IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1473~1485
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1473-1485

Scaler enhanced deformable attention with graph neural network for video compression

Revathi Kasinathaperumal[1], Hosanna Princye Periapandi[2]
[1]Department of Electronics and Communication Engineering, S.E.A. College of Engineering and Technology, Visvesvaraya Technological University, Belagavi, India
[2]Department of Electronics and Communication Engineering, Sri Sairam College of Engineering, Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Aug 1, 2025
Revised Jan 13, 2026
Accepted Jan 25, 2026

Keywords:

Constrained directional enhancement filter
Graph neural network
High efficiency video encoding
Scaler enhanced deformable attention
Video compression
Video quality

ABSTRACT

Video compression is widely used to reduce bandwidth and storage requirements when storing and transmitting videos, most existing neural video compression approaches adopt the predictive residue-coding framework, which is suboptimal for removing redundancy across frames. Additionally, minimizing only the pixel-wise differences between the raw and decompressed frames is ineffective in improving the perceptual quality of the videos, blocking artifacts degrade the visual quality, especially near edges and texture areas. Hence, to solve these problems, this research proposes a scaler enhanced deformable attention graph neural network (SEDA-GNN) to utilized for reduce inter-frame redundancy by employing a deformable attention mechanism that efficiently captures motion and structural changes, thereby minimizing redundancy. Modelling complex temporal dynamics with graph neural networks (GNNs) captures dependencies between frames, thereby facilitating highly efficient video encoding, then constrained directional enhancement filter (CDEF) effectively reduces blocking artifacts while preserving sharp edges through directional and constrained filtering, thereby improving visual quality in compressed video. The SEDA-GNN approach achieved a bjontegaard delta bit rate (BD-BR) reduction of 2.372% on the joint collaborative team on video coding (JCT-VC) database and 3.230% of BD-BR on the ultra video group (UVG) dataset, demonstrating significant performance when compared to invertible neural networks (INNs).

Corresponding Author:

Revathi Kasinathaperumal
Department of Electronics and Communication Engineering, S.E.A. College of Engineering and Technology
Visvesvaraya Technological University
Belagavi, Karnataka, India
Email: revathiselvaraj1229@gmail.com

1. INTRODUCTION

Video encoding, also known as video compression, enables the significant transmission and storage of video content. Standard of video coding follow a strategy which utilizes advanced methods like intra-frame prediction, variable block sizes, prediction of inter-frame, quantization, entropy, and transfer coding to effectively enhance compression efficiency [1]. The high-efficiency video coding (HEVC) standard was launched by the joint collaborative team on video coding (JCT-VC) to differentiate classes based on video quality. Video compression standards have advanced, including advanced video coding (AVC), HEVC, versatile video coding (VVC), and the enhanced compression model (ECM), to deliver higher-quality video with less bit consumption [2], [3]. With the rise of metaverse-enabled online communication, interactive

video coding has made sustainable progress achieving low latency However, an ultralow delay is caused by high-volume data in video communication with raw signals [4]. HEVC/H.265 has increasingly struggled to meet the demands of streaming platforms as a widely adopted digital medium video is highly valued for delivering dynamic and immersive visual experiences [5]. However, producing a video that effectively conveys the idea of aesthetic quality is often more expensive and complex, the video compression method uses an autoencoder-style network for encoding and decoding. Although an autoencoder can capture important information for reconstruction, any neglected information lost during encoding cannot be recovered during decoding, which leads to irreversible degradation [6], [7]. The deep contextual video compression (DCVC) approach is presented for conditional coding-based deep video compression, in DCVC, a valuable context is extracted as a condition to compress the current frame and improve the compression efficiency [8]. The partition structure and intra prediction are performed using the maximum time in VVC intra coding and brute-force recursive (rate/distortion optimization or RDO) search, that determines the optimal partition and choose the best intra-prediction model [9]. In a semantic communication system, raw images are converted into semantic features at the transmitter and then decoded at the receiver to reconstruct the decompressed image. For task-execution applications, only task-specific semantic information is extracted and encoded at the transmitter [10]. In all intra-configurations, every frame is independently encoded by considering an image compression approach, which treats each image as a standalone entity to exploit spatial redundancy within a single frame that is decoded without references to others, enhancing robustness [11]. Traditional methods fail to exploit temporal redundancy, similarity among consecutive frames, and key factors in video coding efficiency, by ignoring the inter frame correlations that avoid significant opportunities to decrease bitrate and enhance effectiveness in compression [12].

The existing deep neural network-based auto encoder approaches are applied to video compression, where optimizing rate distortion tradeoffs to reduce bit rate while preserving reconstructed frame quality. These approached used for non-linear, adaptive representation which is more flexible to capture complex spatial variations across frames [13], [14]. The traditional encoding techniques helps to predict spatiotemporal relationship among block regions in consecutive frames to improve compression efficiency and accuracy of the model performance to reduce minimize residual block error [15]. The high frequency embedding approaches are utilized for frame reconstruction, and also various techniques like adaptive quantization factors. Machine learning (ML) and deep learning (DL) techniques helps to improve the efficiency and compression video quality of the data [16], [17]. These methods solve the challenges like large and complex motions in the videos and in video pushing learned variety of different motion patterns which is further away from practical usability in consumer devices. The using of generic framework to control the scale motion vectors on per frame basis to approximately match the range of motion in the training video [18], [19]. A several researches facing challenges in predicting coding to eliminate the short-term inter-frame and intra-frame redundancies that lead to poor efficiency in compression, existing methods are not effective to generating long term background from the noise [20]. Du *et al.* [21] implemented contextual generative video compression with transformers (CGVC-T) that adopted a generative adversarial network (GAN) to improve coding efficiency by using contextual coding and enhance perceptual quality. GANs reconstruct finer details and manage high visual quality at lower bitrates by enabling higher compression efficiency and reduced bandwidth usage. However, the GAN model was trained by minimizing the mean square error (MSE), which affected the yield in the smoothed frames and led to unsatisfactory perceptual quality. Sheng *et al.* [22] developed a temporal context mining (TCM) approach that propagated features before reconstructing the frame, and these features were stored in a generalized decoded picture buffer. Multi-scale temporal contexts helped to extract multi-scale temporal contexts from the propagated features, which enhanced the accuracy of the temporal information. The temporal context refilling (TCR) method learns the temporal content which is combined into the compression scheme. This involves a contextual encoder-decoder, frame generator, and temporal context encoder. The parallelization-friendly encoding approach eliminates use of an autoregressive entropy model to achieve practical encoding and decoding times. However, the TCM method faced a struggle depending on accurate feature propagation, which caused errors when the method was not properly handled.

