Video Object Detection From Compressed Formats for Modern Lightweight Consumer Electronics

Sangeeta Yadav, Preeti Gulia, Nasib Singh Gill, Ishaani Priyadarshini, Rohit Sharma, *Senior Member, IEEE*, Kusum Yadav, and Ahmed Alkhayyat

***Abstract*—The rapid rise of technological advancements led to the increased consumption of electronic gadgets. This change expedited the requirement for sustainable technologies to meet the growing consumer requirements with minimum computational costs. Nowadays, video content shares a large proportion of the Internet bandwidth. Object Detection from the videos is essential in various real-time applications. Traditionally, the videos are decoded to the raw format for detection tasks. This analytics process can be more efficient if the detection tasks are carried out from compressed video formats instead of raw video. The compressed format of the videos, produced by modern deep learning-based approaches, contains both semantic and motion information in easily consumable formats. Based on the same notion, a video compression cum object detection network has been proposed in this paper, which consumes the compressed videos for carrying out detection tasks. The proposed network comprises an already-designed video compression network, which has been extended to incorporate object detection capabilities. The proposed network has been experimented with a standard ImageNet VID dataset, and the results show fast and efficient object detection from the compressed videos. Coupled with temporal features, the proposed model achieves significantly better mAP of 44.3 *w.r.t.* 36.7 and 39.6 from YOLOv5-s and YOLOX-s models, respectively. The comparative results have shown incremental improvement in the detection tasks from the compressed videos, making it sustainable for its application in modern lightweight consumer electronic devices.**

***Index Terms*—Autoencoder, consumer electronics, deep learning, object detection, video compression.**

I. Introduction

IN CONSUMER electronic devices, video content is heavily consumed. Hence, video compression techniques are one of the primary technologies found in these devices for efficient storage, retrieval, and transmission. Several real-world data sources, such as autonomous driving, human-computer interaction, and visual surveillance, are based on video content. Significant improvements have been made in image analysis using deep learning methods during the past few years [1], [2], [3]. Some novel frameworks for single image object detection have been proposed, including YOLO (You Look Only Once) [4], SSD (Single Shot Multibox Detector) [5], R-FCN (Region-based Fully Convolutional Network) [6], Faster R-CNN (Regions with Convolutional Neural Networks) [7] and FPN (Feature Pyramid Network) [8]. Despite promising static image object detection results, video detection will remain challenging. As image-related distortions are present in frames, several past works have focused on enhancing the frame-wise detection outcomes [9], [10], [11]. The existing image recognition networks are individually applied to take out the features of the dense frames, and the bounding box rescoring or feature aggregation leverages temporal coherence. These methods resulted in improved performance. CNNs-based processing of the dense frames is very computationally expensive and, hence, becomes more complex and unaffordable as the video size goes longer. Several methods have been proposed to decrease the redundant computation [12]. These methods employ expensive feature extractors on sparse vital frames, and then these results are propagated to the other frames. The main idea behind the feature propagation is measuring the pixel-wise displacements using FlowNet [13]. FlowNet comprises multiple convolutional layers, so it takes some extra time for displacement calculation. These methods consider video a collection of consecutive frames, ignoring that videos are transmitted and stored in compressed formats. A video is split into intra-codes, I frame, and predictive P/B frames. An I frame comprises a whole image, but only the changes relative to the reference frame are stored in P/B frames. As the consecutive frames in a video are highly correlated, the changes among them are already encoded in the video stream. The processing of videos as a collection of consecutive frames and then employing diverse methods to retrieve motion cues is cumbersome.

Action recognition plays a vital role in video analytics. Here, the type of action is predicted based on the movements in the given video. As video accumulates richer information than stills, understanding or action recognition from videos stimulates new explorations in vision and deep learning. Traditionally, the server first decodes the video into a sizeable raw format for carrying out analytics from the video content. Then, the analytics engine generates the metadata

Manuscript received 6 April 2023; revised 26 May 2023 and 23 September 2023; accepted 14 October 2023. Date of publication 18 October 2023; date of current version 26 April 2024. *(Corresponding author: Rohit Sharma.)*

Sangeeta Yadav, Preeti Gulia, and Nasib Singh Gill are with the Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak 124001, India (e-mail: sangeeta.rs.dcsa@mdurohtak.ac.in; preeti@mdurohtak.ac.in; nasib.gill@mdurohtak.ac.in).

Ishaani Priyadarshini is with the School of Information, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: ishaani@ischool.berkeley.edu).

Rohit Sharma is with the Department of ECE, SRM Institute of Science and Technology, Ghaziabad 201204, India, and also with the Department of Electronics and Communication Engineering, ABES Engineering College, Ghaziabad, India (e-mail: rohitapece@gmail.com).

Kusum Yadav is with the College of Computer Science and Engineering, University of Ha'il, Ha'il 55476, Saudi Arabia (e-mail: Y.kusum@uoh.edu.sa).

Ahmed Alkhayyat is with the College of Technical Engineering, The Islamic University, Najaf 54001, Iraq (e-mail: ahmedalkhayyat85@gmail.com).

Digital Object Identifier 10.1109/TCE.2023.3325480

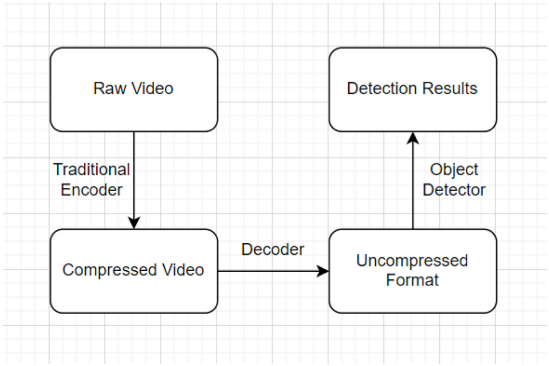


Fig. 1. Traditional scheme of object detection from videos.

