RESEARCH ARTICLE

DOI: 10.54392/irjmt2436

Asian Research Association

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

A Novel Approach for Surveillance Compression using Neural Network Technique

Nikita Mohod [a, b], Prateek Agrawal [a, c, *], Vishu Madan [a]

[a] School of Computer Science and Engineering, Lovely Professional University, Punjab-144411, India

[b] Department of Computer Science and Engineering, Sipna College of Engineering and Technology, Amravati, Maharashtra-444701, India

[c] Faculty of Engineering and Technology, SGT University, Gurugram, Haryana-122505, India

* Corresponding Author Email: dr.agrawal.prateek@gmail.com

DOI: https://doi.org/10.54392/irjmt2436

Received: 23-11-2023; Revised: 30-03-2024; Accepted: 08-04-2024; Published: 23-04-2024

Abstract: The integration of closed-circuit television (CCTV) monitoring is crucial in the field of video processing, which provides an efficient method for comprehensive surveillance. However, a key challenge associated with this practice is its substantial demand for storage space. Typically, surveillance footage is stored in hard disk drives, and due to limited storage spaces, it is deleted after some time. To address this issue, an innovative method for compressing CCTV video, named object detection-based surveillance compression (ODSC), is introduced. Our ODSC model is divided into two steps: -i) depending upon the objects in the video, determine the significant and non-significant frames of surveillance video using the neural network approach YOLOv5s & YOLOv7-tiny and Yolov8s ii) construct the video of significant frames. Following a comprehensive analysis of the experimental outcomes, it is noted that YOLOv8s stands out with a remarkable detection accuracy of 99.7% on the COCO dataset. Our ODSC approach is reducing the storage space greatly and achieving an average compression ratio of up to 96.31% using YOLOv8s, which surpasses the existing state-of-the-art methods.

Keywords: Significant Frames, Non-Significant Frames, Object Detection, YOLOv5, YOLOv7, YOLOv8 and Surveillance Compression

1. Introduction

CCTV holds paramount importance in modern security and surveillance. Acting as vigilant electronic eyes, CCTV systems provide continuous monitoring in various settings, from public spaces to private properties [1]. Their presence deters criminal activities, aids in crime detection, and enhances overall safety [2]. In businesses, institutions, and public areas, CCTV serves as a crucial deterrent against theft and misconduct. The ability to capture real-time footage and the potential for retrospective analysis make CCTV an invaluable tool for ensuring the security of people and assets [3]. In an era where safety is a top priority, CCTV plays an indispensable role in bolstering surveillance measures [4]. With this increasing number of CCTV cameras and devices, the volume of data generated through CCTV cameras continues to grow, and the need for efficient storage and transmission becomes paramount. Due to the limited space, saving CCTV footage often proves challenging, leading to the deletion of surveillance footage at set intervals. However, this deletion results in the loss of significant content. To preserve this valuable data for longer durations, the implementation of efficient surveillance compression techniques becomes imperative [5].

Video compression is a technique used to reduce the size of video files without significantly compromising the quality of the content. It is crucial for efficient storage, transmission, and streaming of videos. Compression is achieved by eliminating redundant or unnecessary information in the video, resulting in a more compact representation of the data. There are two main types of video compression: lossy compression and lossless compression [6]. In lossless compression, the size of the video file is reduced without any loss of video quality. It ensures that the original video data can be perfectly reconstructed from the compressed version. Lossless compression relies on encoding techniques that focus on eliminating redundant data and minimizing data duplication without discarding any information [7]. While, lossy compression aims to significantly reduce the file size by selectively discarding data that is considered less essential, thus accepting some degradation in video quality. The trade-off between quality and compression ratio is a key aspect of lossy compression. Lossy compression employs a range of techniques to achieve compression, including

quantization, subsampling, filtering, intra-prediction, inter-prediction, and the removal of perceptually less important details. H.265 [8] or High-Efficiency Video Coding (HEVC), is a popular lossy compression technique that works on the principles of quantization, motion compensation, estimation, transform coding, and entropy coding.

Video compression is accomplished through various techniques that exploit redundancies in video data. One fundamental method involves reducing spatial redundancy, where similarities between neighboring pixels are identified and encoded more efficiently, optimizing storage [9]. Temporal redundancy reduction is achieved by storing only the differences between successive frames, exploiting the consistency in motion between frames. Transform coding, such as the Discrete Cosine Transform, converts video signals from spatial to frequency domains, enabling the quantization of transformed coefficients and the removal of less critical information. Entropy coding further reduces the bit rate by assigning shorter codes to frequently occurring patterns. These methods collectively ensure that video data is represented in a more compact form without compromising essential visual information. Additionally, the development of deep learning (DL) models, particularly autoencoders, has introduced innovative approaches to compression. These models, with encoder and decoder networks, learn spatial and temporal features automatically, allowing for more adaptive and efficient compression [10]. The integration of deep learning in video compression holds promise for achieving higher levels of efficiency, adaptability, and perceptual quality in the representation of video content.

1.1 Motivation and Approach

In this research work, the compression of surveillance video using the concept of content-aware compression is performed. The Automated Teller Machine (ATM) video footage is a pertinent case study for our research investigation. The CCTV cameras installed within ATM rooms continuously capture video content, forming a continuous, round-the-clock surveillance feed. However, the substantial periods of activity, primarily related to ATM transactions, make up only a fraction of the total recording duration. The rest of the footage predominantly captures the static state of the ATM itself. This situation leads to the inefficient use of storage resources, as a significant portion of the recorded content lacks relevance or serves a non-significant purpose in the surveillance context. To tackle this issue, this article presents deep learning-based object detection methods to identify frames where human presence is detected. Subsequently, we separate the identified significant segments, representing human interactions at the ATM, from the non-significant segments portraying the static ATM environment. Frames in the surveillance video where a person is present are recognized as significant frames, while frames with the firmed ATM are considered non-significant parts. Figure 1 denotes the sample of significant frames and non-significant frames. Later on, the compressed video is constructed using significant frames. This process greatly reduces storage space requirements by eliminating the storage of redundant or inconsequential video content. The result is enhanced efficiency in the management of surveillance video, intentionally retaining pertinent, significant portions for extended durations, while selectively deleting non-significant segments. The major contribution of our research work is summarized as follows: -

1) Gathering surveillance footage from nationalized bank ATMs and dividing the video into a short interval from 10 minutes (Min) To 40 Min.
2) Splits the video into frames using the OpenCV library.
3) Employing a neural network approach to distinguish between significant and non-significant frames in the surveillance video and determine the best network depending on evaluation metrics.
4) Used the significant and non-significant frames of the best network. Discard the non-significant frames and reconstruct the video using the identified significant frames.