Guo *et al.* [23] implemented an invertible encoding-decoding network for deep-video compression. Invertible neural networks (INNs) utilized to preserve both spatial and temporal information. INN is primarily applied in residual nonlinear transformations to enhance the representation power of a network. Invertible encoding helped to manage motion information and reduce artifacts in compressed videos. However, INN faced difficulty in fully optimizing contextual encoding; while motion compression increased, the exploration of contextual encoding lagged. Thang and Bang [24] developed a hierarchical random access coding method which reduced bidirectional temporal redundancy to enhance the significant coding of the traditional deep neural video compression approach. The method applied an interpolation network video frames to enhance the prediction of the inter-frame. The proposed method improved inter-prediction prediction, thereby generating intermediate frames that reduced residual errors. In the optimized bitrate

allocation, frames at higher levels in the group of pictures (GoP) were assigned lower bitrates because they were not used as reference frames. However, the proposed framework mostly focused on the configuration of delay, where the order of frames matches their encoding order, which affects the compression efficiency of the video sequences. Lu *et al.* [25] implemented an end-to-end deep video compression framework, which was utilized for pixel-wise motion information learned from an optical flow network and further compressed by an auto-encoder network to save bits. In rate-distortion optimization, all modules were jointly trained using a single loss function to balance compression efficiency and video quality. An adaptive quantization layer reduced the number of parameters required for variable bitrate coding. Residual compression uses deep networks to transform and encode residual information efficiently. However, the block-based method introduces inaccurate motion information and thus degrades the compression performance. This research has various existing challenges, such as traditional approaches that face many problems, such as poor perceptual quality, which is caused by smoothed frames. The method has difficulty performing feature propagation accurately, which affects the overall model performance. Improper motion information reduces the compression performance. When the motion compression increased, the model faced problems and degraded the system performance, the contributions of this work are as follows:

- The scaler enhanced deformable attention graph neural network (SEDA-GNN) consists of deformable attention mechanisms to dynamically integrate and align information across frames, which efficiently captures motion and structural changes to minimize inter-frame redundancy, attention modules that focus on preserving visually significant features, SEDA-GNN, improve the visual fidelity of compressed videos and address the limitations of pixel-wise loss functions.
- The constrained directional enhancement filter (CDEF) effectively reduces blocking artifacts while preserving sharp edges, and its directional and constrained filtering approach primarily targets ringing artifacts across detected edge directions, thereby improving the visual quality in compressed video.
- SEDA-GNN models capture complex temporal dependencies among frames to provide more efficient and accurate video compression. By dynamically modelling non-local spatio-temporal relationships based on motion cues, it efficiently improves the video quality.

This research is organized as follows. Section 2 describes the research method. Section 3 represents the results and discussion. The section 4 concludes the research.

2. RESEARCH METHOD

In this research, the SEDA-GNN method minimizes the redundancy and pixel-wise differences among the raw frames and enhances the videos perceptual quality and compression process to improve the compression video quality. The proposed SEDA-GNN approach helps to determine the final coding unit (CU) size and reduced the CU side thereby improving overall model performance. The CDEF method significantly decreases the blocking artifacts by preserving sharp edges and texture, Figure 1 represent the flowchart of the proposed method.

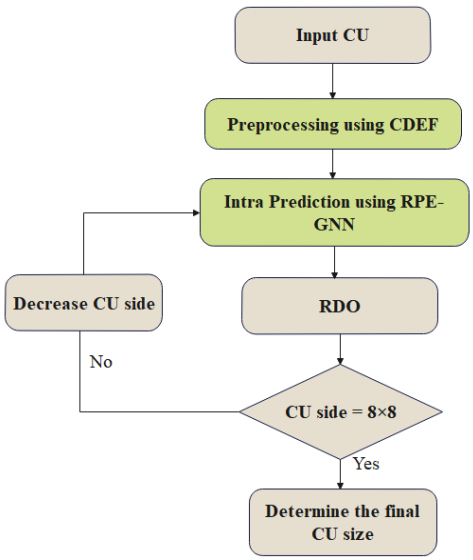

Figure 1. The flow chart of proposed SEDA-GNN

2.1. Dataset description

The ultra video group (UVG) [26] dataset with (1920×1080) HD video sequences high, the high efficiency video quality (HEVC) class B with full-HD videos is 5 with resolution of video sequences (1920×1080). HEVC class C with 4 video sequences having 832×480 resolutions. HEVC class D has 4 video sequences with a resolution of 416×240 resolution, with a YUV420 raw format available with different frame rates from 24 frames per second to 120 fps. Where Y represents the luma component (brightness) and U and V are the chroma components (color information).

The JCT-VC [27] dataset is a group of video coding experts from the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) study group 16 video coding experts group (VCEG), and the International Organization for Standardization (ISO)/ International Electrotechnical Commission (IEC)/Joint Technical Committee 1 (JTC) 1/Subcommittee (SC) 29/ Working Group (WG) 11. The moving picture experts' group (MPEG) is a new-generation standard for creating video coding in 2010, for high-quality video coding minimizes the amount of data required by approximately 50%. The database consists of 34 videos, which are divided into training 14 videos, 6 for validation, and testing 14 videos. Training and testing of randomly selected videos of different resolutions to improve efficiency and assess performance.

2.2. Preprocessing

In this phase, a CDEF is utilized to eliminate artifacts such as ringing around hard edges. Therefore, applying two directional filters it is achieved (45° off) for every pixel, proper filter selection is performed according to the optimization and is based on the minimization of (1). Where the group of pixels in a selected direction is P and the mean of the group p is μ, CDEF is an in-loop restoration filter that is applied to loop restoration units (LRU) with 64×64, 128×128, or 256×256-pixel blocks, every LRU independently chooses the restoration option from one of the three possibilities.

$$E_d^2 = \sum_k \sum_{p \epsilon p_{d,k}} (x_p - \mu_{d,k})^2 \quad (1)$$

2.3. CU partitioning

The inputs gathered and fed into a structured neural network, which trained using a specific loss function, finally HEVC is integrated into the trained model, and the original traversal search process is replaced. The developed GNN needs to go through the metrics of the three layers, which are correlated to the global mean square residual (GMSR), the overall texture complexity is initially evaluated by computing the RMSE, the complexity of the local texture is measured as RMSE $\leq TH_A$. If GMSR $\leq TH_B$, the CU is smooth, and in decision-making, no small size is partitioned, GNN is used to determine whether further partitioning is required to determines CU texture complexity during fast partitioning and selects a better CU partition by considering three metrics.