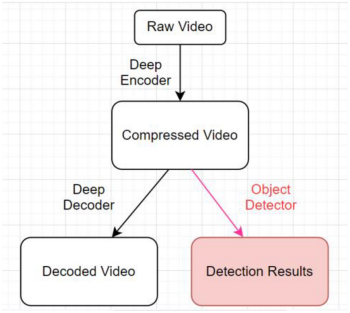


Fig. 2. Proposed scheme of object detection from videos.

using the decoded raw video, which is stored and utilized for the analytics tasks. The server must do decoding and analytics computation as in this analytics process. Hence, it is a time-consuming and less efficient approach to video analytics computation. Figure 1 illustrates the whole process.

Decoding compressed videos for object/action detection increases the computation task of the analytics server. The efficient detection directly from compressed videos will surely enhance the whole process. This same concern has motivated us to use the proposed model. Our model is based on an integrated network comprising compression and object detection sub-networks. It exploits the compressed formats for object detection tasks instead of decoding into raw format. As the efficiency of modern lightweight consumer electronic devices primarily depends on low computation tasks, the proposed model can be effectively used in those devices to provide fast decoding and analytics capabilities. The proposed scheme is presented in Figure 2.

The working of traditional compression codecs considers the similarity between the successive frames. They keep some essential frames and reproduce the remaining frames using the preserved frames' residual error and motion vectors. Our model comprises a flow autoencoder for efficient motion vector compression and processing. The frame autoencoder eradicates the redundancy and insignificant data and makes necessary signals prominent. The motion vectors provide better motion information than simple RGB stills. In addition, motion signals did not consider spatial differences; for example, if two persons do the same thing but in varied lighting and attires, they will generate the same motion signals. Hence, the generalization gets better and improves training efficiency due to lesser variation. The proposed model exploits some time-sensitive variations rather than i.i.d. frames in addition to the spatial features. This way of constraining information helps to address the dimensionality overhead. Avoiding continual processing of near duplicates and using only valid signals enhances the model's efficiency. Lastly, as object detection operates directly on compressed formats of the videos, this saving of decompression overload also adds to the efficiency of the network.

In this study, a deep learning-based technique of video object detection has been proposed. The proposed work is an autoencoder-based video compression network extension [14]. This end-to-end network compresses the frames with frame and motion autoencoders and motion extension networks. This paper has extended this network to explore and carry out the detection task from the compressed format of the videos. The video stream has been compressed using the network as proposed in [14], and then the YOLOX backbone-based object detection network has been used to detect the objects from the compressed format of videos. The comparative results have shown some incremental improvement in the detection from the compressed videos. One of the limitations of current neural network-based compression and decompression is that it is not optimized for performance by hardware accelerators. Traditional Video Codecs such as HEVC are implemented efficiently via hardware accelerators, providing better efficiency in practice. Also, the current method is only suitable for low-resolution images.

The significant contributions of this paper are:

(a) This article discusses detailed literature on object detection in videos and deep learning models for compressed videos.

(b) A novel video compression-cum-object detection model is proposed for lightweight consumer electronics comprising an autoencoder-based video compression network and YOLOX-based object detector, elaborated stepwise.

(c) This study discusses the ablation study of the proposed model.

(d) This study outlines the comparative analysis of the proposed model with the existing ones.

This paper is organized as follows: Section II elaborates on the related literature review that helps to understand the related works that have been done in this field. Section III, describes explains the details of the proposed model in a stepwise manner. It also explains the details of the dataset and the evaluation parameters used in implementation. Section IV presents the experimental results analysis of the proposed model. It also provides a comparative analysis of the proposed and existing models. The conclusion of the research study is given in Section V.

II. Related Work

Generally, the widely used object detection methods comprise detection networks and feature networks. A typical proposal-based object detector has been proposed by Ross Girshick et al. based on the extracted proposals [4]. The proposal generation step is further integrated into CNNs using Faster R-CNN [7]. In comparison to Faster R-CNN, R FCN performs better with higher speed. Earlier object detection methods are based on per-frame detection, and the detection quality is further improved by exploiting temporal coherence. Some methods also attempted to leverage temporal redundancy

to speed up the computation. Several end-to-end learning models have been proposed to achieve high performance and to improve the per-frame features [15].

Dosovitskiy et al. proposed Flownet, a network for learning optical flow with CNNs [13]. Several works used this network to aggregate and align features in their model. Kang et al. efficiently generated spatiotemporal proposals using a tubelet proposal network [16]. Xiao and Lee have computed the relation among neighboring frames, and then their features are aggregated using a memory module [11]. The features of the consequent frames are aligned across time using deformable convolutions [15]. Instead of feature-level aggregation, the models proposed in [9], [15] are based on detecting bounding boxes. The weaker detections have been further enhanced by proposing several mapping strategies to link cross-frame box sequences to still image detections.

Christoph Feichtenhofer et al. carried out the joint learning of the detector and ROI tracker. The cross-frame boxes are also linked, exploiting the same tracker. The works mentioned above employ high computational networks for per-frame feature generation, resulting in higher detection performance. The optical flow networks are used for fast inference by computing the pixel-level correspondence and propagating the extracted deep features from keyframes to the remaining frames [12]. In comparison to feature networks, flow estimation, and feature propagation are quicker; hence, significant speed improvement can be achieved. Box-level temporal propagation has been introduced in [17]. Firstly, the bounding boxes of the keyframes are generated. Then, bounding boxes of other frames are generated using a coarse-to-fine network.

