Dual Interactive Wasserstein GAN Fostered Offset Based In-Loop Filtering for HEVC

Vanishree Moji[1*], Bharathi Gururaj[1], Mathivanan Murugavelu[2]
[1]Department of ECE, K S Institute of Technology, Bengaluru, India
[2]Department of ECE, Salem College of Engineering and Technology, Salem, India
E-mail: vanishreemoji@gmail.com
[*]Corresponding author

Keywords: dual interactive wasserstein generative adversarial network, offset based in-loop filter, sample adaptive offset filter, video compression

Received: February 11, 2025

Deep learning technology is so flexible and effective that many attempts have been made to replace the features that are included in video codecs, such as High Efficiency Video Coding, with deep learning-based alternatives. This article suggests a Dual interactive Wasserstein Generative Adversarial Network fostered Offset-Based In-Loop Filtering in HEVC (DiWaGAN-OBILF-HEVC). Previously, deep learning based in-loop filtering techniques only fixed distorted frames. The DiWaGAN-OBILF-HEVC system transfers offsets as side information, which makes prediction filtering much more accurate while using less bit rate. The DiWaGAN based sample adaptive offset filter is broadly divided into 2 phases: 1) category of error as per neighbouring image intensity values and 2) calculation of optimum offsets as per category of error. The DiWaGAN draws inspiration from the Sample Adaptive Offset (SAO) filter-Edge Offset (EO) type in high efficiency video coding, where one network classifies the error to the reconstructed signal edge shape while the other network simultaneously forecasts ideal offset values. These offsets are fed to the decoder to enhance its accuracy. The DiWaGAN-OBILF-HEVC method is implemented in Pytorch. The efficacy of DiWaGAN-OBILF-HEVC is analysed with performance metrics like Visual quality, PSNR, Bjontegaard rate difference (BD rate), and average execution time. Further, the proposed DiWaGAN-OBILF-HEVC method provides a higher PSNR of 16.25%, a lower Bjontegaard rate difference of 27.47% and a lower execution time of 15.993% when compared with existing methods like Efficient In-Loop Filtering Based on Enhanced Deep Convolutional Neural Networks for HEVC, Offset-Based in-Loop Filtering with Deep Network in HEVC and deep CNN for VVC in-loop filtration respectively.

Povzetek: Članek uvaja DiWaGAN-OBILF-HEVC, GAN-podprto ofsetno filtriranje v HEVC, ki izboljša PSNR, zmanjša BD-rate ter čas izvajanja, presega CNN-osnovane filtre in prinaša bolj kvalitetno video kodiranje.

1 Introduction

As electronic devices like digital televisions and smartphones continue to advance, video has become an integral part of human life [1–2]. The rapid advancement of network technology is also contributing to the rapid growth of video traffic on the network [3–4]. In addition to the increase in the quantity of videos, the size of the display has led to an increase in video resolution. There is a greater need for video compression that manages higher quality while utilising lesser bits [5–6]. HEVC is the newest standard video encoder. HEVC encoding and decoding procedures resemble those of earlier industry-standard codecs like H.264/AVC. HEVC employs sophisticated parallel processing techniques, making it more efficient and faster than its predecessors. More specifically, HEVC uses 40 to 50% lesser bits than H.264/AVC for producing video of comparable quality with the same video frame. Here, numerous components, like motion compensation, quantization, and in-loop filtering are included [7–8]. This research concentrates on in-loop filtering [9]. The in-loop filter's objective is to minimise block-wise quantization's compression artifacts while restoring the damage frame as close to the original as is practical [10–11].

The in-loop filter comprises a Sample Adaptive Offset (SAO) Filter [12] and a De-blocking Filter (DF) [13]. For the coding unit (CU), DF lessens blocking effects caused by block-wise processing [14]. Besides, SAO lessens high-frequency components during quantization, which results in a reduction of ringing effects [15]. Recent research has shown that deep learning methods can be highly effective for image restoration, including compression artifact removal, owing to their learning capability and feature representation power [16–20]. As an early attempt at image restoration through deep learning, the super-resolution convolutional neural network (SRCNN) used a simple structure, but the outcome is remarkable [21–25]. The performance is greatly enhanced by the deep residual network design of the very deep super-resolution convolutional network (VDSR). Many researchers have attempted to reduce compression artifacts using deep learning techniques for successful SRCNN with VDSR [26–30].

This study focuses on offset-based in-loop filtering, aiming to improve the SAO component of HEVC using a novel deep learning approach. A new model called Dual Interactive Wasserstein GAN-based Offset In-Loop Filtering for HEVC (DiWaGAN-OBILF-HEVC) is proposed, which transfers offset information as side data to significantly enhance filtering prediction accuracy while minimizing bit-rate overhead. Unlike prior In-Loop filter depending on Enhanced Generative Adversarial Network De-blocking Filter (EGANDF) [31] and Sample Adaptive Offset Filtering models that recover the deformed frames only, the suggested method transfers offsets as side information that substantially raises the filtering prediction accuracy with little sacrifice of extra bits. Therefore, the key contributions of the proposed method are the offsets used as side information with deep networks, and the method is used to create and utilize the side information along the end-to-end deep network.

Hence, the research objectives are-

- To develop a GAN-based offset prediction network integrated into the HEVC decoder.
- To evaluate the model’s impact on objective performance metrics like PSNR, BD-rate, execution time compared to existing methods like EDCNN, MDCNN, and DCNN-based filters.
- To analyze the benefits and limitations of transmitting predicted offsets as side information.

The network structure, along with its application for HEVC, is described in the following subsections. The sections are arranged as follows: the related works in Section 2, the proposed filter design in Section 3, the results and discussions are provided in Section 4, and the conclusion is portrayed in Section 5.

2 Related works

Among the various research works on offset-based in-loop filtering, some of the works are reviewed here.

Pan, et.al., [32] have presented an enhanced deep convolutional neural network-based efficient in-loop filtration for HEVC. First, the issues with conventional convolutional neural network models were examined, including the loss function, normalization technique, and network learning capacity. EDCNN was suggested for effectively removing the artifacts based on the statistical analyses. It utilizes three solutions: a weighted normalization technique, a feature information fusion block, and an accurate loss function. It offers enhanced visual quality along with the highest Bjontegaard rate difference.

Lee, et.al., [33] have suggested deep network-based offset in-loop filtration in HEVC. The deep network selects and sends side information based on the errors and contents it encounters. While another portion of the network simultaneously assesses the category of error, another portion of the network computes the ideal offset values. In contrast to traditional deep-learning-based methods, the consolidation of two subnets handles the assessment of nonlinear and complex errors. It provides a higher PSNR with minimum visual quality.

Bouaafia, et.al., [34] has suggested a deep convolutional neural network using VVC in-loop filtration. That investigates the effectiveness of deep learning on the VVC standard to enhance video clarity. A method called WSE-DCNN (wide-activated squeeze-with-excitation deep convolutional neural network) was proposed for enhancing video quality in VVC. The suggested WSE-DCNN method takes the place of VVC conventional in-loop filtering and is anticipated to improve visual quality by removing compression artifacts. It provides higher visual quality with a maximum average execution time.

