Pertanika J. Sci. & Technol. 34 (1): 339 - 364 (2026)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

A Unified Framework for Human Detection and Anomaly Identification in Video Surveillance

Mintu Movi[1]*, Abdul Jabbar P[2], Noufal K P[3], and Bindu V R[4]

[1]*School of Computer Sciences, Mahatma Gandhi University, Kottayam, 686560 Kerala, India*
[2]*Graduate School, Stamford International University, Prawet, 10250 Bangkok, Thailand*
[3]*Information Kerala Mission, Public Office, Thiruvananthapuram, 695033 Kerala, India*
[4]*School of Computer Sciences, Mahatma Gandhi University, 686560 Kottayam, Kerala, India*

ABSTRACT

The increasing reliance on surveillance systems underscores the critical need for efficient and automated solutions to detect rare events within residential environments. This study introduces a robust anomaly detection framework for near real-time analysis of recorded CCTV footage. The proposed system addresses law enforcement's significant challenge: the manual, error-prone, and time-consuming review of extensive surveillance footage. The system integrates advanced techniques such as data stream management, human subject detection, and anomaly detection. A pre-trained Mask Region-Based Convolutional Neural Network (Mask R-CNN) ensures precise human detection, even in complex scenes. At the same time, an enhanced unsupervised Isolation Forest algorithm identifies rare or anomalous events by distinguishing them from routine activities in the video data. Results demonstrate that the framework significantly improves the speed and accuracy of crime analysis, offering a reliable, scalable, and automated solution with minimal human intervention. The system empowers law enforcement agencies and security professionals by streamlining detection, facilitating proactive responses, and thorough investigations. Its adaptability across diverse residential settings further strengthens its utility for various security and surveillance applications, ultimately contributing to enhanced safety and security measures.

Keywords: Convolutional neural network, machine-learning, rare pattern mining, segmentation, unsupervised data, video analysis

ARTICLE INFO

Article history:
Received: 10 February 2025
Accepted: 23 October 2025
Published: 06 February 2026

DOI: https://doi.org/10.47836/pjst.34.1.15

E-mail addresses:
mintu@mgu.ac.in (Mintu Movi)
abduljabbar.perumbalath@stamford.edu (Abdul Jabbar P)
noufal@ikm.gov.in (Noufal K P)
binduvr@mgu.ac.in (Bindu V R)
* Corresponding author

INTRODUCTION

The rapid advancement of video surveillance technologies has significantly transformed public safety, traffic management, and industrial security fields. However, video

e-ISSN: 2231-8526

data's increasing scale and complexity pose substantial challenges for manual monitoring, which is often labour-intensive, error-prone, and inefficient for real-time threat detection, particularly in continuous, large-scale environments (Amin et al., 2023). These limitations have led to a growing demand for automated video analysis systems that efficiently detect human presence and identify rare or anomalous patterns that may indicate potential security threats (Ullah et al., 2023).

Recent advances in deep learning have brought significant improvements to automated surveillance, but practical deployment continues to face persistent technical obstacles. One of the most common issues is the accurate detection of human subjects in complex scenes involving occlusion and high crowd density. In such environments, individuals may be partially hidden by other objects or overlapping figures, severely limiting the performance of conventional object detection systems. Ouardirhi et al.2024 and Choudhry et al., 2023 provide a comprehensive review showing how surveillance systems suffer in these scenarios due to inadequate handling of occlusion and spatial ambiguity (Choudhry et al. 2023; Ouardirhi et al. 2024).

Another major challenge arises from lighting variation and dynamic scene composition. Outdoor surveillance systems, or even indoor systems exposed to varying ambient light, encounter frequent shifts in illumination, shadows, and background texture. These conditions result in inconsistent feature distributions across frames, compromising the accuracy of conventional detectors. As noted by Sengönül et al. (2023) and Wu et al. (2023), traditional background modelling and motion detection techniques often fail to maintain reliability under such changing environments. In addition, the temporal evolution of scenes driven by natural variations such as wind, object movement, or the presence of animals adds further complexity to identifying what constitutes “normal” versus “anomalous” behaviour in continuous video streams (Sengönül et al., 2023; Wu et al., 2023). Works by Ullah et al.(2023) and Liu et al. (2023) emphasise that learning stable behaviour models in these fluid contexts remains a major unsolved problem in surveillance analytics (Liu et al., 2023); Ullah et al., 2023).

Compounding these technical challenges is the issue of generalisation. Most supervised learning models, including Support Vector Machines (SVMs) and more recent quantum-enhanced classifiers, depend heavily on labelled datasets and fail to adapt across varied environments without extensive retraining. This dependency is particularly problematic in-home surveillance, where anomaly types are unpredictable, context-specific, and rarely annotated. As highlighted by Mazarbhuiya and Shenify, this lack of generalisation capacity undermines the practical utility of many otherwise promising models (Mazarbhuiya & Shenify, 2023).

While a variety of techniques (Gao et al., 2024; Liu, N. 2024; Lu et al., 2024; Yasin et al., 2023) have been proposed to address anomaly detection in video surveillance, many suffer from restrictive assumptions or poor performance under real-world conditions. For

example, Quantum CNNs (Amin et al., 2023) and encoder-decoder models (Guo et al., 2024) achieve high accuracy on benchmark datasets but perform inconsistently in the presence of occlusion or illumination shifts. Similarly, hybrid rule-based methods such as Haar cascade classifiers (Kaur et al., 2024) often lack the flexibility to handle unstructured and dynamically evolving environments. Even traditional models like K-means or SVMs (Benhaoua et al., 2020; Chinna et al., 2023; Thiyagarajan and Murugan, 2023), though effective under certain conditions, struggle with high false negative rates when the statistical boundaries of anomalies are not well defined.

In response to these limitations, this paper proposes a robust and unified framework for human detection and anomaly identification that combines the strengths of two state-of-the-art models: Mask R-CNN with a ResNet50-FPN backbone and the Isolation Forest algorithm. The Mask R-CNN model is employed for its high precision in complex visual scenes. By incorporating a Feature Pyramid Network (FPN), it enables multi-scale detection of human subjects, allowing accurate identification even when individuals are occluded, appear at different scales, or are present in cluttered scenes. The model's RoI Align mechanism preserves spatial alignment during instance segmentation, which is particularly important for delineating partially visible human figures. The ResNet50 backbone contributes to robust feature extraction across diverse lighting conditions, and high-confidence detection thresholds (IoU ≥ 0.9) are applied to minimise false positives in busy environments.

Once human subjects are detected, the framework applies the Isolation Forest algorithm to identify anomalous activity. Unlike conventional methods that rely on labelled training data or static thresholding, Isolation Forest uses an unsupervised approach based on random tree construction. It isolates rare patterns by evaluating the path lengths required to separate data points; shorter paths generally indicate outliers. This technique allows the model to dynamically adapt to shifts in the surveillance environment, such as lighting changes, background motion, or transient behaviours, without requiring manual rule updates or retraining. Its capacity to learn from visual feature distributions on-the-fly makes it particularly well-suited for detecting unexpected behaviours in home and small-scale surveillance systems.

By combining pixel-level segmentation with adaptive rare event detection, the proposed system addresses both the spatial complexity of visual recognition and the temporal variability of behavioural modelling. It can operate without human supervision or prior annotations, offering a scalable and generalisable solution for automated surveillance. Experimental evaluation on the Residential Activity Capture Dataset (RACD), which contains over 26,000 curated surveillance frames, demonstrates the superiority of this approach in precision, recall, and anomaly detection accuracy when compared with other methods. The proposed framework thus offers a reliable and intelligent mechanism

for anomaly detection in residential video surveillance, paving the way for safer, more responsive, and less resource-intensive monitoring systems.