2.4. Graph neural network

The traditional methods faced difficulties with non-Euclidean data, an GNN is proposed for graph-structured data with a better scope for video compression. A GNN consists of a graph convolutional network (GCN), graph attention networks (GAN), and graph auto-encoders (GAE) for compression. In video summarization, every video is presented as a graph where features are frames that are treated as graph nodes and the relationships among frames are encoded as edge weights which effectively capturing the temporal and spatial dependencies between the video frames, Figure 2 represents the architecture diagram of SEDA-GNN approach.

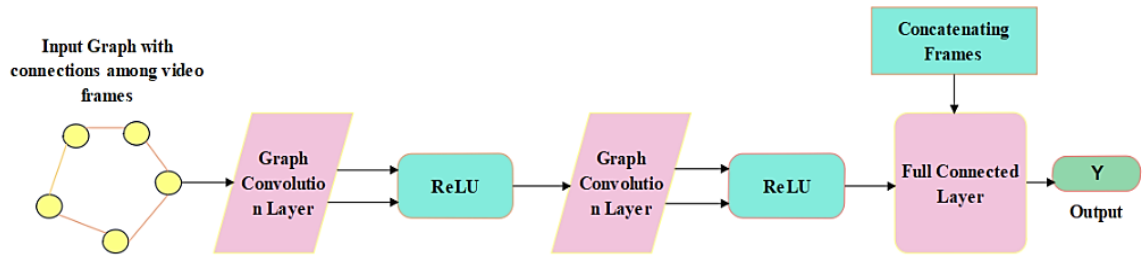

Figure 2. Architecture diagram of SEDA-GNN method

The introduction of GNN has been successful and efficient in the domain of video summarization, Figure 2 the architecture diagram of the proposed method, a video frame input is given as input graph and their temporal connections. A graph is processed via two graph convolution layers, every layer is followed by rectified linear unit (ReLU) activation function to extract spatio-temporal feature levels, subsequently the features from all frames are concatenated and forwarded through a fully connected layer (FCN). Finally, it provides Y as an outcome that represents compressed video information. Spectral graph convolution is adopted by the original GCN, which is defined in (2).

$$g_\theta * x = U g_\theta U^T \tag{2}$$

Where the normalized Laplacian matrix L of the eigenvector graph matrix is U; thus, $L = U\lambda U^T x$, and x is the input data, trainable convolutional kernel parameter is $g_\theta = diag(\theta)$. Because of the large number of parameters and computational complexity, an advanced GCN was calculated using (3).

$$g_\theta * x = \sum_{j=0}^{k-1} \theta_j L^j x \tag{3}$$

To address the issue of high computational costs, two approximation approaches are utilized, Chebyshev convolutional kernel: solving high multiplicative complexity problem of wide dense matrices, approximate are provided using Chebyshev polynomials is $g_\theta(\Lambda)$. The formula for this approximation is given by (4):

$$g'_\theta(\Lambda) \approx \sum_{k=0}^{k} \theta'_k T_k(\tilde{\Lambda}) \tag{4}$$

Where $\tilde{L} = \frac{2}{\lambda_{max}} L - I_N$, this formula is $o(|\epsilon|)$ a computationally complexity and depends on the number of edges in the graph central node first-order neighbors: problems are simplifying and alleviate overfitting because of local node in large size. A set K =1, meaning which only central node and its first-order proximity are substituted, accordingly, $T_0(x) = 1, T_1(\tilde{L}) = \tilde{L}$ and λ_{max} to 2, the convolution with the new approximate formula is given in (5). Where the eigenvalue range of $I_N + D^{-\frac{1}{2}} A D^{\frac{1}{2}}$ is [0, 2], gradient disappearance and multilayer convolution lead to the gradient explosion given in (6), where $\tilde{A} = A + I_N, \tilde{D} = \sum_j \tilde{A}_{ij,}$ the eigenvalue range is normalized among [0, 1], substituting (6) into (7).

$$g_\theta * x \approx \theta'_0 x + \theta'_1 (L - L_N) x = \theta (I_N + D^{-\frac{1}{2}} A D^{\frac{1}{2}}) x \tag{5}$$

$$I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \tag{6}$$

$$g_\theta * x = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \tag{7}$$

2.5. Scale-enhanced deformable attention

Scale-enhanced deformable attention (SEDA) has been proposed, which combines scale aggregation, dilated sampling, position updating, attention calculation, and feature integration mechanism that helps to improve instance awareness across diverse scales and shapes. Scale aggregation collects multi-resolution features; dilated sampling uses learned offsets to sample relevant points at different scales. Position updating adjusts reference points dynamically, attention calculation computes weights over sampled positions, and feature integration fuses the results into richer representations.

Scale aggregation: let X is an input feature of the SEDA, according to the convolutional multi-head self-attention (MHSA), the value is generated $V = [V_1, V_2, \ldots V_M] \epsilon R^{M \times \frac{C}{H} \times H \times W}$ from X using linear projection, Where, $V_M \epsilon R^{M \times \frac{C}{H} \times H \times W}$ is the $m - th$ $(m = 1,2, \ldots, M)$ head. In the convolutional MHSA framework, all the attention heads operate on the same scale. Consequently, the inter-level multiscale property cannot be effectively exploited, which leads to a limited multiscale representation for each feature map. To solve this issue, the principle of a dilated transformer is adopted in which each head sample feature point uses a different dilation rate, however the original dilated transformer failed to facilitate interactions between different scales.

Therefore, developing an aggregation approach that consolidates M heads into R(R<M) scale heads through dynamic selection weights $\{\pi_r\}^R r = 1$ is predicted from the value through global average pooling (GAP), a linear layer, and SoftMax activation. The computations of $\{\pi_r\}^R r = 1$ are summarized as in (8) and let $\tilde{V}_r \in R^{M \times \frac{C}{H} \times H \times W}$ be the value for the $r - th$ scale-head; then, $\tilde{V}_r (r = 1,2, \ldots, R)$ is produced by (9),

Where $\pi_{r,M}$ is the $m-th$ element of π_r. Subsequently, SEDA is applied to the R scale heads $\{\tilde{V}^r\}^R r=1$ respectively, that showing the dynamic presentation of activation functions.

$$\{\pi_r\}^R r=\sigma(Linear(GAP(Linear(X)))) \tag{8}$$

$$\tilde{V}_r=\pi_r \otimes V=[\pi_r, V_1, \pi_{r,2} V_2, \ldots ., \pi_{r,M}, V_M] \tag{9}$$