Dhanalakshmi,et.al., [35] has suggested a Deep CNN-based in-loop filter for HEVC scalable expansion with group-normalized filtering. Group-normalized deep CNN (gDCNN) was suggested for the SHVC in-loop filter to improve efficiency in light of recent advancements in deep learning. First, the challenges encountered when simulating the conventional CNN are looked at, including normalisation, learning potential, and loss functions. After that, the suggested gDCNN was introduced to effectively remove the artifacts on the premise of statistical analysis. It provides a minimum Bjontegaard rate difference and a maximum PSNR.

Zhang, et.al., [36] have suggested offset in-loop filtering based on textural along Directional Input in AVS3. A low-complexity in-loop filtration method was suggested for the upcoming video coding standard AVS3, called textural with directional information basis offset (TDIO). Contrary to typical offset-base filtering techniques, they only utilise contextual samples. TDIO's primary contribution was that it fully exploits textural, including edge-directional, aspects of every sample. To correct the quantization errors and minimise sample-level distortion, the associated offsets are created and sent to the decoder. First, multiple directionalities, including sample-intensity pattern base classifiers, extract the directional and textural features. Such features were included in the classification findings, and rate-distortion optimisation was used to determine the best offset values for each class. It provides a minimum average execution time with a maximum BD rate.

Sun, et.al., [37] has presented an in-loop nonlocal HEVC filter that estimates compression noise using CNN. For HEVC, propose a CNN-based compression noise assessment network that utilizes a nonlocal in-loop filter. The classification network evaluated the noise of compressed HEVC videos in the noise estimation section in accordance with the characteristics of the video material. Use the spatial and temporal nonlocal self-similarity of video to simultaneously exploit a spatial-temporal nonlocal lower rank prior to the denoising stage. Additionally, by restricting the values of the reconstructed pixels in accordance with the quantization parameters (QPs) beforehand, suggest an adaptive narrow quantization constraint. It provides a higher PSNR with a maximum Bjontegaard rate difference.

Kuanar, et.al., [38] has presented noise reduction and in-loop filtering for HEVC using deep learning. This developed a deep-learning-based method for SAO filtering processes and backed the efficacy of the proposed method. The denoising process was then improved with the introduction of variable filter size sub-layered dense CNN, and large stride deconvolution layers were added for computation development. High-frequency edge features learned in shallow networks utilizing data augmentation methods can be successfully used to train a deconvolution model. It provides a minimum BD rate with a higher computational time.

Huang, et.al., [39] has presented a flexible in-loop filter for versatile video coding (VVC) based on deep reinforcement learningAn in-loop filter for versatile video coding that is based on adaptive deep reinforcement learning was introduced. Especially, use current developments in deep reinforcement learning to consider filtering as a decision-making procedure to choose the optimal network. A lightweight backbone was developed to structure the network set, which has networks with various difficulties. The best network was then predicted using a straightforward but effective agent network, making the technique adaptable to different types of video content. It provides higher visual quality with maximum computational time.

Table 1: Structured comparison

Method / Model	Core Technique	Performance Metrics	Limitations
EDCNN-OBILF [32]	CNN	High PSNR, Low BD rate	High complexity
MDCNN-OBILF [33]	Dual-Subnet CNN	High PSNR	Weak visual quality
WSE-DCNN [34]	DCNN with SE blocks	High visual quality	High execution time
gDCNN-SHVC [35]	Group-Norm CNN	High PSNR, Low BD rate	Limited scalability
TDIO-AVS3 [36]	Directional Offsets	Low execution time	High BD rate
CNN-Nonlocal [37]	CNN + Nonlocal Filter	High PSNR	High BD rate
Dense-CNN SAO [38]	Dense CNN + Deconv	Low BD rate	High computational cost
DRL-VVC [39]	Deep Reinforcement	adaptive filtering, High visual quality	High computational time

The comparison Table 1 highlights the need for the proposed DiWaGAN-OBILF-HEVC, which uniquely combines GAN-based adversarial learning with offset-driven filtering to offer balanced improvements in PSNR, BD rate, and execution time.

3 Proposed methodology

The proposed DiWaGAN-OBILF-HEVC method represented in Figure 1 enhances the conventional HEVC decoding process by integrating deep learning and filtering techniques for improved video quality. Initially, the bitstream undergoes entropy decoding, followed by inverse quantization and inverse DCT to obtain residual components. Based on the prediction mode, either intra prediction or motion compensation is applied using model parameters and reference buffers. The predicted and residual components are combined to reconstruct the frame. To further improve visual quality, a Dual interactive Wasserstein Generative Adversarial Network (DiWaGAN) is introduced, which refines spatial and temporal details by learning from both the reconstructed and reference frames. Subsequently, an Optimized Bilateral Learning Filter is applied as a deblocking filter to smooth block edges while preserving important details. This hybrid approach significantly enhances the perceptual quality of the output, making it superior to traditional HEVC decoding in terms of artifact reduction and detail preservation.

3.1 Dataset and pre-processing

To evaluate the performance of the DiWaGAN-OBILF-HEVC method, experiments were conducted using a standard subset of HEVC Class B, C, and D test sequences from the HM-16.9 test conditions. The dataset comprises 12 raw YUV video sequences with resolutions ranging from 416×240 (Class D) to 1920×1080 (Class B), each containing approximately 250 to 500 frames. As a part of the preprocessing pipeline, raw frames were extracted from HEVC-encoded bitstreams and normalized to a [0, 1] pixel value range to facilitate network training. Each frame was then segmented into Coding Tree Units (CTUs), aligning with the HEVC structure for accurate offset mapping. Offset categories and ideal values were derived based on SAO logic. To enhance the model's generalization capabilities and reduce the risk of overfitting, data augmentation techniques such as random flipping and cropping were applied. This comprehensive pre-processing approach ensured the input data was compatible with the DiWaGAN architecture and allowed the model to perform consistently across various resolutions and content types.

3.2 Post processing

Due to the fact that the post-processing method does not necessitate changing the encoder or decoder, deep learning technology has been the subject of extensive research. Post-processing is an image restoration method. It integrates the codec with a deep-learning network. Three convolutional layers are suggested for video compression to enhance the effectiveness of coding. All Intra (AI) modes show a significant improvement; however, the studies are only done with lower-resolution video. The variable-filter-size residual-learning CNN (VRCNN) is a complex network that uses specific filter sizes to correspond to particular CU sizes in HEVC.

