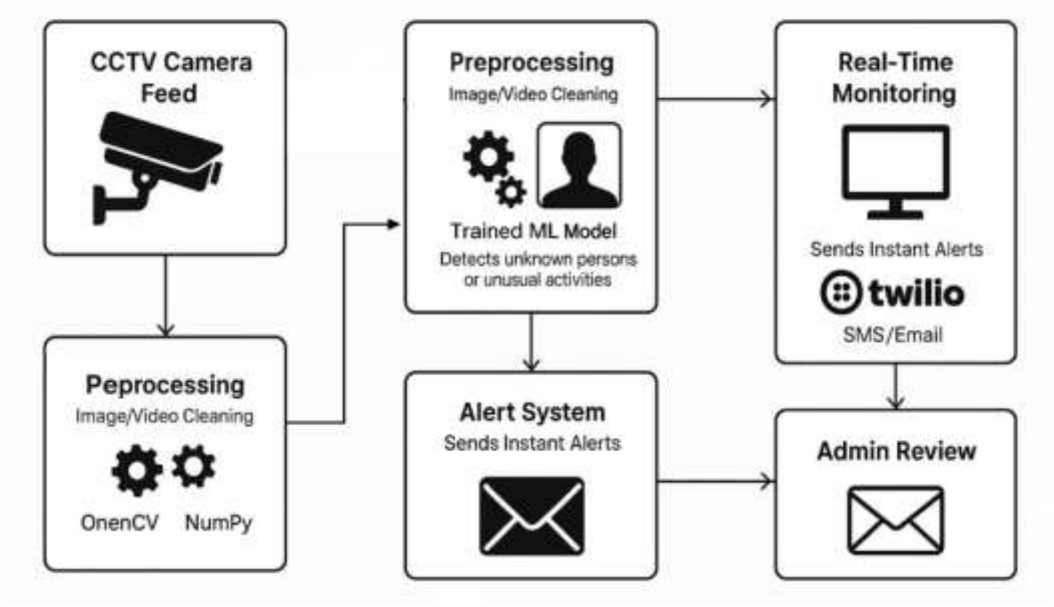


Figure 3.1 System Architecture

The Figure 3.1 describes the comprehensive architecture of the Real-Time Security Monitoring System, commencing with CCTV Camera Feeds as the primary data source, which then undergoes an initial Preprocessing stage utilizing OpenCV and NumPy for fundamental image and video cleaning. These prepared frames are subsequently fed into a more advanced "Preprocessing" phase that refines them for the custom-trained Machine Learning Model, the system's core, which intelligently detects unknown persons or unusual activities by classifying faces against a known database. Upon identifying an unauthorized individual, the Real-Time Monitoring component immediately triggers Instant Alerts through the Alert System, dispatching notifications via Twilio for SMS and sending emails with captured snapshots to designated personnel. This seamless, automated flow ensures prompt security awareness, allowing for timely Admin Review and response to potential threats.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Equation 3.1 Accuracy Formula

The Equation 3.1 illustrates the Accuracy which serves as a fundamental evaluation metric that quantifies the overall correctness of a classification model's predictions. It is calculated as the ratio of the number of correctly classified instances (both true positives and true negatives) to the total number of instances in the dataset. While widely used for its simplicity and intuitive understanding, accuracy indicates the proportion of all predictions that the model got right, providing a quick summary of performance. However, it's important to note that accuracy alone can sometimes be misleading, especially in scenarios with highly imbalanced datasets where a model might achieve high accuracy by simply predicting the majority class.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Equation 3.2 Precision