Dilated sampling: for different-scale heads, different dilation rates are used to sample the entries of the value at the target query point, without loss of generality, the dilation rate r is assigned to the $r-th\ head\ (r=1,2,R)$, given the query position $P_i \in R^2$ of the $i-th$ query. The value sampling strategies for different scale heads are defined as follows: for $r=1$, the value points are sampled within a standard $K \times K$ window centered at P_i, for $r \geq 2$, the sampling positions are determined by applying dilation rate r to the sampling grid. In short dilated sampling set $\Gamma_{r,i}$ for query point $q_{r,i}$ at the $r-th$ scale-head is given by (10), additionally, at the boundary of feature map, zero-padding is used to assist dilated sampling.

$$\Gamma_{r,i}=\{p_{r,j} | p_{r,j}=p_i+r.\omega, \omega \epsilon[-\tfrac{k}{2}, \tfrac{k}{2}]^2\} \tag{10}$$

Position update: the strategy in (10) samples feature points within a regular $K_r \times K_r$ window for the $r-th$ head where $K_r=K+(K-1)(r-1)$, although basic dilated sampling enlarges the receptive field it lacks the flexibility to accurately model the individual instances. To address this limitation, a set of learnable position offsets is introduced to update the sampled positions adaptively thus the sampled positions are updated as (11), where $\Delta p_{r,j} \in R^2$ denotes the offset for positions $p_{r,j}, \phi$ denotes bilinear interpolation, following DAT. The locations $p_{r,j}$ are also normalized to the range [0, 1] the offset $\Delta p_{r,j} \in [0, 1]$ is predicted from the $i-th$ query using a linear layer that is $\{\Delta p_{r,j}\}=Linear\ (q_r, i)$ hence for the $i-th$ query at location $p_{r,j}$, the value points are selected from the set $\Gamma_{r,i}$ to calculate the attention at the $r-th$ scale head.

$$\tilde{\Gamma}_{r,i}=\{\phi(p_{r,j}+\Delta p_{r,j}), p_{r,j} \in \Gamma_{r,i} \tag{11}$$

Attention calculation: based on (8) to (10), the $j-th$ value point for the $i-th$ query at the $r-th$ scale-head, where $\tilde{V}^r=(\phi(p_{r,j}+\Delta p_{r,j}))\ p_{r,j} \epsilon \Gamma_{r,i}$, instead of using dot product similarity, as in typical self-attention. The attention weights are predicted directly from the query using a linear layer and SoftMax function for calculation. Let $A_{r,i,j}$ be the attention weight between the $i-th$ query and the $j-th$ key at the $r-th$ scale head $\{A_{r,i,j}\}^{|\Gamma_{r,i}|}{}_{j=1}$ is obtained by (12), were $|\Gamma_{r,i}|$ components the power of set $\Gamma_{r,i}$ then the attention of the $r-th$ scale-head is calculated as (13).

$$\{A_{r,i,j}\}_{j=1}{}^{|\Gamma_{r,i}|}=\sigma(Linear(q_{r,i})) \tag{12}$$

$$o_{r,i}=\sum_{j=1}^{|\Gamma_{r,i}|} A_{r,i,j}\, \tilde{V}^r(\phi(p_{r,j}+\Delta p_{r,j})) \tag{13}$$

Where $o_{r,i}$ denotes the $i-th$ element in the output of the $r-th$ scale-head ss shown in (12), every scale head in SEDA operates on the complete input feature, X. This operation is different from a typical MHSA, in which each head operates on a partition of the input feature to calculate scale head. Therefore, the dilation rate defined in (13) can be set to any value without the dimensional mismatch issue that may occur in a typical MHSA.

Feature integration: the final output of SEDA is produced through the feature integration of all scale heads specifically, it is operated through the concatenation operation followed by a linear layer WO, which is formulated as (14). Where l has the same definition as in (15) and the attention weights $\left(\sum_{l=1}^{L}\sum_{j=1}^{|\Gamma_{r,i}|} A_{r,i,j}\, \tilde{V}^r(\phi(\varphi_1(p_{r,j}+\Delta p_{r,l,j}).Wo_r=1\right.$ recalling the overall architecture, the SEDA is equipped with the self-attention of the encoder and cross-attention of the decoder where $W_o=[W_{o1}, W_{o2}, \ldots ., W_{oR}]$ is the split of W_o according to the dimensions of $\{o_{r,i}\}^R r=1$. The difference between these two SEDA arrangements is that the queries are different specifically queries in the self-attention of the encoder are from image features, whereas queries in the decoder cross-attention originate from object queries.

$$z_i=Concat\ (O_1, i, O_2, i \ldots ., O_R, i).W_o=\sum_{r=1}^{R} O_{r,i}\,.W_{or} \tag{14}$$

$$z_i = \sum_{r=1}^{R} \left(\sum_{l=1}^{L} \sum_{j=1}^{|\Gamma_{r,i}|} A_{r,i,j} \, \tilde{V}^r (\phi \left(\varphi_1 \left(p_{r,j} + \Delta p_{r,l,j} \right) \right) . Wo_r \quad (15)$$

2.5.1. Algorithm for proposed SEDA-GNN

Input: GNN model f_θ with parameter θ and the training dataset is UVG and JCT-VC and rate-distortion and perception loss are considered, which are represented as $\mathcal{L}(.)$, 100 epochs were included, which is shown as T, output: trained model parameter θ. The training loop updating θ iteratively over 100 epoches thereby minimizing the loss tradeoff among compression rate, perceptual quality and distortion quality. The final trained model parameters θ which balances best when competeing losses across both datasets as shown in Algorithm 1.

The proposed GNN approach used SEDA by considering (4) adaptively samples across dilated rates and multiple scales, dynamically update their position according to learned offsets. This provides model to attend effectively across time and space for salient video features effectively. By (7) quantifies the loss function quality of video compression to calculate rate of distortion. The GNN model helps to predict CU partitioning with lower complexity and it leads to efficient encoding by enhancing quality of video.

