International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Deep Learning-Based Video Compression for Surveillance Footage

Prateek Agrawal[1], Nikita Mohod[2], Vishu Madaan[3]
[1]School of CSE
Lovely Profesional university
Phagwara, Punjab, India
dr.agrawal.prateek@gmail.com
School of CSE
Lovely Professional University, India
Sipna C.O.E.T, Amravati, Maharashtra, India
mohodnikita9@gmail.com
[3]School of CSE
Lovely Profesional university
Phagwara, Punjab, India
dr.vishumadaan@gmail.com

Abstract—The utilization of closed-circuit television (CCTV) monitoring plays a pivotal role in the realm of video processing, providing an effective means for vigilant surveillance. However, a significant challenge associated with this practice is the substantial demand it places on storage space. Traditionally, surveillance footage is stored on hard disk drives, and due to limited storage capacities, it often necessitates periodic deletion. To tackle this issue, we have introduced an innovative method for compressing CCTV video, named "Detection-Based video Compression" (DBVC). Our DBVC model is a two-step process. In the first step, we employ advanced neural network approaches such as Mask-RCNN and YOLOv4 to determine active and idle frames within the surveillance video. These cutting-edge techniques enable precise identification of objects and events of interest in the video feed. In the second step, we construct a new video composed solely of the active frames, eliminating redundant or uneventful segments. After conducting a comprehensive analysis of the experimental results, it is evident that Mask-RCNN stands out with an impressive detection accuracy of 98% on the COCO dataset, making it a robust choice for identifying objects and events in the surveillance footage. Consequently, we chose to leverage the output generated by Mask-RCNN and YOLOv4 for subsequent processing stages. Our DBVC approach is a breakthrough in video compression technology, significantly reducing the storage space required for CCTV footage. In fact, it achieves an average compression ratio of up to 85% when using YOLOv4, surpassing the capabilities of existing state-of-the-art compression methods. This innovation not only optimizes storage efficiency but also maintains a high level of surveillance data integrity, making it a valuable advancement in the field of CCTV video processing and storage management.

Keywords- Active frames; Idle frames; object detection; YOLOv4; Mask-RCNN; and Surveillance compression.

I. INTRODUCTION

In the digital age, the application of closed-circuit television (CCTV) monitoring has witnessed an unprecedented surge, establishing itself as a cornerstone in the domain of video processing. The use of CCTV systems has proven to be an invaluable tool, facilitating robust and vigilant surveillance across a spectrum of settings, ranging from governmental security initiatives to private sector endeavors. As these systems capture an unceasing stream of video content, they provide a steadfast watchful eye that never wavers. With this increasing number of CCTV cameras and devices, the demand for efficient video storage arises. Typically, the substantial volume of surveillance footage is diligently archived on hard disk drives or local storage devices, despite their efficacy, are bound by finite capacities. To circumvent the inevitable constraints of storage space, a recurring practice has emerged: the periodic deletion of recorded content after a specified duration but this technique leads to the loss of important information also. To address this quandary, compression techniques proffer a compelling solution.

Video compression is the process of reducing the size of a video file while maintaining an acceptable level of video quality. This reduction in file size is achieved by employing various encoding techniques to remove redundant or less essential data, ultimately making video content more manageable for storage and transmission. Basically, there are two types of video compression techniques: lossless and lossy compression. 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. 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 [1] 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.

Artificial intelligence (AI), has significantly revolutionized the field of video compression, enhancing efficiency and video quality. AI-driven video compression employs cutting-edge techniques to reduce file sizes while maintaining or even improving video quality. One key application of AI in this domain is content-aware video encoding, where machine learning (ML) and deep learning (DL) models recognize the content of a video, enabling selective compression that prioritizes critical elements. In our research work, we perform the compression of surveillance video using the concept of content-aware compression. We adopted Automated Teller Machine (ATM) video footage as a pertinent use case for our research investigation. The CCTV cameras installed within ATM rooms continuously capture video content, constituting a round-the-clock surveillance feed. However, the actual periods of substantial activity, related to ATM transactions, represent only a fraction of the total recording duration, while the remainder of the footage predominantly depicts the static state of the ATM machine itself. This situation results in the inefficient utilization of storage resources, as a significant portion of the recorded content lacks of relevance or serves an idle purpose within the context of surveillance. Our approach to addressing this issue involves the utilization of deep learning-based object detection methods to identify frames wherein human presence is detected. Subsequently, the identified active segments, corresponding to human interactions at the ATM, are separated from the non-active segments that portray the static ATM environment. The frames of surveillance video, where the person is present are recognized as active frames while the frames where only firmed machines are available are termed as idle parts. This segmentation process enables a substantial reduction in storage space requirements by eliminating the storage of redundant or inconsequential video content. The consequence is an enhanced efficiency in the management of surveillance video, with the deliberate retention of pertinent, active portions for extended durations, while the idle segments are selectively deleted. The major contribution of our research work is summarized as follows: -