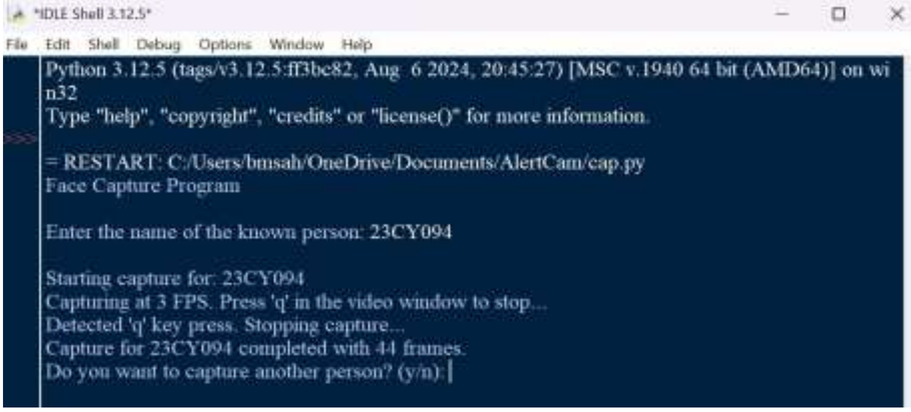
The Equation 3.2 demonstrates Precision in ML which is a crucial evaluation metric that quantifies the accuracy of a classification model's positive predictions. It specifically measures the proportion of correctly identified positive instances out of all instances that the model classified as positive. This metric is particularly vital in applications like security monitoring, medical diagnostics, or spam detection, where the occurrence of a False Positive—an incorrect positive classification—can lead to significant costs, unnecessary interventions, or system inefficiencies. A high precision value indicates that when the model asserts a positive identification, it is highly likely to be correct, thereby minimizing false alarms and enhancing the trustworthiness of positive classifications.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Equation 3.3 Recall

The Equation 3.3 highlights Recall as a crucial evaluation metric in machine learning, measuring a model's ability to identify all relevant positive instances by quantifying the proportion of actual positive cases correctly found. This metric is particularly vital in applications like your security monitoring system or medical diagnosis, where the high cost of a False Negative—failing to detect an actual positive event, such as missing an intruder—is exceptionally high. A high recall value indicates the model is highly effective at capturing nearly all true positive cases, thus minimizing missed risks.

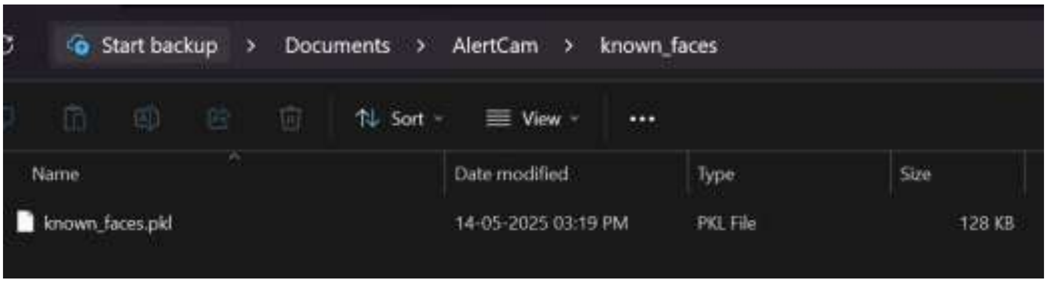
4. RESULT AND DISCUSSION



```
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug  6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on wi
n32
Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:/Users/bmsah/OneDrive/Documents/AlertCam/cap.py
Face Capture Program

Enter the name of the known person: 23CY094

Starting capture for: 23CY094
Capturing at 3 FPS. Press 'q' in the video window to stop...
Detected 'q' key press. Stopping capture...
Capture for 23CY094 completed with 44 frames.
Do you want to capture another person? (y/n):
```

Figure 4.1 Capturing of Frames

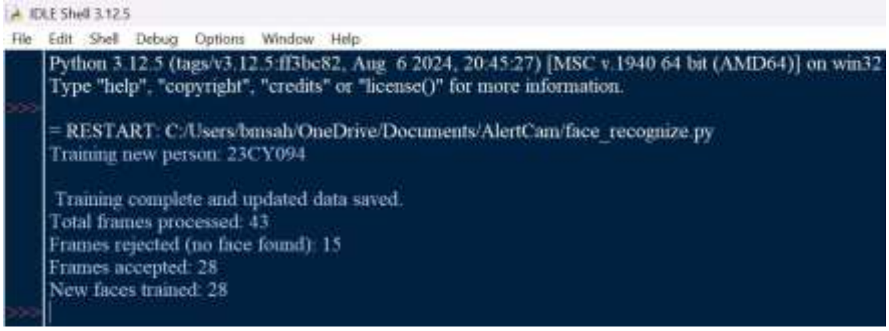
The Figure 4.1 illustrates a critical phase of the Real-Time Security Monitoring project: the systematic data collection and enrollment process for authorized individuals. Specifically, it depicts the execution of the "Face Capture Program", engineered to acquire comprehensive facial data for known personnel. The interface demonstrates the system's prompt for a unique individual identifier, followed by the subsequent capture of a defined sequence of video frames (e.g., 44 frames at 3 FPS). This meticulously controlled capture procedure is fundamental for constructing the robust foundational database of facial profiles, which serves as the reference against which live video feeds are compared, thereby enabling the machine learning model to accurately differentiate between authorized personnel and potential unauthorized intruders in a real-time operational environment.