Algorithm 1. GNN training procedure
1: procedure TrainGNN (f θ, data training, $\mathcal{L}$, optimizer, T)
2: for epoch = 1 to T do
3: for each batch $\mathfrak{P}$ in $\mathcal{D}_{\_train}$ do
4: (X, A, Y) ← ExtractFeatureAndLabels ($\mathfrak{P}$) # node features X, adjacency A, labels Y
5: Optimizer. zero_grad () # clear previous gradients
6: $\hat{Y}$ ← f θ(x, A) using (4) # forward pass through the GNN
7: Loss ← $\mathcal{L}$ ($\hat{Y}$, Y) using (7) # compute loss
8: Loss. backward () # backpropagate gradients
9: Optimizer. step () # update model parameters
10: end for
11: end for
12: return θ
13: end procedure

2.6. Notation table

The notation table summarizes the all variable and their symbols used for the video compression; each notation defines the parameters related to the transfer coefficient and video frames motion estimation. The encoding process like pre-processing, CU partitioning, prediction and entropy coding represented by its related symbols. The video compression mathematical representations are provided in notation table which provides the clarity and consistency to understand in Table 1. Notation and their descriptions used in the video compression.

3. RESULTS AND DISCUSSION

The SEDA-GNN approach is simulated with MATLAB 2020a environment, and, windows 10 OS, i5 processor, and 16 GB RAM are the required system configurations. The SEDA-GNN performance metrics used for evaluating model performance are bjontegaard delta peak-to-signal noise ratio (BD-PSNR), multi-scale – structural similarity index measure (MS-SSIM) and bjontegaard delta bit rate (BD-BR) with percentage of coding time (ΔT), the mathematical expressions are provided in (16) to (18). Where T_{HM} and T_{Pro} denote the encoding time, $PSNR_{HM}$ and $PSNR_{Pro}$ are the PSNR ratio, and $BD_{rate_{HM}}$, $BD_{rate_{Pro}}$ define the bitrate in the HM output of the proposed method, the rate difference is defined as $\Delta D = Dh - Dl$, where Dh and Dl presents the low and high points of the curve range of RD, as presented in l also, the bit rates are r_{actual} and $r_{proposed}$ of SEDA-GNN and the actual method.

$$\triangle T_{Avg} = \frac{1}{4} \sum_{i=1}^{4} \frac{T_{Pro}(QP_i) - T_{HM}(QP_i)}{T_{HM}(QP_i)} \times 100 \quad (16)$$

$$BD - PSNR = PSNR_{Pro} - PSNR_{HM} \quad (17)$$

$$BD - BR = \frac{BD_{rate_{Pro}} - BD_{rate_{HM}}}{BD_{rate_{HM}}} \times 100 \quad (18)$$

3.1. Performance analysis

Table 1 represents the UVG and JCT-VC dataset class distributions with the resolution sizes and different sequences. The JCT-VC database video sequences are divided into various resolutions, with each class having a different resolution. Table 2 provides a detailed explanation of the resolution distributions and sequences and presents the resolution of the dataset classes.

Table 1. Summarizes the symbols and variables used in the video compression process

Symbol	Description
E_d^2	Directional error metrics for filter selection in CDEF
k	Index over directional filters or groups
$p \epsilon p_{d,k}$	Pixel p belonging to the pixel group P in direction d, group k.
x_{p}	Intensity or value of pixel p.
$\mu_{d,k}$	Mean intensity of pixels in group $p_{d,k}$
$CDEF$	Constrained directional enhancement filter.
LRU	Loop restoration unit—processes blocks (e.g., 64×64, 128×128, 256×256).
CU	Coding unit (in HEVC partitioning).
$RMSE$	Root means square error—global texture complexity metric.
$GMSR$	Local texture complexity measure.
$TH_{\mathrm{a}}, TH_{\beta}$	Thresholds A and B for texture complexity decision-making.
GNN, GCN, GAN, GAE	Graph neural network; graph convolutional network; graph attention network; graph auto-encoder.
$L, U, \lambda, g_\theta, x$	L – normalized Laplacian matrix U – eigenvectors of L λ – eigenvalues g_θ– trainable filter kernel x – node feature/data vector
$T_{\mathrm{k}}(\tilde{\Lambda})$	Chebyshev polynomial of order k applied to scaled Laplacian $\tilde{\boldsymbol{\lambda}}$.
$\lambda_{\max}$	Maximum eigenvalue of L.
$\tilde{L} = (2/\lambda_{\max})\, L - I_{\mathrm{n}}$	Scaled Laplacian.
$g'_\theta(\Lambda)$	Chebyshev-based approximation of spectral convolution.
$A, D, \hat{\mathrm{A}}, \dot{\mathrm{D}}$	A – adjacency matrix D – degree matrix $\hat{\mathrm{A}} = \mathrm{A} + \mathrm{I_n}$, $\dot{\mathrm{D}} = \sum_{\mathrm{j}} \hat{\mathrm{A}}_{\mathrm{ij}}$ (degree of $\hat{\mathrm{A}}$).
I_{n}	Identity matrix of size n.
SEDA	Scale-enhanced deformable attention mechanism.
M	Number of attention heads in MHSA.
C, H, W	Feature dimensions: channels, height, width.
$V = [V_1, V_2, \ldots, V_M]$	Value vectors for each attention head.
$R\ (< M)$	Number of aggregated scale-heads after SEDA weighting.
π_r	Weight for the $r-$ th scale-head (after SoftMax).
$\tilde{V}_r$	Aggregated value tensor for the $r-$ th scale-head.
$P_i, p_{r,j}$	$\mathrm{P_i}$ – Query position for $i-$ th query; – , $p_{r,j}$value sampling position for $j-$ th point in head r.
$\Delta p_{r,j}$	Learnable offset applied to sampling position , $p_{r,j}$
ϕ	Position mapping and bilinear interpolation operator.
$q_{r,i}$	Query vector at position i in head r.
$A_{r,i,j}$	Attention weight from query i to key j at head r
$o_{r,i}$	Output value at query i for head r, after weighted summation
$O_1, i, O_2, i \ldots, O_R, i$	Set of outputs from each scale-head at position i.
W_{or}	Linear projection weights for the r-th scale-head.
z_{i}	Final integrated output at position i after concatenation and projection
$\Gamma_{r,i}$	Set of sampled positions at scale-head r for query i

Table 2. Class distribution of UVG dataset

Dataset	Number of video sequences (VS)	Sequence	Resolution
HEVC-B	5	Cactus	1920×1080
		BQ terrace	
		Park scene	
		Kimono	
		Basketball drive	
HEVC-C	4	BQ mall	832×480
		Party scene	
		Race horse	
		Basketball drill	
HEVC-D	4	Basketball pass	416×240
		BQ square	
		Race horses	
		Blowing bubbles	

Table 3 presents the performance of SEDA-GNN approach of various from B to D classes with metrics namely BD-BR, MS-SSIM, BD-PSNR, and ΔT, for average performance in different classes are evaluated, the SEDA-GNN approach attains an BD-BR average of 3.372%, BD-PSNR of -87.5 dB, ΔT of – 153.0%, average and -69.2% of MS-SSIM. The B class consists of 5 full HD sequences with a resolution of 1920×1080, which are used to evaluate the performance of high-resolution coding in both low-delay configurations and random access, class C contains 832×480 mid-resolution sequences that represent moderate-resolution content targets for broadcast and streaming. Class D having 4 lower-resolution 416×240 sequences used to codec behavior under conditions of compact video-like video conferencing, the proposed SEDA-GNN approach is compared based on classes B, C, D, and E with respect to the JCT-VC dataset.