- Collection of nationalize bank ATM surveillance footage
- Divide the video into smaller segments of around 15 minutes and split the video into frames.
- Identify active and idle frames of surveillance video using the neural network approach.
- Remove the idle parts and construct the video back from active frames.
- Compare the performance of neural network approaches Mask-RCNN and YOLOv4.

The subsequent structure of the paper is structured as follows: In Section 2, a comprehensive overview of the pertinent literature pertaining to object detection and video compression is provided. This section furnishes valuable insights into the current landscape of research in these domains. Section 3 expounds upon our proposed DBVC approach, which is further divided into two modules: i) the object detection module and ii) the video compression module. Section 4 is devoted to the presentation and in-depth analysis of the obtained results. Finally, in Section 5, we synthesize our findings into conclusive insights based on our research.

II. LITERATURE SURVEY

A. DL-based object detection

Object detection (OD), a crucial task in computer vision, has witnessed significant advancements, particularly with the emergence of deep learning techniques. This survey delves into the evolution of deep learning-based object detection methods, providing insights into their development, challenges, and applications. The inception of DL-based object detection can be traced back to the development of Convolutional Neural Networks (CNNs). CNNs demonstrated their potential in image classification tasks, but their application to object detection was less straightforward. Region-based Convolutional Neural Network (RCNN) pioneered a two-step approach, first proposing regions of interest and then classifying them [2]. While it suffered from computational inefficiency due to a selective search for region proposals. Building upon the earlier R-CNN, Fast R-CNN addressed the computational inefficiencies and complexities of R-CNN while significantly improving object detection accuracy. Like R-CNN, Fast R-CNN still uses selective search to generate region proposals. However, unlike R-CNN, which applied a separate CNN for each proposed region, Fast R-CNN shares feature extraction across all regions [3].

The demand for real-time object detection led to the creation of single-shot detectors. You Only Look Once (YOLO) and Single Shot Multi-Box Detector (SSD) are prominent examples. YOLO introduced the concept of dividing the image into a grid and predicting bounding boxes and class probabilities directly, making it faster and more suitable for real-time applications [4].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

SSD performs object detection 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 to as anchor boxes [5]. These anchor boxes are pre-defined to cover a range of object sizes and shapes[6]. YOLOv2, also known as YOLO9000, was the second version of the YOLO model. It introduced a significant improvement in terms of accuracy and speed compared to the original YOLO. It has a multi-scale detection approach, which allows it to detect objects of different sizes. It divided the image into grids of different sizes, which helped in detecting both small and large objects in the same image [7]. While YOLOv3 is the third iteration of the YOLO model and introduced further improvements in terms of accuracy and detection speed [8]. It incorporated a Feature Pyramid Network, which improved its ability to detect objects across different scales. YOLOv3 is generally considered to be more accurate than YOLOv2, but it requires more computational resources [9]. To determine objects in surveillance video YOLO version exhibits better accuracy [10], [11]. In 2018 Redmon et al. introduced YOLOv3, which Built upon YOLOv2 and continued to enhance object detection capabilities. It Divided the input image into multiple scales for detecting objects of different sizes and Incorporated the FPN for improved feature representation. YOLOv3 predicted multiple bounding boxes per object to improve localization accuracy and advanced object detection accuracy and versatility [8].