Name	Date modified	Type	Size
known_faces.pkl	14-05-2025 03:19 PM	PKL File	128 KB

Figure 4.2 Pickle File of Known Faces

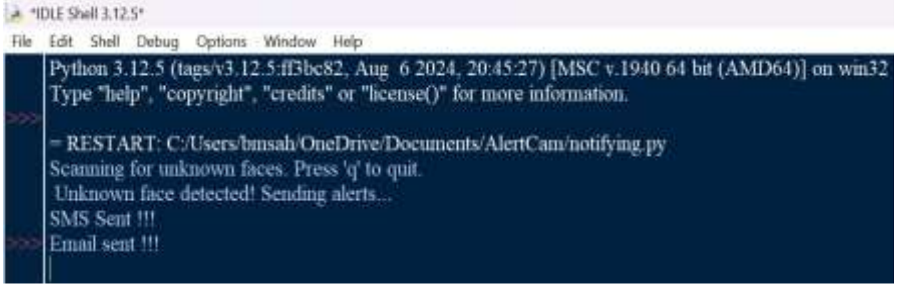
The Figure 4.2 illustrates a critical step in preparing the dataset for your machine learning model: the efficient storage of known facial data. The known_faces.pkl file, located within the AlertCam/known_faces directory, signifies that the collected facial datasets—likely including processed images and their corresponding facial embeddings or encodings—have been serialized into a pickle (.pkl) format. This conversion is essential as it allows for the rapid and convenient loading of the entire dataset into memory for the training phase of your custom machine learning model. By storing the data in this optimized format, the system streamlines the process of feeding pre-processed and ready-to-use information to the model, significantly improving the efficiency of model development and subsequent recognition tasks.



```
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug  6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmsah/OneDrive/Documents/AlertCam/face_recognize.py
Training new person: 23CY094

 Training complete and updated data saved.
Total frames processed: 43
Frames rejected (no face found): 15
Frames accepted: 28
New faces trained: 28
>>>
```

Figure 4.3 Training Dataset

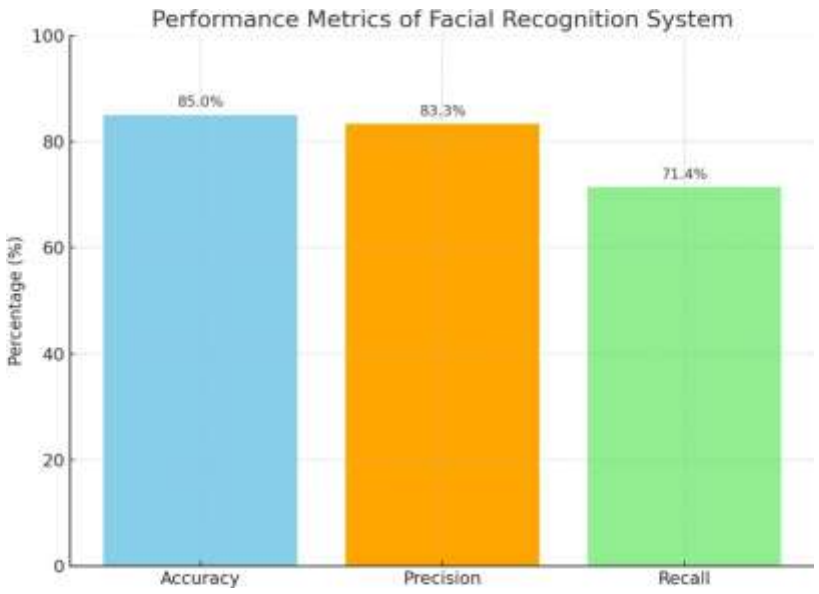
The Figure 4.3 details a crucial step in the ongoing development and maintenance of your Real-Time Security Monitoring system: the enrollment and training of new authorized individuals. It captures the execution of the face_recognize.py script, specifically demonstrating the process of "Training new person." The output indicates that the system processed 43 frames for this individual, rejecting 15 frames where no face could be reliably detected, but successfully accepting and training on the remaining 28 frames. This automated training process is vital for continuously updating the machine learning model's database with new facial profiles, ensuring that the system can accurately recognize all authorized personnel as they are added, thereby enhancing the precision and effectiveness of the overall security surveillance.



```
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug  6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmsah/OneDrive/Documents/AlertCam/notifying.py
Scanning for unknown faces. Press 'q' to quit.
 Unknown face detected! Sending alerts...
SMS Sent !!!
>>> Email sent !!!
```