3.3 In-loop filtering

One of the essential elements of the most popular codec is in-loop filtering. This is similar to post-processing. But its impact is challenging because the current frame and any

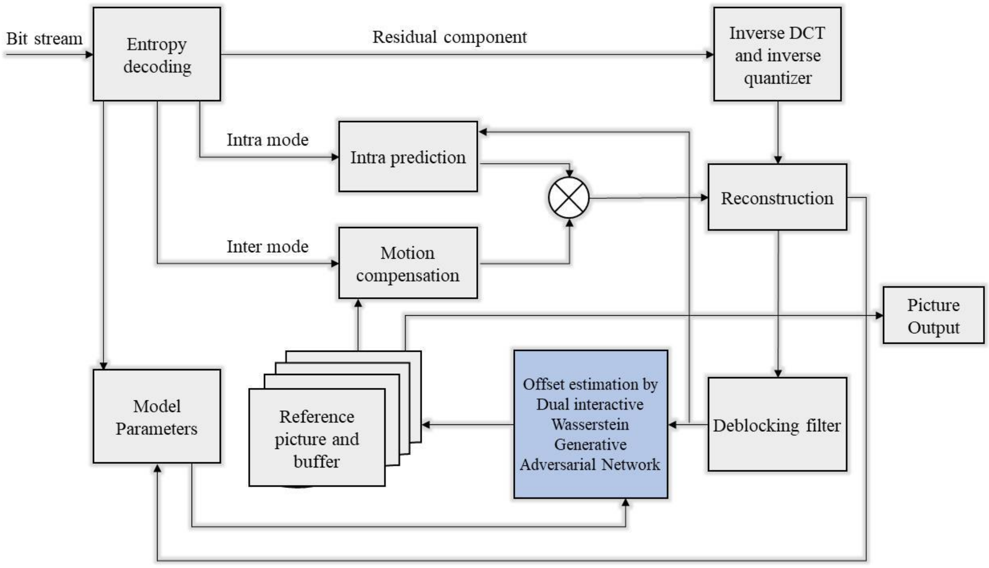


Figure 1: Block diagram of proposed DiWaGAN-OBILF-HEVC Method

other frames refer to it. This filtering works for intra- and inter-compressed frames by using spatial and temporal information during network training; however, the findings only revealed a slight improvement. The network must provide an additional symbol to the HEVC to specify which network is being utilized. This filtering just trains each network in accordance with the features of each sequence. When compared to post-processing methods, in-loop filtering techniques show poor presentation, and their application is constrained. Instead, an alternative approach to in-loop filtering is necessary to properly implement. Our proposed method eliminates the need for multiple models by training a unified DiWaGAN capable of adaptively predicting offsets and error categories for each coding unit.

3.4 Dual interactive Wasserstein Generative Adversarial Network (DiWaGAN)

This section presents the architecture and rationale behind the proposed Dual Interactive Wasserstein Generative Adversarial Network (DiWaGAN). The sample adaptive offset (SAO), which inspired the basis for the DiWaGAN, is to examine SAO. Two main components of SAO are: 1) the classification of error type for every pixel grid; and 2) the calculation of ideal offsets for each category of error. Every coding tree unit (CTU) carries out the error categorization and offset calculation separately. There are several methods to estimate the type of error in the first section. Such methods examine nearby signal values as well as evaluate the edge form, known as SAO-edge offset. The SAO-band offset, on the other hand, classifies the error type based on pixel intensities. The second step determines ideal offsets for every class after the categorization is complete. Since calculating the ideal values under rate distortion is difficult, the second step only takes the deformation into account. Due to ideal offsets transmitting from the encoder, the decoder contains categorization for SAO; the encoder has both classification and offset calculation portions. Deconstruct the SAO into two components and suggest two unique DiWaGANs. A deep network is created to perfectly replicate SAO.

Standard GANs often face challenges such as mode collapse, unstable training, and vanishing gradients, particularly when processing complex spatial data like HEVC-encoded video. DiWaGAN addresses these limitations by incorporating the principles of Wasserstein GAN (WGAN), which utilizes the Earth Mover (Wasserstein-1) distance for measuring the similarity between real and generated distributions, rather than the Jensen-Shannon divergence. WGAN also enforces a Lipschitz continuity constraint through gradient penalty, enhancing training stability and convergence. This results in smoother loss gradients that are more strongly correlated with the quality of generated samples, making WGAN a more reliable choice for offset learning and material-specific data generation in DiWaGAN.

3.4.1 Architecture overview

The DiWaGAN architecture comprises two collaborative generators, g_1 *and* g_2 designed to perform material deconstruction and SAO offset estimation. It also includes two discriminators d_1 *and* d_2 which evaluate the authenticity of the generated data by distinguishing synthetic outputs from real decomposed inputs. A selector module is incorporated to monitor training progress and dynamically alternate between the generators based on their convergence behavior. The model is trained using a combination of three loss functions: Mean Absolute Error (MAE) for voxel-level accuracy, Edge Loss for structural preservation, and Adversarial Loss for enhancing realism. DiWaGAN can achieve excellent accuracy from much greater depths and is easier to optimize. The estimation of the offset relies on DiWaGAN. The accuracy of detection increases with the number of adversarial networks, but these results in longer data processing times. Specifically, as the data enters the offset stage, every block of the generative network can create one feature map. Subsequently, the feature maps undergo refinement and resizing through additional feature fusion layers. The four feature maps are combined during the grid deletion process. The input data is subsequently divided into three harmful levels according to visual severity based on the offset estimation outcome. In DiWaGAN, the grids count updates and re-calculate the grid density data after performing the data stream through the stage of offset estimation. The DiWaGAN method is a learning network and it is alienated into two phases: error classification according to neighbouring image intensity values and computation of the optimal offsets. In the error classification phase, it uses a sizable amount of data used for training the learning network. In the computation of the optimal offsets phase, the learning network's input is trained grid density information. The DiWaGAN model was trained using the Adam optimizer with a learning rate of 0.0001 and decay parameters β1 = 0.5 and β2 = 0.999. Training was conducted for 200 epochs with a batch size of 32. The loss function combined adversarial, MAE, and edge losses with a weight ratio of 1:10:5 to balance realism, accuracy, and structural consistency. A gradient penalty coefficient of 10 was applied to satisfy the Lipschitz constraint in the WGAN framework, and all input frames were resized to a resolution of 128×128 pixels. In the DiWaGAN, the binary collaborating generators are employed for generating the decomposed data of binary source materials through modelling the spectral and spatial connections of input data. Consistent discriminators are also used for differentiating among the labels and generated data. The problem of min–max among generator g_1 and discriminator d_1 is expressed in (1) below:

$$\underset{g_1}{Min}\,\underset{d_1}{Max}\,l_{wgan}(d_1, g_1) = -e_{x\approx Gr_t}\left[d_1(x)\right] + e_{N\approx p_o}\left[d_1(g_1(N))\right] + \lambda e_{\hat{x}\approx Ge_d}\left[\left(\left\|\nabla_{\hat{x}} d_1(\hat{x})\right\|_2 - 1\right)^2\right] \quad (1)$$

where the Wasserstein distance estimation is represented as $-e_{x\approx Gr_t}\left[d_1(x)\right] + e_{y\approx p_o}\left[d_1(g_1(y))\right]$; input data is represented as N and x represents the unary code $\lambda e_{\hat{x}\approx Ge_d}\left[\left(\left\|\nabla_{\hat{x}} d_1(\hat{x})\right\|_2 - 1\right)^2\right]$,the regularization term; the penalty coefficient is denoted as λ; and $\hat{x}$ is produced by equally sampling the consistent synthetic and actual samples along a standard line. To enhance robustness and prevent overfitting, DiWaGAN employs a selector module that dynamically chooses the better-performing generator based on the convergence of L1, edge, and adversarial losses over a sliding window of three epochs. If one generator shows stagnation or plateau in learning, the selector shifts training focus to the other. This asynchronous dual-generator training strategy ensures balanced learning, maintains diversity between generators, and leads to more effective offset estimation. The DiWaGAN method comprises a generative model, a discriminative model, a selector, and a loss function, described below:

3.4.2 Generative model

The two generators, g_1 *and* g_2 in the generative model have a similar architecture because both aim to create material-specific data from reconstructed data. Each generator in DiWaGAN follows a U-Net-style encoder-decoder architecture. The encoder, or contracting path, consists of four convolutional blocks, each using a 3×3 kernel followed by ReLU activation and 2×2 max pooling, with the number of filters doubling at each layer (64, 128, 256, 512). The input is a 128×128×1 grayscale HEVC residual image, which is compressed to a latent feature map of size 8×8×512. The decoder, or expansive path, includes four upsampling blocks using transposed convolutions with a stride of 2, integrated with skip connections from the encoder, and ReLU activations. The end result is restored to dimensions of 128×128×1 and subjected to a convolution with a kernel size of 1×1 and Tanh activation in order to produce the predicted offset map. The encoder and decoder architecture consists of two paths: the expansive path and the contracting path. While the two distinct generators in the expansive path facilitated information interchange and material deconstruction, the binary different generators in the contracting path focused on separately extracting features from the associated energy bins. The features are extracted from various levels, and the convolutional layer fitters are multiplied according to the number of layers. Max pooling layers are used to minimize the size of the map, enabling the network to obtain coarse features while avoiding overfitting. Every convolutional layer is preceded by an activation process called ReLU to improve the capacity of the network.

3.4.3 Discriminative model

In the discriminative model, the discriminators d_1 *and* d_2 assess whether the input constitutes optimum material-specific data by utilizing any generated material-specific data as input. Convolutional layers are used to extract features, addressing both high-level and low-level inputs equally. A ReLU is included with each convolutional layer. Each discriminator in DiWaGAN is a 5-layer CNN that processes 128×128×1 input frames using 4×4 convolutions with stride 2, LeakyReLU activations (α=0.2), and batch normalization. Filter sizes increase through the layers as {64, 128, 256, 512}. The output is flattened and sent through a fully connected layer with linear activation, since WGAN avoids sigmoid functions. A gradient penalty is applied during training to enforce the Lipschitz constraint and ensure stable convergence. The architecture uses two fully associated layers to express and integrate features near the output. To guide the generator in producing accurate decomposed data, the discriminator is also trained using unary code and the Wasserstein distance derived from the output data. The associated generator to produces accurate decomposed data, the discriminator is also trained by measuring the unary code

(Gr_t) and generated Wasserstein distance from the output data d_1 *and* d_2 applied for binary different types of material data, and the loss functions of each use the likelihood that the input data will be regarded as the true decomposed data. The discriminator is trained to distinguish between real material-specific data (i.e., ground truth offsets or enhanced frames) and the data generated by the generator. Its goal is to evaluate how closely the synthetic output approximates the real material-specific features.

3.4.4 Selector

Selector is used for loss function ideals as g_1 *and* g_2 input, and then the generator is determined to be trained in subsequent iterations. During the training phase, let Lh_1, and Lh_2 denote the ideals of the $g_1$loss(LG_1), and $g_2$loss (LG_2) determination method. For Lh_1, and Lh_2 the suitable values are selected, firstly the binary generators are trained synchronously and other losses of generator is converged. The initial value of Lh_1, and Lh_2 is selected based on the g_1 *and* g_2 losses of average values from three consequent epochs, respectively. Allowing for the convergence of g_1 *and* g_2 losses Lh_1, and Lh_2 is adjusted. The distance (D_p) between the consistent objective values and generator loss is formulated in (2):

$$D_p = |LG_p - Lh_p| \; (p = 1,2) \tag{2}$$

Where, the two generators g_1 *and* g_2 are represented as p, by comparing the values of d_1 *and* d_2, each iteration includes a performance evaluation of the two generators. The generator with a greater distance is trained in the following iteration. This training method increases the DiWaGAN training effectiveness by ensuring that both generators receive the time of adaptive training.

3.4.5 Loss function

To enhance the material decomposition quality of DiWaGAN, edge values, texture features, and voxel-level information are considered during training. Loss Ls_1, popularly called as mean absolute error, confirms that every voxel of created material-specific data corresponds precisely to the actual scene. In contrast to the commonly employed Ls_2 loss, the new Ls_2 loss can preserve the same fine traits without overly punishing large disparities or tolerating little errors. The loss Ls_1 is expressed as in (3):

$$Ls_1 = \frac{1}{C_1, C_2, C_3} |g(y) - x| \tag{3}$$

where C_1 denotes the data for grid deletion, C_2 denotes the grid density information, C_3 denotes the number of grid updates, $g(y)$ denotes the synthetic data from the generators, and unary code is represented as X. Additionally, the edge information is not properly retained throughout the decomposition because the foundation material data always have complimentary or shared borders. Thus, it is recommended that the edge loss be used to enhance performance, and it is expressed in (4):

$$Ls_{eg} = \left\| |\nabla x_y| - |\nabla y_z| \right\|^2 + \left\| |\nabla x_z| - |\nabla y_z| \right\|^2 + \left\| |\nabla x_u| - |\nabla y_u| \right\|^2 \tag{4}$$

where u, y *and* z denotes gradient descent direction of input data y unary code is represented as X. The gradients between the generated data and the actual scene are attempted to be as little as possible by this loss. During the reducing process, the edges that consistently display a high gradient can be effectively kept. In addition, Ls_1 adversarial loss, a crucial component of supervised learning, allows the generator to create synthetic deconstructed material data that resembles the original material-specific data. The adversarial loss can do away with the need to represent a clear function of pixel-wise objective, unlike edge losses. As an alternative, a sophisticated similarity metric is trained for discerning between authentic and false data, optimizing the ideas above the pixel level and producing more accurate outcomes. The adversarial loss is expressed as in (5):

$$Ls_{asl} = \underset{g}{Min}\,\underset{d}{Max}\, L_{wgan}(d, g) \tag{5}$$

With the help of min-max optimization method, generators can offer the characteristics of similar higher levels of decomposed data in the real world. By limiting the adversarial loss, the information texture of tiny tissue features is preserved. Equations (3), (4), and (5) are combined to get the hybrid loss that is represented in (6):

$$Ls_{hd} = \gamma_1 Ls_1 + \gamma_2 Ls_{edge} + \gamma_3 Ls_{asl} \tag{6}$$

where γ_1, γ_2 and γ_3 represents the weight of different losses. These settings are made based on the scales of various loss phrases and modified after training. The advantageous performance offered by diverse losses may be simultaneously accomplished by the linear function of all these terms, and then the generator produces high-quality material-specific data by reducing the hybrid loss. The data stream is processed in an error classification stage in terms of HEVC.