The introduction of a region proposal network (RPN) in Faster R-CNN addressed the computational bottlenecks associated with traditional R-CNN. RPN uses a lightweight neural network to generate region proposals, enabling end-to-end object detection with improved speed and accuracy [12]. Faster R-CNN also introduced anchor-based object detection, where predefined anchor boxes of various aspect ratios and scales are used for object localization. This approach improves object detection accuracy by handling objects of different sizes and shapes. To address the problem of variable input sizes when applying CNN to object detection and image classification tasks, He et al. introduce the spatial pyramid pooling layer (SPP), which is designed to work with images of different dimensions. This layer allows a network to accept input images of different sizes and aspect ratios [13]To handle objects at multiple scales, a feature pyramid Network (FPN) was introduced. FPN uses a top-down architecture to combine features from different layers of a CNN, enabling the detection of objects of various sizes more effectively [14]. Mask-RCNN builds upon the Faster R-CNN object detection model and extends it to provide pixel-level instance segmentation, which not only identifies and localizes objects in an image but also delineates the exact outline of each object[15]. The object detection landscape is divided into two categories: two-stage detectors and one-stage detectors. Two-stage detectors, typified by Faster R-CNN, are renowned for their accuracy but are computationally expensive. On the other hand, one-stage detectors, exemplified by YOLO and SSD, excel in real-time performance but may trade off some accuracy.

B. DL-based video compression

Deep learning-based video compression is an emerging area of research that has the potential to significantly improve video compression efficiency and quality. Video compression is essential for reducing the size of video files while maintaining acceptable quality. Traditional video compression techniques, such as H.264 [16] and H.265, have limitations, and deep learning offers a promising alternative. Particularly CNN, Recurrent Neural Networks (RNN), and autoencoders applied to video compression framework. Variational autoencoders (VAEs) and generative adversarial networks (GANs) enhance the efficiency of compression.

H.265, is a video compression standard that was developed to improve upon its predecessor, H.264. H.265 offers better video quality at lower bitrates, making it more efficient for video streaming and storage. Like H.264, H.265 uses block-based compression. The video frames are divided into small blocks, typically 8x8 or 4x4 pixels in size. This hierarchical structure for partitioning video data into coding units (CU), prediction units (PU), and transform units (TU). allows for efficient coding at various levels of granularity. Allowing the encoder to use larger or smaller blocks based on the complexity of the content. This adaptability enhances compression efficiency.

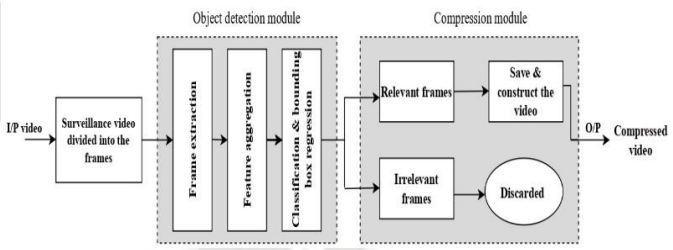

Figure 1. DBVC Model

It also supports both intra-frame and inter-frame compression. Intra-prediction encodes information within a single frame, while inter-prediction exploits similarities between neighboring frames. For inter-frame compression, H.265 uses motion estimation to determine how blocks in the current frame relate to blocks in reference frames (previous and future frames). It estimates the motion vectors to describe how blocks move from one frame to another. Blocks are transformed using a discrete cosine transform (DCT), similar to H.264. The transformed data is then quantized, which reduces the precision of the coefficients, leading to compression. After quantization, entropy coding techniques, such as context-adaptive binary arithmetic coding (CABAC) or context-adaptive variable-length coding (CAVLC), are applied to further compress the data. Also includes Sample Adaptive Offset (SAO) and filtering for

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

reducing distortions and enhancing the visual quality of decoded video. SAO helps mitigate artifacts introduced during the compression process. Specifically, researchers tried to improve the modules of H.265 using a neural network. Research has focused on end-to-end deep learning-based video compression, where the entire compression process is learned from data. This approach can adapt better to various video content and resolutions.

III. DBVC MODEL

This section gives details about the architecture and methods implemented to achieve video compression as shown in Fig.1. Here the input surveillance video is divided into frames and frames are given as input to the object detection module which identifies active/ relevant frames and idle/irrelevant frames of video. Furthermore, the active/relevant frames are used to construct the compressed video. As we did not perform any preprocessing operation on frames, the frames have a 100% similarity index as compared to their original frames.