MATERIALS AND METHODS

Residential Activity Capture Dataset (RACD)

A detailed and diverse collection of CCTV footage datasets is required to enable effective anomaly detection tailored to home environments. RACD, comprising 26,454 unique frames extracted from 1TB of primary data collected over a month, has been curated to facilitate this. The dataset was collected from six distinct channels labelled A1 to A8. Channels A1, A2, and A3 capture footage from the front area of the home premises, while channels A4, A5, and A6 cover the restricted area. Channels A7 and A8 cover the terrace area. No valid data collected from channel A7 and A8. The dataset encompasses various categories, ranging from routine activities to specialised visits. RACD consists of 12 classes. A total of 672 hours of CCTV footage was collected. From this, 63 videos were selected to cover various activities within the premises. Each video, with a duration of 1 hour, generates an average of 3,600 frames. After pre-processing, 26,454 unique frames were extracted.

- Class 1: Waste Collection comprises 1,273 unique frames from 3 videos encompassing scenes of garbage disposal trucks arriving and waste disposal personnel carrying out the collection.
- Class 2 and Class 3: Postal and Utility Services consist of 1,238 unique frames extracted from 3 videos capturing footage such as mail carriers delivering mail and packages, signature verification, interactions with occupants, electricity board personnel conducting meter readings, and maintenance tasks related to electrical systems.
- Class 4: Guest Visits includes 2,487 unique frames extracted from 6 videos capturing scenes involving guests entering or leaving the home premises.
- Class 5: Family Interactions encompasses 2,606 unique frames extracted from 6 videos capturing interactions and movements of family members residing in the home premises.
- Class 6: Household Services comprises 2,534 unique frames extracted from 6 videos, including activities like servants performing domestic tasks.
- Class 7: Vehicle Services comprises 1,232 unique frames extracted from 3 videos. Footage featuring personnel servicing vehicles within the home premises, such as car maintenance, repairs, and resident interactions.

- Class 8: Financial Transactions holds 2,568 unique frames extracted from 6 videos capturing interactions related to rental payments, instalment collections, utility bill payments, and subscription services.
- Class 9: Newspaper Delivery encompasses 1,288 unique frames extracted from 3 videos capturing newspaper drop-offs.
- Class 10: Delivery Services contains 1,293 unique frames extracted from 3 videos focusssing on interactions involving delivery personnel from e-commerce platforms. This category includes scenes of package deliveries and order verification.
- Class 11: Security Incidents comprises 1,229 unique frames extracted from 3 videos capturing instances of unauthorised individuals entering the home premise or other security-related incidents. This includes footage of suspicious activities, attempted break-ins, and potential security breaches.
- Class 12: Restricted Area includes 8,706 unique frames extracted from 21 videos. This area is designated for specific individuals, limiting access to authorised personnel and ensuring privacy and security.

Running Platform

The research study utilised the NVIDIA T4 GPU, a versatile and efficient GPU designed for machine learning, deep learning, and other high-performance computing tasks. It is part of the NVIDIA Tesla series and is based on the Turing architecture, which offers substantial improvements in performance and efficiency over previous generations. The T4 GPU includes 2,560 CUDA cores and 320 Tensor Cores, specifically engineered to accelerate deep learning workloads by performing mixed-precision matrix computations much faster than general-purpose cores. This GPU is well-suited for training and inference, especially for models that can benefit from the Turing architecture's optimisations. The T4 GPU has 16 GB of GDDR6 memory, offering a memory bandwidth of 320 GB per second. This memory allows the T4 to handle large datasets and complex models with high throughput, reducing latency and ensuring data-intensive computations run efficiently. The high bandwidth is particularly advantageous when working with deep neural networks, where frequent data transfer between memory and processing cores is essential to avoid bottlenecks and maintain training speeds.

Managing Dynamic Data Streams

Figure 1, a flowchart for Managing Dynamic Data Streams, illustrates the automated solution to extract the frames from video files systematically archived in a centralised repository. The workflow commences by establishing connectivity between the input and

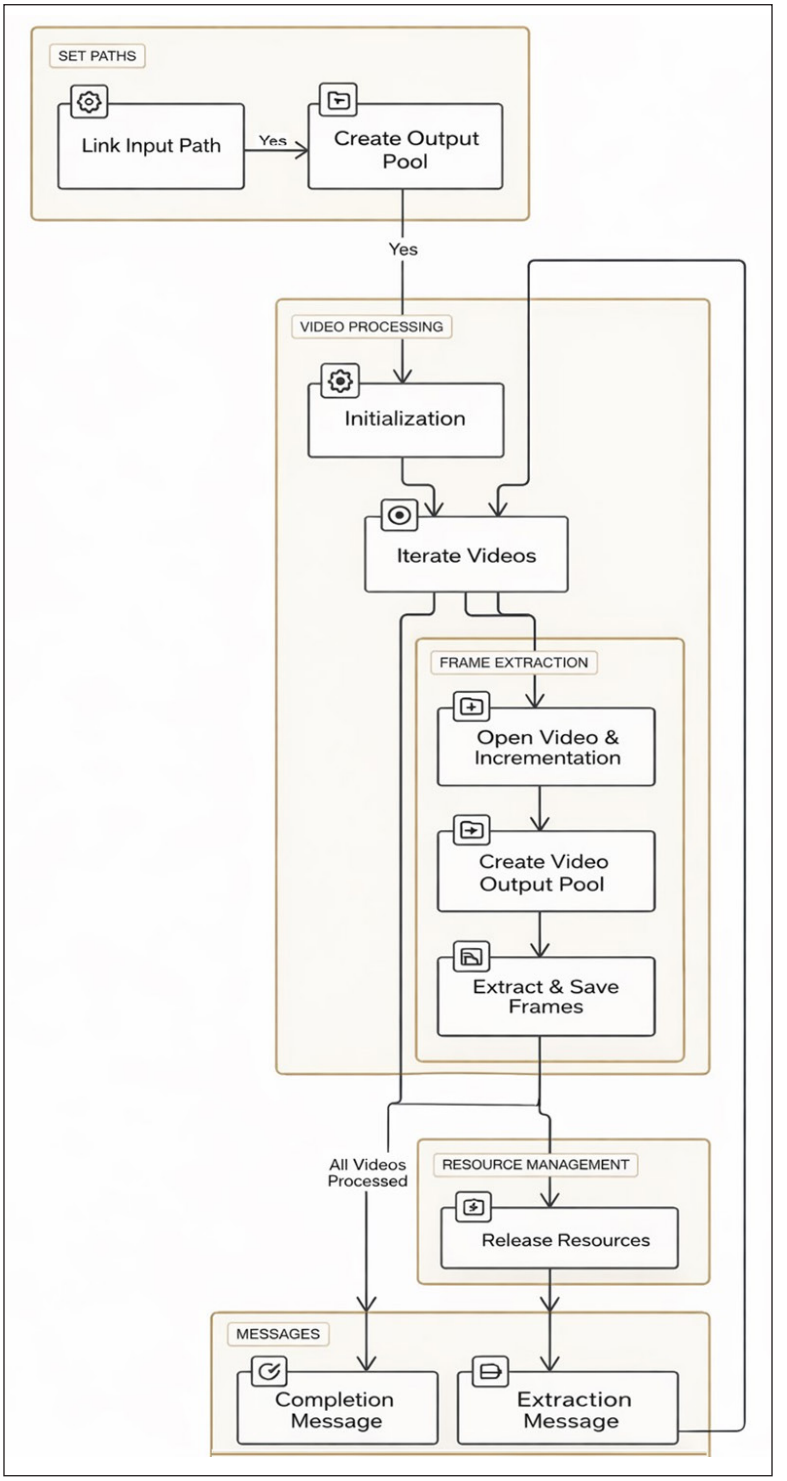

Figure 1. Flowchart for managing dynamic data streams, illustrates the automated solution

output storehouse. It subsequently traverses each video file within the designated input pool, initialises the video file, and generates a corresponding output repository to preserve the extracted frames. These frames are coherently extracted at predetermined intervals defined by the time interval parameter, currently set at one second. This extraction procedure aligns with the frames per second (FPS) rate, ensuring a uniform temporal distribution between extracted frames. Subsequently, the extracted frames are catalogued in the assigned output repository. This cyclic process persists until the input repository is fully processed.