Figure 4.4 Detecting and Alerting Admin

The Figure 4.4 vividly demonstrates the real-time operational phase and core alerting capability of your security monitoring system. It shows the notifying.py script actively "Scanning for unknown faces," indicating the continuous surveillance process. Crucially, the console output "Unknown face detected! Sending alerts..." confirms the successful classification of an unauthorized individual by the machine learning model. This detection immediately triggers the system's proactive response mechanisms, with the subsequent messages "SMS Sent !!!" and "Email sent !!!" validating the successful dispatch of multi-channel alerts via the integrated Twilio API for SMS and the configured email service. This output provides direct evidence of the system's ability to not only identify threats in real-time but also to promptly notify designated administrators, thereby fulfilling its primary objective of intelligent and proactive security.



Graph 4.1 Accuracy, Precision, Recall Graph

The Graph 4.1 visually represents the empirical validation of the developed machine learning model's effectiveness. The results showcase the system's robust capabilities in identifying individuals within a real-time surveillance context, quantified across three critical metrics. An Accuracy of 85.0% indicates a high overall rate of correct classifications, signifying that the model correctly identified both authorized and unauthorized individuals in the vast majority of cases. The Precision of 83.3% is particularly strong for a security application, demonstrating that when the system identifies a face as "unknown" or unauthorized, it is correct over 83% of the time, thereby minimizing disruptive false alarms. Lastly, a Recall of 71.4% illustrates the model's ability to detect a significant proportion of actual unauthorized individuals present. While strong, the balance between precision and recall is crucial, and these metrics collectively validate the system's practical utility and reliability in enhancing security monitoring.

5. CONCLUSION

The developed system unequivocally demonstrates the effective deployment of a lightweight, machine learning-based facial recognition model within demanding real-time surveillance environments, fundamentally enhancing proactive security infrastructure. Its modular architecture and high adaptability render it exceptionally well-suited for diverse institutional settings seeking intelligent and responsive security solutions. This implementation not only validates the capability for rapid, accurate identification but also showcases the system's potential for seamless integration with existing surveillance frameworks, delivering immediate actionable alerts. Looking forward, future work will focus on further augmenting the system's capabilities through significant deep learning upgrades, which could encompass more sophisticated neural network architectures to improve recognition under challenging conditions such as varying illumination, angles, and partial occlusions. Additionally, efforts will be directed towards expanded integration with cloud-connected CCTV systems, enabling more scalable data processing, centralized management, and potentially leveraging cloud AI services for enhanced analytical insights and resilience.

REFERENCES

[1] T, Saranya & D, Jeyamala & Sellamuthu, Suseela & S, Indra. (2024). A Secure Framework for MIoT: TinyML-powered Emergency Alerts and Intrusion Detection for Secure Real-time Monitoring. 13-21. 10.1109/I-SMAC61858.2024.10714760.

[2] Mukto, Md & Hasan, Mahamudul & Mahmud, Md Maiyaz & Ahmed, Md & Jabid, Taskeed & Ali, Md & Rashid, Mohammad & Islam, Mohammad & Islam, Maheen. (2023). Design of a Real- Time Crime Monitoring System Using Deep Learning Techniques. Intelligent Systems with Applications. 21. 200311. 10.1016/j.iswa.2023.200311.