Liu and Zhu utilized convolutional LSTM to propagate feature maps across frames [18]. These works focus on appearance features, only ignoring capturing motion cues explicitly. Though their models are faster than earlier ones, the performance degrades. Some researchers also focused on model accelerating to focus on developing such lightweight deep networks that are unrelated to specific chores.

Advanced Video Coding, MPEG-4 Part 10/H.264 formats are widely used to compress, record, and distribute video frames. This motion compensation-based method is block-oriented. The 3D video compression technique of HEVC has also been proposed [19]. The researchers have explored diverse dimensions of object detection-based video compression and video coding for machines [20], [21]. YOLO is emerged as an efficient and state of technique for object detection [22].

Moreover, several recent standard development activities, evaluation frameworks, and compact visual representation compression schemes for modern video coding have been proposed and experimented for their compatibility and suitability with object detection techniques [23], [24], [25]. The effect of optimization techniques on video coding has also been analyzed [26]. Along with video coding, researchers have also explored the potential and performance of object detection techniques with encoded videos [27], [28]. To improve the performance of the analytics tasks, many different approaches, such as enhanced global-local aggregation and intelligent analytics by collaborative compression, have been proposed and analyzed in terms of their complexity and performance [29], [30].

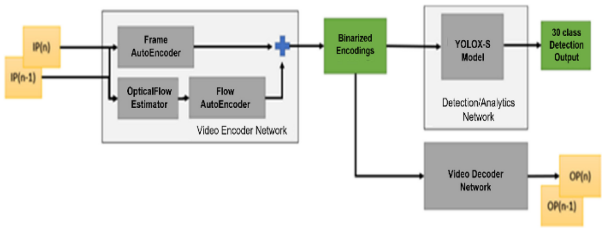


Fig. 3. Overview of proposed work.

The field of video analytics from their compressed formats is less explored, and only limited works and literature are available. Some have generated non-deep features from the signals of the compressed streams, while others have tried to generate video-level features to enhance both performance and speed. In such models with per-frame feature quality requirements, the video object detections must generate per-frame bounding boxes. Some networks comprising QoS-aware and intelligent capabilities have also been proposed [31], [32].

A collaborative framework known as LtC has been introduced to enhance the efficiency of video stream reduction within an analytics pipeline. This framework, detailed in [33], operates by using the full-fledged analytics algorithm on the server side as a teacher to train a lightweight neural network, referred to as the student network. Additionally, recent advancements in video compression techniques have been assessed, considering both inter-frame and intra-frame codecs and their performance metrics like PSNR and encoding time [34]. These evaluations encompass natural and synthetic video content, investigating various optimization strategies for video compression. These optimizations ultimately lead to improved analytics capabilities.

Furthermore, a practical application has been proposed for shop floor environments, leveraging heavy-learning-based methods in combination with unsupervised segmentation and lenient machine learning techniques for classification purposes, as discussed in [35]. Moreover, an object detection and tracking method has been put forward, operating within the video decoding process to ensure swift de-identification. This approach, outlined in [36], achieves computational efficiency by partially identifying personal information elements like faces and vehicle license plates within Groups of Pictures (GoPs) and tracking their locations using object displacement features. This work is also motivated by the recent research in the analytics domain from compressed formats. In this proposed work, YOLOX is used as a baseline, and its computation speed is further improved for video object detection, making it applicable to modern sustainable consumer electronics.

III. Proposed Model

The proposed model is designed to make the detection directly from the compressed format of the videos. Deep learning is emerging as a potential tool for the next generation of pure end-to-end trainable and optimizable video codecs. It already presents a breakthrough in the analytics domain. The proposed model comprises an end-to-end trained pure video compression-cum-detection network. The entire integration of two sub-networks, i.e., compression network and detection. The compression network efficiently compresses the video

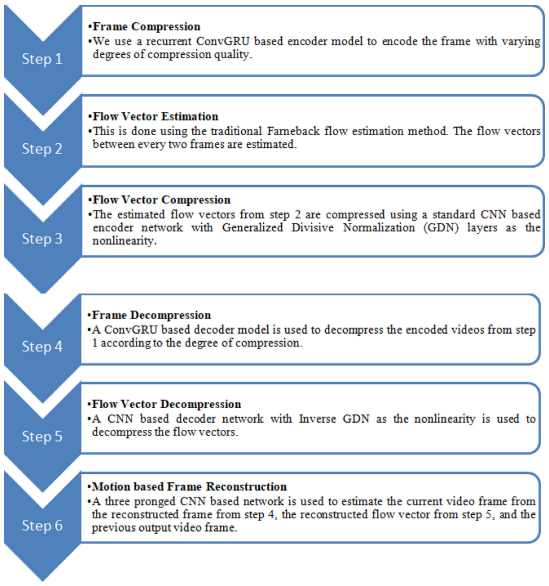


Fig. 4. Flow of operations among different sub-networks of the video compression.

frames and then feeds this compressed representation to the detection network to detect the objects of various categories. The overview of the proposed work is presented in Figure 3.

This figure depicts that the video compression network takes the video frames as the input, and the frame autoencoder and flow encoder components collectively generate their corresponding binarized formats. Then, the YOLOX-S-based detection network detects the objects in the given input video from the binarized formats without decoding them. The detailed architecture and internal configurations of the compression and detection networks are in the below sections.