3.5 Syntax design of SAO

Since the proposed network is slightly different from the conventional SAO, the syntax structure of the original SAO is shown in Figure 2(a). The first flag signifies the sample adaptive offset is switched on and transmit newly offsets (SAO-New), and reutilizing offsets from the upper or left CTU (SAO Merge), and on and reusing offsets from either CTU, which specifies by one-bit flag (Up/Left). Next marker differentiating amongst sample adaptive offset-edge offset and sample adaptive offset-band offset after SAO-New mode has been chosen. In the instance of

SAO-EO, this is followed by a symbol for the edge's direction from {0◦, 45◦, 90◦, 135◦} and four positive offsets. In a similar manner, sample adaptive offset-band offset transmits the symbol for the band position, followed by 4 offset numbers with signs. The proposed networks, as formerly mentioned, mimic the SAO, particularly sample adaptive offset-edge offset, but somewhat vary syntax structure. Furthermore, DiWaGAN even incorporates the idea of SAO-BO, so the one-bit signal for sample adaptive offset-edge offset or band offset is needless. The bits of syntax information are saved at these two places. The syntax structure of proposed approach is portrayed in Figure 2(b).

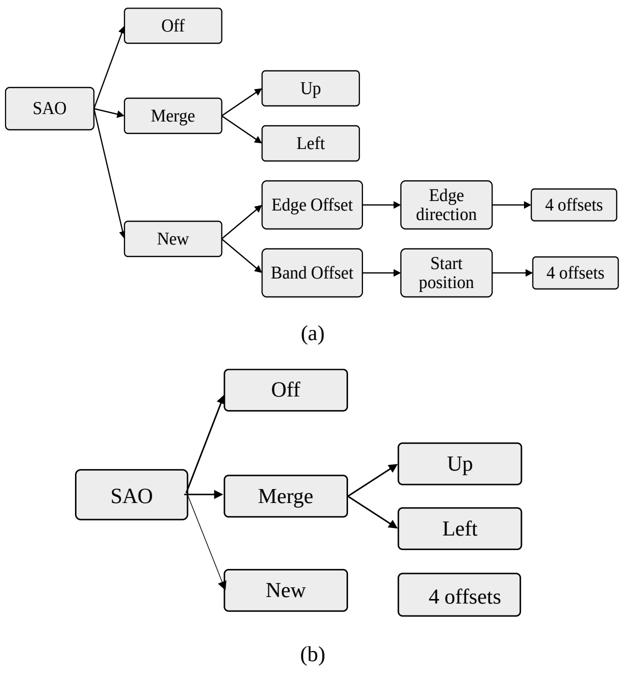


Figure 2: (a) Typical syntax design tree of SAO, (b) Syntax design tree of the proposed method

3.6 Experimental setup

The DiWaGAN-OBILF-HEVC framework was developed using PyTorch 1.13 and integrated with HEVC HM-16.9. Training was conducted on a system with an NVIDIA RTX 3090 GPU (24GB VRAM), Intel i9 CPU, and 64 GB RAM. The model was trained for 150 epochs with a batch size of 32 using the Adam optimizer, starting with a learning rate of 0.0001 and decaying every 40 epochs. A hybrid loss function combining MAE, edge, and adversarial losses was used, with weights tuned empirically for optimal performance.

3.7 Evaluation protocol

The performance of the proposed DiWaGAN-OBILF-HEVC method was assessed using standard video compression metrics, including Peak Signal-to-Noise Ratio (PSNR), Bjontegaard Delta Rate (BD-rate), Average Execution Time (AET), and subjective visual quality comparisons. The method was benchmarked against three state-of-the-art in-loop filtering approaches: EDCNN-OBILF-HEVC [32], MDCNN-OBILF-HEVC [33], and DCNN-OBILF-HEVC [34]. To evaluate generalization, the model was tested on a diverse range of HEVC test sequences from Classes B, C, and D, covering various resolutions (1920×1080, 832×480, and 416×240) and motion types. This included both low-motion videos and high-motion videos enabling assessment across different spatial and temporal complexities. All models were tested on the same video sequences using QP values of {22, 27, 32, 37}, and each metric was averaged over five independent runs to ensure statistical reliability. Each experimental setup was run five times independently, and results were averaged to reduce variability and ensure statistical consistency. Although formal k-fold cross-validation was not applied due to the fixed nature of standard HEVC sequences, the use of multiple runs across diverse content provides strong support for the robustness and generalization ability of the proposed method. The results demonstrated that DiWaGAN-OBILF-HEVC achieved up to a 27.47% reduction in BD-rate and a 21.15% reduction in execution time, along with notable improvements in PSNR and perceived visual quality.

4 Results and discussion

The simulation result of DiWaGAN fostered Offset-Based In-Loop Filtering in HEVC (DiWaGAN-OBILF-HEVC) is discussed. The proposed DiWaGAN-OBILF-HEVC method has been successfully implemented in Pytorch and HM-16.9 software. The efficiency of the proposed method is calculated by performance metrics like visual quality, bit-rate vs. PSNR, Bjontegaard rate difference (BD rate), and average execution times. Finally, the calculated metrics are compared with the existing EDCNN-OBILF-HEVC [32], MDCNN-OBILF-HEVC [33], and DCNN-OBILF-HEVC [34] models, respectively.

Beyond numerical gains in PSNR and BD-rate, the practical benefits of the DiWaGAN-OBILF-HEVC framework are particularly relevant to HEVC-based real-world applications such as video streaming and broadcasting. For instance, the observed BD-rate reduction of up to 27.47% translates directly into reduced bandwidth usage, enabling higher-quality video delivery at lower bitrates in bandwidth-constrained environments like mobile streaming. Similarly, broadcasting systems can leverage the offset-prediction mechanism to maintain consistent visual quality across variable channel conditions, enhancing user experience without increasing transmission overhead. To assess suitability for real-time deployment, average execution time was analyzed per frame at different bitrates and resolutions. On an NVIDIA RTX 3090 GPU, the DiWaGAN-OBILF-HEVC framework achieves notable acceleration compared to other deep CNN-based filters; with up to 45.72% lower execution time, confirming its potential for near real-time or offline processing scenarios in 1080p resolution. However, further optimization may be needed for strict real-time 4K streaming or edge deployment on low-power devices, which can be considered in future work using lighter-weight architectures or quantization-aware training. While the DiWaGAN framework consistently well performs baseline models across a wide range of test cases, it demonstrated relatively modest performance gains in static or low-texture sequences. This is likely due to limited offset variation and low residual complexity in such content, which reduces the effectiveness of offset-based enhancement. In these scenarios, the model may slightly over-smooth fine details or introduce negligible improvements compared to SAO. The deep network appears biased toward learning high-frequency structures, favoring more complex motion or texture-rich regions. This highlights a limitation of the current training set and opens the door to further improvements through offset sparsity control or content-aware filtering. Despite this, the DiWaGAN-OBILF-HEVC approach remains robust and consistent across intra and inters prediction scenarios, ensuring enhanced prediction accuracy and visual quality under diverse conditions.

4.1 Performance metrics

The performance matrices, such as visual quality, bit-rate vs. PSNR, Bjontegaard rate difference (BD rate), and average execution times, are estimated to analyse the performance of the proposed approach.