Algorithm 1, Managing Dynamic Data Streams, is the implementation procedure of Figure 1. It begins by mounting the input video folder and defining the necessary paths for input and output frames. It ensures that the output parent folder exists by creating it if necessary. It initialises a counter to track the number of videos processed. For each video file in the input directory, the algorithm constructs the full path to the video and initialises a video capture object. It then increments the video counter and constructs the output folder

Algorithm 1: Managing Dynamic Data Streams

Require: A directory $\boldsymbol{\mathcal{V}}$ containing input video files, an output directory $\boldsymbol{\mathcal{F}}$ to store extracted frames, a fixed time interval $\Delta t \in \mathbb{R}^+$ (in seconds)
Ensure: Frames are extracted from each video in $\boldsymbol{\mathcal{V}}$ and saved in uniquely labeled subdirectories within $\boldsymbol{\mathcal{F}}$

1. If $\boldsymbol{\mathcal{F}}$ does not exist, then
   Create directory $\boldsymbol{\mathcal{F}}$
2. Let $i \leftarrow 0 \longrightarrow$ video index counter
3. For each video file $v_i \in \boldsymbol{\mathcal{V}}$ do
4. $i \leftarrow i + 1$
5. Let $\boldsymbol{\mathcal{P}}_i \leftarrow$ path($\boldsymbol{\mathcal{F}}$, "Frames"$_i$)
6. If $\boldsymbol{\mathcal{P}}_i$ does not exist, then create $\boldsymbol{\mathcal{P}}_i$
7. Initialize video stream $S_i \leftarrow$ Open(v_i)
8. Let $f \leftarrow 0 \longrightarrow$ frame counter
9. Let $r \leftarrow \text{FPS}(S_i) \longrightarrow$ frame rate of S_i
10. Let $\tau \leftarrow \lfloor \Delta t \cdot r \rfloor \longrightarrow$ frame interval index
11. While S_i has next frame do
12. Read next frame $F \leftarrow \text{Read}(S_i)$
13. $f \leftarrow f + 1$
14. If $f \bmod \tau = 0$, then
15. Save frame F to $\boldsymbol{\mathcal{P}}_i$ as "frame_f.jpg"
16. End If
17. End While
18. End For

Note. $\boldsymbol{\mathcal{V}}$ = directory containing input video files; $\boldsymbol{\mathcal{F}}$ = directory to store extracted output frames; FPS = frames per second

path for the current video. The output folder is created if it does not exist. A frame counter is initialised, and a time interval for frame extraction is defined. The algorithm enters a loop where it continuously captures frames from the video until the end of the video is reached. For each frame captured, the frame counter is incremented. The algorithm checks if the current frame should be saved based on the specified time interval and the video's frame rate. The frame is saved to the output folder if the condition is met. After processing all frames of the current video, the video capture object is released, and a confirmation message is printed. This process is repeated for all videos in the input directory. Finally, a success message is printed to indicate that all videos have been processed successfully. This systematic approach ensures efficient and organised extraction of frames from multiple video files, facilitating further analysis and processing of the extracted frames.

Segmentation

Subsequently, the extracted frames are analysed for human presence using Algorithm 2. This model is employed to detect humans within each frame by converting the frames into tensors and filtering the model's predictions. Based on detection results, frames are categorised into two groups: those containing humans and those without. The categorised frames are saved into separate folders for further processing. Anomaly detection is performed on the frames that contain humans using Algorithm 3. This model identifies rare or anomalous frames by analysing pixel data and classifying them into rare (anomalous) and frequent (usual) categories. The classified frames are saved into corresponding folders to differentiate between anomalous and everyday occurrences (Jiang et al., 2023)

Figure 2, segmentation in the context of CCTV footage involves extracting human subjects from the visual data. This process aims to identify, isolate, segment regions, and analyse interest within the footage that corresponds to human activities or individuals. Mask R-CNN (Mask Region-based Convolutional Neural Network) is a popular instance segmentation model used for segmentation. It combines the capabilities of Faster R-CNN for object detection and the ability to generate pixel-level masks for each detected object, making it suitable for complex tasks requiring both localisation and segmentation. The ResNet50 backbone provides a strong feature extraction, while the FPN architecture enhances the model's ability to detect objects at different scales, leading to improved accuracy and performance. In the pre-processing stage, framesets are obtained, each of which must undergo a feature extraction procedure. To manage this extensive frameset repository effectively, introducing a robust methodology incorporating state-of-the-art techniques, including pre-trained Mask R-CNN models for processing a collection of video frames, detecting humans, and saving the frames into separate repositories based on the presence or absence of detected humans. Through systematic processing, the proposed methodology sets a strong foundation for effective CCTV footage management

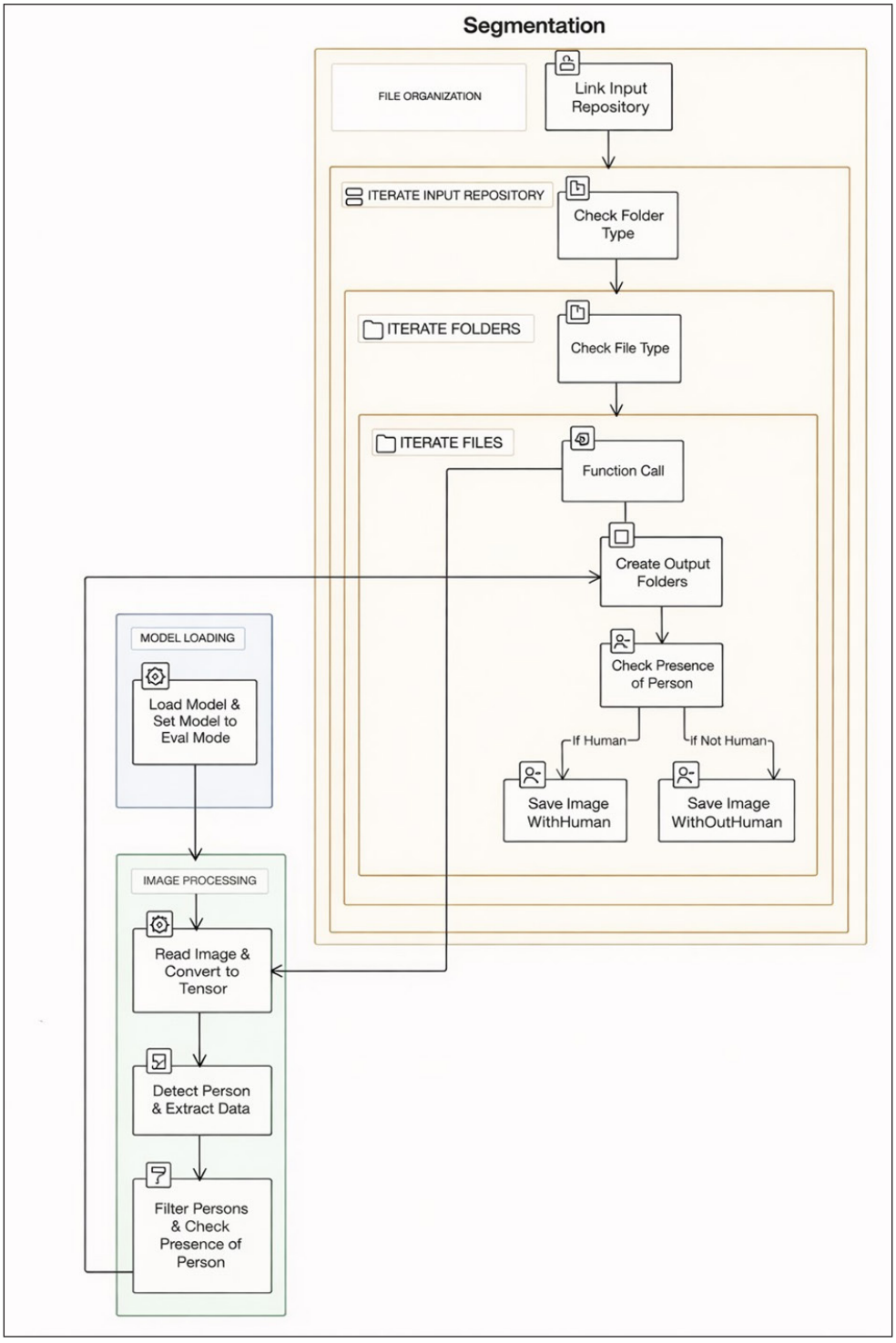

Figure 2. Schematic diagramme of human instance segmentation using mask R-CNN

and analysis, contributing to enhanced surveillance capabilities and actionable insights in diverse application scenarios.