A. Video Compression

The proposed work is an experimental approach to detecting the objects from the compressed video frames. The whole work is carried out in two phases. In the first phase, the video frames are compressed using a deep compression network, and then the detection network detects objects from the compressed formats produced by the earlier network. The same authors proposed and experimented with a deep video compression network to compress the video frames. This deep compression model is an integrated network of three sub-networks: frame autoencoder, flow autoencoder, and motion extension network. The frame and flow autoencoder are used for frame and flow vector compression, and the motion extension network is used to reconstruct the next frame based on various inputs. The various operations and their flow among these modules are given in Figure 4.

The organization of different modules in the video compression network is shown in Figure 5. The network design depicts that frame and flow autoencoders comprise multiple encoder and decoder networks. The frame autoencoder compresses the video frames, and the flow autoencoder compresses the flow vector. Consequently, taking these two outputs, the motion extension network reconstructs the next frame by referencing the previous frame.

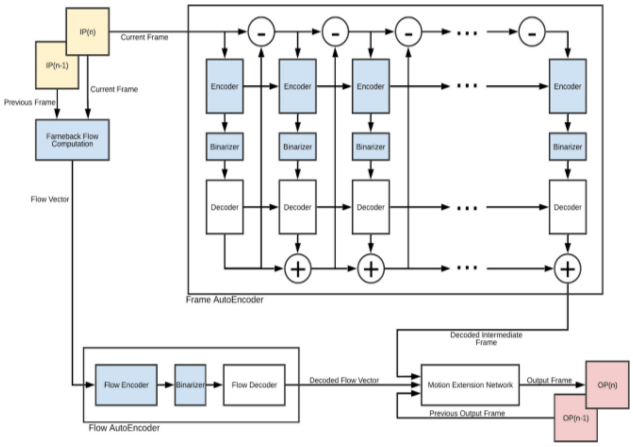


Fig. 5. Video Compression architecture overview [42].

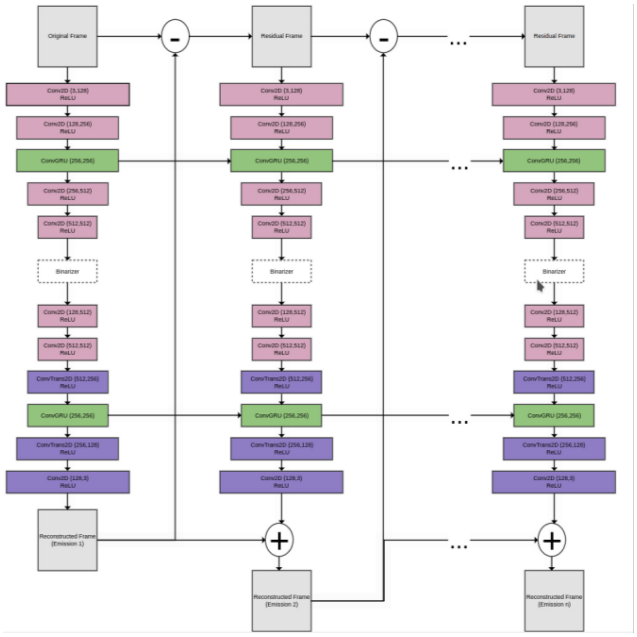


Fig. 6. Frame Autoencoder.

The frame autoencoder comprises Conv and ConvGRU layers. Multiple emission steps are used to produce frames of varying quality. An input video frame passed through the multiple layers of the encoder network and finally encoded in binary format. This binary format is then quantized, followed by the deconvolution process to regenerate the video frames accordingly. The detailed design of the frame autoencoder is given in Figure 6. The flow autoencoder and motion extension networks are similarly designed and comprise deep convolution layers. For flow vector computation, the traditional Farneback flow estimation method is used. A three-pronged CNN-based network has been used to predict the next video frame. It employs CNN blocks to transform inputs and concatenate blocks to merge the two prongs.

The binary-coded output of the frame encoder and flow encoder are used to reproduce the video frames. The size of frame encoder data is controlled by the number of emission steps and the signal's bit rate. The frame decoder computes the intermediate frame using the binary-coded frame

Algorithm 1 Pseudocode of Proposed Deep Encoder Model

Input: $I_{aw}\{I_0, I_1, I_2, \ldots\ldots I_i\}$ = Sequence of raw video frames, i = number of frame in video sequence.
Output: Encoded video frame sequence output $I_{encoded}$ and encoded flow sequence output $F_{encoded}$

Begin
For I_i of frames I_{raw}
Step 1: Previous Frame Retrieval
- Return previous raw frame $I_{i,1}$ if stored in the cache as I_{pre}
- Return current raw frame I_0 if no frame is stored in the cache as I_{mec}

Step 2: Flow Estimation
- Resize images to small resolution.
- F_{raw} = Farneback flow estimation (I_i, I_{prc})

Step 3: Encoding Frame input I_{stoded}
- Normalize raw frame Ifrom 0-255 range to 0-1 range.
- Decide on emission steps (S) based on desired quality and bit rate.
- I_{scoded} = VideoEncoderNetwork.predict (I_i, S)

Step 4: Encoding Flow input $F_{encoded}$
- Normalize raw frame F_{from} 0-255 range to 0-1 range.
- $F_{encoded}$ = FlowEncoderNetwork.predict (F_{raw})

Step 5: Store current frame as the previous frame in cache
- $I_{pre} = I_i$

Step 6: Store or Transmit Encoded Video Stream
- Store or transmit $I_{encoded}$ and $F_{encoded}$ in sequence.