4.1.1 Peak Signal to Noise Ratio (PSNR)

This is the main test employed to measure the robustness of the proposed method. It is divulged as in (7),

$$PSNR = 10\log_{10}\left[\frac{maax(I(i,j))^2}{MSE}\right] \tag{7}$$

where $maax\,(I(i,j))$ represents higher values in the image.

4.1.2 Bjontegaard rate difference

Bjontegaard rate difference represents the average bit-rate saves with similar quality.

$$BD\ rate = 100(e^k - 1) \tag{8}$$

where $k = \frac{\text{int}_2 - \text{int}_1}{y_2 - y_1}$

4.2 Performance evaluation

Figures 3a - 3d depict the efficacy of the proposed DiWaGAN-OBILF-HEVC method with metrics like visual quality, PSNR, Bjontegaard rate difference (BD rate), and average execution time. The obtained results are compared with existing models EDCNN-OBILF-HEVC [32], MDCNN-OBILF-HEVC [33] and DCNN-OBILF-HEVC [34].

Figure 3a shows the visual quality performance of proposed method and is compared with existing methods. The proposed method provides upto 16.7%, 39%, 35.6%, 46.58%, and 42.3% higher visual quality for the bit rates of 10000 kpbs, 20000 kpbs, 30000 kpbs, 40000 kpbs, and 50000 kpbs respectively.

Further, Figure 3b shows the performance analysis of PSNR wherein the proposed method is compared with the existing methods. The proposed method provides upto 44.75%, 39.51%, 41.77%, 36.18% and 35.67% higher PSNR for the bit rates of 10000 kpbs, 20000 kpbs, 30000 kpbs, 40000 kpbs and 50000 kpbs respectively.

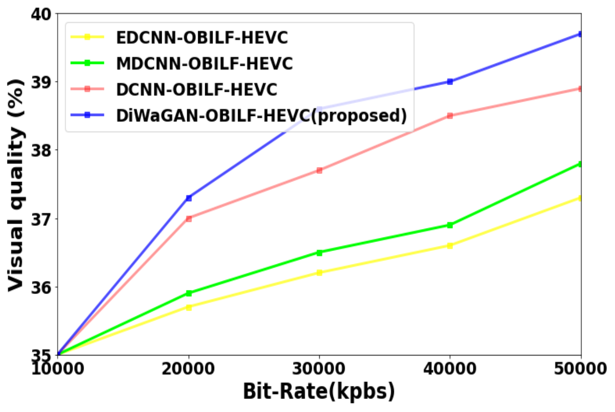
Furthermore, Figure 3c shows the analysis of Bjontegaard rate difference wherein the proposed method is compared with the existing methods. The proposed method provides upto 44.12%, 44.03%, 45.78%, 37.29% and 38.71% lower Bjontegaard rate difference for the bit rates of 10000 kpbs, 20000 kpbs, 30000 kpbs, 40000 kpbs, and 50000 kpbs respectively.

Finally, Figure 3d shows the average execution time of proposed method that is compared with existing methods. The proposed method provides upto 40.11%, 43.21%, 40.99%, 45.72%, and 45.39% lower execution time for bit rates of 10000 kpbs, 20000 kpbs, 30000 kpbs, 40000 kpbs, and 50000 kpbs respectively.

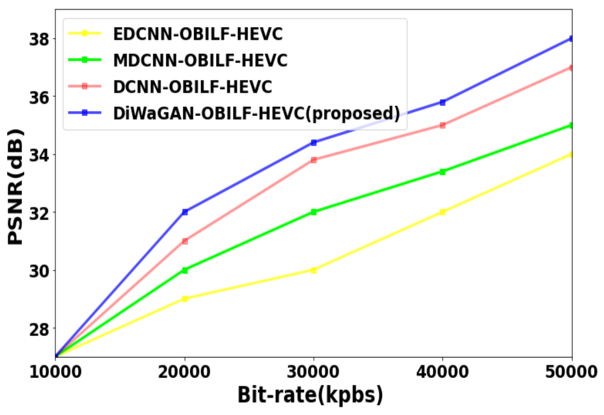
Hence, the outcomes show that the proposed DiWaGAN-OBILF-HEVC approach well performs the existing works such as. On the other hand, existing models mentioned occasionally reach a better presentation, but they provide negative gain. However, DiWaGAN-OBILF-HEVC attains better performance and is stable for all sequences and configurations. The DiWaGAN-OBILF-HEVC forecasts better for intra- and inter-coded deformation and the complicated characteristics of error are significantly better forecasted through the offsets. The DiWaGAN-OBILF-HEVC scheme shows better performance for PSNR, i.e., high-resolution sequences. Finally, for all rate points, the DiWaGAN-OBILF-HEVC performs better than the HEVC anchor. The suggested approach uses fewer bits. When compared to HEVC, the PSNR is higher. The improvement in visual quality is clearly visible. The DiWaGAN is able to forecast the complex pattern of the residual signal, which contains a ratio with a lesser QP than traditional SAO in HEVC. For higher QP or lower resolution, DiWaGAN reduces the count of SAO-New. According to the analysis, the deep network is trained to ignore less complex patterns in favour of highly complex ones.

4.3 Computational complexity analysis

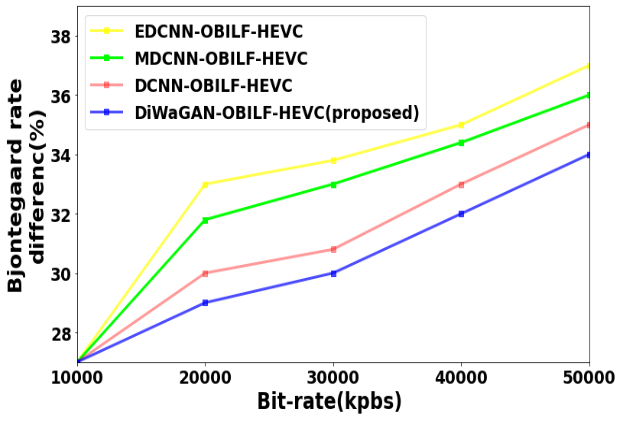
To assess the practicality of DiWaGAN-OBILF-HEVC for real-world deployment, its computational complexity in terms of training time, memory usage, and inference performance was evaluated and compared with existing deep learning-based in-loop filters. The model was trained on an NVIDIA RTX 3090 GPU (24GB VRAM) with an Intel i9 CPU and 64 GB RAM. Training for 200 epochs with a batch size of 32 required approximately 18 hours, which is moderately higher than MDCNN (14 hours) and slightly faster than DCNN-based models (21 hours), primarily due to the parallel training strategy of the dual generators. During inference, DiWaGAN processes 1080p frames at an average rate of ~18 frames per second (FPS), indicating its suitability for near real-time or offline processing applications such as video streaming and broadcasting. For higher resolution (e.g., 4K), frame rates dropped to approximately 9–11 FPS, suggesting that further model optimization (e.g., pruning, quantization-aware training) would be beneficial for real-time 4K scenarios.



