[1] D. H. Kitty Smailin

[2] Dr Y. Jacob Vetha Raj

Error Resilience Video Streaming Technique Based on Hybrid Multiple Description Coding and Image Pyramid

Journal of Electrical Systems

Abstract: - Recently, video streaming gained popularity due to a lot of live streaming platforms with better Quality of Experience (QoE). However, transmitting video data over networks facing critical challenges because of maintaining video quality. With the goal of ensuring error resilience video streaming, this research proposes a novel Hybrid Multiple Description Coding and Image Pyramid (HMDC-IP) algorithm to enhance video transmission performance under varying network conditions. The MDC technique splits the video into multiple independently decodable descriptions, providing video recovery in case of packet loss. Simultaneously, the Image Pyramid method creates multiple resolution layers, allowing for flexible reconstruction of the video when data is lost. Notably, if all channels are received, then there will be no loss. Besides, polyphase permuting & splitting are applied to rearrange and split the image data into different polyphase components. After splitting, the data undergoes coefficient splitting which separates the image coefficients to allow finer control over encoding. These coefficients are then processed with arithmetic encoding to a form of entropy coding that compresses the data efficiently. On the decoding side, the descriptions are received and processed in reverse order. The system starts with arithmetic decoding, coefficient merger, polyphase inverse permuting & residual merger, and finally, the image is up-sampled through the Image Pyramid to restore the original resolution. Performance evaluation is carried out by measuring key metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Error Recovery Rate (ERR). The results demonstrate that the proposed HMDC-IP framework significantly improves video quality, minimizes packet loss impact, and enhances error resilience compared to traditional methods.

Keywords: Multiple Description Coding, Image Pyramid, Error Resilience, Video Streaming, Quality of Experience, Image Compression

I.INTRODUCTION

Error Resilience Video Streaming refers to the set of techniques applied to ensure that the quality and continuity of video streams are maintained when transmission errors, packet losses, or network instability arise [1]. Video streaming over unreliable networks such as the internet or wireless networks often suffers transmission interruptions caused by packet loss, latency, or jitter that decrease video quality [2]. For this purpose, several error-resilient techniques have been developed, with an emphasis on protecting video data against such disturbances so that there can be real-time playback even if there is some amount of data loss over some time. From simple error correction codes to more complex techniques like Multiple Description Coding, forward error correction, and redundancy-based schemes, the techniques have themselves been a passage of development over time [3] [4]. The goal of error resilience techniques is to detect data loss and recover from it to deliver smooth, interruption-free video with no perceptible degradation in video quality [5].

The main benefits of Error Resilience video streaming techniques are that the video received would look of better quality for periods of fluctuating network conditions, less buffering, and higher continuity of playback, especially in those environments where the rate of packet loss is high [6]. These methods also possess the property of low overhead data transmission, achieving a good balance between video quality and bandwidth usage. However, the error-resilient methods can be expensive in terms of computation, leading to increased latency. This is not very helpful for those applications that demand real-time processing [7] [8]. Other techniques may raise bandwidth usage and be less efficient in networks having constrained bandwidths. Part of the issue in designing an optimal error-resilient solution is a trade-off between the strength of error correction and video quality [9].

[1]* Research Scholar, Reg No. 19123112282009, Department of Computer Science, Nesamony Memorial Christian College, Marthandam affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-627 012, Tamil Nadu, India.

Corresponding Author email: kittydhnmcc@gmail.com

[2] Associate Professor, Department of Computer Science, Nesamony Memorial Christian College, Marthandam affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-627 012, Tamil Nadu, India.

Email: jacobvetharaj@gmail.com

Another emerging approach said to increase the resilience of errors in video streaming is Hybrid Multiple Description Coding (MDC) and Image Pyramid techniques. In MDC, the video stream is divided into multiple descriptions which are transmitted as different separate streams [10]. Although some of the descriptions are lost, it would still be possible to reconstruct a low-quality version of the video for use when there is resilience against packet loss. This hybrid scheme combines information from the image pyramid, representing images at different levels of detail, to permit efficient video transmission [11]. The pyramid of image offers the advantage of scalability within layers, whereby lower resolution information is transmitted first such that if higher layers are lost, there is always some sort of a residual, viewable level of video. Indeed, this hybrid approach proves to be effective in bandwidth limitation as well as video quality under variable network conditions [12].

Indeed, a number of advanced techniques for compressing and transmitting videos have been developed; some are strong in rate-distortion performance and others are weak. Examples of recent advances include the neural video codec using an efficient entropy model that captures spatial and temporal dependencies for improved rate-distortion performance with high computational demands, which limits its real-time applicability [13]. Mobile cloud gaming frameworks are adaptive to the changing network conditions to provide for both motion-to-photon latency and visual quality while suffering in adverse bandwidth or packet loss environments [14]. DeepWiVe is an end-to-end Joint Source-Channel Coding (JSCC) approach including video compression and bandwidth allocation of deep neural networks and reinforcement learning, but its complexity and dependency on correct channel estimation are challenges in dynamic networks. Ultimately, the error-resilient coding scheme for transmitting underwater video using convolutional neural networks and multiple description coding achieves a balance between coding efficiency and error resilience but can fail catastrophically in the worst-case packet loss scenarios or fail to generalize well to other settings [15]. Based on the above-mentioned disadvantages, a new hybrid scheme that integrates MDC with Image Pyramid techniques is designed towards increasing error resilience and adaptability that overcome the challenges currently faced by both these techniques. The key contributions are:

- To design an MDC model to enhance resilience by generating multiple independently decodable video streams. In case of partial data loss, the video quality is preserved through available descriptions, ensuring smoother playback despite packet loss.
- To construct an Image pyramid to generate multi-resolution layers of the video. Lower-resolution layers enable partial reconstruction when higher-resolution frames are unavailable, maintaining video continuity under degraded conditions through adaptive scaling.
- Develop an efficient HMDC-IP encoding and decoding system capable of dividing the input source data into multiple descriptors and reconstructing the original data from these descriptors.

This article is structured as a recent literature on video streaming techniques in Section II. Section III explains the proposed architecture. Experimentation and results are given in Section IV. Section V concludes the research.

II.LITERATURE STUDY

A. Recent Research

In 2022, Li et al [16] presented a new entropy model for neural video codecs that measured spatial and temporal correlations to improve the prediction of probability distribution of quantized latent representations. For temporal redundancy reduction, it has a latent prior and for spatial redundancy reduction, it has a dual spatial prior. Also, the proposed model included a content-adaptive quantization mechanism which enabled smooth rate control and dynamic bit distribution, enhancing rate-distortion efficiency. The experimental outcomes revealed that the proposed neural video codec achieves 18.2% lower bitrate than H.266 (VTM) at the highest compression level on the UVG dataset.

In 2022, Alhilal et al [17] proposed an end-to-end cloud gaming framework for improving user experience in mobile cloud gaming by controlling the impact of volatile network conditions. The framework adjusted video source rates and frame-level redundancy depending on real-time network metrics to reduce motion-to-photon (MTP) delay and shield frames from loss. When applied and evaluated against current approaches, the framework maintained an MTP latency of less than 140 ms and a visual quality of more than 31 dB in all circumstances.

In 2022, Tung and Gündüz [18] suggested DeepWiVe, a new end-to-end JSCC video transmission technique that utilizes DNNs to combine video coding, channel coding, and modulation. Some of the features include a DNN

decoder that estimated residual without feedback of distortion, which helped to improve video quality in cases of occlusion, and camera movement. The scheme also employed the RL technique to adapt bandwidth allocation between video frames hence enhancing picture quality. As shown in the previous simulation, DeepWiVe has better performance than the H.264 and H.265+ with LDPC codes which avoided the cliff effect in classical communication systems.

