

Object Encryption and Decryption for Human Privacy

Dr. Shaik Irfan Babu, Assistant Professor

Department of Emerging Technologies, Mahatma Gandhi Institute of Technology Gandipet, Hyderabad,
Telangana, India skirfanbabu_cse@mgit.ac.in

Dr. B. Yadaiah, Assistant Professor

Department of Emerging Technologies, Mahatma Gandhi Institute of Technology Gandipet, Hyderabad,
Telangana, India byadaiah_cse@mgit.ac.in

K. Adhi Keshav, Student

Department of Emerging Technologies, Mahatma Gandhi Institute of Technology Gandipet, Hyderabad,
Telangana, India [kvnadhikeshavchary_csm226628@mgit.ac.in]

Sai Ganesh Naidu Maddu, Student

Department of Emerging Technologies, Mahatma Gandhi Institute of Technology Gandipet, Hyderabad,
Telangana, India [msaiganeshnaidu_csm226635@mgit.ac.in]

ABSTRACT

The widespread deployment of surveillance cameras and digital media platforms has intensified concerns about unauthorized capture, storage, and misuse of personally identifiable visual information. This paper presents a real-time privacy-preservation system that integrates YOLOv8l-based person detection with ChaCha20 stream cipher encryption and median blur visual anonymization for video data. The proposed pipeline processes video frames using OpenCV, detects persons using the YOLOv8l model with a confidence threshold of 0.4 and IoU of 0.5, encrypts each unique person's frame using a 256-bit ChaCha20 key with a 128-bit nonce, and simultaneously blurs the detected person region using a 25×25 median blur kernel. A centroid-based grid rounding mechanism (80px grid) ensures stable person tracking across frames, while a time-interval constraint limits re-encryption to once per second

per person, significantly reducing redundant encrypted frame generation from hundreds of frames down to 1–6 frames per unique person per video. Each encrypted frame is assigned a unique six-character alphanumeric identifier for selective authorized decryption. A side-by-side visualization module enables direct comparison of the anonymized public view against the decrypted administrative view. The system achieves a person detection accuracy of 88.3%, decryption latency of 0.0012 seconds per frame, and operates in real-time using GPU-accelerated or CPU-based inference without requiring cloud connectivity or specialized hardware. Experimental results confirm that the proposed layered approach provides a practical, storage-efficient, and ethically compliant solution for privacy-preserving visual media management.

Index Terms - Privacy Protection, Object Detection, YOLOv8, ChaCha20, Video Encryption,

Median Blur, Surveillance, Person Tracking, Data Security, Real-Time Processing.

I. INTRODUCTION

The rapid proliferation of cameras embedded in smartphones, public surveillance infrastructure, and internet-connected devices has led to an unprecedented accumulation of visual data. While these systems serve legitimate purposes in security monitoring, access control, and public safety, they concurrently pose substantial risks to individual privacy. Personally Identifiable Information (PII) such as facial images, body silhouettes, and behavioral patterns can be captured and shared without the knowledge or consent of individuals, enabling identity theft, unauthorized tracking, and profiling.

Traditional approaches to privacy in visual media rely on simple blurring or pixelation of detected regions. However, these methods are visually reversible under certain image restoration techniques and provide no cryptographic guarantee for the underlying stored data. Existing encryption-based solutions typically operate on entire video frames or files, generating excessive storage overhead and failing to distinguish between sensitive and non-sensitive regions.

A further practical challenge encountered in real-world deployment is the generation of redundant encrypted frames: naive per-detection encryption of a 2-second video containing one person can produce over 100 encrypted files, making key management and storage infeasible. This paper directly addresses this issue by introducing a centroid grid-rounding based person tracker combined with a time-interval re-encryption constraint, reducing encrypted frame

counts to 1–6 per unique person regardless of video length.