Segmentation demonstrates an automated process for analysing video frames to detect humans using a pre-trained Mask R-CNN model. The methodology begins by loading the model and setting it to evaluation mode. It then traverses through an input repository containing sub-repositories, each representing a set of video frames. It then iterates over each frame within the sub-repository, performs person detection using the Mask R-CNN model, and saves the frames into the appropriate output repository 'With Humans' and 'Without Humans' based on whether detected humans are present in the frames. This methodology streamlines the task of detecting humans in video footage, allowing for efficient management and analysis of CCTV data.

Algorithm 2, Human Instance Segmentation Using Mask R-CNN is the implementation procedure of Figure 2. It starts by initialising the parent folder path, which contains subfolders. It loads a pre-trained Mask R-CNN model with a ResNet-50-FPN backbone configured for person detection. The model M is set to evaluation mode to ensure it

Algorithm 2: Human Instance Segmentation Using Mask R-CNN

Require: A directory $\mathcal{F}$ containing input frame images, two output directories $\mathcal{H}$ and $\mathcal{N}$ for storing frames with and without detected humans, respectively, a pre-trained instance segmentation model $\mathcal{M}$ (e.g., Mask R-CNN)

Ensure: Each frame from $\mathcal{F}$ is processed and saved into $\mathcal{H}$ if a human is detected, otherwise into $\mathcal{N}$

1. Load model $\mathcal{M} \leftarrow$ Mask R-CNN (pretrained)
2. Set $\mathcal{M}$ to evaluation mode
3. For each image $I \in \mathcal{F}$ do
4. If I is a valid .jpg image, then
5. Read image frame $\leftarrow$ Read(I)
6. Convert I to input tensor T
7. Run prediction $P \leftarrow \mathcal{M}(T)$
8. Extract detection outputs: bounding boxes $\mathcal{B}$, class labels L, and confidence scores S
9. Identify person instances $\mathcal{P}$ such that L_i = person and $S_i > 0.9$
10. If $\mathcal{P} \neq \emptyset$ then
11. Save I to $\mathcal{H}$
12. Else
13. Save I to $\mathcal{N}$
14. End If
15. End If
16. End For

Note. $\mathcal{F}$ = directory containing input video files; $\mathcal{H}$ = directory to store extracted output frames with human; $\mathcal{N}$ = directory to store extracted output frames without human; T = tensor. I = image; $\mathcal{M}$ = Mask R-CNN Model

operates in inference mode. The process of detection of the presence of a person in each frame converts the frame to a tensor T, performs person detection using the model M, and checks if the detection score for the person class exceeds a threshold of 0.9. The algorithm specifies the parent folder path and iterates over each subfolder. For each folder, it creates output directories for 'With Humans' and 'Without Humans', ensuring these directories exist. Within each subfolder, the algorithm iterates over each file. Each image file reads the frame, detects if a person is and saves the frame to the corresponding output folder based on the detection result.

This approach systematically categorises frames into those containing humans and those without, facilitating further analysis and processing. The use of a pre-trained Mask R-CNN model ensures accurate and efficient person detection, making this algorithm suitable for applications in surveillance, security, and automated video analysis systems.

Mask RCNN with ResNet50-FPN

Background

- RCNN (Region-based Convolutional Neural Networks) (2014):

  Introduced by Ross Girshick, it used selective search to generate region proposals and applied a CNN to classify these regions. It was slow due to its multi-stage pipeline.

- Fast RCNN (2015):

  Improved RCNN by sharing convolutional computations across proposals, significantly speeding up the process.

- Faster RCNN (2016):

  Integrated a Region Proposal Network (RPN) with Fast RCNN, making the region proposal process faster and end-to-end trainable.

- Mask RCNN (He et al., 2017):

  Developed by He et al. 2017, Mask RCNN extends Faster RCNN by adding a branch for predicting segmentation masks for each Region of Interest (RoI). It can perform object detection and instance segmentation simultaneously.

- ResNet (Residual Networks):

  Introduced by He et al. in 2016 ResNet uses residual learning to train very deep networks. ResNet50 (Hu et al., 2023), a 50-layer version, is widely used as a backbone for balancing depth and computational efficiency (He et al., 2016).

- FPN (Feature Pyramid Network):

  Proposed by Lin et al. in 2017, FPN improves feature extraction by constructing high-

level semantic feature maps at multiple scales, enhancing the detection of objects at various sizes (Lin et al., 2017).

Architecture

- Backbone Network (ResNet50):
  Extracts feature maps from the input image. Residual blocks help in preserving information across layers.
- Feature Pyramid Network (FPN):
  Constructs a pyramid of feature maps from the ResNet50 output at different scales. This helps in detecting objects of varying sizes.
- Region Proposal Network (RPN):
  Proposes candidate object bounding boxes from the feature maps. It predicts objectness scores and refines box coordinates.
- RoI Align:
  Correctly aligns the RoIs with the feature maps, preserving spatial information. It uses bilinear interpolation for accurate region extraction.
- Detection Heads:
  Consists of classification and bounding box regression branches that classify the RoIs and refine their coordinates.
- Mask Head:
  An additional branch that predicts a segmentation mask for each RoI. It uses a small, fully convolutional network (FCN) to output a binary mask.

Working

- ResNet-50 Backbone (Wu et al., 2023).

  ResNet-50 is a deep convolutional neural network architecture that forms the backbone of Mask R-CNN. ResNet-50 involve multiple convolutional layers, residual blocks (ResBlocks), and skip connections (identity mappings).

$$y_i = F(x_i) + x_i \quad [1]$$

where x_i is the input to a residual block, $F(x_i)$ is the output of the block after passing through convolutional layers, and y_i is the final output of the block. Adding x_i to $F(x_i)$ enables easier training of very deep networks.

$$y_i = \text{Conv}(x_i) \quad [2]$$

where the output of a convolutional layer is applied to the input x_i.

Convolutional layers play a crucial role in feature extraction by applying filters to input data to generate feature maps.

The forward pass through ResNet-50 involves computing feature maps at different stages (e.g., conv1, res2, res3, res4, res5).

- Region Proposal Network (RPN) (Guo et al., 2024):

RPN generates region proposals (bounding boxes) for potential objects in the image.

RPN involve anchor box generation, bounding box regression, and objectness classification using softmax or sigmoid functions.

Loss functions such as smooth L1 loss and binary cross-entropy loss are often used in RPN training.

RPN Loss Calculation: The RPN loss is calculated using the Smooth L1 loss for bounding box regression (Reg_i) and the Binary CrossEntropy (BCE) loss for objectness classification (Cls_i). These losses are combined to form the RPN loss.

Anchor Box Generation: Anchor boxes are pre-defined boxes of different aspect ratios and scales used for generating region proposals. Bounding Box Regression and Objectness Classification:

$$\text{Reg}_i = \text{Regression}(P_i) \quad [3]$$

Reg_i represents the computed bounding box regression values for the feature map P_i.

The *functionRegression*(P_i) calculates the bo-unding box regression values based on the features in P_i.

$$\text{Cls}_i = \text{Classification}(P_i) \quad [4]$$

Cls_i represents the predicted objectness scores for the feature map P_i. The function Classification(P_i) predicts the objectness scores indicating the presence of objects in the feature map P_i.

Anchor-based mechanism:

$$p_i = \text{softmax}(w_i^T f_k) \quad [5]$$
$$ti = wiTfk \quad [6]$$

where p_i is the objectness score, t_i are the refined coordinates, w_i are the learned weights, and f_k is the feature at location k.