END

Algorithm 2 Pseudocode of Proposed Video Decoder Model From Encoded Video Stream

Input: $I_{encoded}\{I_0, I_1, I_2, \ldots\ldots\ldots I_i\}$ = Sequence of encoded video frames and $F_{encoded}\{F_0, F_1, F_2, \ldots\ldots\ldots F_i\}$ = Sequence of encoded flow vectors where i = number of frame in video sequence.
Output: $I_{decoded}\{I_0, I_1, I_2, \ldots, I_i\}$ = Sequence of decoded video frames, i = number of frame in video sequence.

Begin
For I_i of frames $I_{encoded}$ and F_i of flow $F_{encoded}$
Step 1: Previous Frame Retrieval
- Return previously decoded frame $I_{i,1}$ if stored in the cache as I_{pre}
- Return reference raw frame I_0 if no frame is stored in the cache as I_{pre}

Step 2: Frame Decoder Network
- $I_{inermediate}$ = FrameDecoderNetwork.predict $(I_{encoded}, S)$

Step 3: Flow Decoder Network
- $F_{inermediate}$ = FlowDecoderNetwork.predict $(F_{encoded})$

Step 4: Motion Extension Network
- $I_{decoded}$ = MotionExtNetwork.predict(I_{pre}, $I_{inermediate}$ $F_{inermediate}$)

Step 5: Store curent frame as the previous frame in cache
- $I_{pre} = I_{decoded}$

Step 6: Visualize Decoded Video Stream
- Convert $I_{decoded}$ from 0-1 range to 0-255 range
- Visualize $I_{decoded}$ in sequence.

END

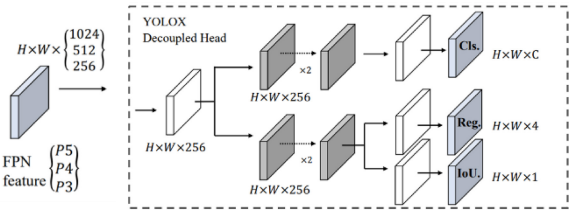


Fig. 7. The detection network employed on compressed streams.

data. Similarly, the flow decoder network decodes the flow information. Finally, the Motion Extension produces the intermediate representation of the current image using the previous image and decoded flow information.

The intermediate representations merged on the decoded intermediate frame to produce a high-quality current frame. The same has been represented by equation (1).

$$I_{decoded} = f_{decoder}\left(I_{encoded}\ \ F_{encoded}\ \ I_{decoded_{prev}}\right) \tag{1}$$

where "$I_{encoded}$ and $F_{encoded}$ are binarized encoding of current frame and flow vectors, $I_{decoded_{prev}}$ is previously decoded frame and $f_{decoder}$ is the representation of decoder neural network."

The pseudo-code for encoding and decoding the video frames is given in Algorithm 1 and 2.

The video compression network is designed to minimize the structural distortion between the input video and the output. Mean Squared Error (MSE) minimizes the color distortion in decompressed images.

$$L = L_{ssim} + \alpha\ L_{mse} \tag{2}$$

where MSE error is evaluated as:

$$L_{mse}(y, y') = \frac{1}{N}\sum_{0}^{n}(y - y'_i)^2 \tag{3}$$

Three comparison measurements, namely structure (s), contrast (c), and luminance (l), are used to compute the SSIM error.

$$L_{ssim}(y, y') = \left[l(y, y') \cdot c(y, y') \cdot s(y, y')\right] \tag{4}$$

The Flow MotionNet video compression model has been trained with a randomized emission steps training strategy by taking emission steps ranging from 1 to 10.

B. Detection Network for Compressed Video Streams

The proposed detection network utilizes the compressed representation of the videos for detection. YOLOX-S has been used as a single-stage object detector backbone with the addition of GRU layers for feature aggregation across the temporal domain. The design of the detection network employed on compressed streams is given in Figure 7.

The design of the network is given in Figure 7. Firstly, a 1 × 1 conv layer is employed to decrease the feature channel to 256 for each level of FPN feature. Then, two parallel branches are added with two 3 × 3 conv layers for regression and classification. IoU branch is also added to the regression branch. The algorithm and corresponding pseudo-code used for the detection task are given in Algorithm 3.

IV. Experiment and Results

A. Training and Implementation

The experimentation has been carried out by using the large-scale ImageNet VID dataset. This dataset comprises 3862 video streams for model training. 937 and 555 video streams are used for testing and validation, respectively. The frame rate is 30 or 25 frames per second (fps). The frame-level bounding box annotations are also available for validation and

Algorithm 3 Multi-Label Video Object Detection

Input

1. Encoded frames $I_{encoded}$

Output

1. Predicted video labels/tags $y'_{labels_{thresholded}}$

Steps:

1. Label encoded frames $I_{encoded}$ using frame labeler network $f_{labeler}$.
2. Filter labels with confidence greater than the threshold and yield final predicted video labels $y'_{labels_{thresholded}}$.