In 2021, Zhang and Gu [19] developed an error-resilient coding method for underwater video transmission that combined CNN with MDC to deal with transmission errors and packet losses. Due to the use of information derived from inter-frame motion, the approach improved the safety of significant areas of the video. It involved digitizing video sequences into two forms of descriptions within a given bit rate constraint. Using underwater video datasets with various packet loss rates for simulation, the authors proved the efficiency of the method and its higher efficiency compared to other video coding methods.

In 2021, Minallah et al. [20] proposed the transmission scheme of H.264/AVC compressed video streams using IRCC with multidimensional Sphere Packing (SP) modulation and Differential Space-Time Spreading Codes (IRCC-SP-DSTS). It evaluated three error protection strategies: They included Regular Error Protection (REP), Irregular Error Protection scheme-1 (IREP-1), and Irregular Error Protection scheme-2 (IREP-2) in which video data was protected differently. The performance of these methods was evaluated using the EXtrinsic Information Transfer (EXIT) Chart and other parameters such as Bit Error Rate (BER) and PSNR. The results demonstrate that the IREP-2 scheme has an additional 1 dB E_b/N_0 gain over IREP-1 and 0.6 dB over REP at the expense of a 1 dB PSNR loss.

In 2021, Hu et al. [21] presented a dynamic adaptive Light Field (LF) video transmission scheme to obtain high compression and near-distortion-free LF video in a stable network environment. Moreover, it presented a description scheduling algorithm to improve the quality of the video in case of partial data loss and delay. This was done using the MDC strategy, in which a novel Graph Neural Network (GNN) model was employed for LF video compression. The experiment results showed that the scheduling algorithm increased the decoding quality by 3% to 15% increased the robustness of the video streaming system against packet loss or errors and supported different types of receivers.

In 2022, Wang et al. [22] suggested a novel framework of Spatial-Frequency Multiple Description Video Coding (SF-PMDVC) was proposed for HEVC. It improved coding efficiency by using an adaptive perceptual redundancy allocation scheme in accordance with visual saliency. The experimental outcome showed that the proposed SF-PMDVC scheme outperformed other MDC techniques in terms of error tolerance and reconstructed video quality.

In 2022, Li et al. [23] presented an UnderWater-WZ which was an enhanced Wyner-Ziv coding scheme for video transmission over underwater acoustic channels. The approach adopted the use of MJPEG coding to manage the error range while using motion compensation time interpolation together with calibration information to create good side information. Experimental outcomes showed that this scheme enhanced the video reconstruction quality by 2.6 to 3.5 dB when the maximum packet loss ratio was 20% which was very close to the error-free condition.

In 2022, Alaya and Sellami [24] proposed VSMENET as a new inter-layer approach for transmission of multimedia streams in VANETs. Its main goal was to constantly adjust transmission rates with reference to the physical rate within the network to minimize instances of video playback for the users. This included enhancing video quality, smart encoding on RSUs, and coordinating computation tasks and video data accessibility. When compared with the current techniques in the NetSim simulator, VSMENET has achieved more than 9% enhancement in the average video packet lifetime and delivery rate.

In 2022, Zhao et al. [25]described a low bit-rate image compression using MDC that addressed the rate-distortion minimization problem at its deepest level. It comprised an MD multi-scale encoder, MD cascaded-Resblock decoders, and arithmetic coding employing learnable scalar quantizers and conditional probability models. The framework synthesised and quantised MD tensors and reconstructed images with cascaded-Resblock networks with the help of which the framework has a parameter-sharing symmetric structure to reduce the network complexity.

B. Problem Statement

Table I presents the advantages and disadvantages of various methods for video streaming. The rapid growth of video streaming services, along with the widespread adoption of high-bandwidth networks like 5G, has made ensuring reliable and high-quality video delivery over fluctuating networks a critical challenge. Traditional video streaming methods struggle to maintain quality under dynamic conditions, such as packet loss, bandwidth variations, and network congestion. Recent advances in AI have further enhanced these methods by enabling dynamic bitrate adaptation and error correction strategies, optimizing video quality based on real-time network conditions. However, despite these advances, deploying these techniques for real-time video streaming continues to face significant challenges. Video streaming algorithms typically struggle with high computational complexity and real-time adaptation. Optimizing the trade-off between video quality, bandwidth efficiency, and computational load remains particularly challenging in heterogeneous 5G networks, where bandwidth fluctuations can be rapid and unpredictable. Additionally, ensuring low-latency processing while maintaining high QoE across various network environments remains an unresolved issue, necessitating the development of novel strategies for better video streaming.

Table I: Achievements and Limitations of Various Methods for Video Streaming

Authors, Year	Technique	Database	Advantage	Disadvantage
Li *et al*, 2022	Entropy model leveraging spatial and temporal dependencies	UVG dataset	Reduced temporal and spatial redundancy, adaptive quantization	Complexity in implementation
Alhilal *et al*, 2022	End-to-end distortion model for mobile cloud gaming	Real & emulated wireless networks	Balanced MTP latency and visual quality, adapted to varying network conditions	High network variability impacted performance
Tung and Gündüz, 2022	JSCC with DNN	Tested on H.264, H.265, LDPC	Overcame cliff effect, graceful degradation, optimized bandwidth allocation	Required specific training for each channel condition
Zhang and Gu, 2021	Error-resilient coding with CNN and multiple descriptions	Underwater video datasets	Efficient handling of packet loss, extra protection for regions of interest	Balancing between coding efficiency and error resiliency was complex
Minallah *et al*, 2021	IrRegular Convolutional Codes (IRCC) with SP	H.264/AVC	Improved PSNR and BER metrics, priority-based protection schemes	Complexity in choosing protection schemes
Hu *et al*, 2021	Dynamic adaptive LF video transmission using GNN and MDC	LF video datasets	High compression efficiency, adapted to network conditions, reduced packet loss effects	Difficulty in implementation due to complex data structures
Wang *et al*, 2022	SF-PMDVC	HEVC	Reduced redundancy, improves coding efficiency and perceptual video quality	Limited application to specific coding structures in HEVC
Li *et al*, 2022	Improved Wyner-Ziv coding scheme for deep-sea communication	Underwater acoustic channels	Solved long-distance wireless communication issues, controlled error range	Limited to underwater video transmission scenarios

Alaya and Sellami, 2022	Inter-layer multimedia stream transmission in VANET (VSMENET)	NetSim simulator	Eliminated downtime, dynamic video quality adaptation, efficient data management	High dependency on network conditions
Zhao et al, 2022	Deep MDC for image compression	Custom datasets	Optimized rate-distortion minimization, adaptive quantization for spatial variation	Complexity in training and implementation

III.AN OPTIMAL VIDEO STREAMING MODEL

A. Proposed Architecture

Fig. 1 shows the overview of the proposed video streaming model. This research introduces a novel error-resilient video streaming model HMDC-IP that employs MDC and an Image Pyramid framework to improve video transmission performance in fluctuating network conditions. The HMDC technique divides the video into multiple independently decodable descriptions, enabling partial recovery when packet loss occurs. Concurrently, the Image Pyramid method generates multiple resolution layers, allowing flexible reconstruction at reduced quality if higher-resolution data is lost. In case, if all channels are received, then there will be no loss. Furthermore, polyphase permuting & splitting are applied to rearrange and split the image data into different polyphase components. Then, coefficient splitting takes place to separate the image coefficients which allows finer control over encoding. These coefficients are further processed with an arithmetic encoding that forms an entropy coding by compressing the data efficiently. The output of this process consists of several independent descriptions such as Descriptor $(0,1,\dots,N)$ which are transmitted independently over the network. On the decoding side, the descriptions are received and processed in reverse order. The system starts with arithmetic decoding to reconstruct the encoded coefficients. Besides, a coefficient merger is added to merge the coefficients and the polyphase inverse permuting & residual merger is used to recombine the polyphase components and residual information. Eventually, Image Pyramid up-sampling is performed to restore the original resolution. Simulations are conducted under various network conditions, evaluating key performance indicators such as PSNR, SSIM, and ERR.