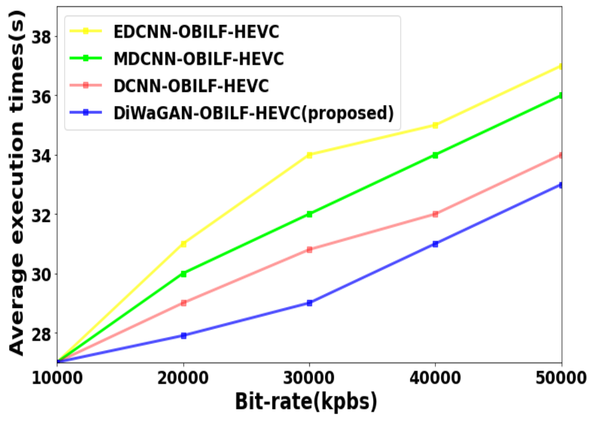
(a)



(b)



(c)



(d)

Figure 3: Comparison charts for the metrics (a) visual quality (b) PSNR (c) Bjontegaard rate difference and (d) Average Execution Time of proposed method along with existing methods

Memory consumption during training peaked at approximately 18 GB of VRAM, which is higher than EDCNN but comparable to other GAN-based methods. During inference, memory requirements are significantly reduced, making the system deployable on modern GPU-equipped edge servers. Overall, DiWaGAN demonstrates a balanced trade-off between quality enhancement and computational overhead, making it a viable solution for practical HEVC-based video systems, especially where quality is prioritized over ultra-low latency.

4.4. Discussion

The proposed DiWaGAN-OBILF-HEVC method demonstrates significant performance gains when evaluated against prior deep-learning-based in-loop filters. Compared with EDCNN-OBILF-HEVC, MDCNN- OBILF-HEVC, and DCNN-OBILF-HEVC, our method achieves superior PSNR and visual quality at various bitrates while maintaining lower Bjontegaard rate Difference (BD-rate) and reduced execution time. The integration of dual interactive Wasserstein GANs contributes to the higher PSNR by effectively modeling edge-aware residuals, while the use of optimized bilateral learning filtering ensures enhanced deblocking without sacrificing texture details. In terms of BD-rate, DiWaGAN achieves up to 46.58% reduction over DCNN-based approaches, which is a substantial improvement in compression efficiency. The execution time is also significantly lower, indicating that our model generalizes well without computational overhead scaling linearly with complexity. This makes it suitable for real-time video decoding scenarios. However, the dual-generator architecture adds training complexity and initial inference cost. This trade-off between training overhead and inference speed is balanced through feature fusion layers and adaptive loss selection in the Selector module. While DiWaGAN excels in high-motion and complex texture regions, its performance gain in static scenes is modest, likely due to limited offset variation. This suggests potential future work in adaptive offset sparsity control to further reduce bitrate in simpler content.

5 Conclusion and future work

Dual interactive Wasserstein Generative Adversarial Network fostered Offset-Based In-Loop Filtering in HEVC (DiWaGAN-OBILF-HEVC) is successfully implemented using Pytorch and its efficiency is evaluated under different performance metrics, like visual quality, PSNR, Bjontegaard rate difference (BD rate), and average execution time. The performance of the DiWaGAN-OBILF-HEVC method provides higher PSNR 15.86%, 15.26% and 16.25%; lower Bjontegaard rate difference 23.34%, 15.06%, and 27.47%; and lower execution time 12.566%, 12.075% and 15.993% when compared with the existing EDCNN-OBILF-HEVC [32], MDCNN-OBILF-HEVC [33] and DCNN-OBILF-HEVC [34] models respectively.

Future work will focus on improving the scalability and adaptability of DiWaGAN to other modern video codecs such as VVC (H.266), AV1, and AVS3. Since the architecture is modular and data-driven, it can be retrained or fine-tuned for different coding standards by adjusting offset generation logic and loss calibration accordingly. This flexibility opens avenues for applying the method to a broader range of in-loop filtering scenarios. Additionally, adapting the model to handle underspecified visual data such as low-light, noisy, or surveillance video poses a promising challenge. These scenarios may require enhanced robustness through techniques like content-aware filtering, uncertainty modeling, or self-supervised learning on low-resource codecs. Integration of lightweight variants of the model or quantization-aware training could further enable real-time deployment on embedded devices or edge computing hardware. Exploring these challenges will not only validate the generalizability of DiWaGAN but also enhance its practicality in real-world multimedia applications, including streaming, broadcasting, and video conferencing under varying network and hardware constraints.

Ethical approval

The authors are not aware of this article including any research that involves humans or animals.

Funding

We would like to state that this research was not funded by any agencies in the public, commercial, or non-profit sectors.

Declaration of competing interest

The authors state that they have no known competing financial interests or personal relationships that could affect the work described in this paper.

References

[1] Huang, Z., Sun, J., Guo, X., & Shang, M. (2021). One-for-all: An efficient variable convolution neural network for in-loop filter of VVC. *IEEE Transactions on Circuits and Systems for Video Technology, 32*(4), 2342–2355.

[2] Jia, W., Li, L., Li, Z., Zhang, X., & Liu, S. (2021). Residual-guided in-loop filter using convolution neural network. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 17*(4), 1–19.

[3] Ding, D., Ma, Z., Chen, D., Chen, Q., Liu, Z., & Zhu, F. (2021). Advances in video compression system using deep neural network: A review and case studies. *Proceedings of the IEEE, 109*(9), 1494–1520.

[4] Shao, T., Liu, T., Wu, D., Tsai, C. Y., Lei, Z., & Katsavounidis, I. (2022). PTR-CNN for in-loop filtering in video coding. *Journal of Visual Communication and Image Representation, 88*, 103615.

[5] Jia, M., Gao, Y., Li, S., et al. (2022). An explicit self-attention-based multimodality CNN in-loop filter for versatile video coding. *Multimedia Tools and Applications, 81*, 42497–42511. https://doi.org/10.1007/s11042-021-11214-2

[6] Ding, D., Kong, L., Wang, W., & Zhu, F. (2021). A progressive CNN in-loop filtering approach for inter frame coding. *Signal Processing: Image Communication, 94*, 116201.

[7] Ma, D., Zhang, F., & Bull, D. R. (2021). BVI-DVC: A training database for deep video compression. *IEEE Transactions on Multimedia, 24*, 3847–3858.

[8] Xu, Q., Jiang, X., Sun, T., & Kot, A. C. (2021). Detection of transcoded HEVC videos based on in-loop filtering and PU partitioning analyses. *Signal Processing: Image Communication, 92*, 116109.

[9] Su, L., Cao, M., Yu, Y., Chen, J., Yang, X., & Wu, D. (2023). Dynamic convolutional capsule network for in-loop filtering in HEVC video codec. *IET Image Processing, 17*(2), 439–449.

[10] Lee, J. K., Kim, N., Cho, S., & Kang, J. W. (2020). Deep video prediction network-based inter-frame coding in HEVC. *IEEE Access, 8*, 95906–95917.