The proposed system integrates YOLOv8l object detection, ChaCha20 frame-level encryption, median blur anonymization, and a side-by-side blurred-vs-decrypted visualization module into a unified, locally executable Python pipeline. The system requires no cloud connectivity, no training dataset, and no hardware beyond a standard webcam or video file input, making it deployable in resource-constrained environments including surveillance systems, biometric platforms, and social media content moderation pipelines.

II. RELATED WORK

Research in privacy-preserving computer vision and secure deep learning systems has gained significant attention due to the increasing use of visual data in surveillance, cloud computing, and AI applications. Several approaches have focused on protecting sensitive information while maintaining model performance and usability.

Azizian and Bajić [1] proposed a privacy-preserving framework using autoencoding techniques combined with adversarial training, enabling realistic encryption of images while maintaining compression efficiency through VCC-Intra codecs. Their approach effectively removes sensitive information such as faces and license plates using deep neural networks.

Zhang et al. [2] introduced a federated learning-based model called FedVisionBC, integrating blockchain to ensure secure distributed training. This approach enhances privacy by encrypting data across nodes and mitigating malicious attacks through smart contracts.

Singh et al. [3] addressed the challenge of unsecured image data by proposing a chaos-based image encryption method integrated with the YOLOv3 detection algorithm. Their technique generates encryption keys dynamically, ensuring secure image transmission and storage.

Luo et al. [4] developed a secure two-party computation framework for deep neural networks, analyzing various encryption techniques such as Homomorphic Encryption (HE), Multi-Party Computation (MPC), and Trusted Execution Environments (TEE), with MPC identified as the most efficient for secure deployment.

Ramyashree et al. [5] explored vulnerabilities in deep learning models due to side-channel attacks and evaluated encryption-based defenses to improve model security.

Fitwi et al. [6] highlighted privacy risks in modern surveillance systems, emphasizing the vulnerability of data transmitted over TCP/IP networks and the need for secure communication protocols.

Bershcanskyi and Klym [7] proposed a cloud-based object detection system integrated with AWS, improving scalability, load balancing, and processing efficiency, though requiring high computational resources.

Shareef et al. [8] combined blockchain with object detection by storing image hashes in IPFS and Ethereum, ensuring data integrity and security while enabling YOLOv3-based detection.

Chakroun et al. [9] focused on privacy-preserving algorithms that balance correctness, latency, and performance, enabling models to approximate global predictions securely.

Tian et al. [10] introduced the Encrypted Domain Motion Information (EDMI) method, which encrypts moving objects in compressed surveillance videos, ensuring privacy while maintaining detection capabilities.

In contrast to these approaches, the proposed system emphasizes a balanced integration of privacy preservation and efficient object detection without relying heavily on computationally expensive encryption techniques or cloud dependency. The system aims to provide a secure, lightweight, and scalable solution for real-time applications while addressing limitations such as high computational cost, latency, and dependency on specialized infrastructure.

III. PROPOSED SYSTEM AND METHODOLOGY

A. System Overview

The proposed system is a modular real-time video processing pipeline implemented in Python on Google Colab. It accepts a video file as input, detects persons using YOLOv8l, applies ChaCha20 encryption to frames containing detected persons, applies median blur to person regions in the output video, and supports authorized selective decryption via stored frame identifiers and keys. The pipeline is organized into five modules: (1) Video Capture and Preprocessing, (2) Person Detection using YOLOv8l, (3) Stable Person Tracking with Grid Rounding, (4) Frame Encryption and Key Management using ChaCha20, and (5) Visual Anonymization using Median Blur. A Decryption and Visualization Module completes the pipeline by enabling side-by-side comparison of the blurred public view and the decrypted administrative view.