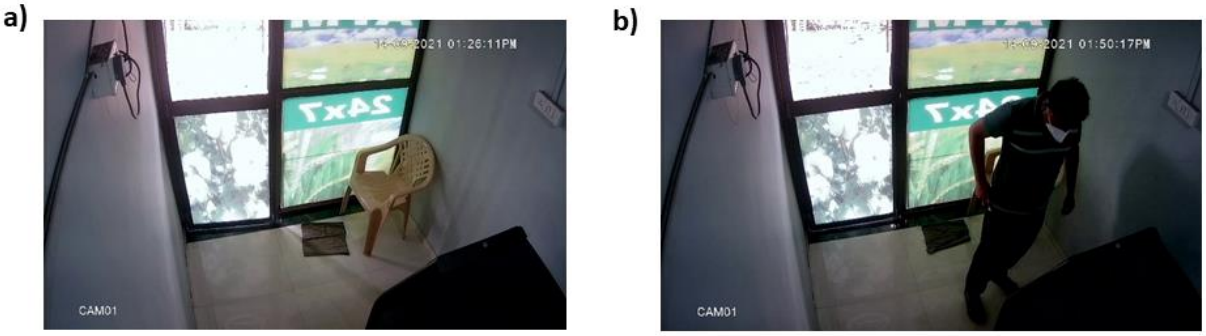

Figure 1. Samples frames of ATM surveillance (a) non-significant frame (b) Significant frame

The subsequent structure of the paper is as follows: Section 2 presents a thorough review of relevant literature in the realms of object detection and video compression, offering valuable insights into the current research landscape within these domains.

Section 3 elaborates on our proposed ODSC approach, delineated into two modules: i) the object detection module and ii) the video compression module. The ensuing Section 4 is dedicated to presenting and delving into a comprehensive analysis of the resuls obtained. Lastly, Section 5 synthesizes our findings, culminating in conclusive insights drawn from our research.

2. Literature Survey

This section deals with the popular deep learning (DL) based object detection framework and video compression approaches.

2.1. DL-based object detection

The field of computer vision has seen notable progress in object detection (OD), a pivotal task, especially with the advent of deep learning techniques. OD aims to identify the objects and their location within images or video frames. Unlike image classification, which determines the presence of a single object category in an entire image, OD detects multiple objects of varying classes and accurately delineates their boundaries with bounding boxes [11]. Traditional approaches to OD relied on handcrafted features and machine learning algorithms like Support Vector Machines or Haar cascades. These methods often required manual feature engineering and struggled with complex scenarios [12].

The advent of deep learning revolutionized object detection, notably with the introduction of Convolutional Neural Networks (CNN) [11]. Neural network-based OD is divided into two approaches (i) region-based (two-stage) and (ii) region-free (one-stage) [12]. In 2014 Girshick *et al.* introduced region-based OD named RCNN which is pioneering of two-stage OD framework [13]. To recognize objects, the RCNN framework is divided into two parts. In the first part, it proposes regions of interest (RoIs) within an image using selective search, and in the second part, RoIs are individually processed through a CNN to extract features, followed by classification and bounding box regression to detect objects. R-CNN achieves high accuracy but is computationally intensive. Fast R-CNN mitigated the computational inefficiencies and intricacies of R-CNN wh ile notably enhancing object detection accuracy. In contrast to the R-CNN approach of employing distinct CNN for individual proposed regions, Fast R-CNN integrates feature extraction across all regions while retaining the use of selective search for region proposals [14].

Though Fast RCNN overcomes the limitation of RCNN, still the network is slow due to the use of a selective search approach for region proposals. To address this, Ren *et. al.* present Faster-RCNN which introduces a Region Proposal Network (RPN) that shares convolutional layers with the subsequent detection network, enabling the model to generate region proposals and perform object detection simultaneously [15]. Faster RCNN uses the anchor box method, to detect and locate objects at various scales with different aspect ratios which improves the detection accuracy of the network. He *et al.* developed spatial pyramid pooling (SPP) which addresses the challenge of variable input sizes in CNN by enabling them to accept images of different dimensions. This layer divides the input feature map into a grid and applies pooling operations independently within each grid cell. It then concatenates the pooled features from all grid cells into a fixed-length vector, which can be fed into subsequent layers for classification or detection tasks [16]. To handle objects at multiple scales, Lin *et. al.* introduced a feature pyramid network (FPN) in OD [17]. FPN employs a top-down architecture, where it fuses features from different layers of a CNN. This fusion process generates a hierarchical pyramid of features, with higher-level features containing semantic information and lower-level features providing finer details.

In 2017, He *et al.* presented Mask-RCNN which is an extension of Faster R-CNN. It integrates an additional branch for pixel-level segmentation masks alongside OD [18]. Mask-RCNN employs a ResNet-50/101 as the backbone network for feature extraction, RPN for candidate object proposals, and a mask head for predicting segmentation masks, to enable precise instance segmentation. This region-based OD approaches excel in complex scenarios but suffers from computational complexity and slower inference speeds compared to region-free techniques [12].

The demand for real-time OD led to the development of single-shot detectors [19]. You Only Look Once (YOLO) and Single Shot Multi-Box Detector (SSD) are prominent real-time detectors. Redmon et. al presents You Only Look Once (YOLO) [20]. YOLO divides the image into a grid and predicts bounding boxes and class probabilities which makes it faster and more suitable for real-time applications [20]. Liu developed a single-shot detector (SSD) to perform OD in a single forward pass through a CNN. It utilizes a set of default bounding boxes or "priors" of various sizes and aspect ratios, referred as anchor boxes [21]. These anchor boxes are pre-defined to cover a range of object sizes and shapes [19]. In 2017, Redmon *et. al.* introduced YOLOv2, to detect the smaller objects in an image [22]. Yolov2 incorporates batch normalization, anchor boxes, and a finer grid for better localization accuracy. YOLOv3 is the third iteration of the YOLO model and introduced further improvements in terms of accuracy and detection speed [23]. It uses FPN, to

improve its ability to detect objects across different scales. YOLOv3 is generally considered to be more accurate than YOLOv2, but it requires more computational resources [24]. To determine objects in surveillance video YOLO version exhibits better accuracy [25, 26]. To enhance the accuracy and speed of the detection model in real-time YOLOv4 marks success. It incorporates advanced features such as the CSPDarknet53 backbone and PANet, enhancing the model's capability to detect objects with high precision across various scales. Overall, YOLO has evolved from its initial concept to become one of the most popular and influential OD frameworks, continually pushing the boundaries of speed, accuracy, and efficiency in real-time object detection tasks. Hence, in our research paper variants of YOLO are used to differentiate significant and non-significant frames of ATM surveillance video.