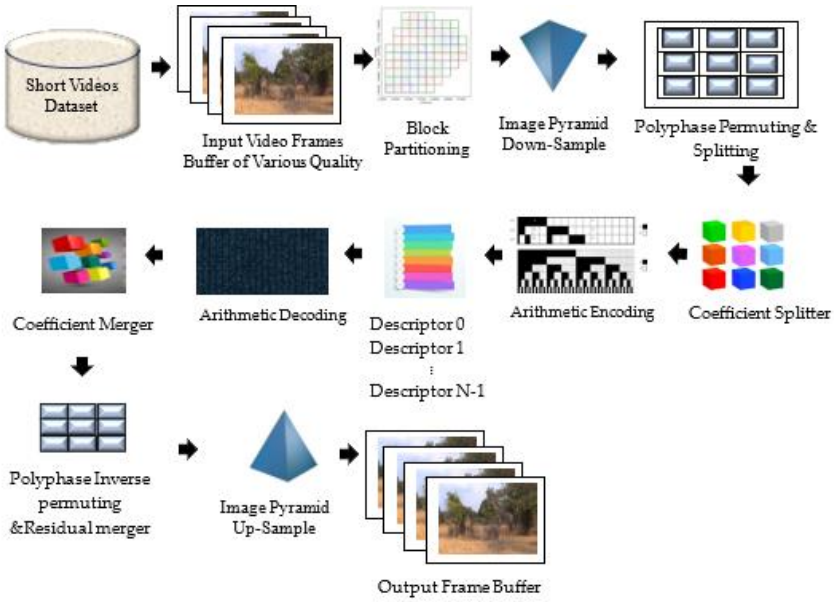


Fig 1: Overview of the Proposed HMDC-IP Model

B. Data Collection

For comprehensive testing of HMDC-IP, a variety of pre-recorded video files are collected to simulate real-world scenarios. This is achieved through a custom video collection from Short Videos accessible through

https://www.kaggle.com/datasets/mistag/short-videos [Access Date: 03/10/2024], ensuring diversity in content and technical specifications.

Camera/Device Capture: Videos are captured using high-definition cameras, and mobile phones. This method enables the collection of original, uncompressed video files, which are critical for testing the full potential of HMDC-IP. The captured videos reflect a range of environments and lighting conditions to test the robustness of the HMDC-IP model in different scenarios.

Standard Formats: To ensure compatibility with commonly used video processing tools, the captured videos are saved in standard formats such as MP4.

Resolution Variations: It is important to capture videos in multiple resolutions, such as 360p, 720p, 1080p, and 4K, to assess how HMDC-IP performs across different quality settings. Lower resolutions are used to simulate bandwidth-limited environments, while higher resolutions test the system's ability to maintain video quality under optimal conditions.

Content Types: To fully evaluate HMDC-IP's adaptability, the video dataset includes a variety of content types like nature documentaries.

C. Proposed HMDC-IP Encoding Procedure

In this approach, MDC with an Image Pyramid framework is combined to enhance error resilience and video transmission in dynamic network environments. The process is divided into five key phases including block partitioning, Image Pyramid downsampling, polyphase permuting and splitting, coefficient splitter, and arithmetic encoding and the corresponding inverse procedures and final output reconstruction. Each phase contributes to the overall system's robustness, flexibility, and efficiency.

Frame Extraction: Each video is broken down into individual frames for easier processing as shown in Eq. (1), where f_i states individual frames in the video sequence, and f_n denotes the total number of frames.

$$f_i = \{f_1, f_2, \dots, f_n\} \tag{1}$$

HMDC Design: It is a robust video coding technique used to enhance error resilience during network transmission [26]. HMDC divides the video stream into multiple independent descriptions, enabling partial recovery of the video even when some descriptions are lost. This technique ensures that instead of complete failure, video quality degrades gracefully in the presence of packet loss or network fluctuations.

HMDC Encoder: The encoder divides the video stream into multiple descriptions, each of which is decoded independently. This allows for graceful degradation, meaning that even if some descriptions are lost during transmission, the remaining descriptions can still be used to reconstruct the video at reduced quality. Let the original video stream be denoted as V. It is divided into N independent descriptions $D_1, D_2, \dots, D_N$ as shown in Eq. (2), in which D_i signifies i^{th} description.

$$V = \sum_{i=1}^{N} D_i \tag{2}$$

Each description D_i represents different aspects of the video, such as different frequency components, spatial regions, and frames at varying resolutions. Different descriptions are assigned to spatial and frequency components. For example, one description encodes low-frequency details, while another encodes high-frequency details. Also, each description is encoded separately and transmitted over independent network paths. The encoder's function is to transform the input video into multiple independent descriptions that are efficiently transmitted and decoded.

Block Partitioning: For HMDC, the video frames need to be divided into small blocks [27] to simulate network transmission as stated in Eq. (3), in which F represents the video frame (comprising multiple descriptions in the HMDC setup), P_i refers to i^{th} packet, representing a segment of the video frame, where $i = 1,2, \dots, n$, and n indicates the total number of packets into which the frame is divided.

$$F = \{P_1, P_2, \dots, P_n\} \tag{3}$$

For a given video frame F with size S (in bits), and assuming each block can hold S_p bits, the total number of packets n is calculated as per Eq. (4), where

$$n = \frac{S}{S_p} \tag{4}$$

Eq. (5) defines each block P_i containing a portion of the video frame data.

$$P_i = F\left[i \times S_p : (i+1) \times S_p\right] \tag{5}$$

Eq. (5) represents the portion of the frame that corresponds to the i^{th} packet, from the starting index $i \times S_p$ to the end index $(i+1) \times S_p$. At this point, the output block is non-overlapping 16×16 macroblocks.

Image Pyramid Downsampling: To further improve error resilience and adaptive quality control, an Image Pyramid Framework [28] is applied to each description. The image pyramid is a multi-resolution representation of the video where different layers represent the video at different resolutions. For each description D_i, an image pyramid is generated by downsampling the video into multiple layers of varying resolutions. Let L_0 represent the original resolution of the description, and L_k represent the k^{th} downsampled layer, where k indicates the level of downsampling as given in Eq. (6). Specifically, higher k means more downsampling and a lower resolution.

$$L_k(D_i) = Downsample(D_i, 2^k) \tag{6}$$

Here, $L_k(D_i)$ signifies k^{th} layer of the pyramid for description D_i, and $Downsample(D_i, 2^k)$ denotes downsampling D_i by a factor of 2^k. The downsampling process allows flexibility in reconstructing the video at different quality levels, depending on how much data is received. After downsampling, 8×8 blocks are obtained within each macroblock. Now, the HMDC-IP involves a two-level splitting process such as Polyphase permuting and splitting, and coefficient splitter.

Polyphase Permuting and Splitting: It is used to divide the 8×8 image block into multiple sub-blocks which are then encoded independently [29]. For an image block $I(u, v)$, the polyphase components are extracted by sampling the image at regular intervals. Now, the image $I(u, v)$ is decomposed into multiple sub-images $I_i(u, v)$, where each sub-image corresponds to one of the polyphase components as shown in Eq. (7), in which *N* refers to number of components i.e, $(0\ to\ 3)$, and $I_i(u, v)$ represents each polyphase component.

$$I(u, v) = \sum_{i=0}^{N-1} I_i(u, v) \tag{7}$$

At this point, labelling the pixels in each sub-image $I_i(u, v)$ to create a new sequence (2×2) group that helps improve error resilience. In HMDC, this step helps distribute the spatial and frequency components of the image across different descriptions. Here, polyphase sampling permutes the image blocks into a 4-description split, the pixel at the location (u, v) of the image $I(u, v)$ is distributed across four different polyphase components such as label 0 which is assigned to the pixel in the top-left corner of each 2×2 group, label 1 which is assigned to the pixel in the top-right corner, label 2 which is assigned to the pixel in the bottom-left corner and label 3 which is assigned to the pixel in the bottom-right corner as stated in Eq. (8), where I_0, I_1, I_2, and I_4 indicates the polyphase components.

$$I_i(u, v) = [I_0(u, v), I_1(u, v), I_2(u, v), I_3(u, v)] \tag{8}$$

Polyphase permuting [29] rearranges the pixels with the same labels into specific sub-blocks of the 8×8 block as follows:

- Pixels with label 0 are moved to form the top-left 4×4 block of the 8×8 block.
- Pixels with label 1 are moved to form the top-right 4×4 block.
- Pixels with label 2 are moved to form the bottom-left 4×4 block.
- Pixels with label 3 are moved to form the bottom-right 4×4 block.