A. Object detection module

To predict the active and idle frames of video, we used YOLOv4 from a one-stage detector and Mask-RCNN from a two-stage detector which is summarized as follows: -

1) *YOLOv4:-*

In 2016, Redmon et al. presented YOLOv4, an object detection algorithm that builds upon the success of previous YOLO versions in real-time object detection [17]. YOLOv4 uses a backbone network architecture, typically based on the CSPDarknet53 or CSPResNeXt53 architecture. This backbone network extracts features from the input image. It Uses FPN to capture features at different scales. This helps in detecting objects of varying sizes within an image. YOLOv4 includes a novel neck architecture that 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. With the aid of anchor boxes and a grid cell system, YOLOv4 efficiently localizes and classifies objects within images. To optimize object localization and classification, it uses the intersection over union (IoU) loss. YOLOv4 is trained on substantial datasets like COCO and can be further enhanced through data augmentation techniques. Striving for real-time performance, it optimizes network architecture and supports hardware acceleration.

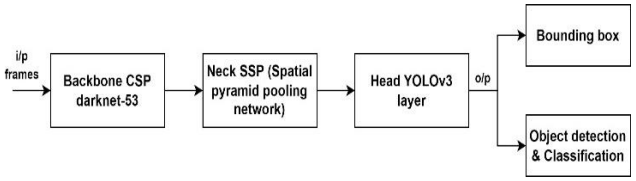

Figure 2. Represents the framework of YOLOv4

2) *Mask-RCNN: -*

He et al. introduced Mask-RCNN which is a state-of-the-art deep learning architecture for object instance segmentation. It is an extension of the Faster R-CNN object detection framework that incorporates pixel-level segmentation masks for each object in the image. The process begins with a region proposal network (RPN) that generates potential bounding box proposals for objects in the input image. These proposals are then passed through a backbone network to extract feature maps. It employs a Region of Interest (RoI) Align operation to preserve fine-grained details in these regions. The model simultaneously performs object detection to predict bounding boxes and assigns class labels to the objects. However, it doesn't stop at detection; it goes further by generating segmentation masks for each object, which depict the precise pixel-wise boundaries of the objects in the image. During training, Mask R-CNN is supervised with both bounding box and pixel-wise mask annotations, optimizing the model using appropriate loss functions. This framework employs several loss functions for training like classification loss, bounding box regression loss and mask loss. During inference, the model takes an input image, performs object detection to identify RoIs, classifies objects, refines bounding boxes, and finally generates pixel-wise masks for each detected object. Figure 3 represents the architecture of the Mask-RCNN module.

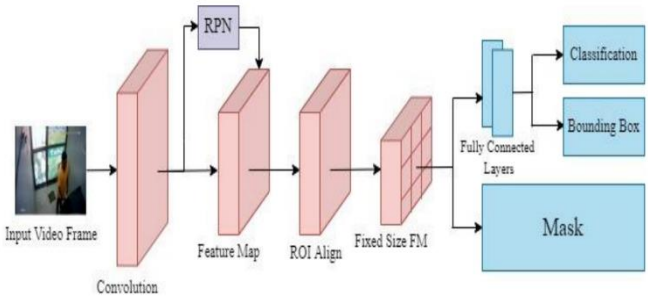

Figure 3. Mask-RCNN Framework

B. Video Compression module

In the compression module of the DBVC framework, compression is achieved by preserving the active frames of surveillance video and eliminating the idle frames. When considering ATM surveillance as an exemplary use case, the active frames pertain to the time frame during which an individual enters an ATM room, engages in transactional activities, and subsequently departs after a certain period. In our DBVC framework, this specific time interval is retained, while instances involving a static, unoccupied ATM machine within the room are deemed nonessential and are consequently excluded to facilitate compression. This discernment is guided by the recognition that, in a 24-hour cycle, merely two or three hours are characterized by any form of activity, with the majority of storage space being squandered on maintaining the superfluous segments. To rectify this issue, the DBVC

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

framework disposes of the idle duration within the surveillance video, thereby enabling the retention of active frames for extended periods. The idle frames in the video hold no prospective value, and the compression model discards these frames, while the active frames are harnessed for the construction of the compressed video.

C. Algorithm and Sequence flow diagram

The sequence flow diagram of the DBVC model is shown in Figure 4. Here, the framework is divided into the following steps: -

- 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.
- One-phase YOLOv4 model and two-phase Mask-RCNN model trained on COCO dataset and then applied on surveillance video for testing purposes.
- Depending on the object recognized in surveillance video using YOLOv4 and Mask-RCNN, frames are split into:- active frames and idle frames.
- In the last step idle frames are discarded and active frames are stored to construct the compressed surveillance video.