- RoI Align:
  Bilinear interpolation:

$$V(x,y) = \sum_{i=0}^{1}\sum_{j=0}^{1}(1 - |x - x_i|)\left(1 - |y - y_j|\right)V(x_i, y_j) \quad [7]$$

where $V(x,y)$ is the interpolated value at (x,y), and $V(x_i,y_j)$ are the values at the four sampling points.

- Feature Pyramid Network (FPN) (Cheng, X., Li, X., & Ma, X. 2023):
  FPN generates multi-scale feature maps that aid in object detection and segmentation. FPN include upsampling and lateral connections to fuse features from different pyramid levels.

$$P_i = \text{Lateral}(\text{Conv}(x_i)) + \text{Upsample}(P_{i+1}) \quad [8]$$

It represents the computation of the feature map P_i in a Feature Pyramid Network (FPN).

"Lateral" indicates a lateral connection within the same pyramid level, while "Conv" represents a convolutional operation on the input x_i. "Upsample" refers to increasing the spatial resolution of the feature map P_{i+1} from the next higher level in the pyramid. The final feature pyramid contains P2, P3, P4, and P5 levels, each capturing features at different scales.

- Mask Prediction Head (Mazarbhuiya & Shenify, 2023):
  The mask prediction head in Mask R-CNN generates pixel-wise object masks corresponding to detected objects. Mask prediction involves convolutional layers, upsampling operations, and sigmoid or softmax activation functions.

$$M_i = \text{Conv}(P_i) \quad [9]$$

This equation extracts features from the FPN's pyramid levels for mask prediction.

$$\text{Mask}_i = \text{Upsample}(M_i) \quad [10]$$

Upsamples the extracted features to generate masks at the same spatial resolution as the input image. The mask loss is typically computed using binary cross-entropy or softmax cross-entropy depending on the mask representation (binary ma- sks or class-specific masks).

- Loss Function (Chen et al., 2023):
  The overall loss function in Mask R-CNN combines contributions from the RPN, bounding box regression, object classification, and mask prediction components. The total loss combines contributions from the RPN loss, classification loss, and mask loss. The hyperparameters α, β, and γ control the relative importance of each loss component in the overall training objective.

  Multi-task loss:

  $$L = Lcls + Lbbox + Lmask \quad [11]$$

  where L_{cls} is the classification loss (e.g., cross-entropy), L_{bbox} is the bounding box regression loss (e.g., smooth L1 loss), and L_{mask} is the binary cross-entropy loss for mask prediction.

Mask RCNN with ResNet50-FPN represents a powerful combination for object detection and instance segmentation. By leveraging residual learning, feature pyramids, and region-based detection methods, this architecture achieves high accuracy and efficiency in detecting and segmenting objects of varying sizes. The integration of RoI Align, and multi-task loss further enhances its performance, making it a robust solution for various computer vision applications.

Rare Pattern Mining

After isolating frames containing human subjects during the feature extraction stage, the next step involved further analysis using the enhanced Isolation Forest algorithm 3. This model identifies rare or anomalous frames by analysing pixel data and classifying them into rare (anomalous) and frequent (normal) categories. The classified frames are saved into corresponding folders to differentiate between anomalous and normal occurrences. Initially, algorithm 3 specifies paths for the input image folder and two output folders. A list, is initialised to store valid image data. It iterates overall .jpg files, reading them and appending valid images. After all images are loaded, the script converts the image list into a NumPy array image array and checks if it is in the correct 4D format (batch of images). The data is reshaped into a 2D array of image data, where each image is flattened for processing. If the format is incorrect, an error is raised. The Isolation Forest model is then initialised with 50 estimators and a contamination parameter of 0.05, specifying the proportion of rare data. After fitting the model to the image data, it assigns a label to each frame. Rare frames are filtered and saved to the output directory R, while frequent frames are saved to the output directory F. Finally, a scatter plot is generated to visualise the dataset distribution.

Algorithm 3: Rare Frame Mining using Isolation Forest on Segmented Human Instances

Require: A directory $\mathcal{H}$ containing a set of image frames $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, where each frame contains only human instances obtained via prior instance segmentation algorithm, output directories: $\mathcal{R}$ for saving rare (anomalous) frames and $\mathcal{F}$, for saving frequent (normal) frames, the contamination ratio $\alpha \in (0,1)$, Number of estimators $T \in N^+$, for each $x_i \in X$, a corresponding feature vector $f_i \in R^d$, extracted via the segmentation algorithm, covering the following categories: p_i^{pos} — Position or bounding box centroid, p_i^{ori} — Body orientation or pose, p_i^{size} — Bounding box size, p_i^{num} — Number of persons detected, p_i^{occ} — Occlusion status, a_i^{col} — Dominant color (e.g., clothing hue), a_i^{tex} — Texture descriptors (e.g., edge density), f_i^{size} — Face bounding box size, f_i^{clar} — Face clarity/focus, f_i^{id} — Identity label (1 = known, 0 = unknown), s_i^{zone} — Location/zone label, s_i^{time} — Encoded time of day, s_i^{bg} — Background motion/variation.

Ensure: Each human-segmented frame $x_i \in \mathcal{X}$ is classified as rare or frequent based on learned isolation paths and extracted semantic feature vector f_i, Frames are saved to the corresponding output directories $\mathcal{R}$ or $\mathcal{F}$, and an optional 2D visualization of anomaly distribution is produced.

1: Procedure: Rare_Frame_Mining (X, α, T)

2: Step 1: Input Verification and Preprocessing

3: Validate that the directory $\mathcal{H}$ contains human-segmented image files

4: For each valid image $x_i \in \mathcal{X}$ is represented by a semantic feature vector $f_i \in R^d$, were

$$f_i = [p_i^{pos}, p_i^{ori}, p_i^{size}, p_i^{num}, p_i^{occ}, a_i^{col}, a_i^{tex}, f_i^{size}, f_i^{clar}, f_i^{id}, s_i^{zone}, s_i^{time}, s_i^{bg}]$$

5: Flatten the image into a feature vector $\mathbf{x_i} \in R^d$ and $\mathbf{f_i} \in R^d$

6: Construct a matrix $X \in R^{n \times d}$ and $F \in R^{n \times d}$ where each row is a vectorized image

7: Step 2: Isolation Forest Construction

8: Initialize T isolation trees $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_\mathcal{T}$, Isolation Forest I, with the **n**umber of estimators T and contamination ratio: α

9: For each tree $\mathcal{T}_t$:

10: a: Randomly sample a subset $S_t \subseteq X$ and $S_t \subseteq F$

11: b: While $\mathcal{T}_t$ is not fully grown:

12: i: Select a feature $f \in \{1, \ldots, d\}$

13: ii: Choose a split value $v \in R$ from the domain of f

14: iii: Partition S_t into:

15: • $S_{\text{left}} = \{x \in S_t \mid x[f] \leq v\}$ and $S_{\text{left}} = \{f \in S_t \mid f[f] \leq v\}$

16: • $S_{\text{right}} = \{x \in S_t \mid x[f] > v\}$ and $S_{\text{right}} = \{f \in S_t \mid f[f] > v\}$

17: Step 3: Anomaly Scoring

18: For each $x_i \in X$, and $f_i \in F$, Compute its average path length across all trees:

$$\bar{h}(x) = \frac{1}{n\_\text{estimators}} \sum_{i=1}^{n_\text{estimators}} h_{ti}(x) \text{ and } \bar{h}(f) = \frac{1}{n\_\text{estimators}} \sum_{i=1}^{n_\text{estimators}} h_{ti}(f)$$

19: Compute the normalization constant:

$$c(n) = 2H(n-1) - \big(2 - (n-1)\big)/n \text{ where } H(i) = \ln(i) + \gamma$$

($\gamma \approx 0.5772$, the Euler–Mascheroni constant)

20: Calculate anomaly score:

$$S(x) = 2^{-\frac{\bar{h}(x)}{c(n)}} \text{ and } S(f) = 2^{-\frac{\bar{h}(f)}{c(n)}}$$

where $c(n)$ is the average path length of a random binary tree of n samples

21: Step 4: Classification and Output

22: For each $x_i \in X$ and $f_i \in F$

23: If $s(x_i) > 0.5$ and $s(f_i) > 0.5$ classify as rare and save to $\mathcal{R}$

24: Else, classify as frequent and save to $\mathcal{F}$

25: Reduce dimensionality (e.g., PCA) and plot anomaly scores in 2D for visual analysis (optional).