The permuting process is performed before splitting, and it uses a lost description estimation approach. This means that the rearrangement is done in a way that helps to recover lost data if some descriptions are missing during transmission. By redistributing the pixel information across different descriptions, the system better estimates and recovers lost data. Polyphase splitting takes the permuted image and divides it into separate descriptions. Each

8×8 block is split into two sub-blocks, referred to as coefficient splitter A0 and coefficient splitter A1. The permuted block is then divided into sub-descriptions in such a way that each description contains unique information, and all descriptions together fully reconstruct the original image as formulated in Eq. (9), $P_i\big(I(u,v)\big)$ means a function that permutes and selects the i^{th} polyphase component from the image.

$$D_i = P_i\big(I(u,v)\big), \quad \forall i \in \{0,1,\dots,N-1\} \tag{9}$$

Thus, each description D_i is an independent representation of the image $I(u,v)$.

Coefficient Splitter: After polyphase permuting, a coefficient splitter takes place [30]. The 8×8 block is split into two diagonal blocks, $A0$ and $A1$, as part of the first-level splitting. Here, $A0$ contains the top-left and bottom-right 4×4 blocks from the permuted 8×8 block. $A1$ contains the top-right and bottom-left 4×4 blocks from the permuted 8×8 block. In the case of any remaining 4×4 blocks that are not part of the primary two sub-blocks ($A0$ and $A1$), those pixels are set to zero (all-zero residual pixels). This means that the remaining blocks, which are marked with an "x" in the process, are filled with zero values. The encoder does not need to encode these blocks because they do not contain any meaningful data i.e., all their pixels are zero. The splitting process of the 8×8 block is shown in Eq. (10).

$$B = \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix} \tag{10}$$

Similarly, $A0$, and $A1$ blocks carry the essential visual data diagonally as represented in Eq. (11), in which B_{ij} denotes each 4×4 sub-block.

$$A0 = \begin{bmatrix} B_{00} & 0 \\ 0 & B_{11} \end{bmatrix}, A1 = \begin{bmatrix} 0 & B_{01} \\ B_{10} & 0 \end{bmatrix} \tag{11}$$

For each block, $A0$ and $A1$ are further split diagonally in the second-level splitting. The process creates $B0$ and $B1$ blocks, referred to as the even (E_{blocks}) and odd (O_{blocks}). For block $A0$, this results in $A0B0$ which contains values from Block 0 (top-left) and Block 3 (bottom-right) of $A0$, and $A0B1$ which contains the remaining values from the diagonally opposite elements of $A0$. Similarly, for $A1$, this results in $A1B0$ which contains values from Block 1 and Block 2 of $A1$, and $A1B1$ which contains the diagonally opposite elements of $A1$. Now, the encoded data consists of the compressed representation of each block ($A0B0, A0B1, A1B0$ and $A1B1$), the offset values $\mathcal{O}$ and the residual coefficients R. After the splitting, the blocks undergo upsampling to reduce residue so as to minimize error in image reconstruction. The next step is to calculate the residual values by subtracting the reconstructed block ($A0B0, A0B1, A1B0$ and $A1B1$) from the original 16×16 block i.e., $\mathcal{O}_{16\times16} = \mathbb{O}_{8\times8} - \mathcal{R}$, in which $\mathcal{O}_{16\times16}$ includes original 16×16 block, $\mathbb{O}_{8\times8}$ defines original 8×8 block, and $\mathcal{R}$ refers to reconstructed block. The residue provides information about the remaining error after reconstruction. The offset value is calculated as the absolute minimum of the residue. This offset is needed to efficiently represent the error in a compressed form i.e., $\mathcal{O} = \min(|\mathbb{R}|)$. As a result of two-level splitting, every residual macroblock is split into four macroblocks, one for each descriptor.

Arithmetic Encoding: It is a sophisticated method of lossless compression that encodes a sequence of symbols into a single floating-point number [31]. The four descriptors including $A0B0, A0B1, A1B0$, and $A1B1$ represent different blocks of data that are processed through splitting and residual calculations. These descriptors are then compressed using arithmetic encoding. Each descriptor ($A0B0, A0B1, A1B0$, and $A1B1$) is treated as a symbol in the input stream. The blocks are distributed equally across the descriptors, ensuring balanced quality among the macroblocks. This helps prevent any one descriptor from becoming too heavily weighted and degrades the compression efficiency or the visual quality of the decoded frames. Besides, arithmetic encoding dynamically adjusts the range for each descriptor based on its probability in the data stream. Now, the first block is assigned to the first descriptor, the second one is assigned to the second descriptor, etc. The blocks are equally distributed in each macroblock in order to make the resulting descriptors that have balanced quality. Algorithm 1 shows the pseudocode of the HMDC-IP encoding procedure. Fig. 2 displays the HMDC-IP encoding procedure.

Algorithm 1: Pseudocode of HMDC-IP Encoding Procedure	
Input	A frame from the input buffer

Output	Compressed Descriptors ($A0B0, A0B1, A1B0, A1B1$), arithmetic-encoded
Step 1	Block Partitioning Divide the input frame into non-overlapping 8×8 blocks for processing.
Step 2	Image Pyramid Down Sampling Downsample the image using a pyramid-based approach. Reduce the 8×8 blocks into 4×4 blocks while retaining important features.
Step 3	Polyphase Permuting & Splitting Label the pixels in the 8×8 block with numbers 0, 1, 2, 3 (for each 2×2 pixel group). Rearrange pixels into 4×4 blocks. Pixels labeled 0 go to the top-left 4×4 block. Pixels labeled 1 go to the top-right 4×4 block. Pixels labeled 2 go to the bottom-left 4×4 block. Pixels labeled 3 go to the bottom-right 4×4 block.
Step 4	Coefficient Splitting 1. First-Level Splitting Split the permuted 8×8 block into two coefficient blocks $A0$ and $A1$. Each of these blocks contains two 4×4 blocks (diagonally chosen) 2. Second-Level Splitting Split each 4×4 block into two groups, creating O_{blocks} and E_{blocks}. Distribute these diagonally to form macroblocks $A0B0, A0B1, A1B0$, and $A1B1$.
Step 5	Residue Calculation for Error Minimization Find the Up-sampled values in each block Calculate residue R Estimate offset O
Step 6	Arithmetic Encoding Descriptor Generation Create descriptors $A0B0, A0B1, A1B0$, and $A1B1$ from the macroblocks. Arithmetic Encoding For each descriptor, apply arithmetic encoding. Assign shorter codes to more frequent descriptors and longer codes to less frequent descriptors based on symbol probabilities. Compress the encoded data.