2.2. DL-based video compression

DL-based video compression represents a burgeoning field of study poised to enhance compression efficiency and quality. Traditional methods like H.264 and H.265 have constraints, prompting exploration into deep learning alternatives [27-28]. Traditional methods consist of inter-prediction, intra-prediction, sampling, coding optimization, and filtering techniques [28]. To improve the traditional techniques, researchers apply CNN, Recurrent Neural Networks (RNN), and autoencoders, including Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) to video compression frameworks and bolster compression efficiency.

In H.265, despite featuring 35 intra-prediction modes, there's a limitation in capturing the deeper spatial information between the current and its neighboring blocks. To address this issue Cui et. al. integrates CNN with H.265 [29]. Inter-frame prediction estimates the motion compensation (MC) and motion estimation (ME) between two consecutive video frames to eliminate duplication towards the temporal axis and also determines video compression efficiency. Lin et al. integrate GAN with MC and ME to predict the next frame using a previously compressed frame [30]. To obtain final coded bits traditional methods used discrete cosine or sine cosine transform in H.265 and H.264 due to which the quality of compressed video reduces. To enhance this, Zhang *et al.* introduced implicit dual domain CNN (IDCNN) to reduce the artifacts in compressed video [31]. To improve the rate distortion of video, Bouurtsoulatze *et. al.* present deep pre-coding for video distribution. Here, a multi-scale pre-coding CNN downscales high-resolution frames over many scale factors and is trained to reduce post-processing and blurring artifacts caused by typical linear up-scaling filters. Before encoding, an adaptive pre-coding mode selection algorithm is proposed, which adaptively determines the best resolution. In this way, the neural network is combined with traditional approaches to achieve better compression.

While few researchers have tried to perform compression in an end-to-end manner. Lu et. al. used an optical flow module to obtain the motion information between the current frame and the previous compressed frames for each frame to be compressed [32]. A trained network performs MC to produce a prediction signal for the current frame. Two auto-encoders are used to compress. the prediction residues and motion information, respectively.

2.3. DL-based Surveillance video compression

The rapid use of CCTV cameras over recent decades, emphasizes its widespread adoption for surveillance purposes. As surveillance cameras have become essential for safety and security, there has been an expansion in video parameter space, including higher spatial resolutions, frame rates, and dynamic ranges. Consequently, there's a considerable increase in the storage space required to accommodate this higher-quality footage. Hence, a domain-specific compression technique to store significant data while minimizing storage space usage is required.

Zonglei *et. al.* introduces a deep compression technique that employs object detection to differentiate moving and stationary objects in apron surveillance videos. Extracted object images, their positions, and background images are stored in a linked list. During decompression, objects are restored to the background, maintaining original video details, and significantly reducing storage space [33]. Wu *et. al.* presents a method to compress foreground and background parts of surveillance

An adaptive background updating and interpolation module improves the compression ratio by sharing background information among adjacent frames to achieve foreground compression, a motion-based approach with residual encoding is used [34]. Ghamsarian *et. al.* introduced a novel approach for cataract surgery video compression, by detecting and compressing relevant frames containing instrument actions separately from irrelevant frames using CNN [35]. Matha *et. al.* proposed a relevance-based compression scheme, informed by clinician input, which achieves up to 95.94% storage reduction witH.264/AVC and up to 98.82% with AV1, excluding idle phases [36].

3. ODSC Model

This section outlines the ODSC model which is divided into two steps: - 1) the object detection module and ii) the compression module. Figure 2 represents the proposed ODSC model. The use case for the ODSC model is ATM surveillance video, so the surveillance

video is divided into the frames using OpenCV library, and frames are given as input to the OD module. The working details of the modules are explained in the following subsections.

3.1 Object Detection Module

To predict the significant and non-significant frames of ATM surveillance video, we used YOLOv5s, YOLOv7-tiny, and YOLOv8s from a one-stage detector which are summarized as follows.

3.1.1 YOLOv5s

In 2020, Ultralytics presented YOLOv5s, an object detection algorithm in several variants, which includes YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, differ in terms of model size and computational requirements. For the ODSC model, YOLOv5s is used. It has CSPDarknet53 architecture as the backbone network which extracts features from the input image. It Uses FPN to capture features at different scales which helps in detecting objects of varying sizes within an image. Neck architecture fuses features from various network levels, improving its ability to detect objects at multiple scales. The detection head comprises three detection scales, each predicting bounding boxes, object probabilities, and class probabilities. This architecture enables YOLOv5s to efficiently process entire images for rapid and accurate object detection. Figure 3 denotes the architecture of the YOLOv5s framework.

In YOLOv5, the loss function consists of three main components: localization loss, objectness loss, and classification loss. Localization loss used smooth L1 loss function denoted in equation 1, where x is the difference between the predicted value and the target value.

$$L_{Loc}(x) = \begin{cases} 0.5x^2, & x < 5 \\ |x| - 0.5, & otherwise \end{cases} \quad (1)$$

Objectness loss and classification loss used binary cross-entropy (BCE), where p_i predicted objectness score, t_{iis} the true objectness score and N_{obj} is the total number of anchor boxes shown in equation 2.

$$L_{Obj}/L_{Cls} = -\frac{1}{N_{obj}}\sum_{i=0}^{N_{obj}}[t_i \log p_i + (1 - t_i)\log(1 - p_i)] \quad (2)$$

The overall loss is determined using equation 3, which is a weighted sum of these individual losses, with different weights assigned to maintain a balance during training.