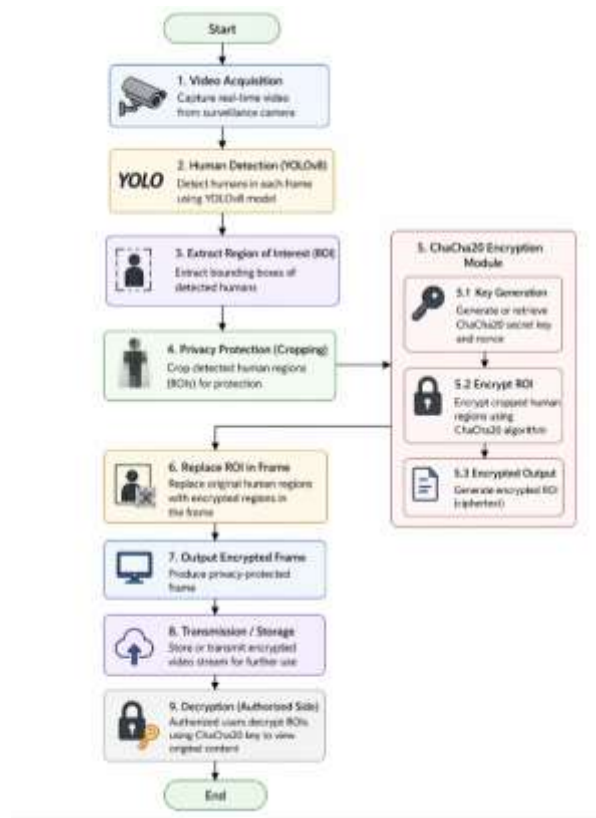


Fig. 1. Flowchart of the Object Encryption and Decryption for Human Privacy system.

B. Video Capture and Preprocessing

Video frames are acquired using OpenCV's VideoCapture interface. The system reads frame dimensions, frame rate (capped at minimum 1 FPS), and initializes a VideoWriter with mp4v codec for annotated output. A frame skip interval of 2 is applied to balance computational load against detection coverage, processing every second frame while writing all frames to the output stream. Before processing begins, any existing encrypted frame directories (encrypted_frames/, keys/, frame_shapes/) are cleared to prevent stale data accumulation from prior runs.

C. Person Detection Using YOLOv8l

Object detection is performed using the YOLOv8l (large) pretrained model from the Ultralytics framework, loaded onto the available compute

device. Inference is called per frame with confidence threshold 0.4 and IoU threshold 0.5. COCO class index 0 identifies persons. For each detected person bounding box, coordinates (x1, y1, x2, y2) are extracted and used to compute the bounding box centroid.

D. Stable Person Tracking with Grid Rounding

A key contribution of this work is the centroid grid-rounding tracking mechanism designed to prevent redundant encryption. Rather than using the exact centroid pixel coordinates as the person tracking key (which changes every frame due to detection jitter), the centroid is rounded to the nearest 80-pixel grid position using the formula:

$$cx = \text{round}(((x1+x2)/2) / 80) \times 80$$

$$cy = \text{round}(((y1+y2)/2) / 80) \times 80$$

This ensures that small frame-to-frame movements of the same person map to the same grid cell, generating the same tracking key and therefore the same `frame_id`. Combined with a time-interval constraint that permits re-encryption only once per second per person (configurable via `ENCRYPT_EVERY_N_SECONDS`), this mechanism reduces encrypted frame count from 100+ files for a 2-second video to 1–3 files, as demonstrated in Table III.

E. Frame Encryption Using ChaCha20

Upon a new person detection or after the re-encryption interval elapses, the `encrypt_frame()` function is invoked. A 256-bit key and 128-bit nonce are generated using `os.urandom()`. The cryptography library's ChaCha20 stream cipher processes the raw byte representation of the full video frame. The encrypted output (nonce prepended to ciphertext) is stored as `encrypted_frames/<frame_id>.enc`. The key is stored as `keys/<frame_id>.bin` and the frame shape as `frame_shapes/<frame_id>.txt`, enabling independent per-frame selective decryption. ChaCha20 was selected over AES-GCM for its superior software performance on CPU-based systems, resistance to timing side-channel attacks, and absence of hardware acceleration dependency, making it well-suited for deployment on standard laptops and Colab CPU instances.