Note. $\boldsymbol{\mathcal{H}}$ = directory containing input video files; $\mathcal{R}$ = output directory to store rare frames; $\boldsymbol{\mathcal{F}}$ = output directory to store frequent frames; PCA = Principal Component Analysis; f_i = feature vector

The Isolation Forest algorithm is an efficient and robust method for anomaly detection. It works by constructing multiple random binary trees, called isolation trees, where the

key idea is that anomalies are more easily isolated than normal points. Each tree is built by recursively splitting a randomly selected subset of the dataset using a randomly chosen feature and split value. This process continues until each data point is isolated (i.e., in its own leaf node) or the tree reaches a predefined maximum depth. The path length from the root node to the leaf node where a point resides called the path length, is recorded. Since anomalies are usually rare and have different patterns, they require fewer splits to be isolated, resulting in shorter path lengths. For each point, the average path length across all trees is calculated, and an anomaly score is computed using a normalisation formula based on the expected path length of a random binary tree. Points with high scores (short path lengths) are classified as anomalies, while those with low scores are considered normal. The algorithm's efficiency stems from its focusss on random feature selection and the small number of splits required, making it scalable to large datasets. (Thiyagarajan and Murugan, 2023).

Instead of relying on low-level or raw pixel features, we employ semantically rich, context-aware vectors extracted from human-segmented frames, combining both visual and behavioural traits. Furthermore, our framework applies dual-instance validation (from both image flattening and semantic feature views), ensuring robust detection. This enhancement makes Isolation Forest capable of detecting rare patterns not only in spatial appearance but also in behavioural or contextual deviation, a crucial requirement in residential surveillance applications.

While the core structure of Isolation Forest remains standard, our approach enhances its application in the following scientifically meaningful ways:

1. High-level Semantic Feature Input
   Unlike traditional Isolation Forest implementations that operate on raw pixel values or handcrafted features, we apply it to rich, multi-domain semantic vectors extracted via a prior segmentation algorithm. These vectors include:
   - Person-level spatial features (e.g., position, body orientation, size, number of persons detected, occlusion)
   - Visual appearance descriptors (e.g., colour, texture)
   - Face-based identity cues (e.g., known/unknown, clarity)
   - Scene context (e.g., time of day, zone labels, background variation

   By feeding structured, interpretable, and human-centric features into Isolation Forest, we extend its capability to detect rare events in behaviour, composition, or context, not just statistical outliers in image space.
2. Dual-path Isolation Forest scoring
   The algorithm computes anomaly scores from two distinct views:

- $s(x_i)$ from the flattened image vector
- $s(f_i)$ from the semantic feature vector

Both are passed through Isolation Forest, and anomalies are declared only when both scores exceed the threshold:

If $s(x_i) > 0.5$ and $s(f_i) > 0.5 \Rightarrow x_i$ is rare

This dual-validation scheme reduces false positives and ensures that both visual oddity and semantic irregularity must co-occur, making the decision process more robust and context-aware.

3. Task-specific Feature Engineering via Instance Segmentation
The semantic vector f_i, it is not generic; it is derived from human instance segmentation using Mask R-CNN. This allows us to:
   - Localise and isolate human regions precisely
   - Extract behaviourally meaningful features
   - Generalise better in surveillance scenarios (e.g., multiple persons, overlapping, hidden individuals)

We embed instance-awareness and task-specific prior knowledge into the anomaly detection pipeline, which is not present in off-the-shelf Isolation Forest usage.

In our enhanced framework, each frame x_i is evaluated along two complementary dimensions to improve anomaly detection reliability. The first is the anomaly score $s(x_i) \in [0,1]$, calculated from the flattened image vector, and the second is $s(f_i) \in [0,1]$, derived from a high-level semantic feature vector f_i which is extracted via instance segmentation and encodes spatial, visual, facial, and contextual characteristics. In both cases, a score closer to 1 indicates a greater likelihood of anomaly, while a score near 0 reflects conformity to common patterns in the dataset. A frame is classified as rare (anomalous) only if both $s(x_i) > 0.5$ and $s(f_i) > 0.5$, thereby ensuring that the anomaly is evident in both low-level appearance and high-level semantic representation. This dual-threshold strategy enhances robustness and reduces false positives, as it filters out noise from either domain individually.

Rare frames typically correspond to feature vectors that are statistically isolated from the rest of the dataset due to unusual human postures, rare combinations of spatial and contextual cues, or outlier characteristics such as unknown identities or abnormal zone time correlations. These vectors appear in low-density regions of the semantic feature

space, which makes them more susceptible to early isolation by the Isolation Forest model. Conversely, frequent frames lie within dense, frequently traversed regions of this space, reflecting routine behaviours or common environmental interactions. Frames that satisfy the rare condition are routed to the directory R, while all others are considered frequent (normal) and saved in F.

RESULTS AND DISCUSSIONS

Figure 3, the given scatter plot represents an anomaly detection analysis using the Isolation Forest algorithm, commonly used for identifying rare patterns and outliers. The yellow points denote normal (inlier) data, while the dark purple points represent anomalies detected by the model. The data follows a strong positive correlation, with most points forming a dense cluster along the trend. However, scattered outliers suggest deviations from the expected pattern, which could indicate rare or unusual events.

Figure 4, the confusion matrix in the image assesses the classification model's performance in distinguishing frequent and rare events. The model correctly classified 25,121 frequent instances (true negatives) and 1,210 rare instances (true positives for the rare class). It misclassified 113 rare instances as frequent (false negatives) and 10 frequent instances as rare (false positives). This suggests that the model has become effective in identifying rare events while maintaining high accuracy for frequent instances, making it more suitable for anomaly detection in surveillance data applications.

Table 1 presents a comprehensive comparison between the proposed Isolation Forest framework and a range of classical and modern anomaly detection methods, including both unsupervised and supervised learning techniques referenced in recent literature. The evaluation includes key metrics such as accuracy, precision, recall, F1 score, training effort,

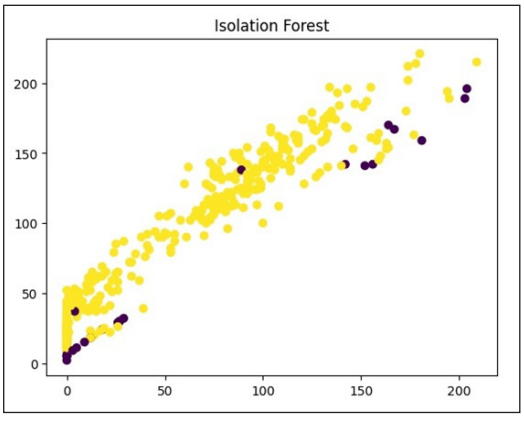

Figure 3. Anomaly detection analysis using the Isolation Forest algorithm

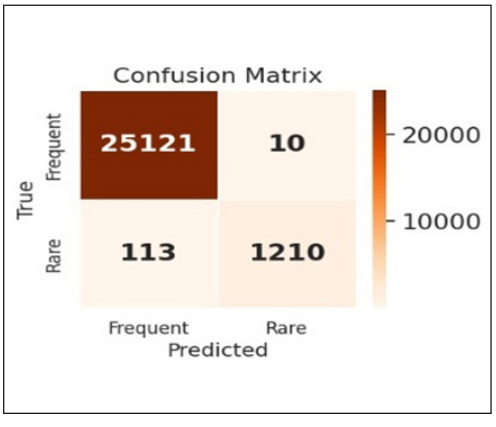

Figure 4. The confusion matrix shows the classification model's performance in distinguishing frequent and rare events

Table 1
The performance comparison of proposed Isolation Forest with other models