$$L = L_{Loc} + L_{Obj} + L_{Cls} \quad (3)$$

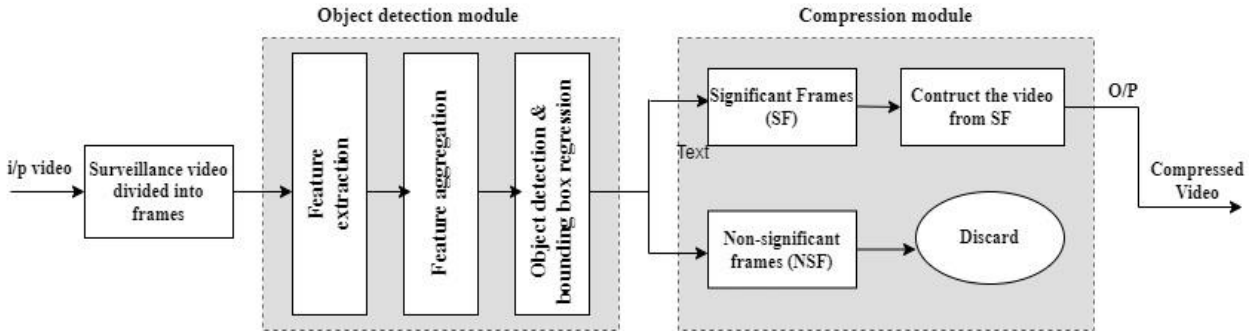

Figure 2. ODSC model overview

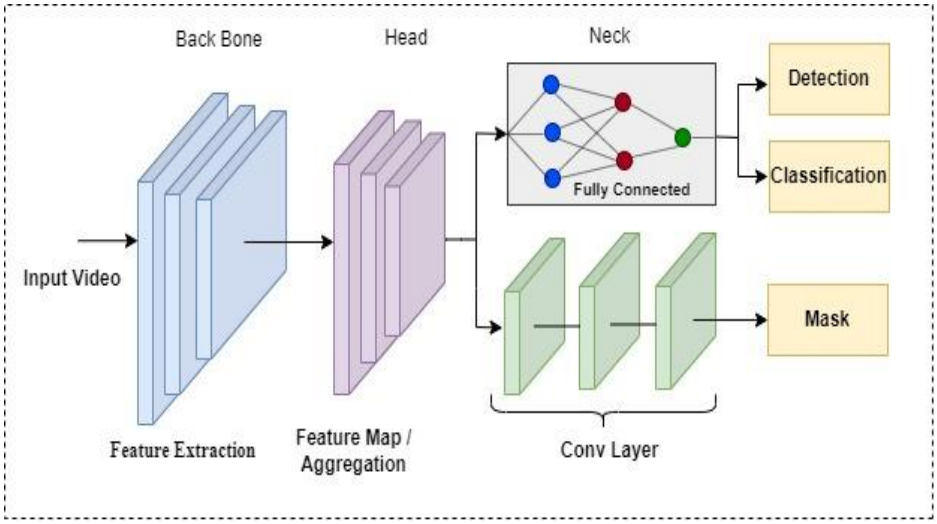

Figure 3. YOLOv5s framework

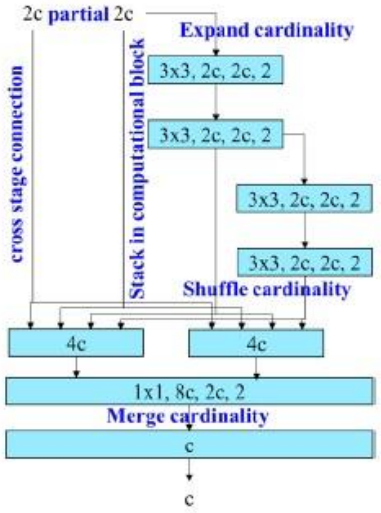

Figure 4. YOLOv7 framework

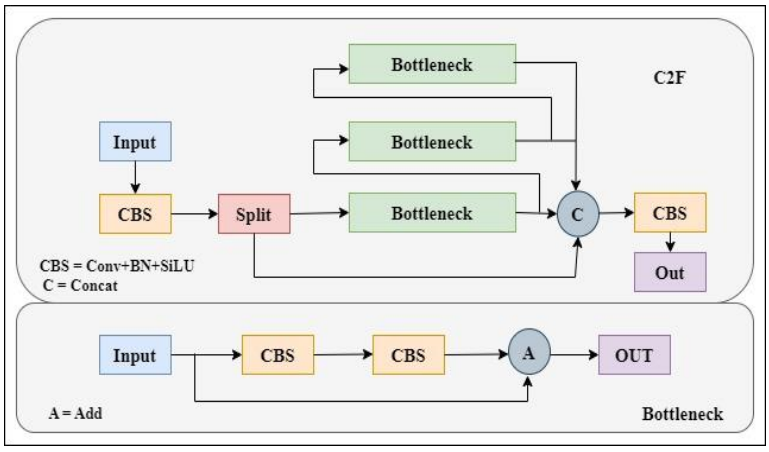

Figure 5. YOLOv8s framework

3.1.2 YOLOv7

In 2023 Wang et al. introduced YOLOv7, a popular and efficient OD algorithm that detect and classify objects in images and videos [37].

Researchers developed fundamental models tailored for different GPU settings, categorizing them as YOLOv7-tiny for edge GPUs, YOLOv7 for regular GPUs, and YOLOv7-W6 for cloud GPUs. YOLOv7 uses a modified YOLOv3 architecture as its backbone, but it incorporates several architectural improvements to enhance detection accuracy and speed. The architecture comprises several key elements, including compound scaling, the extended efficient layer aggregation network (EELAN), a collection of useful enhancements through planned and reparameterized convolution, coarseness for auxiliary loss, and fineness for primary loss. The trainable bag-of-freebies approach improves detection accuracy without incurring additional inference costs. It also focuses on the re-parameterization module which replaces the original module and the dynamic label assignment strategy which manages the assignment of labels to different output layers is a key aspect in improving object detection techniques. This approach performs stack scaling to the neck component and utilizes the proposed compound scaling method to increase the depth and width of the entire model, resulting in YOLOv7-X. Figure 4 represents the architecture of the YOLOv7 model.

3.1.3 YOLOv8

In 2023, Ultralytics again presented YOLOv8 a region-free OD framework that consists of several key components: an input segment, a backbone, a neck, and an output segment [38]. The input segment is responsible for pre-processing the input image. It applies mosaic data augmentation, adaptive anchor calculation, and adaptive grayscale padding to enhance the input data. The backbone network and neck module form the core structures of YOLOv8. The backbone network processes the input image using Conv and C2f modules to extract feature maps at different scales. It incorporates the ELAN structure from YOLOv7 [37], reducing one standard convolutional layer and enhancing gradient flow through the Bottleneck module. This approach maintains lightweight characteristics while capturing more gradient flow information. The output feature maps

are processed by the SPPF module, which employs pooling with varying kernel sizes to combine feature maps effectively. Finally, the results are passed to the neck layer for further processing. The neck layer of YOLOv8 is designed to improve the model's feature fusion capability by incorporating the FPN along with the Path Aggregation Network (PAN) structure like YOLOv5. This combination enables better integration of features from different scales and enhances the model's ability to capture contextual information across the entire image. The detection head of YOLOv8 adheres to the standard approach of dividing the classification head from the detection head. This process includes performing loss computation and filtering out target detection boxes. The loss calculation consists of two main parts: classification and regression, with the Objectness branch excluded. For classification, BCE loss is utilized, while Distribution Focal Loss (DFL) is employed for regression. Figure 5 represents the YOLOv8s framework.