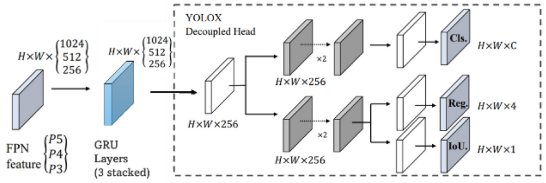


Fig. 8. The detection network with GRU temporal feature module employed on compressed streams.

training purposes. The video streams of the ImageNet DET sub-dataset contain the objects of 30 categories. The training set of the proposed model comprises both ImageNet DET and VID training sets. For the evaluation, only the VID validation set is taken. Firstly, the deep encoder converts all frames into a binarized compressed format. The mean average precision (mAP) performance metric reports the results. The images from the ImageNet DET and ImageNet VID datasets are taken in the ratio of 2:1 from each mini-batch, which is then used to train the model with no motion encodings. A Titan X GPU and Titan Xp GPU are used to perform comparative analysis and ablation study. The model is trained with 4 Titan Xp GPUs in 120 iterations and an SGD optimizer. 2.5×10^{-5} is the learning rate for the last 40K iterations, while 2.5×10^{-4} for the first 80K.

Training with Motion Encodings: The Training with motion encodings uses only the ImageNet VID dataset. The Motion encodings are calculated from videos, compressed into binarized encodings using a motion encoder, and then concatenated with binarized frame encodings.

Training GRU temporal feature module: The proposed model has only been trained and evaluated on the ImageNet VID dataset. 3,862 videos of the ImageNet VID training set have been used for training purposes. The temporal feature module uses three layers of a GRU module with 256 features. We unroll the GRU to 10 steps and train on sequences of 10 frames. The object detection backbone network is frozen; only GRUs with heads are trained. This helps in the overfitting problem. Figure 8 depicts the model configuration with the temporal feature module.

The detection network used in the proposed model is a single-stage object detector. It is developed by making several modifications to the backbone and YOLOv3 model. Mainly, a decoupled head is used instead of YOLO's head. For each level of FPN (Feature Pyramid Network) feature, a 1×1 conv layer is used to decrease the feature channel to 256. Moreover, two parallel branches are added with two 3×3 conv layers each for regression and classification.

TABLE I
ABLATION STUDY – MAP, PARAMETERS, AND LATENCY OF MODEL WITH INCREMENTAL FEATURES

Methods	mAP	Parameters	Latency
YOLOX-S backbone	34%	9M	10ms
Motion Encodings	39.2%	9M	10ms
GRU	44.3%	10.5 M	16 ms
Post-processing	45.1%	10.5 M	52 ms

TABLE II
COMPARISON OF MODEL WITH YOLOV5-S AND YOLOX-S DETECTION MODELS

Methods	mAP	Parameters	Latency
YOLOv5-S	36.7%	7.3M	9 ms
YOLOX-S	39.6%	9M	10ms
Proposed work	44.3%	10.5 M	16 ms
Proposed work + Post-processing	45.1%	10.5 M	52 ms

B. Detection Results and Comparative Analysis

The final model comprises a YOLOX-S backbone model with a GRU temporal feature module. The combination of frame and motion binarized encoding as input provides an mAP of 44.3%. We also employ sequence-based NMS as post-processing, with results in 0.8% mAP gain to 45.1%. Table I depicts the final model's mAP results with the total parameters and latency numbers.

Ablation Study: In an incremental approach, we study the effect of Motion encodings, temporal modules, and seq-NMS. Model parameters, latency, and mAP have been measured at each stage. Table I summarizes the results at each stage.

With YOLOX-S backbone on binarized frame encodings, we achieve 34% mAP. With the addition of motion encodings, mAP improves by 5.2% to 39.2. As it increases channel input of only the initial layer without affecting the rest of the model, latency and model size stay approximately the same. With the addition of a GRU-based temporal feature module, mAP improves to 44.3%. It increases the parameters by 1.5M and latency to 16ms from 10ms. Sequence NMS-based post-processing step further improves mAP to 45.1%.

Comparison with image detection methods: For real-time or faster analysis of videos, primarily frame-based fast object detectors are used. YOLOv5 small and YOLOX small are preferred object detectors in such scenarios. As the preference for detection over compressed video formats without decompression is also suitable in this scenario, we compare the results with YOLOv5-s and YOLOX-s models. Table II compares the results of the proposed model with these models.

The experimental results infer that as the proposed detection network is coupled with temporal features, it achieves significantly better mAP of 44.3 *w.r.t.* 36.7 and 39.6 from

YOLOv5-s and YOLOX-s models, respectively. Moreover, post-processing further improves the mAP to 45.1, but the latency time significantly increased to 52 ms. The proposed work results in better accuracy with marginal increased parameter size and latency overhead.

Comparison with specialized video detection methods: Specialized detection methods for video streaming have also been proposed to utilize temporal frame sequences more effectively. The proposed method includes warping of frame feature based on motion vector before aggregation, using tubelet proposal network to efficiently generate spatiotemporal proposals, using spatiotemporal-based attentions, identifying feature degradation before aggregation, using tracking methods, etc. With these specializations, this approach improves mAP to 80% and more [43]. All these networks employ a computationally expensive network to produce the per-frame features and achieve high detection performance.

The temporal feature modules and seq-NMS-based post-processing steps are also employed to showcase that these methods can be employed and improve our approach. As the main scope of this research is limited to the evaluation of deep learning compressed formats for detection tasks only, we did not exhaustively test and improve on these specialized tricks.

V. Conclusion

Video compression techniques are commonly found in modern consumer electronic devices. The advancements in computational power and compression efficiency have made the devices more lightweight and efficient. The proposed work is an experimental approach to explore the possibilities and evaluate the performance outcomes of deep detection models over deep learning-based compressed video streams for lightweight consumer electronic devices. The video object detection is carried out over the compressed video streams, which are compressed by an autoencoder-based deep network. This analytics approach is efficient as it utilizes more semantically compressed formats to avoid decompression overhead, making analytics faster. The experimental results show competitiveness and improvement in the detection outcomes and also motivate exploring enhanced and efficient deep models for video stream specialized approaches, ultimately resulting in practical, applicable models for consumer electronic devices.