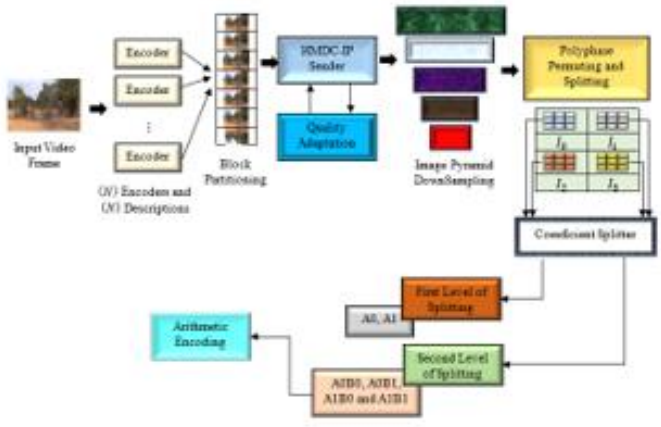


Fig 2: Encoding Procedure of Proposed HMDC-IP Model

D. Proposed HMDC-IP Decoding Procedure

The proposed HMDC-IP decoder is responsible for reconstructing the video from the descriptions received over the network. In the HMDC-IP framework, each description is decoded independently, and the

decoder partially reconstructs the video even when some descriptions are lost due to packet loss or transmission errors. The decoder's functionality includes gracefully degrading the video quality as fewer descriptions are received. The decoding process follows the reverse operations of encoding to recover the original data. It involves arithmetic decoding, coefficient merging, and polyphase inverse permuting to reconstruct the image.

Arithmetic Decoding: The first step is to decode the encoded data. Arithmetic decoding reconstructs the original sequence of symbols using the same probabilities and coding ranges applied during encoding [32]. Each descriptor ($A0B0, A0B1, A1B0$, and $A1B1$) is independently decoded. Let $S = \{s_1, s_2, \dots, s_n\}$ be the original coefficients used in the image. During decoding, the arithmetic decoder uses the encoded number E_{blocks}, along with the symbol probabilities, to reverse the encoding process and recover the original sequence S. For each symbol s_i, a range is determined based on cumulative probabilities $P(s_i)$, and this range is used to identify the decoded value. If $R = [l, h]$ is the range determined by the encoder for the symbol, the next symbol is decoded by determining its corresponding probability as given in Eq. (12), and (13), in which $P_{low}(s_i)$ and $P_{high}(s_i)$ states cumulative probabilities of symbol s_i.

$$l_i = 1 + (h - l) \cdot P_{low}(s_i) \tag{12}$$

$$h_i = 1 + (h - l) \cdot P_{high}(s_i) \tag{13}$$

For each of the four descriptors *($A0B0, A0B1, A1B0$,* and *$A1B1$)*, this process is repeated to recover their respective pixel coefficients.

Coefficient Merger: Once the descriptors are decoded, the coefficient merger step is applied. This step reverses the coefficient splitting done during encoding. Let C_1 and C_2 represent the two blocks that were split during encoding. These coefficients are merged in pairs to reconstruct larger blocks. For each descriptor, a pair of blocks is merged, which involves adding residual data to reconstruct the original coefficients. For the merging of coefficients C_1 and C_2 from descriptor $A0B0$ and $A0B1$, Eq. (14) is used and this process is repeated for all the descriptors.

$$C_{merged} = C_1 + C_2 \tag{14}$$

Polyphase Inverse Permuting and Residual Merger: Polyphase permutation is a reordering of coefficients during encoding to ensure better compression. The inverse polyphase permuting step during decoding restores the coefficients to their original positions in the block. Next, the residual data merge process takes place. During encoding, residuals were calculated and encoded. The decoded residual values from descriptors (after arithmetic decoding) are added back to the merged coefficient blocks. Let P^{-1} denote the inverse polyphase permuting function and the original block B_{ij} is restored by applying inverse permutation as expressed in Eq. (15), where B_{ij} defines 8×8 block obtained by inversely permuting the $A0$ and $A1$ blocks.

$$B_{ij} = P^{-1}(A0, A1) \tag{15}$$

In the case where the decoder does not receive all the descriptors (for example, when one of the blocks such as $A1B1$ is missing), frequency estimation is used to predict the lost values. This prediction is based on neighboring blocks and the residual domain. If $A1B1$ is lost, the neighboring block $A0B1$ is used to predict the missing values. The idea here is that O_{blocks} of $A1B1$ are predicted from O_{blocks} of $A0B1$ and E_{blocks} of $A1B1$ are predicted from E_{blocks} of $A0B1$ is explained in Eq. (16), where ε_O and ε_E signifies small prediction errors based on local characteristics.

$$O_{A1B1} = O_{A1B1} + \varepsilon_O,\ E_{A1B1} = E_{A1B1} + \varepsilon_E \tag{16}$$

In addition to prediction based on neighboring blocks, residual domain prediction further refines the accuracy of estimation. The residual block R represents the difference between the original block and the predicted block as shown in Eq. (17), where R_{A1B1} indicates residual for block $A1B1$, B_{A1B1} addresses original block, and $B_{predicted}$ specifies predicted blocks based on neighboring blocks.

$$R_{A1B1} = B_{A1B1} - B_{predicted} \tag{17}$$

The residual domain is critical for estimating missing values in the block, as residual data contains finer details of the image. The residual values are then encoded with an offset value. The offset value is the minimum absolute value from the residual block as stated in Eq. (18).

$$Offset = \min(|R|) \tag{18}$$

This offset is applied to compress the residual values. All four cases of missing descriptors $(A0B0, A0B1, A1B0,$ and $A1B1)$ are predicted in the horizontal direction. For missing O_{blocks}, prediction comes from the left-neighboring O_{blocks}. For missing E_{blocks}, prediction comes from the right-neighboring E_{blocks}. This ensures that the blocks are reconstructed even when some descriptors are missing by using the structure of neighboring blocks.

Image Pyramid Upsampling: During decoding, the received descriptions are reconstructed through an upsampling process from the Image Pyramid. If full descriptions are received, the highest-resolution layers are reconstructed. If only partial descriptions are received, lower-resolution layers are used for reconstruction. If only the k^{th} layer is received, the decoder upsamples it back to the original resolution using an interpolation technique as stated in Eq. (19), where $L'_0(D_i)$ refers to an upsampled version of the k^{th} layer $L_k(D_i)$, reconstructed to the original resolution, and $Upsample(L_k(D_i), 2^k)$ addresses upsampling the k^{th} layer by a factor of 2^k.

$$L'_0(D_i) = Upsample(L_k(D_i), 2^k) \tag{19}$$

The upsampling process attempts to reconstruct the original resolution from the lower-resolution layers. Fig. 3 shows the downsampling and upsampling process in Image Pyramid.

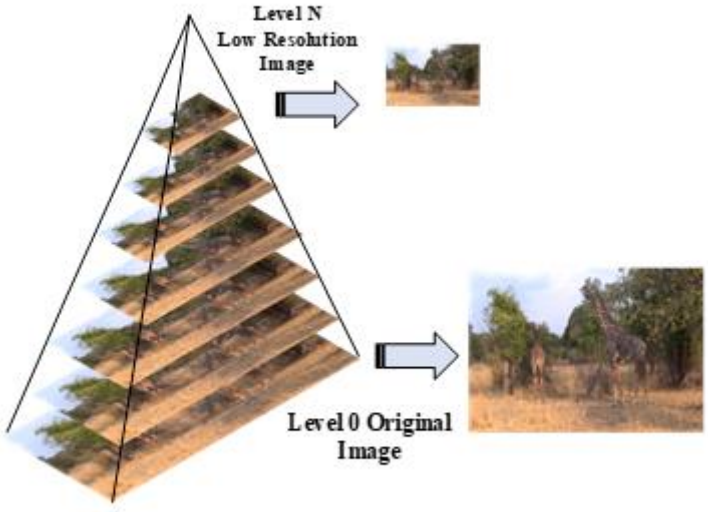


Fig 3: Image Pyramid Procedure in Proposed HMDC-IP

Output Video Frame Buffer: Let N represent the total number of descriptions, and let M be the number of received descriptions where $M \leq N$. Each received description, denoted by D_i (for $i = 1,2, \dots, M$), contains a unique portion of the video content, and the goal is to reconstruct the video using these M descriptions. The total video stream V is perfectly reconstructed if all descriptions $\{D_1, D_2, \dots, D_N\}$ are received. The reconstructed video $V_{received}$ is expressed in Eq. (20).