The developed SEDA-GNN model helps in reducing the computational complexity with high fidelity and maintains a superior rate-distortion trade-off. The SEDA-GNN exhibited superior compression efficiency with substantial ΔT reduction and significant MS-SSIM improvement. Table 4 comparison between the proposed method and the proposed SEDA-GNN with other existing DL approaches respectively.

Table 3. Class-wise performance comparison of the SEDA-GNN approach on JCT-VC dataset

Classes	BD-BR (%)	BD-PSNR (%)	ΔT (%)	MS-SSIM (%)
B	2.461	-0.096	-94.760	-53.71
C	2.983	-0.104	-94.136	-53.38
D	3.209	-0.138	-93.845	-52.16
E	3.123	-0.192	-94.786	-53.81
Average	3.372	-87.5	-153.0	-69.2

Table 4. Performance analysis of different DL techniques with SEDA-GNN

Method	BD-BR (%)	BD-PSNR (dB)	ΔT (%)	MS-SSIM (%)
CNN	1.461	-0.906	-95.760	-51.71
RNN	3.893	-0.401	-93.148	-52.49
LSTM	2.029	-0.381	-94.836	-53.18
ViT-ret	3.483	-0.234	-92.654	-54.54
SEDA-GNN	2.372	-87.5	-153.0	-69.2

The proposed model balances visual quality and compression gain, leading to high perceptual quality and lower bit rates. Compared to CNN, RNN, LSTM, and ViT-ret, SEDA-GNN achieved the best trade-off across all key metrics it delivers enhanced visual quality and compression efficiency, thereby validating its robust architecture. Table 5 represents the comparison between the existing method and with proposed SEDA-GNN to attains better MS-SSIM rate.

The SEDA-GNN approach helps in message passing by capturing both temporal and spatial dependencies when compared to other traditional methods. SEDA-GNN maintains consistent performance across various content classes, with an average BD-BR of only 3.32%, the high MS-SSIM scores across scenes further confirmed their adaptability to different video characteristics. Table 6 represents a comparison between the traditional method with proposed SEDA-GNN approach handles BD-BR across all UVG classes and the ability to generalize compression visual fidelity and efficiency based on scene complexity.

Table 5. Performance analysis of SEDA-GNN with existing methods

Method	BD-BR (%)	BD-PSNR (dB)	ΔT (%)	MS-SSIM (%)
CNN	1.461	-0.906	-95.760	-51.71
RNN	3.893	-0.401	-93.148	-52.49
LSTM	2.029	-0.381	-94.836	-53.18
ViT-ret	3.483	-0.234	-92.654	-54.54
SEDA-GNN	2.372	-87.5	-153.0	-69.2

Table 6. Performance of SEDA-GNN approach for classes using UVG dataset

Classes	BD-BR (%)	BD-PSNR (dB)	ΔT (%)	MS-SSIM (%)
Beauty	3.810	-0.843	-84.324	-43.26
Jockey	3.477	-0.263	-85.549	-44.88
CityAlley	2.824	-0.929	-85.704	-42.74
FlowerFocus	3.989	-0.862	-86.326	-45.39
SunBath	3.736	-0.432	-86.817	-45.68
Average	3.320	-0.203	-85.673	-44.79

3.2. Comparative analysis

In this section, the SEDA-GNN approach is compared with traditional methods, such as CRLVC [21] and TCM [22], on the UVG dataset, in Table 6 the SEDA-GNN approach was evaluated using the database JCT-VC, whereas Table 7 shows the results for the JCT-VC dataset. In Tables 7 and 8, for video sequences in the JCT-VC dataset, SEDA-GNN attains less BD-BR of 5.85% on average, and for the traffic sequences, attains 4.32 % respectively, by utilizing a graph-based context. SEDA-GNN helps to achieve both sustainable MS-SSIM and BD-rate, thereby providing its capability to compress the data more efficiently.

Table 7. Comparative analysis of the existing methods with SEDA-GNN in terms of the UVG dataset

Datasets	Methods	BD-rate (%)			MS-SSIM (%)
		FID	KID	LPIPS	
UVG	CRLVC [21]	-43.1	-31.6	-49.9	21.8
		85.2	30.0	57.1	25.2
		-17.7	-20.6	15.8	-32.5
		5.7	-57.2	18.5	-18.5
		-54.8	-63.4	25.7	-25.6
		-48.5	-47.8	-30.5	-45.2
		-11.7	-31.8	-32.5	-65.3
	SEDA-GNN	3.230	-84.5	-43.6	-76.2
JCT-VC	CRLVC	-10.5	-65.3	-12.7	-12.4
		-12.8	-15.9	-32.6	-32.5
		-15.2	-25.2	-32.8	-24.6
		-32.5	-36.7	-65.2	-50.3
	SEDA-GNN	-47.8	-47.8	-77.1	+72.4

In Table 8, the proposed SEDA-GNN method achieves efficient BD reduction and MS-SSIM outperforming the TCM [22] method which highlights its efficiency in compressing HEVC-RGB data while preserving high perceptual and reconstruction quality. In the table, the proposed method comparison is given between PSNR value and MS-SSIM rate with the traditional method with respect to the classes. While managing the PSNR and BD-rate, the proposed SEDA-GNN method attains efficiency in perceptual quality by optimizing the graph-based representation across temporal and spatial domains.

Table 8. Performance of the SEDA-GNN method for the JCT-VC dataset in terms of classes

Dataset	Method	PSNR (ΔBD-rate)	MS-SSIM (ΔBD-rate)
HEVC-RGB	TCM [22]	-15.2	-48.5
		+4.7	-47.2
		-10.2	-55.1
		-6.4	-53.9
		-23.0	-51.4
		-21.0	-35.0
Overall (PSNR)	SEDA-GNN	-14.4	-37.7

In Table 9, the SEDA-GNN approach is compared with other existing deep learning methods using different performance metrics which have the highest MS-SSIM gains and the lowest BD-BR and △T values. This comparison showed better visual quality and superior rate-distortion efficiency for the JCT-VC and UVG datasets with respect to another existing algorithm. The superior rate-distortion efficiency of compressed videos is more effective when managing and enhancing the quality bit rate when compared with other existing methods.