The future of object detection from compressed video formats is poised for significant growth and innovation, driven by technological advancements, increased demand for real-time applications, and the need for more efficient and privacy-aware solutions. Developing such algorithms will be crucial for real-time processing and reducing computational overhead. The demand for real-time object detection from compressed video will continue to grow in various domains, including surveillance, autonomous vehicles, and robotics. Improved algorithms and hardware acceleration techniques will be required to meet these applications' stringent latency and performance requirements. As edge computing capabilities grow, object detection from compressed video can be performed directly on edge devices, reducing latency and bandwidth usage. This is especially valuable for applications like smart cameras and autonomous vehicles. Training object detection models on compressed video data with limited labeled samples will be essential. Semi-supervised and self-supervised learning techniques will reduce the annotation burden and make object detection more accessible for various applications. Handling challenging conditions, such as low-light environments, occlusions, and adverse weather, will be essential for object detection from compressed video. Research in making object detectors more robust to these conditions will continue to advance.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[2] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[3] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[5] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[6] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99,

[8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[9] K. Kang et al., "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2896–2907, Oct. 2018.

[10] S. Wang, Y. Zhou, J. Yan, and Z. Deng, "Fully motion-aware network for video object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 542–557.

[11] F. Xiao and Y. J. Lee, "Video object detection with an aligned spatial-temporal memory," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 485–501.

[12] X. Zhu, J. Dai, X. Zhu, Y. Wei, and L. Yuan, "Towards high performance video object detection for mobiles," 2018, *arXiv:1804.05830*.

[13] A. Dosovitskiy et al., "Flownet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2758–2766.

[14] S. Yadav, P. Gulia, and N. S. Gill, "Flow-MotionNet: A neural network based video compression architecture," *Multimed. Tools Appl.*, vol.81, pp. 42783–42804, Dec. 2022, doi: 10.1007/s11042-022-13480-0.

[15] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 331–346.

[16] K. Kang et al., "Object detection in videos with tubelet proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 727–735.

[17] K. Chen et al., "Optimizing video object detection via a scale-time lattice," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7814–7823.

[18] M. Liu and M. Zhu, "Mobile video object detection with temporally-aware feature maps," 2021, *arXiv:1711.06368*.

[19] G. Van Wallendael, S. Van Leuven, J. De Cock, F. Bruls, and R. Van de Walle, "3D video compression based on high efficiency video coding," *IEEE Trans. Consum. Electron.*, vol. 58, no. 1, pp. 137–145, Feb. 2012.

[20] M.-J. Kim and Y.-L. Lee, "Object detection-based video compression, *Appl. Sci.*, vol. 12, no. 9, p. 4525, 2022, doi: 10.3390/app12094525.

[21] Y. Zhang, L. Yu, J. Lee, M. Rafie, and S. Liu, "Draft use cases and requirements for video coding for machines," ISO/IEC, Geneva, Switzerland, document N133 of ISO/IEC JTC 1/SC 29/WG 2, Oct. 2021.

[22] "YOLO: Real-time object detection." YOLO Website. Accessed: Jan. 1, 2023. [Online]. Available: https://pjreddie.com/darknet/yolo/

[23] W. Gao, S. Liu, X. Xu, M. Rafie, Y. Zhang, and I. Curcio, "Recent standard development activities on video coding for machines," 2021, *arXiv:2105.12653*.

[24] M. Rafie, Y. Zhang, and S. Liu, "Evaluation framework for video coding for machines," ISO/IEC, Geneva, Switzerland, document N134 of ISO/IEC JTC 1/SC 29/WG 2, Oct. 2021.
[25] W. Yang, H. Huang, Y. Hu, L.-Y. Duan, and J. Liu, "Video coding for machine: Compact visual representation compression for intelligent collaborative analytics," 2021, *arXiv:2110.0924*.
[26] K. Fischer, F. Brand, C. Herglotz, and A. Kaup, "Video coding for machines with feature-based rate-distortion optimization," in *Proc. IEEE 22nd Int. Workshop Multimed. Signal Process.*, Sep. 2020, pp. 21–24.
[27] L. Jiao et al., "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.
[28] L. Jiao et al., "New generation deep learning for video object detection: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3195–3215, Aug. 2022, doi: 10.1109/TNNLS.2021.3053249.
[29] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 10334–10343, doi: 10.1109/CVPR42600.2020.01035.
[30] L.-Y. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: A paradigm of collaborative compression and intelligent analytics," 2020, *arXiv:2001.03569v2*.
[31] J. Ali, M. Adnan, T. R. Gadekallu, R. H. Jhaveri, and B.-H. Roh, "A QoS-aware software defined mobility architecture for named data networking," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 444–449, doi: 10.1109/GCWkshps56602.2022.10008563.
[32] R. C. Meena et al., "Enhancing software-defined networks with intelligent controllers to improve first packet processing period" *Electronics*, vol. 12, no. 3, p. 600, 2023, doi: 10.3390/electronics12030600.
[33] Q. Z. Alam, I. Haque, and N. Abu-Ghazaleh, "Learn to compress (LtC): Efficient learning-based streaming video analytics," 2023, *arXiv:2307.12171v2*.
[34] R. Strukov and V. Athitsos, "Evaluation of video compression methods for network transmission on diverse data: A case study," in *Proc. 16th Int. Conf. Pervasive Technol. Related Assistive Enviro.*, Jul. 2023, pp. 300–305.
[35] N. Mandischer, T. Huh, M. Husing, and B. Corves, "Efficient and consumer-centered item detection and classification with a multicamera network at high ranges," *Sensors* vol. 21, no. 14, p. 4818, 2021, doi: 10.3390/s21144818.
[36] J. Lee and S. Lee, "Fast and energy-efficient object detection and tracking for de-identification in video," in *Proc. IEEE Int. Conf. Consum. Electron.-Asia (ICCE-Asia)*, 2022, pp. 1–2, doi: 10.1109/ICCE-Asia57006.2022.9954651.