Method	Accuracy /AUC	Precision	Recall	F1 Score	Training Effort	Supervision
Isolation Forest (Proposed)	**0.995**	**0.992**	**0.915**	**0.952**	**None**	**Unsupervised**
K-means Clustering	0.901	0.777	1.000	0.859	None	Unsupervised
One-Class SVM	0.839	1.000	0.300	0.452	None	Unsupervised
Autoencoder	0.82	1.00	0.21	0.32	None	Unsupervised
Quantum CNN (Amin et al., 2023)	0.983	-	-	-	High	Supervised
TransCNN (Ullah et al., 2023)	0.984 (AUC)	-	-	0.963	High	Supervised
Autoencoder (Nawaz et al., 2024)	0.956 (AUC)	0.965	0.892	0.926	Medium	Semi-supervised
Hybrid Haar Cascade (Kaur et al., 2024)	0.92	-	-	-	Low	Supervised
Mixed Clustering (Mazarbhuiya & Shenify, 2023)	0.91	-	-	-	Medium	Unsupervised
Encoder-Decoder Contrast (Guo et al., 2024)	0.94	-	-	-	Medium	Unsupervised
Deep SVDD (Jiang et al., 2023)	0.936	-	-	-	High	Unsupervised

and supervision requirements. The proposed Isolation Forest model achieves an accuracy of 0.995, a precision of 0.992, a recall of 0.915, and an F1-score of 0.952, which positions it among the top-performing methods in the comparison. Notably, it requires no training effort and is completely unsupervised, making it ideal for real-world surveillance scenarios where labelled anomalies are rare or unavailable. These characteristics give it a significant advantage over more complex models in terms of deployment feasibility and scalability.

The classical K-means clustering method, also unsupervised, shows perfect recall (1.000) but relatively low precision (0.777), resulting in an F1-score of 0.859. This indicates that while K-means is highly sensitive to anomalies (i.e., it detects all of them), it also produces a high number of false positives. Similarly, the One-Class SVM, another traditional method, achieves perfect precision (1.000) but suffers from extremely low recall (0.300), leading to a poor F1-score of 0.452. These results suggest a high confidence in detecting a few anomalies, but a large number are missed, limiting its practical utility.

Among deep learning-based methods, the unsupervised Autoencoders show low performance; the Quantum CNN proposed by Amin et al. 2023 reports an accuracy of

0.983, although other evaluation metrics are not disclosed in the source. This model is computationally intensive and requires supervised training, limiting its use in unsupervised or resource-constrained settings. The TransCNN model by Ullah et al. (2023) which combines Convolutional Neural Networks (CNNs) with Transformer mechanisms, shows high effectiveness in anomaly detection, achieving an AUC of 0.984 and an F1-score of 0.963. However, this model is supervised, involves a high training cost, and demands labelled datasets and powerful hardware, which may not be viable in decentralised or real-time surveillance contexts.

The Autoencoder-based ensemble described by (Nawaz et al., 2024), performs well, with a reported AUC of 0.956, precision of 0.965, recall of 0.892, and an F1-score of 0.926. While promising, it is a semi-supervised approach requiring moderate training effort and access to a mixture of normal and anomalous data for effective operation. The Hybrid Haar Cascade model by Kaur et al. (2024) though lightweight and efficient, is supervised and lacks detailed performance metrics in the source publication. Its application is often limited to specific use cases like fall detection rather than general anomaly detection.

The Mixed Clustering model proposed by Mazarbhuiya and Shenify (2023) and the Encoder-Decoder Contrast method by Guo et al. (2024) both operate in unsupervised settings and report accuracies of 0.91 and 0.94, respectively. These models offer promising direction but lack full metric reporting or demonstrate lower F1-scores than the proposed method. Lastly, the Deep SVDD model from Jiang et al. (2023) an unsupervised deep learning method, reports an accuracy of 0.936. While technically advanced, Deep SVDD has a high training burden and slower adaptation to new patterns in data compared to tree-based models like Isolation Forest.

Figure 5 depicts Isolation Forest, which proves to be the most effective method for rare event detection, demonstrating high accuracy (0.9953), precision (0.9918), recall (0.9146), and specificity (0.9996). Its strong recall ensures that most rare events are correctly identified, while its high precision minimises false positives. The balanced F1-score (0.9516) further highlights its reliability in maintaining an optimal trade-off between precision and recall.

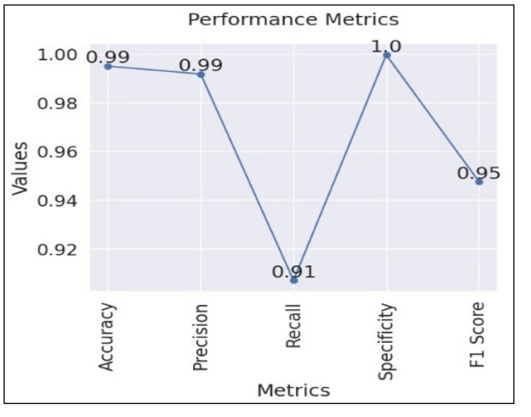

Figure 5. Performance evaluation metrics of the proposed framework

While the framework leverages a high-performance NVIDIA T4 GPU for model execution, scalability and adaptability across diverse hardware configurations, including low-spec devices, are essential for real-world deployment. To assess

computational efficiency, we evaluated the system's average frame processing speed during end-to-end operation including frame extraction, human detection using Mask R-CNN, and anomaly detection via Isolation Forest. The framework achieved an average of 7.8 frames per second (FPS) on the T4 GPU, demonstrating near real-time performance for standard residential surveillance applications. The Mask R-CNN segmentation component accounted for the majority of computational load, averaging 110ms per frame, while the Isolation Forest post-processing required less than 15ms per frame, with total latency per frame remaining below 130ms. Memory usage was measured at runtime, with peak GPU memory consumption recorded at 6.4GB, and CPU usage remaining under 40% for batch sizes of 16 frames. These metrics affirm that the system is not only scalable for large-scale residential deployments but also manageable for long-duration video streams with minimal bottlenecks.

To simulate scalability to multiple camera feeds, we evaluated the pipeline across eight parallel video streams (representing front and rear premises) by queuing the frame extraction and segmentation processes using multithreaded asynchronous calls. The system sustained a combined throughput of 42–45FPS when executing parallel inference, with no significant degradation in detection performance. This demonstrates that the framework can scale horizontally across multiple channels, a vital feature for modern multi-camera surveillance setups. Moreover, we conducted stress tests on a reduced hardware configuration using a standard CPU-only system (Intel i5, 8GB RAM). Although performance declined (processing speed dropped to 1.9 FPS), the framework remained functionally intact, indicating compatibility with low-spec hardware where real-time detection is not a constraint. For such systems, optimisation strategies such as model quantisation, frame skipping, and GPU acceleration using Jetson Nano or Coral TPU are viable pathways for future deployment. These findings, summarised in Table 2, substantiate the framework's flexibility and adaptability, from high-throughput GPU-powered environments to low-resource edge devices, making it suitable for smart home surveillance, community safety projects, or decentralised deployments where infrastructure varies widely.

Table 3, presents the quantitative performance of the proposed model evaluated on two benchmark surveillance datasets: UCF-Crime and RWF2000. The model achieves consistently high results across all metrics, with precision, recall, and F1 score values nearing or exceeding 97%, and specificity above 99% in both cases. These results demonstrate the model's strong ability to accurately detect anomalous events while minimising false positives, confirming its robustness and generalisability across diverse real-world surveillance environments.