3.2 Video Compression Module

In the compression module of the ODSC framework, the output of the OD module is used to perform compression.

Compression is achieved by selectively preserving the frames deemed significant in surveillance videos and discarding those considered non-significant. Here, ATM surveillance is a use case; where the significant frames encompass the duration when an individual enters, performs transactions, and exits the ATM room, while frames depicting a static, unoccupied ATM are considered nonessential and thus excluded to facilitate compression. This determination is based on the observation that, within a 24-hour cycle, only a few hours involve activity, while the majority of videos contain firmed ATM. To address this inefficiency, the ODSC framework eliminates non-significant segments from the surveillance video, allowing for the retention of important frames over extended periods while significant frames are utilized to construct the compressed video

3.3 Algorithm and Sequence flow diagram

The sequence flow diagram (SFD) of the ODSC model is shown in Figure 6. Here, the framework is divided into the following steps: -

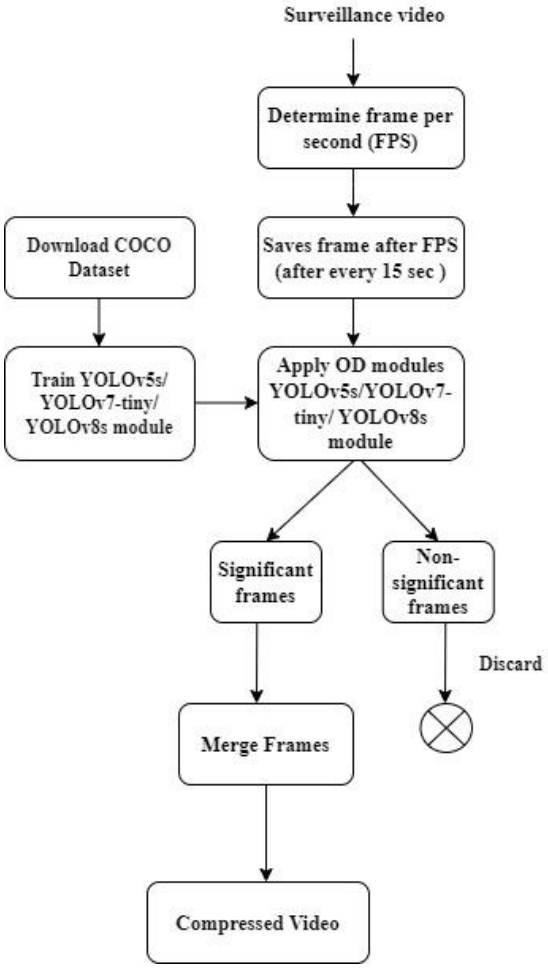

Figure 6. SFD of ODSC model

Algorithm 1: ODSC algorithm

	Input: ATM surveillance footage
	Dataset: COCO
	Output: Compressed surveillance video
1	ODSC_FPS←Get FPS (Input ATM video)
2	ODSC_Split← Split video into the frames
3	ODSC_Save← Save frames after every ODSC_FPS
4	do until all frames
	ODSC_OD← Apply YOLOv5s, YOLOv7-tiny and YOLOv8 framework
5	ODSC_Significant←Predict frames
6	ODSC_non-significant←Predict frames
	end

Table 1. Characteristics of Tested Video

Clip Name	FPS	Size	Interval	Resolution
Clip1.mp4	15	527 MB	30 Minutes (Min)	1280 × 720
Clip2.mp4	15	451 MB	20 Min	1280 × 720
Clip3.mp4	15	700 MB	40 Min	1280 × 720
Clip4.mp4	15	250 MB	10 Min	1280 × 720

1 In the first step, the frame per second (FPS) value of the input surveillance video is calculated. Later, depending on the value of FPS frames of surveillance is saved.

2 The ODSC model used YOLOv5s, YOLOv7-tiny, and YOLOv8 modules, to detect the objects in the surveillance video.

3 Depending on the object recognized in surveillance video frames are split into: - significant frames and non-significant frames.

In the last step non-significant frames are discarded and significant frames are stored to construct the compressed surveillance video.

In this algorithm, the FPS of surveillance video is calculated using the Open CV library and saved in variable ODSC_FPS. Then the video is divided into frames and saved in ODSC_Split. Further frames are saved using the value of FPS (if FPS = 15 then the model saved 1st, 16th, 32th.....frames) in variable ODSC_Save. Then we apply the three D modules to identify significant and non-significant frames of surveillance video and significant frames save in variable ODSC_Significant and non-significant into ODSC_Non-significant variable. At last, the video is constructed using ODSC_Significant and saved in varable ODC Compressed, and non-significant frames are deleted from the network.

4. Results and Analysis

4.1. Datasets

For our research work, we used two datasets: - (i) The Common objects in context (COCO) dataset for training purposes and (ii) The ATM surveillance dataset for testing. COCO is a large-scale dataset with thousands of images and 80 object categories which is used for object detection, segmentation, and captioning tasks in computer vision [39]. YOLOv5s, YOLOv7 and YOLOv8s modules are trained on the COCO dataset to identify significant frames and non-significant frames of ATM surveillance videos.

As the COCO dataset contains 80 classes, to train our OD modules only 3 classes which include person, cat, and dog were used. If the frames contain any of these objects, then OD modules recognize that frame as the significant frame otherwise it is the non-significant frame. Here a threshold value is set to 0.5 means that only detections with confidence scores greater than or equal to 0.5 are considered significant and retained. ATM footage is given as input to trained YOLO modules to differentiate significant frames (SF) and non-significant frames (NSF) of video. Table 1 represents the characteristics (frame per second (FPS), size duration, and resolution of frames) of four different tested ATM surveillance Videos

4.2. Evaluation of OD Modules

To implement YOLOv5s, YOLOv7-tiny, and YOLOv8s, Google Collab Pro is used as these models are computationally intensive and require GPU resources. The parameter specification required for the implementation of YOLO modules is mentioned in Table 2. YOLOv5s uses a backbone network called "CSP-DarkNET," which is a modified version of the Darknet architecture. It includes features like Cross-Stage Partial connections for improved performance and efficiency.