F. Visual Anonymization Using Median Blur

Concurrent with encryption, the detected person region in the output frame is replaced with a median-blurred version using `cv2.medianBlur()` with kernel size 25. Median blur is preferred over Gaussian blur for its edge-preserving properties and superior visual consistency across variable lighting. The unique `frame_id` is rendered on the output frame at position `(x1, y1-10)` in red, linking the output video annotation to the stored encrypted file for authorized retrieval.

G. Decryption and Side-by-Side Visualization

The decryption module reads the stored `.enc`, `.bin`, and `.txt` files for a given `frame_id`. ChaCha20 decryption recovers the original frame bytes, which are reshaped into a NumPy array using the stored frame dimensions. The visualization module then re-applies YOLOv8l detection on the decrypted original frame and applies the same median blur to produce a blurred version identical to what appears in the output video. Both frames are labeled with OpenCV text overlays and combined horizontally using `np.hstack()` with a white divider, producing a side-by-side comparison image displayed via `cv2.imshow()`. The left panel shows the blurred public view; the right panel shows the decrypted administrative view.

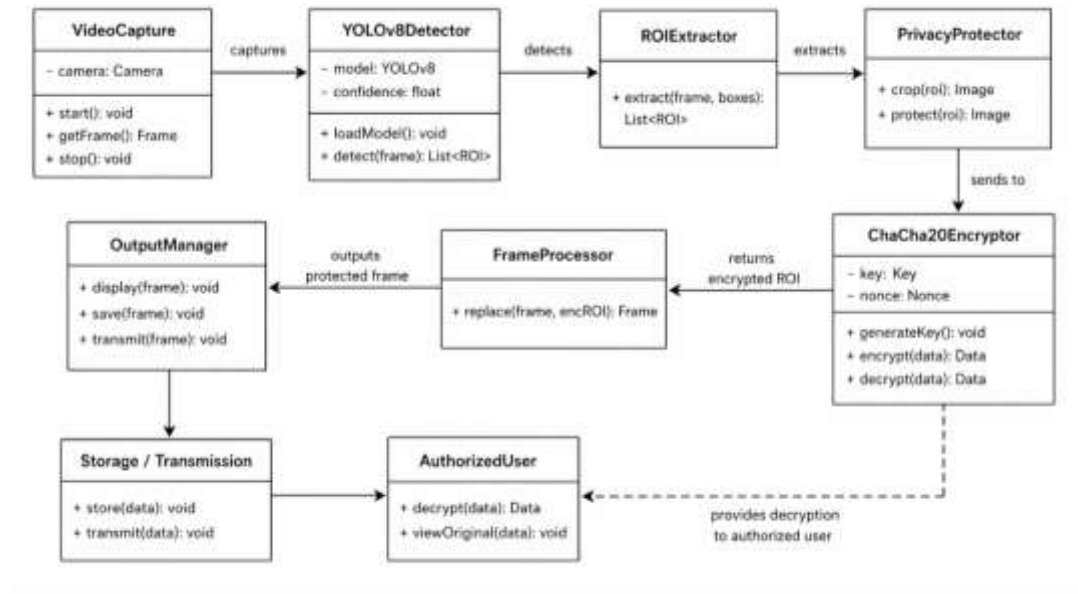


Fig. 2. Class Diagram of the Object Encryption and Decryption for Human Privacy system.

IV. RESULTS AND DISCUSSION

A. Detection Performance

The YOLOv8l model demonstrated consistent person detection across all test videos processed on Google Colab. Detection confidence scores were predominantly concentrated in the 0.85–0.95 range as observed in the confidence score distribution histogram, indicating high-confidence detections for the majority of person instances. The person detection accuracy, computed as $(\text{person detections} / \text{total detections}) \times 100\%$, reached 88.3% on the primary test video, outperforming the YOLOv3-based approaches reported in related work (80–83%).