Table 9. Performance of the SEDA-GNN method with other deep learning algorithms

Method	BD-BR (%)	BD-PSNR (dB)	△T	PSNR (%)	MS-SSIM (%)	VMAF
AlexNet	8.43	-0.56	-68.96	-29.82	-29.67	73.40
ResNet	7.08	-0.49	-71.31	-27.93	-26.45	82.38
CNN	5.68	-0.48	-65.53	-21.92	-31.34	73.48
ViT	8.85	-0.47	-63.23	-29.02	-23.87	69.03
Swin transformer	12.32	-0.39	-62.20	-28.08	-23.26	72.82
Hybrid CNN-ViT	12.67	-0.45	-65.43	-52.18	-47.76	87.67
SWDA-CNN	7.30	-0.38	-85.87	-42.19	-37.76	87.67
HDVC	6.49	+0.35	-89.76	+32.93	+40.92	-
Proposed SEDA-GNN for UVG	5.85	+0.25	-91.88	+34.12	+42.30	90.89
Proposed SEDA-GNN for JCT-VC	4.32	0.24	-90.77	+24.21	-32.03	89.08

3.3. Statistical analysis

A comparative evaluation of various algorithms using non-metric tests, such as the rank and Friedman tests is known as statistical analysis. Table 10 represents the statistical analysis of the video compression which is presented in statistical analysis for comparison among the rank and mean rank. The mean rank is the average of the ranks assigned to the SEDA-GNN algorithm across all video sequences providing high mean consistency for better performance of compression in ranking.

Table 10. Statistical analysis for proposed SEDA-GNN

Algorithm	Friedman	
	Mean rank	Rank
CNN-LSTM	1.83	1
DNN	2.67	2
CNN	3.50	3
RNN	4.17	4
ViT-Ret	4.92	5
SEDA-GNN	5.92	6
Friedman Q	14.62	
DoF	5	
P-value	0.012	

The ordinal is ranked based on the mean rank, the Friedman Q test statistic is calculated from the rank sums is used to assess significant differences between sequences, the degree of freedom is known as DoF. The probability that ranking differences are observed by chance and significant differences is measured by the p-value. The Friedman test helps to rank the performance across various video samples without depending on equal variance assumptions and normality, which makes it more suitable for diverse video compressions in video data.

3.4. Discussion

The SEDA-GNN framework used for video compression by combining a SEDA mechanism within the graph neural network architecture, this method helps dynamically adjust to varying spatial and temporal contexts in video data, thereby efficiently capturing both local and global dependencies. SEDA-GNN selectively focuses on discriminative features across different scales, which helps identify variations in motion and texture within video sequences by constructing a graph-based representation of video frames, the model effectively captures spatial and temporal dependencies and intricate interframe relationships, thereby providing a more significant redundancy reduction and better-quality preservation. Moreover, the GNN component excels in modelling more contextual information and long-range dependencies, the proposed method surpasses the traditional convolutional architecture in capturing nuanced variations in video content, the limitations of existing methods make it difficult to maintain temporal coherence and structural integrity, particularly in scenes with complex textures and motions. The proposed SEDA-GNN approach addresses these challenges in video compression by integrating deformable attention with GNN modelling, this approach provides superior performance in terms of video quality and compression efficiency compared with traditional methods, SEDA-GNN approach in JCT-VC achieved a BD-BR of 2.372% and 3.230% on the UVG dataset, respectively.

4. CONCLUSION

The proposed SEDA-GNN introduces an efficient advancement in optimizing the CU partitioning process within a video-compression framework, the integration of SEDA-GNN captures the feature representation, which is important for accurate CU partitioning. During training, the proposed SEDA attention mechanism provides a stable activation and faster convergence which leads to partition decision accurately. The optimal CU partitions are predicted by using proposed method that removes the artifacts and helps to evaluate the all-possible partition combinations thereby decreasing the number of computations occurs during the compression. The proposed SEDA-GNN model efficiency provides a better performance by considering performance metrics such as 3.230% on the UVG and 2.372% on JCT-VC dataset that indicates enhanced compression performance without compromising the quality. In future work, exploring alternative deep learning-based methods combined with the attention mechanism and transformer architecture used for CU partitioning further improves the compression efficiency and prediction accuracy and it also provides additional improvements in adapting encoding scenarios and diverse video content.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Revathi Kasinathaperumal	✓	✓	✓	✓	✓	✓		✓	✓		✓			✓
Hosanna Princye Periapandi		✓				✓	✓			✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

– The data that support the findings of this study are openly available in UVG dataset in https://www.kaggle.com/datasets/minhngt02/ultra-video-group-uvg [26], JCT-VC dataset in https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/jctvc.aspx [27].