For training, YOLOv5s has a learning rate of 0.001, over 100 epochs. A batch size of 16 is often employed during training. YOLOv5s employs a multi-task loss function that combines objectness loss, class loss, and localization loss shows in equation 3. Yolov5 uses the swish activation function which is shown in Equation 4.

$$f(x) = x.Sigmoid(\beta.x) \quad (4)$$

Where x is input to function, β is hyperparameter, and sigmoid is another activation function represented in equation 5 and also used in YOLOv8. while YOLOv7 uses the LeakyRelu function shown in Equation 6.

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \quad (5)$$

$$Leaky\ ReLU(x) = \begin{cases} x, & x > 0 \\ \alpha x, & otherwise \end{cases} \quad (6)$$

YOLOv7 employs EELAN as its backbone network to enhance the performance of OD. It has a lower learning rate of 0.0002, which helps fine-tune the model with more precision during training with 100 epochs and 16 batch sizes. The specific loss functions in YOLOv7 may include Localization Loss, Objectness Loss, Classification Loss, Coarseness Loss, and Fineness Loss. For YOLOv8 C2F is used as the backbone with 0.01 learning rate having 16 batch size and 100 epochs.

We tested the performance of YOLOv5s, YOLOv7-tiny, and YOLOv8s on the ATM surveillance dataset and received the result mentioned in Table 3. The YOLOv8s module exhibits a high level of precision of 98.7% followed by YOLOv7 and Yolov5 meaning that a significant portion of the positive predictions it makes is accurate. The recall of YOLOv7 and YOLOv8 stands at 98.2%, indicating its effectiveness in capturing most of the actual positive instances followed by YOLOv5 at 97.9 %. The F1 score, which balances precision and recall, is 98.1% for YOLOv7 and YOLOv8, showing a harmonious trade-off between accuracy and completeness. In terms of overall correctness, YOLOv8s boasts an accuracy of 99.7%, implying that the majority of its predictions, both positive and negative, are correct. On the other hand, YOLOv7-tiny exhibits 99.5 % and 98.3 % for YOLOv5. YOLOv7 and YOLOv8 models give similar promising performance but the accuracy of YOLOv8 is 0.2% greater than YOLOv7. Though both models exhibit better accuracy, the time taken to detect significant frames and non-significant frames using YOLOv8 is far better as compared to YOLOv7. For further processing, the output of the YOLOv8 module is used. Figure 7 represent the output of the OD module.

Table 4 denotes the output of YOLOv7 on four ATM surveillance video clips where the first clips contain 27000 frames and the FPS of all frame clips are 15, so only 1800 (27000/15) frames are given as input to the OD module. The YOLOv7-tiny recognized 539 frames as SF and 1261 as NSF. The second clip contains 18000 total frames where 1200 (18000/15) frames are considered as preferred frames. Out of which 355 frames are detected as SF and 845 NSF. The same logic is applied for the rest of the clips and SF and NSF are separated from the preferred frames using the YOLOv7-tiny OD module.

4.3 Compression Module

In the compression module, we utilized the output of YOLO8s to merge all SF at the original FPS value. Table 5 presents an analysis of the compression module. For instance, the size of clip1.mp4 is 527 MB and of interval 30 Min.

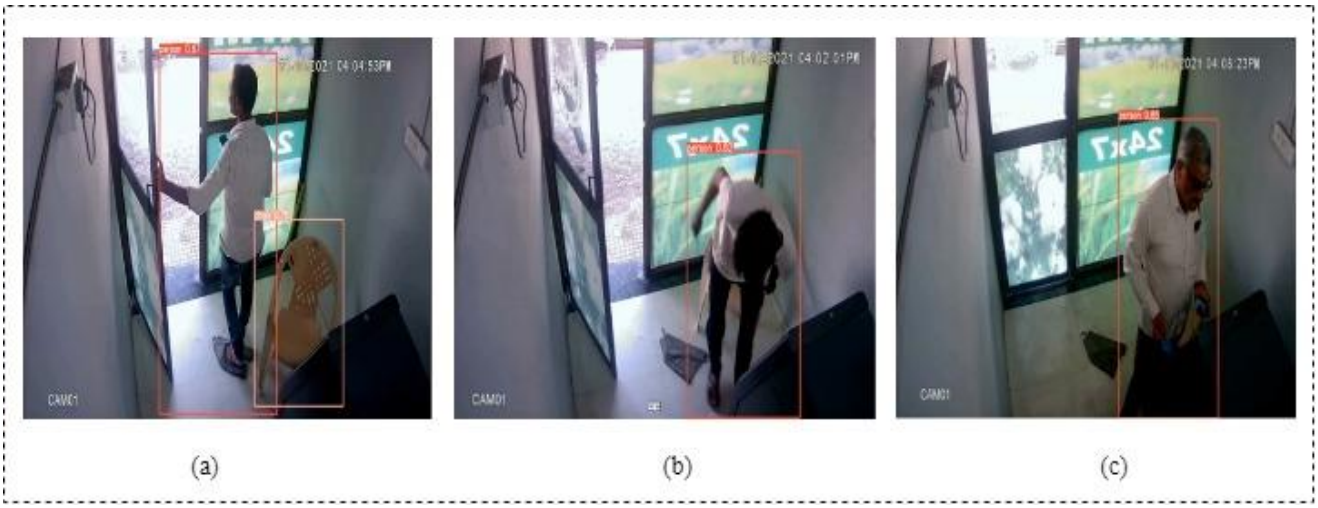

Figure 7. The output of YOLO module (a) YOLOv5s (b) YOLOv7-tiny (c) YOLOv8s

Table 2. Parameter Specification

Parameter	YOLOv5	YOLOv7	YOLOv8
Backbone Network	CSP-DarkNET	EELAN	C2F
Learning Rate	0.01	0.02	
0.01			
Epoch	100	100	100
Batch Size	16	16	16
Loss Function	Multi-task loss	Multi-task loss	Multi-task loss
GPU	Yes	Yes	Yes
Activation function	Swish	Leaky ReLU	Sigmoid