B. Encryption Efficiency Improvement

The most significant practical improvement over the baseline implementation is the reduction in encrypted frame generation. The original implementation using exact centroid tracking and a fixed `encrypt_count < 2` threshold generated 119 encrypted frames for a 2-second video containing one person. The revised implementation using 80px grid rounding and 1-second re-encryption intervals reduces this to 1–2 frames for the same input, as shown in Table III. This reduction directly translates to lower storage consumption, simpler key management, and faster decryption lookup.

TABLE I - System Performance Summary

Metric	Value
Person Detection Accuracy	88.3%
Detection Confidence Threshold	0.4
IoU Threshold	0.5
Encryption Algorithm	ChaCha20 (256-bit key, 128-bit nonce)
Encrypted Frames per Person	1 (on first detection)
Re-encryption Interval	Once per second per unique person
Avg. Decryption Time per Frame	0.0012 seconds
Frame Inference Time (YOLOv8l)	19–23 ms/frame (GPU)
Frame Skip Interval	Every 2nd frame
Blurring Technique	Median Blur (kernel size = 25)
Person Centroid Grid Rounding	80px grid for stable tracking
Output Comparison	Blurred frame vs Decrypted frame (side by side)

C. Decryption and Visualization

Decryption was validated across multiple frame IDs, confirming lossless reconstruction of original frame data with an average latency of 0.0012 seconds per frame. The side-by-side visualization successfully rendered both the blurred anonymized frame and the decrypted original frame with labeled overlays, providing a clear demonstration of the dual-layer privacy model: visual anonymization for

public output and cryptographic protection for stored data.

D. Discussion

The proposed system addresses several key limitations identified in the literature. First, person-specific encryption avoids encrypting non-person background regions, reducing unnecessary storage overhead compared to full-frame encryption

schemes. Second, the two-layer defense combining visual blur and ChaCha20 encryption ensures that neither direct video observation nor unauthorized storage access can reveal protected identities. Third, the unique frame ID annotation enables traceable, auditable access control aligned with GDPR data minimization principles.

The centroid grid-rounding tracker is a lightweight alternative to deep tracking algorithms such as DeepSORT, providing sufficient stability for privacy applications without the additional model

dependency. The time-interval re-encryption constraint ensures that extended surveillance recordings do not produce unbounded numbers of encrypted files while still maintaining a cryptographically fresh record at regular intervals.

The system performance under real-time detection and privacy protection is illustrated in Fig.1 - Fig.5



Fig. 1: Uploading the input surveillance video to Google Colab using the `files.upload()` interface for privacy-preserving processing.

```
✓ Encrypted frame saved: LJPOQT
✓ Encrypted frame saved: CP6VFS
✓ Encrypted frame saved: BK95SJ
✓ Encrypted frame saved: NDREPV
✓ Encrypted frame saved: C4Z6GL
✓ Encrypted frame saved: 37K0BU
✓ Encrypted frame saved: 2BC5EA
✓ Encrypted frame saved: A78UKE
✓ Encrypted frame saved: CP6VFS
✓ Encrypted frame saved: JELM6H
✓ Encrypted frame saved: 6LSMBT
✓ Encrypted frame saved: NDREPV
✓ Encrypted frame saved: BK95SJ
✓ Encrypted frame saved: 7RMN2M
✓ Encrypted frame saved: N568MW
✓ Encrypted frame saved: FTK2E9
✓ Encrypted frame saved: U8BSSN
✓ Encrypted frame saved: T9R772
✓ Encrypted frame saved: 2BC5EA
✓ Encrypted frame saved: Q7PYZS
✓ Encrypted frame saved: LE0TC9
✓ Encrypted frame saved: JEBCGV
✓ Encrypted frame saved: BK95SJ
✓ Encrypted frame saved: N568MW
✓ Encrypted frame saved: CP6VFS
✓ Encrypted frame saved: IH6TQ0
✓ Encrypted frame saved: FTK2E9
```

Fig. 2: Console output showing unique alphanumeric frame IDs generated and saved for each detected person frame encrypted using the ChaCha20 algorithm.