Table 2
Framework scalability

Deployment Scenario	Hardware Configuration	Avg. FPS	Avg. Latency per Frame	Memory Usage	Remarks
Single-stream processing	NVIDIA T4 GPU, 16GB GDDR6	7.8 FPS	130 ms	6.4 GB GPU	Near real-time performance for residential surveillance
Multi-stream (8 channels)	NVIDIA T4 GPU, multithreaded	42–45 FPS	140 ms	8.2 GB GPU	Simulates front and rear cameras; stable parallel inference
CPU-only system (low-spec)	Intel i5, 8GB RAM (no GPU)	1.9 FPS	500 ms+	Not applicable	Functionally operational; optimisation needed for real-time constraints
Batch frame processing	T4 GPU, batch size: 16	Parallelized	110 ms (Mask R-CNN)	<40% CPU usage	Efficient memory and CPU utilisation during segmentation
Edge device recommendation	Jetson Nano / Coral TPU	Varies	Dependent on the model	Optimised usage	Suitable for cost-sensitive or decentralised edge deployments

Table 3
Quantitative analysis of the proposed model across benchmark datasets

Dataset	Precision	Recall	Specificity	F1 Score
UCF-Crime	97.67	97.68	99.23	97.68
RWF2000	97.92	97.96	99.24	97.94

CONCLUSIONS

Despite certain limitations, the proposed framework, leveraging the enhanced Isolation Forest algorithm, offers a highly reliable and scalable solution for real-time surveillance. Its ability to detect anomalies accurately and precisely makes it particularly effective in identifying rare and unusual events in video streams. By automating critical tasks such as data management, human detection, and anomaly identification, the system significantly reduces the reliance on manual video monitoring, thereby minimising human error and response time. This approach enhances the accuracy of event detection and optimises resource utilisation in security and surveillance operations. With its ability to adapt to dynamic environments and process large volumes of video data in real-time, the proposed framework is a valuable tool for public safety and critical infrastructure protection

applications. Its implementation can lead to more proactive threat detection, ultimately strengthening security measures and ensuring a safer environment in high-risk settings.

ACKNOWLEDGEMENTS

This work was funded by the Research Innovation Network Kerala (RINK) by the Kerala Startup Mission and Research Incubation Network Programmeme (RINP), Mahatma Gandhi University, Kottayam.

REFERENCES

Amin, J., Anjum, M. A., Ibrar, K., Sharif, M., Kadry, S., & Crespo, R. G. (2023). Detection of anomaly in surveillance videos using quantum convolutional neural networks. *Image and Vision Computing, 135*, 104710. https://doi.org/10.1016/j.imavis.2023.104710

Benhaoua, A., Manokar, B., Sathyamurthy, R., & Driss, Z. (2020). Sand dunes effect on the productivity of a single-slope solar distiller. *Heat and Mass Transfer, 56*(4), 1117-1126. https://doi.org/10.1007/s00231-019-02786-9

Chen, Y., Debnath, T., Cai, A., & Song, M. (2023). Circular silhouette and a fast algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(11), 1-13. https://doi.org/10.1109/TPAMI.2023.3310495

Cheng, X., Li, X., & Ma, X. (2023). A method for battery fault diagnosis and early warning combining isolated forest algorithm and sliding window. *Energy Science & Engineering, 11*, 4493-4504. https://doi.org/10.1002/ese3.1456

Chinna Rao, B., Raju, K., Ramesh Babu, G., & Pittala, C. S. (2023). An improved Gabor wavelet transform and rough k-means clustering algorithm for MRI brain tumor image segmentation. *Multimedia Tools and Applications, 82*, 28143-28160. https://doi.org/10.1007/s11042-023-14485-z

Choudhry, N., Abawajy, J., Huda, S., & Rao, I. (2023). A comprehensive survey of machine learning methods for surveillance video anomaly detection. *IEEE Access, 11*, 114680-114687. https://doi.org/10.1109/ACCESS.2023.3241234

De Donato, L., Marrone, S., Flammini, F., Sansone, C., Vittorini, V., Nardone, R., Mazzariello, C., & Bernaudin, F. (2023). Intelligent detection of warning bells at level crossings through deep transfer learning for smarter railway maintenance. *Engineering Applications of Artificial Intelligence, 118*, 106405. https://doi.org/10.1016/j.engappai.2023.106405

Gao, W., Feng, R., & Sheng, X. (2024). Lightweight multi-stage temporal inference network for video crowd counting. *Frontiers in Physics, 12*, 1489245. https://doi.org/10.3389/fphy.2024.1489245

Guo, J., Lu, S., Jia, L., Zhang, W., & Li, H. (2024). Encoder–decoder contrast for unsupervised anomaly detection in medical images. *IEEE Transactions on Medical Imaging, 43*, 1767-1778. https://doi.org/10.1109/TMI.2023.3327720

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2961-2969. https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.https://doi.org/10.1109/CVPR.2016.90

Hu, W., Hu, T., Wei, Y., Lou, J., & Wang, S. (2023). Global plus local jointly regularised support vector data description for novelty detection. *IEEE Transactions on Neural Networks and Learning Systems, 34*, 3756-3769. https://doi.org/10.1109/TNNLS.2021.3129321

Jiang, R., Yang, Z., & Zhao, J. (2023). A complete deep support vector data description for one-class learning. *IEEE Access, 11*, 114688-114694. https://doi.org/10.1109/ACCESS.2023.3325734

Kaur, N., Rani, S., & Kaur, S. (2024). Real-time video surveillance-based human fall detection system using hybrid Haar cascade classifier. *Multimedia Tools and Applications, 83*(1), 3-20. https://doi.org/10.1007/s11042-024-18305-w

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2117-2125. https://doi.org/10.1109/CVPR.2017.106

Liu, L., Delnevo, G., & Mirri, S. (2023). Unsupervised hyperspectral image segmentation of films: A hierarchical clustering-based approach. *Journal of Big Data, 10*, 31. https://doi.org/10.1186/s40537-023-00713-8

Liu, N. (2024). CCTV cameras at home: Temporality experience of surveillance technology in family life. *New Media & Society*. Advance online publication. https://doi.org/10.1177/14614448241229175

Lu, H., Xu, H., Wang, Q., Gao, Q., Yang, M., & Gao, X. (2024). Efficient multi-view k-means for image clustering. *IEEE Transactions on Image Processing, 33*, 273-284. https://doi.org/10.1109/TIP.2023.3340609

Mazarbhuiya, F. A., & Shenify, M. (2023). A mixed clustering approach for real-time anomaly detection. *Applied Sciences, 13*(7), 4151. https://doi.org/10.3390/app13074151

Nawaz, A., Khan, S. S., & Ahmad, A. (2024). Ensemble of autoencoders for anomaly detection in biomedical data: A narrative review. *IEEE Access, 12*, 3360691. https://doi.org/10.1109/ACCESS.2024.3360691

Ouardirhi, Z., Mahmoudi, S. A., & Zbakh, M. (2024). Enhancing object detection in smart video surveillance: A survey of occlusion handling approaches. *Electronics, 13*(3), 541. https://doi.org/10.3390/electronics13030541

Sengönül, E., Samet, R., Abu Al-Haija, Q., Alqahtani, A., Alturki, B., & Alsulami, A. A. (2023). An analysis of artificial intelligence techniques in surveillance video anomaly detection: A comprehensive survey. *Applied Sciences, 13*, 4956. https://doi.org/10.3390/app13084956

Thiyagarajan, S. K., & Murugan, K. (2023). Arithmetic optimisation-based k-means algorithm for segmentation of ischemic stroke lesion. *Soft Computing*. Advance online publication. https://doi.org/10.1007/s00500-023-08225-6

Ullah, W., Hussain, T., Ullah, F. U. M., Lee, M. Y., & Baik, S. W. (2023). TransCNN: Hybrid CNN and transformer mechanism for surveillance anomaly detection. *Engineering Applications of Artificial Intelligence, 123*, 106173. https://doi.org/10.1016/j.engappai.2023.106173

Wu, Z., Paoletti, M. E., Su, H., & Tao, X. (2023). Background-guided deformable convolutional autoencoder for hyperspectral anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing, 61*, 3341-3354. https://doi.org/10.1109/TGRS.2023.3334562

Yasin, A., Tahir, S. B., & Frnda, J. (2023). Anomaly prediction over human crowded scenes via associate-based data mining and k-ary tree hashing. *International Journal of Intelligent Systems, 2023*, Article ID 9822428. https://doi.org/10.1155/2023/9822428