Table 3. Comparison of Od Modules

Parameter (%)	Yolov5	YOLOv7	YOLOv8
Precision	97.6	98.6	98.7
Recall	97.9	98.2	98.2
F1 Score	98.1	98.3	98.3
Accuracy	98.3	99.5	99.7

Table 4. Count of Significant and Non-Significant Frames using The Yolov8s Module

Clip Name	Total frames	Preferred frames	SF	NSF
Clip1.mp4	27000	1800	539	1261
Clip2.mp4	18000	1200	355	845
Clip3.mp4	36000	2400	1420	980
Clip4.mp4	9000	600	437	163

Table 5. The output of ODSC model with resolution 1280 × 720 on YOLOv8s

	Original clip		Compressed clip		
Name	Size	Interval	Size	Interval	% of Compression Achieved
Clip1.mp4	527 MB	30 Min	19.4 MB	36 sec	96.31%
Clip2.mp4	351 MB	20 Min	45.2 MB	25 Sec	87.12 %
Clip3.mp4	700 MB	40 Min	48.5 MB	65 Sec	93.07 %
Clip4.mp4	250 MB	20 Min	9.7 MB	30 Sec	96.12 %
Overall Compression	--	--	--	--	93.15 %

Table 6. Comparison with State of the Arts Approaches

Parameter	Arora *et. al.,* [41]	Ghamsarian *et. al.*, [36]	Zonglei *et. al.*, [34]	ODSC framework
DL approach	No	Yes	Yes	Yes
OD approach	No	Yes (Mask-RCNN)	Yes (Faster RCNN)	Yes (YOLOv8s)
Type of Compression	Lossy Compression	Lossy -Compression	Lossy - Compression	Lossy - Compression
Compression achieved	64.32 %	68%	Upto 88 %	96.31%

After passing through the ODSC model the size of the clip is greatly reduced to 19.4 MB of duration 36 sec and achieved 96.3% compression by maintaining the same resolution of frames as that of the original clip. Similarly, the second clip achieved 87.2 % compression, the third clip 93.07%, and the fourth clip 96.12%. Depending upon the count of SF, detected using YOLOv8 the clips are greatly compressed. Achieved the highest compression of 96.31% for clip1 and 87.12% lowest for clip2. The overall compression achieved using the ODSC model is 93.15%. Such compression ratios denote the proportionate decrease in file size, thereby facilitating efficient storage and transmission of video content without loss in quality.

4.4. Comparison with a state-of-the-art approach

The proposed ODSC model is compared with three other approaches which are summarized in Table 6. We implement the approach of Arora *et al.* [40], Ghamsarian *et al*. [35], and Zonglei *et. al.* [33]. Arora *et al.* [41] developed a model to perform compression of surveillance video by removing the redundant frames using the mean squared error concept. This is a lossy compression and when it is implemented on an ATM surveillance dataset it achieves 64.32% compression. Ghamsarian et al. used the Mask-RCNN approach to detect the relevant and irrelevant frames of cataract surgery video from an ophthalmologist's point of view. The relevant frames of video are further compressed using higher quantization parameters. we trained this network on the COCO dataset and tested it on ATM surveillance video which achieved 68% compression. Zonglei *et.al.* presents a compression and decompression network for apron surveillance video. In this approach initially object is detected using the deep learnings Faster RCNN model [33]. Later on, these objects are placed in a linked list with its co-ordinate's information and background. At the time of decompression, the objects with their information are retrieved from the linked list and stored on the background image. This approach is trained on the COCO dataset and tested on ATM surveillance which achieves a maximum of 88% compression. Our ODSC framework detects the SF and NSF using the YOLOv8s approach and achieves the highest 96.31% compression. An experimental analysis states that our model outperforms the existing state-of-the-art approaches and achieves a remarkable compression ratio of 96.31%.

5. Conclusion

We present the novel Object Detection-Based Surveillance Compression (ODSC) model to compress ATM CCTV video while preserving critical surveillance information effectively. The model consists of two crucial steps: (i) utilization of advanced neural network approaches, namely YOLOv5s, YOLOv7-tiny, and YOLOv8s to differentiate between significant and non-significant frames in the surveillance video. Second, it reconstructs the video by retaining only the significant frames. The results of comprehensive experiments indicate that YOLOv8s achieves a remarkable detection accuracy of 99.7% on the COCO dataset, establishing its proficiency. The ODSC approach successfully reduces storage space demands, achieving an impressive average compression ratio of up to 96.31% with YOLOv8s, surpassing existing state-of-the-art methods.

References

[1] A. Hope, CCTV, school surveillance and social control. British Educational Research Journal, 35(6), (2009) 891-907. https://doi.org/10.1080/01411920902834233

[2] E. L. Piza, B. C. Welsh, D. P. Farrington, A. L. Thomas. CCTV surveillance for crime prevention: A 40-year systematic review with meta-analysis. Criminology & public policy, 18(1), (2019) 135-159. https://doi.org/10.1111/1745-9133.12419

[3] M. P. J. Ashby, The Value of CCTV Surveillance Cameras as an Investigative Tool: An Empirical Analysis. European Journal on Criminal Policy and Research, 23(3), (2017) 441-459. https://doi.org/10.1007/s10610-017-9341-6

[4] C. Norris, G. Armstrong, The maximum surveillance society: The rise of CCTV. Routledge, London. https://doi.org/10.4324/9781003136439

[5] L. Zhao, S. Wang, S. Wang, Y. Ye, S. Ma, W. Gao, Enhanced surveillance video compression with dual reference frames generation. IEEE Transactions on Circuits and Systems for Video Technology, 32(3), (2021) 1592-1606. https://doi.org/10.1109/TCSVT.2021.3073114

[6] P. Kavitha, A survey on lossless and lossy data compression methods. International Journal of Computer Science & Engineering Technology, 7(3), (2016) 110-114.