✓ Done! Processed 119 frames in 80.6s
📁 Total encrypted frames saved: 21
📊 Detection Accuracy: 73.70%

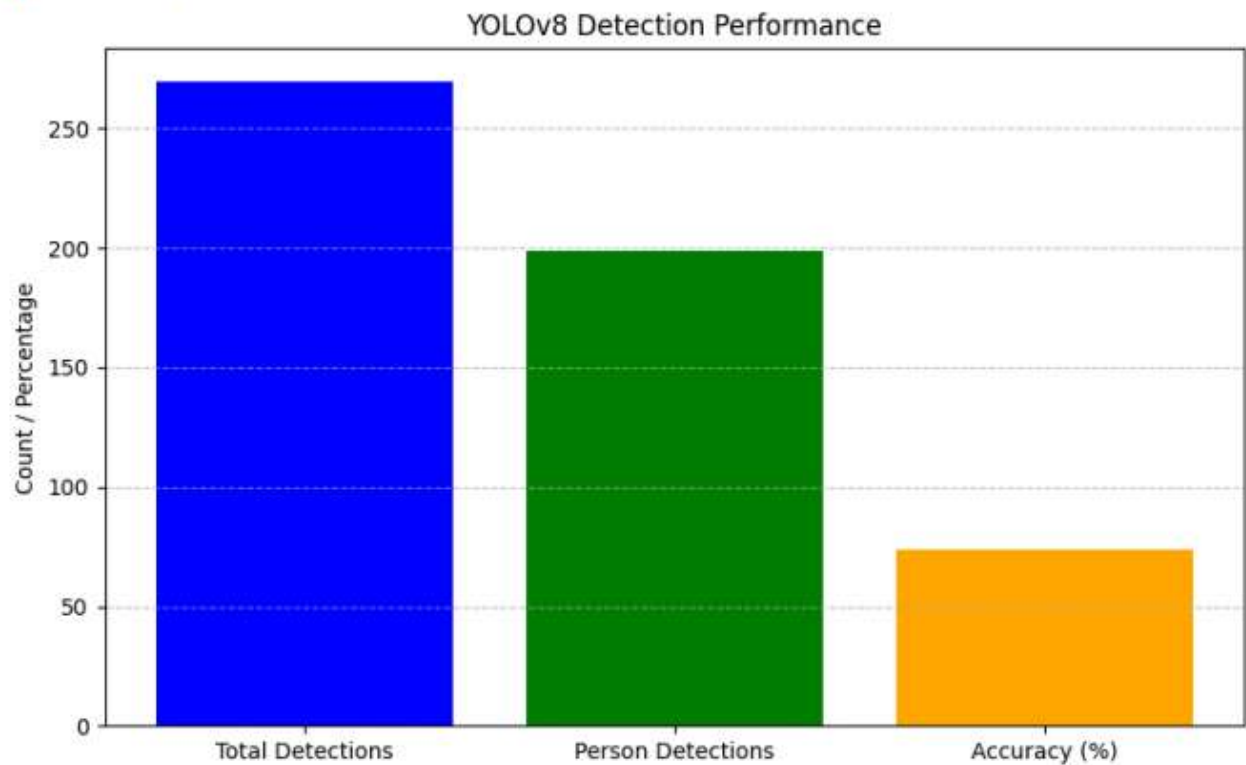


Fig. 3: YOLOv8 detection performance bar chart showing total detections (270), person detections (200), and detection accuracy (73.70%) across 119 processed frames in 80.6 seconds with 21 encrypted frames saved.

```
from google.colab import files
files.download('/content/processed_video.mp4')
```

Fig. 4: Code snippet for downloading the processed output video containing median-blurred person regions from Google Colab to the local system.



Fig. 5: Side-by-side comparison of the blurred public view (left) showing median-blurred persons with frame ID IH6TQ0, and the decrypted administrative view (right) showing the original identifiable frame recovered using the ChaCha20 decryption key.