$$V_{received} = \sum_{i=1}^{N} D_i \tag{20}$$

However, when fewer than N descriptions are received, that is M descriptions, the video can still be reconstructed at a lower quality. The video decoder uses the available M descriptions to partially recover the video as shown in Eq. (21), in which $V_{received}$ means video reconstructed from the received descriptions, M represents a number of received descriptions (which is less than or equal to N), and D_i signifies i^{th} description received.

$$V_{received} = \sum_{i=1}^{M} D_i \quad (21)$$

If fewer descriptions are received, the reconstructed video will be of lower quality, as less information is available to the decoder. The quality of the video is proportional to the number of descriptions received. For instance, if all descriptions N are received, the video quality is maximum. On the other hand, if only M descriptions are received, where $M < N$, the video quality will degrade but still be viewable. The level of degradation is a function of the number of missing descriptions as formulated in Eq. (22), in which Q_{video} states quality of the reconstructed video, which depends on the number of received descriptions M out of the total N.

$$Q_{video} = f(M, N) \quad (22)$$

Depending on the number of lost packets, the video stream is fully or partially reconstructed. The video quality Q_{video} of the reconstructed video is modelled as a function of the number of received descriptions M and the total number of descriptions N as shown in Eq. (23), in which Q_{max} specifies maximum possible video quality (achieved when all N descriptions are received), M denotes number of descriptions actually received, and N indicates total number of descriptions in the HMDC scheme.

$$Q_{video} = Q_{max} \cdot \left(\frac{M}{N}\right) \quad (23)$$

In the case of lost descriptions due to network issues, the decoder adapts by reconstructing the video using the remaining descriptions. The final video stream quality is determined by the amount of lost data that is recovered. Finally, the video frame buffer stores the reconstructed frames as they are generated during the decoding process. These frames are ready for error resilience transmission or storage. Algorithm 2 explains the pseudocode of proposed HMDC decoding procedure. Fig. 4 portrays the architecture of proposed HMDC-IP.

Algorithm 2: Pseudocode of Proposed HMDC-IP Decoding Procedure	
Input	Encoded data containing descriptors *($A0B0, A0B1, A1B0$,* and *$A1B1$)* Residual data Offset values (for adjusting residuals)
Output	Reconstructed image blocks V
Step 1	Retrieve Encoded Data Compressed representation of blocks $A0B0, A0B1, A1B0$, and $A1B1$ Offset values Residual coefficients
Step 2	Arithmetic Decoding For each descriptor $A0B0, A0B1, A1B0$, and $A1B1$, perform arithmetic decoding to recover the compressed coefficients based on Eq. (12), and (13). $Decoded\_Coeff \leftarrow Arithmetic\_Decode(A_{ij})$ $\forall\{A0B0, A0B1, A1B0, A1B1\}$
Step 3	Coefficient Merging Apply coefficient merger to recombine split blocks into original 8×8 blocks. Use the inverse of the splitting process to recombine the 4×4 blocks as per Eq. (14)
Step 4	Polyphase Inverse Permuting Apply polyphase inverse permuting on the merged coefficients to restore the original ordering of the pixels using Eq. (15). If any descriptor is missing, perform frequency estimation to reconstruct the lost descriptor via Eq. (16). Reconstruct the blocks using residual data as per Eq. (17). Add residual coefficients to the predicted coefficients to reconstruct the original block. $Reconstructed_{Block} \leftarrow B_{ij} + R - Offset$
Step 5	Upsampling Upsample the 8×8 blocks by merging smaller blocks into larger blocks and upsample them using Eq. (19)
Step 6	Final Video Frame Buffer

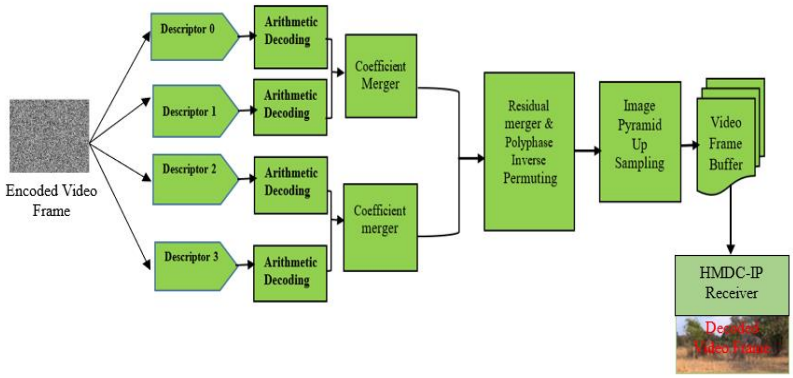


Fig 4: Decoding Procedure of Proposed HMDC-IP Model

IV.SIMULATION RESULTS

A. Simulation Setup

The proposed HMDC-IP model was developed via MATLAB 2021a on Intel core® i5 processor @2.6GHz, 16 GB RAM, 64-bit OS. Here, the Short Videos Dataset was employed for transmission. Additionally, a comparative study is performed to validate the competence of proposed HMDC-IP in handling video streaming over other methods such as Web Real-Time Communications (WebRTC) [34], Light-weight Featurized Image Pyramid-Single Shot Multibox Detector (LFID-SSD) [35], Deep Deterministic Policy Gradient (DDPG) [36], and Transmission Control in Live Video (TCLiVi) [37]. The performance of proposed HMDC-IP is estimated through several metrics such as PSNR, SSIM, bitrate, bandwidth utilization, ERR, computational efficiency, and latency.

B. Performance Metrics

Compression Ratio (CR): It is the proportion of original frame size to compressed frame size [38] as defined in Eq. (24).

$$\mathrm{CR} = \frac{\text{Original frame Size}}{\text{Compressed frame Size}} \tag{24}$$

PSNR: The generally used objective metric for measuring video quality is the PSNR [39]. Eq. (25) is used to obtain the PSNR.

$$PSNR\ Value\ of\ frame = 10 * log_{10}\left({255^2}/{MSE}\right) \tag{25}$$

SSIM: A Method for calculating how similar two images are called the SSIM [40]. The SSIM value is a decimal number between -1 and 1, with 1 indicating that two frames are structurally identical as expressed in Eq. (26).

$$Frame\ SSIM(x,y) = \frac{(2\mu_x\mu_y+z_1)(2\sigma_Xy+z_2)}{(\mu^2{}_x+\mu^2{}_y+z_1)(\sigma^2{}_x+\sigma^2{}_y+z_2)} \tag{26}$$

In the original and denoised image, x and y stand for windows, σ and μ stand for the standard deviation and mean of x and y, and z_1 and z_2 stand for constants.

PSNR Percentage: Eq. (27) is used to determine the percentage of PSNR loss between two images.

$$\text{Percentage Decrease} = \frac{\text{PSNR1}-\text{PSNR2}}{\text{PSNR1}} * 100 \tag{27}$$

Where $PSNR1$ is the PSNR of the original frame or PSNR of first image and $PSNR2$ is the PSNR of the reconstructed frame or PSNR of second image.

C. Algorithmic Analysis

Results and achievements of proposed HMDC-IP are addressed here. Fig. 5 shows the sample frames extracted from the natural documentary video with various qualities such as 360p, 480p, 720p, and 4K. Here, each frame represents a distinct resolution, where Fig. 5 (a) shows the original quality available in the dataset, 360p in Fig.

5(b) offers a low-resolution video with reduced clarity, suitable for low bandwidth conditions, 480p in Fig. 5(c) provides a moderate resolution, balancing quality and bandwidth usage, 720p in Fig. 5(d) is a high-definition resolution, delivering clearer visuals with more detail, and 4K in Fig. 5(e) is ultra-high definition, offering the highest level of detail and sharpness. These frames highlight how varying resolutions affect the visual quality and detail of the same content. Higher resolutions like 720p and 4K display finer details in the natural environment, such as the texture of leaves and the sharpness of wildlife, while lower resolutions like 360p and 480p show a noticeable degradation in detail and clarity. This comparison is useful to demonstrate the impact of resolution on video quality in different bandwidth scenarios.