[11] Wang, L., Xu, X., & Liu, S. (2022). Neural network based in-loop filter with constrained memory. *IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. https://doi.org/10.1109/ICME52920.2022.9859910

[12] Fu, C. M., Alshina, E., Alshin, A., Huang, Y. W., Chen, C. Y., Tsai, C. Y., Hsu, C. W., Lei, S. M., Park, J. H., & Han, W. J. (2012). Sample adaptive offset in the HEVC standard. *IEEE Transactions on Circuits and Systems for Video Technology, 22*(12), 1755–1764. https://doi.org/10.1109/TCSVT.2012.2221529

[13] Norkin, A., Bjøntegaard, G., Fuldseth, A., Narroschke, M., Ikeda, M., Andersson, K., Zhou, M., & van der Auwera, G. (2012). HEVC deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology, 22*(12), 1746–1754. https://doi.org/10.1109/TCSVT.2012.2223053

[14] Karimzadeh, H., & Ramezanpour, M. (2016). An efficient deblocking filter algorithm for reduction of blocking artifacts in HEVC standard. *International Journal of Image, Graphics and Signal Processing, 8*(11), 18–24. https://doi.org/10.5815/ijigsp.2016.11.03

[15] Shen, W., Fan, Y., Bai, Y., Huang, L., Shang, Q., Liu, C., & Zeng, X. (2016). A combined deblocking filter and SAO hardware architecture for HEVC. *IEEE Transactions on Multimedia, 18*(6), 1022–1033. https://doi.org/10.1109/TMM.2016.2532606

[16] Wang, D., Xia, S., Yang, W., & Liu, J. (2021). Combining progressive rethinking and collaborative

learning: A deep framework for in-loop filtering. *IEEE Transactions on Image Processing, 30*, 4198–4211.

[17] Liu, C., Sun, H., Katto, J., Zeng, X., & Fan, Y. (2020). A learning-based low complexity in-loop filter for video coding. *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 1–6. https://doi.org/10.1109/ICMEW46912.2020.9106015

[18] Fu, H., Liang, F., Liang, J., Li, B., Zhang, G., & Han, J. (2023). Asymmetric learned image compression with multi-scale residual block, importance scaling, and post-quantization filtering. *IEEE Transactions on Circuits and Systems for Video Technology*.

[19] Bidwe, R. V., Mishra, S., Patil, S., Shaw, K., Vora, D. R., Kotecha, K., & Zope, B. (2022). Deep learning approaches for video compression: A bibliometric analysis. *Big Data and Cognitive Computing, 6*(2). https://doi.org/10.3390/bdcc6020044

[20] Dong, L., Yue, L., Lin, J., Li, H., & Wu, F. (2020). Deep learning-based video coding: A review and a case study. *ACM Computing Surveys, 53*(1). https://doi.org/10.1145/3368405

[21] Narayanan, N. V., Arjun, T., & Logeshwari, R. (2023). Surveillance image super resolution using SR generative adversarial network. *Advances in Science and Technology, 124*, 125–136.

[22] Chen, W., Xiu, X., Wang, X., Chen, Y. W., Jhu, H. J., & Kuo, C. W. (2021). Quality-aware CNN-based in-loop filter for video coding. In *Applications of Digital Image Processing XLIV* (Vol. 11842, p. 1184203). SPIE. https://doi.org/10.1117/12.2593380

[23] Norkin, A. (2022). Generalized deblocking filter for AVM. *Picture Coding Symposium (PCS)*, 355–359. https://doi.org/10.1109/PCS56426.2022.10018081

[24] Idriss, E. M., Laghrib, A., Hadri, A., & Hakim, A. (2022). A theoretical study of a bilateral term with a tensor-based fourth-order PDE for image super-resolution. *Advances in Computational Mathematics, 48*(6), 83.

[25] Bordes, P., Galpin, F., Dumas, T., & Nikitin, P. (2021). Revisiting the sample adaptive offset post-filter of VVC with neural-networks. *Picture Coding Symposium (PCS)*, 1–5. https://doi.org/10.1109/PCS50896.2021.9477457

[26] Cheng, G., Matsune, A., Li, Q., Zhu, L., Zang, H., & Zhan, S. (2017). Encoder-decoder residual network for real super-resolution. Retrieved from https://github.com/

[27] Khani, M., Sivaraman, V., & Alizadeh, M. (2021). Efficient video compression via content-adaptive super-resolution. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4521–4530. https://doi.org/10.1109/ICCV48922.2021.00448

[28] Toderici, G., Vincent, D., Johnston, N., Hwang, S. J., Minnen, D., Shor, J., & Covell, M. (2017). Full resolution image compression with recurrent neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5306–5314. https://doi.org/10.1109/CVPR.2017.577

[29] Umale-Nagmote, A., Goel, C., & Lal, N. (2025). Enhanced intelligent video monitoring using hybrid integration of spatiotemporal autoencoders and convolutional LSTMs. *Informatica, 49*(18). https://doi.org/10.31449/inf.v49i18.7502

[30] Gong, J., & Xu, Q. (2025). Temporal transformer-based video super-resolution reconstruction with cross-modal attention. *Informatica, 49*(10). https://doi.org/10.31449/inf.v49i10.7146

[31] Moji, V., & Murugavelu, M. (2023). Adaptive and efficient hybrid in-loop filter based on enhanced generative adversarial networks with sample adaptive offset filter for HEVC/H-265. *Periodica Polytechnica Electrical Engineering and Computer Science, 67*(2), 216–228. https://doi.org/10.3311/PPee.20881

[32] Pan, Z., Yi, X., Zhang, Y., Jeon, B., & Kwong, S. (2020). Efficient in-loop filtering based on enhanced deep convolutional neural networks for HEVC. *IEEE Transactions on Image Processing, 29*, 5352–5366.

[33] Lee, S. Y., Yang, Y., Kim, D., Cho, S., & Oh, B. T. (2020). Offset-based in-loop filtering with a deep network in HEVC. *IEEE Access, 8*, 213958–213967.

[34] Bouaafia, S., Messaoud, S., Khemiri, R., & Sayadi, F. E. (2021). VVC in-loop filtering based on deep convolutional neural network. *Computational Intelligence and Neuroscience*.

[35] Dhanalakshmi, A., & Nagarajan, G. (2022). Group-normalized deep CNN-based in-loop filter for HEVC scalable extension. *Signal, Image and Video Processing*, 1–9.

[36] Zhang, J., et al. (2023). Textural and directional information based offset in-loop filtering in AVS3. *IEEE Transactions on Multimedia, 25*, 5957–5971. https://doi.org/10.1109/TMM.2022.3201747

[37] Sun, W., He, X., Chen, H., Xiong, S., & Xu, Y. (2022). A nonlocal HEVC in-loop filter using CNN-based compression noise estimation. *Applied Intelligence*, 1–19.

[38] Kuanar, S., Rao, K. R., Conly, C., & Gorey, N. (2021). Deep learning based HEVC in-loop filter and noise reduction. *Signal Processing: Image Communication, 99*, 116409.

[39] Huang, Z., Sun, J., Guo, X., & Shang, M. (2021). Adaptive deep reinforcement learning-based in-loop filter for VVC. *IEEE Transactions on Image Processing, 30*, 5439–5451.