[3] Kumar, Akshat & Agrawal, Renuka & Singh, Akshra & Noorani, Aaftab & Jaiswal, Yashika & Hemnani, Preeti & Hamdare, Safa. (2024). Real-Time Monitoring and Analysis Through Video Surveillance and Alert Generation for Prompt and Immediate Response. International Journal of Advanced Computer Science and Applications. 15. 10.14569/IJACSA.2024.0151272.

[4] AK, Sreejith & Nath, Keshab. (2024). A Next-Gen Real-Time Video Alert System with Machine Learning Sensitivity. Procedia Computer Science. 235. 447-455. 10.1016/j.procs.2024.04.044.

[5] Thomas, R & Sanjay, G & Pandeeswaran, C & Raghi, K.. (2024). Advanced CCTV Surveillance Anomaly Detection, Alert Generation and Crowd Management using Deep Learning Algorithm. 1-6. 10.1109/AIIoT58432.2024.10574731.

[6] Selvi, Esakky & Adimoolam, Malaiyalathan & Govindharaju, Karthi & Thinakaran, Kandasamy & Balamurugan, Nagaiah & Raju, Kannadasan & Wechtaisong, Chitapong & Khan, Arfat Ahmad. (2022). Suspicious Actions Detection System Using Enhanced CNN and Surveillance Video. Electronics. 11. 1-20. 10.3390/electronics11244210.

[7] Gandapur, Maryam. (2022). E2E-VSDL: End-to-end video surveillance-based deep learning model to detect and prevent criminal activities. Image and Vision Computing. 123. 104467. 10.1016/j.imavis.2022.104467.

[8] Rahimi Ardabili, Babak & Danesh Pazho, Armin & Alinezhad Noghre, Ghazal & Neff, Christopher & Bhaskararayuni, Sai & Ravindran, Arun & Tabkhi, Hamed. (2023). Understanding Policy and Technical Aspects of AI-Enabled Smart Video Surveillance to Address Public Safety. 10.48550/arXiv.2302.04310.

[9] Alam, Hina & Burhan, Muhammad & Gillani, Anusha & Haq, Ihtisham & Arshed, Muhammad Asad & Shafi, Muhammad & Ahmed, Saeed. (2023). IoT Based Smart Baby Monitoring System with Emotion Recognition Using Machine Learning. Wireless Communications and Mobile Computing. 2023. 10.1155/2023/1175450.

[10] Sharara, Leila & Politis, Alexandros & Ismail, Mohammed & Thelen, Klaus. (2023). A Real- Time Automotive Safety System Based on Advanced AI Facial Detection Algorithms. IEEE Transactions on Intelligent Vehicles. 1. 1-22.

[11] Chen, Jingdi & Zhang, Lei & Riem, Joseph & Adam, Gina & Bastian, Nathaniel & Lan, Tian. (2023). RIDE: Real-time Intrusion Detection via Explainable Machine Learning Implemented in a Memristor Hardware Architecture. 1-8. 10.1109/DSC61021.2023.10354120.

[12] Zhou, Hanhan & Chen, Jingdi & Mei, Yongsheng & Adam, Gina & Aggarwal, Vaneet & Bastian, Nathaniel & Lan, Tian. (2024). Real-time Network Intrusion Detection via Importance Sampled Decision Transformers. 82-91. 10.1109/MASS62177.2024.00022.

[13] Vishwakarma, Monika & Kesswani, Nishtha. (2022). DIDS: A Deep Neural Network based real- time Intrusion detection system for IoT. Decision Analytics Journal. 5. 100142. 10.1016/j.dajour.2022.100142.

[14] Kale, Rahul & Lu, Zhi & Fok, Kar & Thing, Vrizlynn. (2022). A Hybrid Deep Learning Anomaly Detection Framework for Intrusion Detection. 10.48550/arXiv.2212.00966.

[15] Laghrissi, Fatimaezzahra & Douzi, Samira & Khadija, Douzi & Hssina, Badr. (2021). Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data. 8. 10.1186/s40537- 021-00448-4.