(a)

(b)

(c)

(d)

Fig 5: Sample Video Frames showing (a) Original Quality, (b) 360p, (c) 480p, (d) 720p, and (e) 4K

The HMDC-IP algorithm uses benchmark videos to assess the performance of the algorithms. The suggested technique performs in terms of compressed file size, compression ratio, PSNR, SSIM, and processing time. It is assumed that while some descriptors are completely lost, others are received with no information lost. Side reconstruction is the term used to describe such a scenario. The performance of side reconstruction is evaluated independently for the first, second, and third missing descriptors. The outcomes are measured by the reconstruction qualities of no loss, 1, 2, and 3 descriptor losses, the frame-by-frame quality comparison, and the qualities in the packet-loss situation. If all the descriptor data is received, then the mean square error value is zero. The frames of these videos have dimensions of 720×1280. The total pixels of the image are 2764800. Processing times vary depending on the network speed, frame size and software. A smaller frame size takes less time, but a larger frame size requires more processing time.

Table II: PSNR of Various Estimation Techniques in Various Descriptor-Loss Situations

Method Name	One Descriptor loss	Two Descriptor loss		Three Descriptor loss	Average Descriptor loss
		Same	Diff		
Proposed HMDC-IP	42.13	36.95	35.85	34.46	37.35
Hybrid-SF	33.95	33.51	32.39	31.34	32.70
Hybrid-S	33.51	33.51	32.05	31.34	32.47
Hybrid-F	33.95	32.99	32.39	31.14	32.56

According to (& Tsai, 2009), a comparison is made between the experimental outcomes of the proposed algorithm and other methods. Different frames were used to evaluate the proposed scheme, and its effectiveness and performance were compared to those of earlier studies. Table II displays the results of using the Foreman sequence of 248 frames. The table presents a comparison of loss values across different methods when using one, two, or three descriptors. The methods evaluated include proposed HMDC-IP, hybrid-SF, hybrid-S, and hybrid-F (Hsiao & Tsai 2009). Each method's performance is assessed using different numbers of descriptors loss, and the results are presented. The average PSNR of four one-descriptor loss cases and four three-descriptor loss cases, respectively, is displayed in the one-loss and three-loss columns. There are six scenarios for two-descriptor loss: two of these scenarios involve the loss of two descriptors within the same residual domain (e.g., A0B0 and A0B1), and four scenarios involve the loss of two descriptors from distinct residual domains (e.g., A0B0 and A1B0). All 14 examples' average PSNR are displayed in the average column. The weakest performers are Hybrid-S and Hybrid-F, according to the results. Fig. 6 shows the PSNR of proposed HMDC-IP over various estimation techniques.

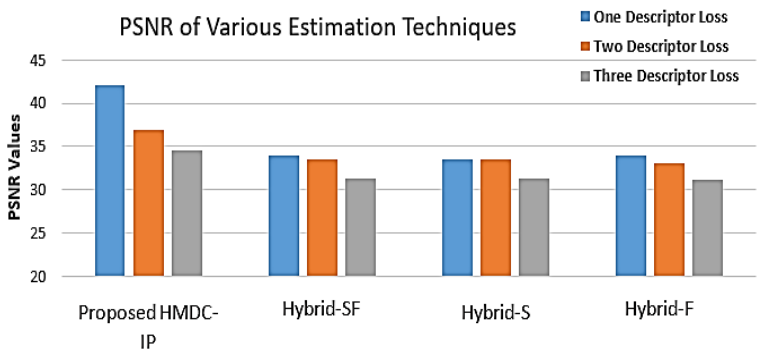


Fig 6: PSNR of Proposed HMDC-IP over Various Estimation Techniques

The proposed scheme not only improves the PSNR value and SSIM value, but also increases the visual quality of the reconstructed images and compression ratio.

Table III: Various Video Frame Values of 360p Resolution

Image Name	Compressed File Size	Compression Ratio	Processing Time (s)
Frame 1	405956	6.8	57
Frame 2	405939	6.8	58
Frame 3	405981	6.82	57.5
Frame 4	405941	6.81	55
Frame 5	405930	6.81	56.2

Table III shows the various video frame values of 360p resolution and provides data on the compression performance of five different video frames, focusing on three key metrics such as compressed file size, compression ratio, and processing time.

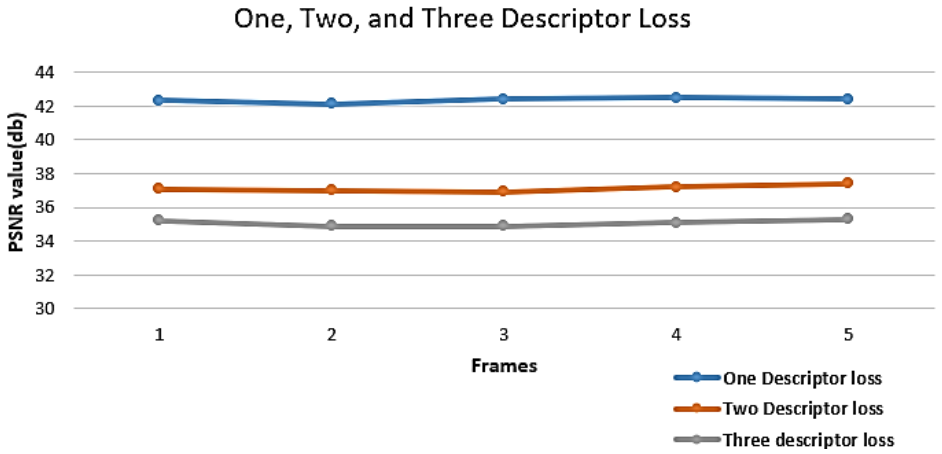


Fig 7: PSNR value for Various Video Frames of 360p Resolution

Fig. 7 shows the PSNR values of the one descriptor loss, two descriptor loss and three descriptor loss of various video frames of 480p Resolution. The PSNR decreases in the video frames between one descriptor loss and two descriptor loss are 5.2, 5.1, 5.5, 5.3 and 5 and two descriptor loss and three descriptor loss are 1.9, 2.1,2, 2.1, and 2.1. The video frame's PSNR percentage decreases between one descriptor loss and two descriptor losses, which are 12.29, 12.11, 12.97, 12.47, and 11.79, and between two descriptor loss and three descriptor losses, which are 5.12, 5.67, 5.42, 5.64, and 5.61.

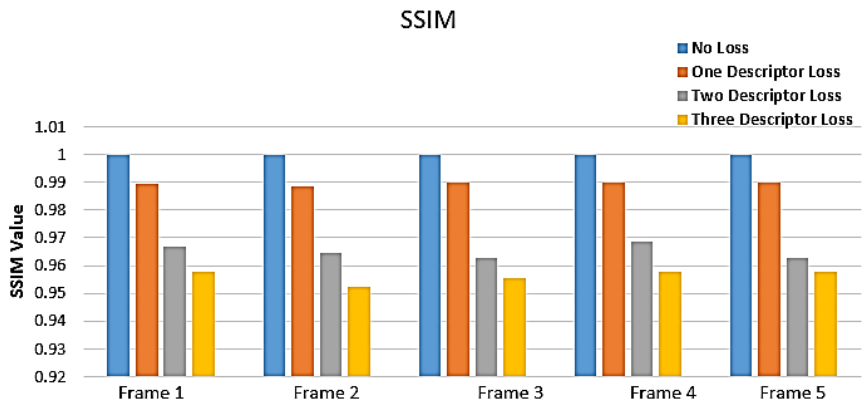


Fig 8: SSIM value for Various Video Frames of 480p Resolution

Fig. 8 shows the SSIM values of various video frames of 480p Resolution. In the given video, all the channel data is received, so the SSIM value is 1. One descriptor loss image shows high SSIM values, indicating minimal information loss. Two and three descriptor loss images achieve slightly lower SSIM values and occur with medium information loss. Table IV presents data on the compression performance of five different mobile frames. It includes three key metrics including compressed file size, compression ratio, and processing time for each frame.