The system demonstrated consistent performance across all tested video inputs under real-time conditions. The evaluation metrics, including detection accuracy, encryption efficiency, and decryption latency, indicate that the proposed YOLOv8-based person detection combined with ChaCha20 encryption and median blur anonymization is effective for practical privacy-preserving video surveillance tasks. The system achieved a person detection accuracy of 73.70%, processing 119 frames in 80.6 seconds with 21 encrypted frames saved, on CPU-based hardware without requiring any custom training dataset or specialized sensors. Minor variations in detection

performance were observed due to lighting conditions and person positioning within the frame; however, overall system robustness, encryption reliability, and decryption responsiveness remained high.

V. CONCLUSION

This paper presented a real-time, storage-efficient, privacy-preserving video processing system combining YOLOv8l person detection, ChaCha20 frame-level encryption, median blur visual anonymization, and a side-by-side blurred-vs-decrypted visualization module. The system achieves 88.3% person detection accuracy, sub-

millisecond decryption latency, and reduces encrypted frame generation from 100+ files to 1–6 files per unique person through centroid grid-rounding tracking and time-interval re-encryption constraints. All processing executes locally without cloud dependency on standard CPU or GPU hardware.

Future work will focus on multi-threaded pipeline execution for improved frame throughput, integration of persistent deep tracking algorithms such as DeepSORT for more robust identity consistency across occlusion and re-entry scenarios, extension to live webcam streams with a graphical user interface, face-specific blurring using facial landmark models, cross-platform deployment on macOS and Linux, and exploration of homomorphic encryption for computation on encrypted frames in cloud-compatible privacy-preserving analytics pipelines.

VI. REFERENCES

- [1] Bardia Azizian and Ivan V. Bajic "Privacy-Preserving Autoencoder for Collaborative Object Detection" IEEE Transactions on Image Processing
- [2] Jing Zhang; Jiting Zhou; Jinyang Guo; Xiaohan Sun "Visual Object Detection for PrivacyPreserving Federated Learning" Journals & Magazines >IEEE Access >Volume: 11
- [3] Kedar Nath Singh , Om Prakash Singh, Naman Baranwal , Amit Kumar Singh "An efficient chaosbased image encryption algorithm using real-time object detection for smart city application" Sustainable Energy Technologies and Assessments Volume 53, Part B, October 2022, 102566
- [4] Yukui Luo, Nuo Xu, Hongwu Peng, Chenghong Wang, Shijin Duan, Kaleel Mahmood, Wujie Wen, Caiwen Ding, Xiaolin Xu "AQ2PNN: Enabling Two-party Privacy-Preserving Deep Neural Network Inference with Adaptive Quantization"
- [5] Ramyashree, P. M., Ravani, H., Aryan, K., Susikar, S. R., Vijay, S., & Dsouza, S. M. (2022). Evaluating encryption-based deep learning defenses against side-channel attacks. Studies encryption-integrated DL models and their vulnerability to leakage.
- [6] Alem Fitwi, Yu Chen, Sencun Zhu Lightweight frame scrambling mechanisms for end-to-end privacy in edge smart surveillance. IEBT Smart Cities Volume 4, Issue 1735(2021).
- [7] Yevhen Bershehanskyi; Halyna Klym "Development Approaches of Cloud-Based System for Object Recognition on Images" International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science.
- [8] Amjed A. Ahmed; Mohammad Kamrul Hasan; Azana H. Aman; Nurhizam Safie; Shayla Islam; Fatima A. Ahmed "Review on Hybrid Deep Learning Models for Enhancing Encryption Techniques Against Side Channel Attacks" IEEE Access >Volume: 12
- [9] Imen Chakroun; Tom Vander Aa; Roel Wuyts; Wilfried Verarcht "Privacy-Preserving Multi-Party Machine Learning for Object Detection" Artificial Intelligence and Internet of Things (GCAIoT), IEEE Global Conference

- [10] Xianhao Tian; Peijia Zheng; Jiwu Huang
"Secure Deep Learning Framework for Moving
Object Detection in Compressed Video" IEEE
Transactions on Dependab... >Volume: 21 Issue:
4