[7] H. Rhee, Y. Il Jang, S. Kim, N. I. Cho, Lossless Image Compression by Joint Prediction of Pixel and Context Using Duplex Neural Networks. IEEE Access, 9, (2021) 86632-86645. https://doi.org/10.1109/ACCESS.2021.3088936

[8] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand. Overview of the high efficiency video coding (HEVC) standard. IEEE Transactions on circuits and systems for video technology, 22(12), (2012) 1649. https://doi.org/10.1109/TCSVT.2012.2221191

[9] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, Learning Image and Video Compression Through Spatial-Temporal Energy Compaction. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (2019) 10071-10080. https://doi.org/10.1109/CVPR.2019.01031

[10] J. Mao, L. Yu, Convolutional neural network based bi-prediction utilizing spatial and temporal information in video coding. IEEE Transactions on Circuits and Systems for Video Technology, 30(7), (2019) 1856-1870. https://doi.org/10.1109/TCSVT.2019.2954853

[11] Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey. Proceedings of the IEEE, 111(3), (2023) 257-276. https://doi.org/10.1109/JPROC.2023.3238524

[12] X. Wu, D. Sahoo, S.C.H. Hoi, Recent advances in deep learning for object detection. Neurocomputing, 396, (2020) 39-64. https://doi.org/10.1016/j.neucom.2020.01.085

[13] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, (2014) 580-587. https://doi.org/10.1109/CVPR.2014.81

[14] R. Girshick, (2015) Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, IEEE, Chile. https://doi.org/10.1109/ICCV.2015.169

[15] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks. in IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), (2017) 1137 - 1149. https://doi.org/10.1109/TPAMI.2016.2577031

[16] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), (2015) 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, Feature pyramid networks for object detection. in Proceedings of the IEEE conference on computer vision and pattern recognition, (2017) 2117-2125. https://doi.org/10.1109/CVPR.2017.106

[18] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, (2017) 2961-2969. https://doi.org/10.1109/ICCV.2017.322

[19] W. Liu et al., SSD: Single shot multibox detector. In Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, (2016) 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[20] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, (2016) 779-788. https://doi.org/10.1109/CVPR.2016.91

[21] Y. Zhong, J. Wang, J.Peng, L. Zhang. Anchor box optimization for object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, (2020) 1286-1294. https://doi.org/10.1109/WACV45572.2020.9093498

[22] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, (2017) 7263-7271. https://doi.org/10.1109/CVPR.2017.690

[23] J. Redmon and A. Farhadi, Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, (2018). https://doi.org/10.48550/arXiv.1804.02767

[24] T. Diwan, G. Anirudh, J. V. Tembhurne, Object detection using YOLO: Challenges, architectural successors, datasets and applications. multimedia Tools and Applications, 82(6), (2023) 9243-9275. https://doi.org/10.1007/s11042-022-13644-y

[25] N. Mohod, P. Agrawal, V. Madaan, YOLOv4 Vs YOLOv5: Object Detection on Surveillance Videos. In International Conference on Advanced Network Technologies and Intelligent Computing, (2022) 654-665. https://doi.org/10.1007/978-3-031-28183-9_46

[26] N. Mohod, P. Agrawal, V. Madan, Human Detection in Surveillance Video using Deep Learning Approach. In 2023 6th International Conference on Information Systems and Computer Networks (ISCON), (2023) 1-6. https://doi.org/10.1109/ISCON57294.2023.10111951

[27] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, Overview of the H. 264/AVC video coding standard. IEEE Transactions on circuits and systems for video technology, 13(7), (2003) 560-576. https://doi.org/10.1109/TCSVT.2003.815165

[28] L. Dong, Y. Li, J. Lin, H. Li, F. Wu, Deep learning-based video coding: A review and a case study.

ACM Computing Surveys (CSUR), 53(1), (2020 1-35. https://doi.org/10.1145/3368405

[29] W. Cui, T. Zhang, S. Zhang, F. Jiang, W. Zuo, D. Zhao, (2017) Convolutional Neural Networks Based Intra Prediction for HEVC. Data Compression Conference Proceedings, USA. https://doi.org/10.1109/DCC.2017.53

[30] J. Lin, D. Liu, H. Li, F. Wu, (2018) Generative adversarial network-based frame extrapolation for video coding. In 2018 IEEE Visual Communications and Image Processing (VCIP), IEEE, Taiwan. https://doi.org/10.1109/VCIP.2018.8698615

[31] Y. Zhang, L. Chen, C. Yan, P. Qin, X. Ji, Q. Dai, weighted convolutional motion-compensated frame rate up-conversion using deep residual network. IEEE Transactions on Circuits and Systems for Video Technology, 30(1), (2018) 11-22. https://doi.org/10.1109/TCSVT.2018.2885564

[32] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, Z. Gao, (2019) Dvc: An end-to-end deep video compression framework. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, USA.https://doi.org/10.1109/CVPR.2019.01126

[33] L. Zonglei, X. Xianhong, Deep compression: A compression technology for apron surveillance video. IEEE Access, 7, (2019) 129966-129974. https://doi.org/10.1109/ACCESS.2019.2940252

[34] L. Wu, K. Huang, H. Shen, L. Gao, Foreground-background parallel compression with residual encoding for surveillance video. IEEE Transactions on Circuits and Systems for Video Technology, 31(7), (2021) 2711-2724. https://doi.org/10.1109/TCSVT.2020.3027741

[35] N. Ghamsarian, H. Amirpourazarian, C. Timmerer, M. Taschwer, K. Schöffmann, Relevance-Based Compression of Cataract Surgery Videos Using Convolutional Neural Networks. In Proceedings of the 28th ACM International Conference on Multimedia, (2020) 3577-3585. https://doi.org/10.1145/3394171.3413658

[36] N. Mathá, K. Schoeffmann, S. Sarny, D. Putzgruber-Adamitsch, Y. El-Shabrawi, Evaluation of Relevance-Driven Compression of Regular Cataract Surgery Videos. In 2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS), (2022) 429-434. https://doi.org/10.1109/CBMS55023.2022.00083

[37] C.Y. Wang, A. Bochkovskiy, H.Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, (2023) 7464-7475. https://doi.org/10.1109/CVPR52729.2023.00721

[38] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics

[39] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context. In Computer Vision-ECCV 2014: 13th European Conference, (2014)740-755. https://doi.org/10.1007/978-3-319-10602-1_48

[40] S. Arora, K. Bhatia, V. Amit, Storage optimization of video surveillance from CCTV camera. 2nd International Conference on Next Generation Computing Technologies, (2016) 710-713. https://doi.org/10.1109/NGCT.2016.7877503

Authors Contribution Statement

Nikita Mohod: Conceptualization, Methodology, Dataset Collection, Analysis, Drafting and Editing. Prateek Agrawal: Conceptualization, Supervision, Editing, Validation. Vishu Madaan: Drafting and Editing, Validation.

Funding

The authors declare that no funds, grants, or other support were received during this work.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

Csc the code data set is available at https://cocodataset.org/#download and on-demand, ATM Surveillance dataset will be provided

Has this article screened for similarity?

Yes

About the License