Algorithm 1, provides outlines of the DBVC algorithm which is a process for compressing ATM surveillance footage using neural network techniques. Here's a step-by-step explanation of the algorithm:

Algorithm Steps:

1. DBVC_FPS (Frames per Second): This step involves determining the frames per second (FPS) of the input ATM surveillance footage. It's important to know the FPS to control the compression process.

2. DBVC_Split (Split video into frames): The surveillance video is split into individual frames. Each frame represents a single image from the video sequence.

3. DBVC_Save (Save frames after every DBVC_FPS): This step involves saving frames after every specified number of frames per second (DBVC_FPS). It's done to ensure the algorithm processes a manageable number of frames at a time.

4. Loop over frames (do until all frames empty):

    - DBVC_OD (Object Detection): In this part of the loop, the algorithm applies object detection modules, specifically YOLOv4 and Mask-RCNN, to each frame. These modules are responsible for identifying objects and events in the surveillance footage. Object detection helps determine which frames contain objects or activities of interest.

    - DBVC_Active (Predict active frames): This step involves using the results from the object detection modules to predict which frames are considered "active." Active frames likely contain objects or events that require retention in the compressed video.

    - DBVC_Idle (Predict idle frames): Conversely, this step uses the object detection results to predict frames that are "idle" or do not contain relevant objects or activities. These frames can be considered for deletion in the compression process.

5. DBVC_Compressed (Construct DBVC_Active): After the loop over all frames is completed, the algorithm constructs the compressed surveillance video using the frames identified as "active." These active frames are selected for retention in the compressed video.

6. DBVC_Discard (Delete DBVC_Idle): Finally, frames identified as "idle" are deleted, as they are considered unnecessary for the compressed video.

Algorithm 1: DBVC algorithm

Input: ATM surveillance footage
Dataset: COCO
Output: Compressed surveillance video

1 DBVC_FPS←Get FPS (Input ATM surveillance)
2 DBVC_Split← Split video into the frames
3 DBVC_Save← Save frames after every DBVC_FPS
4 *do until all frames empty*
DBVC_OD← Apply YOLOv4 and MaskRCNN modules

DBVC_Active←Predict active frames

DBVC_Idle←Predict idle frames

end
5 DBVC_Compressed← Construct (DBVC_Active)

6 DBVC_Discard←Delete (DBVC_Idle)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

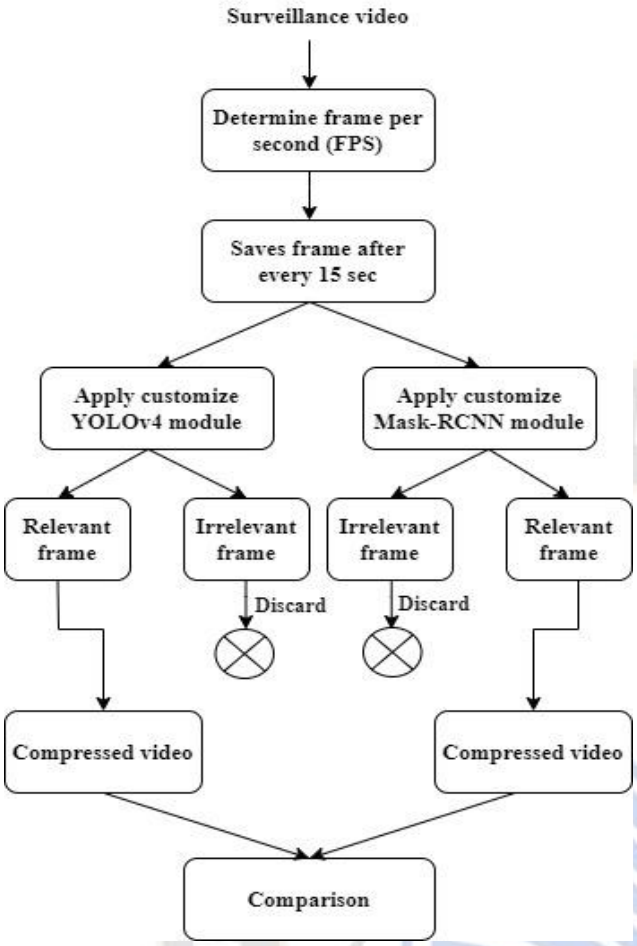

Figure 4. Sequence flow diagram of DBVC model

IV. RESULTS AND ANALYSIS

A. Evaluation metrics

Several evaluation metrics are commonly used to assess the accuracy and reliability of object detection systems. Here are some of the key evaluation metrics for object detection:

1) *Confusion metrics*

A confusion matrix is a tabular representation that summarizes the performance of a classification model in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

2) *Precision*

Precision known as positive predictive value, measures the accuracy of object detection. It calculates the ratio of correctly detected objects (true positives) to the total number of objects the model predicted as positive. Precision tells us the percentage of objects that the model correctly identified among its positive predictions. High precision means that the model has a low rate of false positives.

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

3) *Recall*

Recall, called as sensitivity or true positive rate, measures the ability of the model to find all relevant objects within an image. It calculates the ratio of correctly detected objects (true positives) to the total number of actual objects present in the image.

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

4) *F1 Score*

The F1 score is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance by taking into account both false positives and false negatives.

$$\text{F1 Score} = \frac{2*Precision8Recall}{Precision+Recall}$$

5) *Accuracy*

Accuracy is a more general metric that measures the overall correctness of object detection. It calculates the ratio of correctly detected objects (true positives) and correctly detected background regions (true negatives) to the total number of predictions.

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

B. 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 widely used dataset for object detection, segmentation, and captioning tasks in computer vision [18]. It contains a diverse range of images with complex scenes and multiple objects, making it suitable for challenging object detection tasks. Each image in the dataset is annotated with precise object segmentations, object categories, and key points (for human poses). It is a large-scale dataset with tens of thousands of images and over 80 object categories, including common everyday objects, and serves as a benchmark for evaluating the performance of object detection models and is frequently used in research and competitions. The dataset is divided into 80:20 ratio for training purposes An ATM surveillance dataset typically refers to a collection of videos captured by surveillance cameras placed in ATM rooms. We collected 10 different ATM footage and tested our DBVC model. The results are shown in Table II.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

C. Mask-RCNN vs YOLOv4

We implement YOLOv4 and Mask R-CNN in Google Colab as these models are computationally intensive and require GPU resources. The parameter specification required for the implementation of YOLOv4 and Mask-RCNN model is mentioned in Table 1.

TABLE I. PARAMETER SPECIFICATION

Parameter	YOLOv4	Mask-RCNN
Backbone Network	CSP-DarkNET	ResNet-50 / ResNet 101
Pre-trained weights	yolov4.weights	mask_rcnn_coco.h5
Learning Rate	0.001	0.001
Epoch	300	1000
Batch Size	32	2
Loss Function(s)	Multi-task loss	CIoU-loss
GPU	Yes	Yes

YOLOv4 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. The pre-trained weights for YOLOv4 are typically provided in the file named "yolov4.weights." For training YOLOv4 on custom data, a commonly used learning rate is 0.001, and training is typically conducted over 300 epochs. A batch size of 32 is often employed during training. YOLOv4 employs a multi-task loss function that combines components like objectness loss, class loss, and localization loss. GPU support is essential for the computational demands of the model, making it suitable for parallel processing and accelerating training.

Mask R-CNN employs a ResNet-50 or ResNet-101 backbone network architecture for its feature extraction. The pre-trained weights for Mask R-CNN are usually found in the file "mask_rcnn_coco.h5,". A common choice for the learning rate in Mask R-CNN is 0.001, and training often extends over a more extended period of 1000 epochs. The batch size for training Mask R-CNN is relatively smaller, typically set to 2. Mask R-CNN uses a CIoU (Complete Intersection over Union) loss function, designed to optimize both bounding box localization and mask segmentation tasks concurrently. GPU support is essential to ensure efficient training and inference in Mask R-CNN, especially for the computationally intensive task of instance segmentation.