Sangeeta Yadav received the B.Tech. and M.Tech. degrees from the Department of Computer Science and Engineering, DCRUST, Murthal, India, and the Ph.D. degree from the Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak, India, where she was a Senior Research Fellow. She is currently working with the Ch. Ranbir Singh State Institute of Engineering and Technology, Jhajjar, India. Previously, she had also worked as an Assistant Professor with the Central University of Haryana, Mahendergarh, India, for 4.5 years. She is a UGC NET-JRF Jan-2017 qualified and GATE-2011, 2012, and 2013 qualified scholar. She has authored many research papers of international indexing. Her area of research includes machine learning, artificial intelligence, and deep learning.

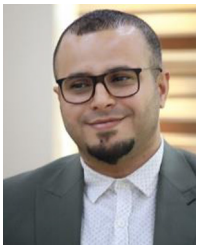
Preeti Gulia received the Doctoral degree in 2013. She is currently working as an Assistant Professor with the Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak, India, where she has been serving the department since 2009. She has published more than 65 research papers and articles in journal and conferences of national/international repute, including ACM and Scopus. Her area of research includes data mining, big data, machine learning, deep learning, IoT, and software engineering. She is an active Professional Member of IAENG, CSI, and ACM. She is also serving as an editorial board member active reviewer of international/national journals. She has guided one research scholar as well as guiding four Ph.D. research scholars from various research areas.

Nasib Singh Gill is currently working as a Professor and the Head of the Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak, India. He has more than 27 years of teaching experience. His publications comprises of Books—4 (Independent), 1 (Joint); Courseware: 6 Courses Papers: 237, published in international/national journals and Books: 149 2, published in Intl. Conf. Proc., Mag., etc.: 28, Published in National Conf. Proceedings: 60, Papers Reviewed: 12. He was the Commonwealth Fellow for the year 2001–2002 (01.10.2001–30.09.2002) and awarded the Commonwealth Fellowship Award by ACU, London at Brunel University, U.K. He got the Best Paper/Article Award by the Computer Society of India in 1994.

Ishaani Priyadarshini received the bachelor's degree in computer science engineering and the master's degree in information security from the Kalinga Institute of Industrial Technology, India, and the master's degree in cybersecurity and the Ph.D. degree from the University of Delaware, USA. She works as a Lecturer with the School of Information, University of California at Berkeley, Berkeley, USA. She has authored several book chapters for reputed publishers and is also an author to several publications for SCIE indexed journals. Her areas of research include cybersecurity, artificial intelligence, and HCI. As a certified reviewer, she conducts peer review of research papers for prestigious IEEE, Elsevier, and Springer journals and is a part of the editorial board for *International Journal of Information Security and Privacy*.

Rohit Sharma (Senior Member, IEEE) working as a Dean (Research) with ABES Engineering College Ghaziabad, India. He serves as a Book Editor for seven different titles to be published by CRC Press, Taylor & Francis Group, USA, and Apple Academic Press, CRC Press, Taylor & Francis Group, USA, and Springer. He has received the Young Researcher Award in the "2nd Global Outreach Research and Education Summit & Awards 2019" hosted by the Global Outreach Research and Education Association. He is serving as a Guest Editor in the SCI journal of *Computers and Electrical Engineering* (Elsevier). He is an Editorial Board Member and a Reviewer of over 12 international journals and conferences, including the topmost journal IEEE ACCESS and IEEE INTERNET OF THINGS JOURNAL. He has actively been an organizing end of various reputed international conferences. He is an Active Member of ISTE, ICS, IAENG, and IACSIT.

Kusum Yadav is currently working with the Department of Information and Computer Science, College of Computer Science and Engineering, University of Ha'il, Saudi Arabia. She has more than 15 years of experience in academic and research in addition to three years of industry experience. She has published several papers in SCI/Scopus/High impact factored journals, international conferences, and book chapters. She has 15 published patents and nine granted copyrights. Her research interests include big data, cyber security, machine learning, soft computing, data mining, hybrid systems, and feature selection. She has guided and is guiding M.Sc. (Artificial Intelligence) students and Ph.D. scholars. She has served as the session chair, a member of the advisory/technical program committee, and a reviewer for many international/national conferences. She has organized many events like faculty development programs, hackathons, and workshops.



Ahmed Alkhayyat received the B.Sc. degree in electrical engineering from AL KUFA University, Najaf, Iraq, in 2007, the M.Sc. degree from the Dehradun Institute of Technology, Dehradun, India, in 2010, and the Ph.D. degree from the University of Cankaya, Ankara, Turkey, in 2015. He contributed in organizing several IEEE conferences, workshop, and special sessions. He is currently the Dean of international relationship and a Manager of the world ranking with Islamic University, Najaf. His research interests include AI, machine learning, deep learning, security, IoT healthcare based, network coding, cognitive radio, efficient-energy routing algorithms and efficient-energy MAC protocol in cooperative wireless networks and wireless local area networks, as well as cross-layer designing for self-organized network. He acted as a reviewer for several journals and conferences.