REFERENCES

[1] B. Chen, Z. Wang, B. Li, S. Wang, and S. Wang, "Interactive face video coding: a generative compression framework," *IEEE Transactions on Image Processing*, vol. 34, pp. 2910–2925, 2025, doi: 10.1109/TIP.2025.3563762.
[2] X. Sheng, L. Li, D. Liu, and H. Li, "Prediction and reference quality adaptation for learned video compression," *IEEE Transactions on Image Processing*, vol. 34, pp. 2285–2300, 2025, doi: 10.1109/TIP.2025.3555401.
[3] A. M. Kamoona, A. K. Gostar, A. B. -Hadiashar, and R. Hoseinnezhad, "Multiple instance-based video anomaly detection using deep temporal encoding–decoding," *Expert Systems with Applications*, vol. 214, 2023, doi: 10.1016/j.eswa.2022.119079.
[4] A. Tissier, W. Hamidouche, S. B. D. Mdalsi, J. Vanne, F. Galpin, and D. Menard, "Machine learning based efficient QT-MTT partitioning scheme for VVC intra encoders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 4279–4293, 2023, doi: 10.1109/TCSVT.2022.3232385.
[5] H. Fei, S. Wu, M. Zhang, M. Zhang, T. S. Chua, and S. Yan, "Enhancing video-language representations with structural spatio-temporal alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 7701–7719, 2024, doi: 10.1109/TPAMI.2024.3393452.
[6] Y. Wang, T. Liu, J. Zhou, and J. Guan, "Video anomaly detection based on spatio-temporal relationships among objects," *Neurocomputing*, vol. 532, pp. 141–151, May 2023, doi: 10.1016/j.neucom.2023.02.027.
[7] V. Shanmugam and B. U. Maheswari, "A semantic-aware compression strategy for intelligent vehicles," *Procedia Computer Science*, vol. 258, pp. 2544–2553, 2025, doi: 10.1016/j.procs.2025.04.516.
[8] V. Galiano, H. Migallón, M. M.-Rach, O. L.-Granado, and M. P. Malumbres, "Correction to: on the use of deep learning and parallelism techniques to significantly reduce the HEVC intra-coding time," *Journal of Supercomputing*, vol. 79, no. 7, pp. 8148–8149, 2023, doi: 10.1007/s11227-022-04918-1.
[9] V. D. Huszar, V. K. Adhikarla, I. Negyesi, and C. Krasznay, "Toward fast and accurate violence detection for automated video surveillance applications," *IEEE Access*, vol. 11, pp. 18772–18793, 2023, doi: 10.1109/ACCESS.2023.3245521.
[10] V. T. Le and Y. G. Kim, "Attention-based residual autoencoder for video anomaly detection," *Applied Intelligence*, vol. 53, no. 3, pp. 3240–3254, 2023, doi: 10.1007/s10489-022-03613-1.
[11] Y. Liu *et al.*, "AMP-net: appearance-motion prototype network assisted automatic video anomaly detection system," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 2843–2855, 2024, doi: 10.1109/TII.2023.3298476.
[12] B. Jin *et al.*, "ADAPT: action-aware driving caption transformer," in *2023 IEEE International Conference on Robotics and Automation*, May 2023, pp. 7554–7561. doi: 10.1109/ICRA48891.2023.10160326.
[13] Y. Wang *et al.*, "TokenCut: segmenting objects in images and videos with self-supervised transformer and normalized cut," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 15790–15801, 2023, doi: 10.1109/TPAMI.2023.3305122.
[14] J. Wensel, H. Ullah, and A. Munir, "ViT-ReT: vision and recurrent transformer neural networks for human activity recognition in videos," *IEEE Access*, vol. 11, pp. 72227–72249, 2023, doi: 10.1109/ACCESS.2023.3293813.
[15] Z. Yu *et al.*, "PhysFormer++: facial video-based physiological measurement with slowfast temporal difference transformer," *International Journal of Computer Vision*, vol. 131, no. 6, pp. 1307–1330, 2023, doi: 10.1007/s11263-023-01758-1.

[16] L. Yu, Z. Li, J. Xiao, and M. Gabbouj, "High-frequency enhanced hybrid neural representation for video compression," *Expert Systems with Applications*, vol. 281, 2025, doi: 10.1016/j.eswa.2025.127552.
[17] V. Shanmugam and B. U. Maheswari, "Optimizing semantic-aware video compression using particle swarm optimization technique for automotive applications," *IEEE Access*, vol. 13, pp. 106091–106102, 2025, doi: 10.1109/ACCESS.2025.3580151.
[18] A. Bilican, M. A. Yilmaz, and A. M. Tekalp, "Content-adaptive inference for state-of-the-art learned video compression," *IEEE Open Journal of Signal Processing*, vol. 6, pp. 498–506, 2025, doi: 10.1109/OJSP.2025.3564817.
[19] B. Cardone and F. D. Martino, "Fuzzy-based video compression using bilinear fuzzy relation equations," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 4, pp. 2215–2225, 2024, doi: 10.1007/s12652-023-04748-w.
[20] C. Jiang, J. Xu, and L. Yin, "Improved aerial video compression for UAV system based on historical background redundancy," *Tsinghua Science and Technology*, vol. 30, no. 6, pp. 2366–2383, 2025, doi: 10.26599/TST.2024.9010110.
[21] P. Du, Y. Liu, and N. Ling, "CGVC-T: contextual generative video compression with transformers," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 14, no. 2, pp. 209–223, 2024, doi: 10.1109/JETCAS.2024.3387301.
[22] X. Sheng, J. Li, B. Li, L. Li, D. Liu, and Y. Lu, "Temporal context mining for learned video compression," *IEEE Transactions on Multimedia*, vol. 25, pp. 7311–7322, 2023, doi: 10.1109/TMM.2022.3220421.
[23] H. Guo, S. Kwong, and M. Zhou, "Exploring invertible encoding for deep video compression," *IEEE Transactions on Broadcasting*, vol. 71, no. 2, pp. 517–528, 2025, doi: 10.1109/TBC.2025.3541869.
[24] N. V. Thang and L. V. Bang, "Hierarchical random access coding for deep neural video compression," *IEEE Access*, vol. 11, pp. 57494–57502, 2023, doi: 10.1109/ACCESS.2023.3283277.
[25] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao, and D. Xu, "An end-to-end learning framework for video compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3292–3308, 2021, doi: 10.1109/TPAMI.2020.2988453.
[26] M. N. Tong "UVG-RGB," *Kaggle.* 2025. [Online]. Available: https://www.kaggle.com/datasets/minhngt02/ultra-video-group-uvg
[27] J. -R. Ohm and G. Sullivan, "JCT-VC - joint collaborative team on video coding," *ITU*. Accessed: Jun. 10, 2025. [Online]. Available: https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/jctvc.aspx

BIOGRAPHIES OF AUTHORS

Revathi Kasinathaperumal received her B.E. (Electrical and Electronics Engineering) from PSNA College of Engineering, Madurai Kamaraj University, Tamilnadu, India in the year 1999; completed her masters in VLSI Design and Embedded Systems from Visvesvaraya Technological University in the year of 2008; and pursuing Ph.D. in the field of Signal Processing under Visvesvaraya Technological University. Since then, actively involved in teaching has nineteen years of experience. At present she is working as an assistant professor in S. E. A College of Engineering and Technology, Bangalore affiliated to Visvesvaraya Technological University, her area of interest is the field of image processing, signal processing, and VLSI and embedded systems. She can be contacted at email: revathiselvaraj1229@gmail.com.

Hosanna Princye Periapandi received her B.E. (Electronics and Instrumentation Engineering) from Sapthagiri College of Engineering from Periyar University, Tamilnadu, India in the year 2002 and completed her masters in Engineering from Anna University in the year of 2004. Since then, she is actively involved in teaching and research and has sixteen years of experience in teaching. She obtained his Ph.D. in the field of Information and Communication Engineering from Anna University in the year of 2018. At present she is working as an associate professor in Sri Sairam College of Engineering, Bangalore affiliated to Visvesvaraya Technological University. Her area of interest is the field of medical image processing, signal processing, and VLSI. She can be contacted at email: hosannaprincye@gmail.com.