We tested the performance of YOLOv4 and Mask-RCNN on the ATM surveillance dataset and received the result mentioned in Table 2. The YOLOv4 module exhibits a high level of precision at 96.6%, meaning that a significant portion of the positive predictions it makes is accurate. Its recall stands at 97.1%, indicating its effectiveness in capturing most of the actual positive instances. The F1 score, which balances precision and recall, is 96.8%, showing a harmonious trade-off between accuracy and completeness. In terms of overall correctness, YOLOv4 boasts an accuracy of 97.3%, implying that the majority of its predictions, both positive and negative, are correct. On the other hand, Mask-RCNN performs even better in terms of precision, with a precision score of 98.2%, indicating a very low rate of false positives. Its recall at 97.8% is also impressive, signifying its ability to capture a high percentage of actual positive instances. The F1 score is 98.1%, suggesting an excellent balance between precision and recall. Mask-RCNN's accuracy at 98.2% indicates a high degree of overall correctness in its predictions. Mask-RCNN, with its slightly higher precision and accuracy, appears to be slightly more accurate and consistent in its predictions, while YOLOv4 maintains a strong balance between precision and recall, making it suitable for tasks where minimizing both false positives and false negatives is crucial. So we used the output of both modules for the compression module. Figures 5 and 6 represent the output of the object detection module.

TABLE II. COMPARISON OF YOLOV4 AND MASK-RCNN

Parameter	Yolov4	Mask-RCNN
Precision (%)	96.6	98.2
Recall (%)	97.1	97.8
F1 Score (%)	96.8	98.1
Accuracy (%)	97.3	98.2

Figure 5. Output of YOLOv4 object detection module

The confidence score for person detection with YOLOv4 is indicated as 85% shown in Figure 5, whereas when utilizing Mask-RCNN, it is depicted as 100% shown in Figure 6.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Figure 6. Output of Mask-RCNN object detection module

In the compression module, we utilized the output of YOLOv4 and Mask-RCNN to conduct compression. Table 3 presents an analysis of the compression module. In this context, we utilized an ATM surveillance video with a size of 351 MB and a duration of 20 minutes. Initially, we ascertained the video's FPS and divided it into 18000 frames. Since the ATM surveillance video had an FPS value of 15 FPS, we saved frames every 15 seconds, resulting in a total of 1200 (18000/15) frames passed as input to the OD modules. Using the YOLOv4 model, we recognize 365 active frames and 835 idle frames within the surveillance video, and simultaneously, a video consisting of the active frames is also constructed of size 47.4 MB and of duration 25 seconds. The total time required to perform this task using the YOLOv4 network is 220 seconds. On the other hand, using Mask-RCNN we detect 403 active frames and 797 idle frames. The size of the reconstructed video from relevant frames is 52.2 MB of interval 26 seconds. Mask-RCNN network takes 4384 seconds to achieve compression. The Mask-RCNN takes more time it detect the object accurately than the YOLOv4 network. While YOLOv4 achieved a higher compression ratio as compared to Mask-RCNN. With the help of the YOLOv4 network, we achieved 86.61% compression percentage and 85.12% using Mask-RCNN. From this, we can conclude that the YOLOv4 network is faster while Mask-RCNN is more accurate and efficient.

TABLE III. COMPARISON OF YOLOV4 AND MASK-RCNN FOR COMPRESSION MODULE ON VIDEO 1

Parameters	Original Video	Yolov4	Mask-RCNN
Size	351 MB	47.4 MB	52.2 MB
Duration	20 Min	25 Sec	26 Sec
Total Frames	1200 (18000/15)	NA	NA
Active frames	NA	365	403
Idle Frames	NA	835	797
Time	NA	220 Sec	4384 Sec
Compression percentage	NA	86.61 %	85.12 %

TABLE IV. COMPARISON OF YOLOV4 AND MASK-RCNN FOR COMPRESSION MODULE ON VIDEO 2

Parameters	Original Video	Yolov4	Mask-RCNN
Size	438 MB	75 MB	78.2 MB
Duration	24 Min 58 Sec	75 Sec	76 Sec
Total Frames	1499 (22485/15)	NA	NA
Active frames	NA	1133	1147
Idle Frames	NA	366	352
Time	NA	394 Sec	5701 sec
Compression percentage	NA	84.01 %	82.14 %

For video 2, where the size of the original video is 438 MB, we are able to reduce it to 75 MB using the YOLOv4 module and 80 MB using the Mask-RCNN module. Using our DBVC model, we are able to achieve an average compression ratio of 85.31 % with YOLOv4 and 83.63 % with the Mask-RCNN approach.

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

We compare our DBVC model with two other approaches which are summarized in Table 5

TABLE V. COMPARISON WITH OTHER MODELS

Parameter	Arora et al. [19]	Patra et al. [20]	DBVC framework
DL approach	No	No	Yes
OD approach	No	No	Yes
Type of Compression	Lossy - Compression	Lossy - Compression	Lossy - Compression
Compression achieved	64.32 %	15.71 %	85.31%

We implement the approach of Arora et al. and Patra et al. on our ATM surveillance dataset. Arora et al. perform compression by finding the similarity between frames while Patra et al. use down sampling of frame concept. Neither approach used DL techniques or OD approach in their framework. An experimental analysis states that our model outperforms the existing state-of-the-art approaches and achieves a remarkable average compression ratio of 85% using neural network approaches. As per our knowledge, we are the first to implement compression using a neural network approach on ATM surveillance video.

V. CONCLUSION

We present the novel Detection-Based video Compression (DBVC) model to effectively compress CCTV video while preserving critical surveillance information. The model consists of two crucial steps: (i) it utilizes advanced neural network approaches, namely Mask-RCNN and YOLOv4, to differentiate between active and idle frames in the surveillance video. Second, it reconstructs the video by retaining only the active frames. The results of comprehensive experiments indicate that Mask-RCNN achieves a remarkable detection accuracy of 98% on the COCO dataset, establishing its proficiency. The DBVC approach successfully reduces storage space demands, achieving an impressive average compression ratio of up to 85% with YOLOv4, surpassing existing state-of-the-art methods.

The results of our comprehensive experiments validate the efficacy of DBVC, and the high compression ratio achieved, coupled with the outstanding detection accuracy, positions our model as a game-changer in the realm of video compression for surveillance. By significantly optimizing storage efficiency while ensuring vital surveillance data is retained, DBVC offers a powerful tool for businesses, organizations, and security agencies seeking to enhance their surveillance capabilities. Furthermore, this work sets a new standard for the industry, highlighting the potential of advanced neural network approaches in addressing critical challenges in video compression and storage management.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Transactions on circuits and systems for video technology, vol. 22, no. 12, p. 1649, 2012.

[2] R. Girshick, J. Donahue, T. Darrell, and 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, pp. 580–587.

[3] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[5] Y. Zhong, J. Wang, … J. P.-P. of the I., and undefined 2020, "Anchor box optimization for object detection," openaccess.thecvf.com, Accessed: Oct. 28, 2023. [Online]. Available: http://openaccess.thecvf.com/content_WACV_2020/html/Zhong_Anchor_Box_Optimization_for_Object_Detection_WACV_2020_paper.html

[6] W. Liu et al., "Ssd: Single shot multibox detector," in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, 2016, pp. 21–37.

[7] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.

[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[9] T. and A. G. and T. J. V Diwan, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," Multimed Tools Appl, vol. 82, no. 6, pp. 9243–9275, 2023.

[10] N. Mohod, P. Agrawal, and V. Madaan, "YOLOv4 Vs YOLOv5: Object Detection on Surveillance Videos," in International Conference on Advanced Network Technologies and Intelligent Computing, 2022, pp. 654–665.

[11] N. Mohod, P. Agrawal, and V. Madan, "Human Detection in Surveillance Video using Deep Learning Approach," in 2023 6th International Conference on Information Systems and Computer Networks (ISCON), 2023, pp. 1–6.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Adv Neural Inf Process Syst, vol. 28, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Trans Pattern Anal Mach Intell, vol. 37, no. 9, pp. 1904–1916, 2015.

[14] 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, pp. 2117–2125.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[16] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," IEEE Transactions on circuits and systems for video technology, vol. 13, no. 7, pp. 560–576, 2003.

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[18] T.-Y. Lin et al., "Microsoft coco: Common objects in context," in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, 2014, pp. 740–755.

[19] S. Arora, K. Bhatia, V. A.-2016 2nd I. Conference, and undefined 2016, "Storage optimization of video surveillance from CCTV camera," ieeexplore.ieee.orgS Arora, K Bhatia, V Amit2016 2nd International Conference on Next Generation Computing, 2016•ieeexplore.ieee.org, Accessed: Oct. 28, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7877503/

[20] A. Patra, K. Gain, L. Rana, and K. Biswas, "A Novel Method of CCTV Video Compression using Down Sampling," academia.eduA Patra, K Gain, LC Rana, K Biswasacademia.edu, Accessed: Oct. 28, 2023. [Online]. Available: https://www.academia.edu/download/68590621/a_novel_method_of_cctv_video_IJERTCONV9IS11006.pdf