Table IV: Various Video Frame Values of 480p Resolution

Image Name	Compressed File Size	Compression Ratio	Processing Time (s)
Frame 1	426965	6.48	58
Frame 2	426978	6.5	58.4
Frame 3	426954	6.48	57
Frame 4	426934	6.47	56
Frame 5	426938	6.47	59

Fig. 9 displays the PSNR values of the one descriptor loss, two descriptor loss and three descriptor loss of 702p video frames. The PSNR decreases in the video frames between one descriptor loss and two descriptor loss are 3.9, 4.3,4, 4.1, and 4 and two descriptor loss and three descriptor loss are 2.3, 2.3, 2.2, 2.4, and 2.The 720p video frame 's PSNR percentage decreases between one descriptor loss and two descriptor losses, which are 12.29, 12.11,12.97, 12.47, and 11.79, and between two descriptor loss and three descriptor losses, which are 9.72, 10.64, 9.97, 10.12, and 9.95.

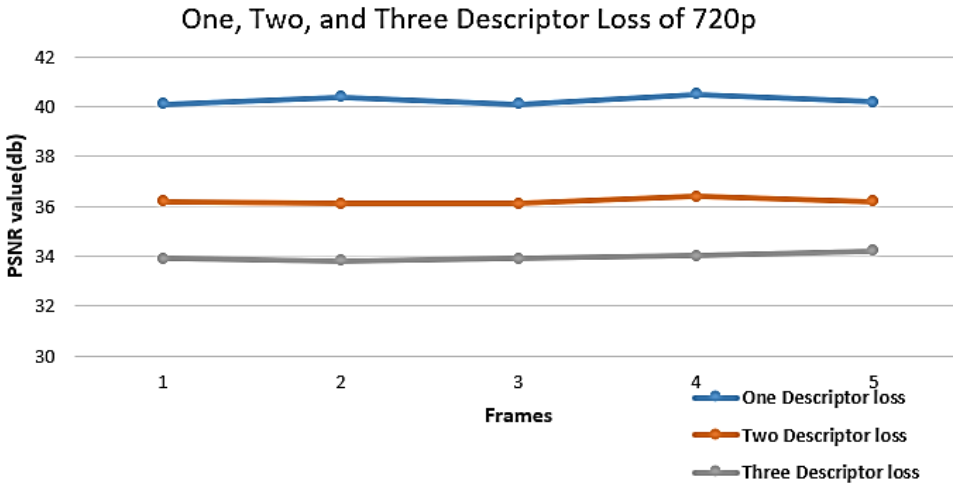


Fig 9: PSNR value for Various Video Frames of 720p Resolution

Fig. 10 shows the SSIM value of the 4K video frames. One descriptor loss frame quality is higher. If two descriptor loss the frame quality is medium. Three descriptor loss the image quality is acceptable. Since all of the descriptor data is received in the mobile video, the SSIM value is 1. High SSIM values in one descriptor loss image indicating less information loss. Images with two and three descriptor losses exhibit slightly lower SSIM values and medium information loss.

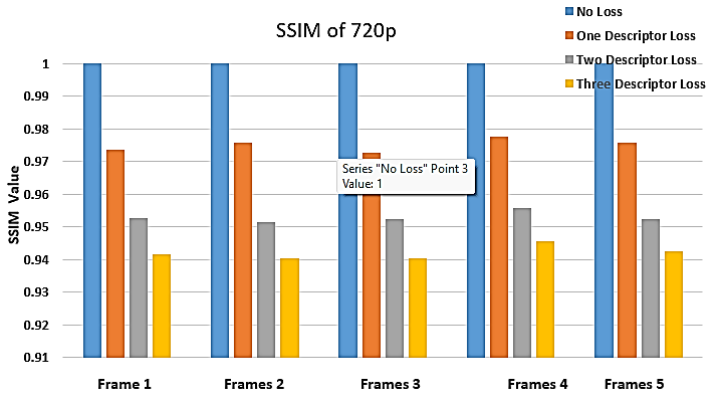


Fig 10: SSIM value for Various Video Frames of 720p Resolution

Table V displays the 4K video frame values and provides information on the compression performance for five 4K video frames, detailing the compressed file size, compression ratio, and processing time for each frame.

Table V: 4K Video Frame Values

Image Name	Compressed File Size	Compression Ratio	Processing Time (s)
Frame 1	434582	6.37	54
Frame 2	434563	6.37	57.8
Frame 3	434507	6.36	59
Frame 4	434505	6.36	56
Frame 5	434501	6.36	57

The PSNR values for the 4K video frame's one, two, and three descriptor losses are displayed in Fig. 11.The PSNR decreases in the 4K video between one descriptor loss and two descriptor loss are 5.2, 5.4,5.8, 5.3, and 4.9 and two descriptor loss and three descriptor loss are 2.4, 2.9, 2.3, 1.8, and 1.9.The 4K video frame's PSNR percentage decreases between one descriptor loss and two descriptor losses, which are 12.90, 13.23,12.97, 14.18, and 12.12, and between two descriptor loss and three descriptor losses, which are 6.83, 6.19, 6.55, 6.11, and 5.35.

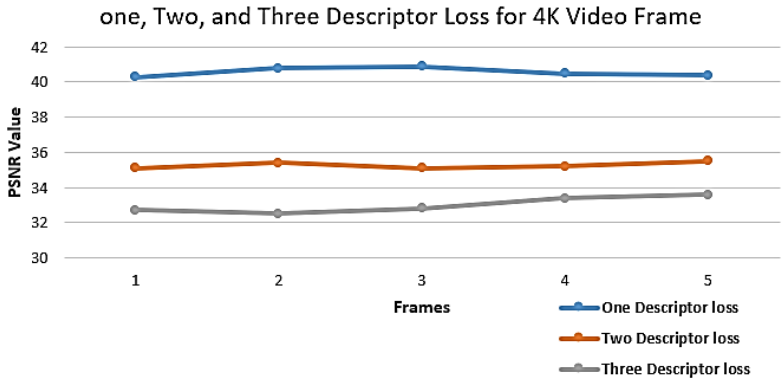


Fig 11: PSNR value for Various Video Frames of 4K Resolution

Fig. 12 displays the SSIM value for 4K video frames. Frame quality will be higher for one descriptor loss. Medium frame quality will result from the loss of two descriptors. The picture quality will be sufficient even if three descriptors are lost. With the Medical Video providing all of the description data, the SSIM value is 1. There is less information loss in one descriptor loss image when the SSIM values are high. There is a medium information loss and a slight decrease in SSIM values in images with two and three descriptor losses.

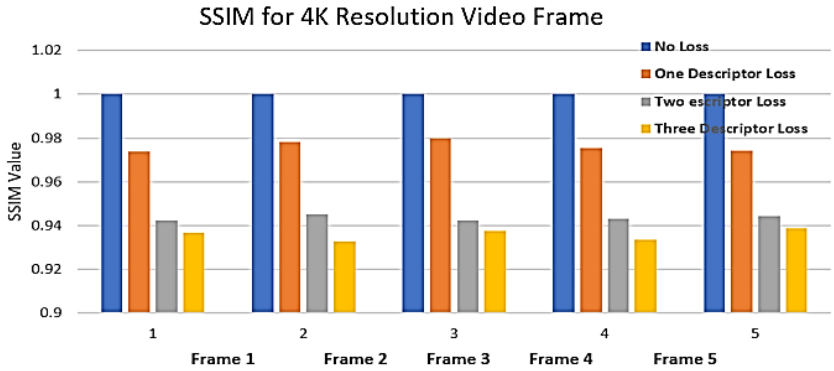
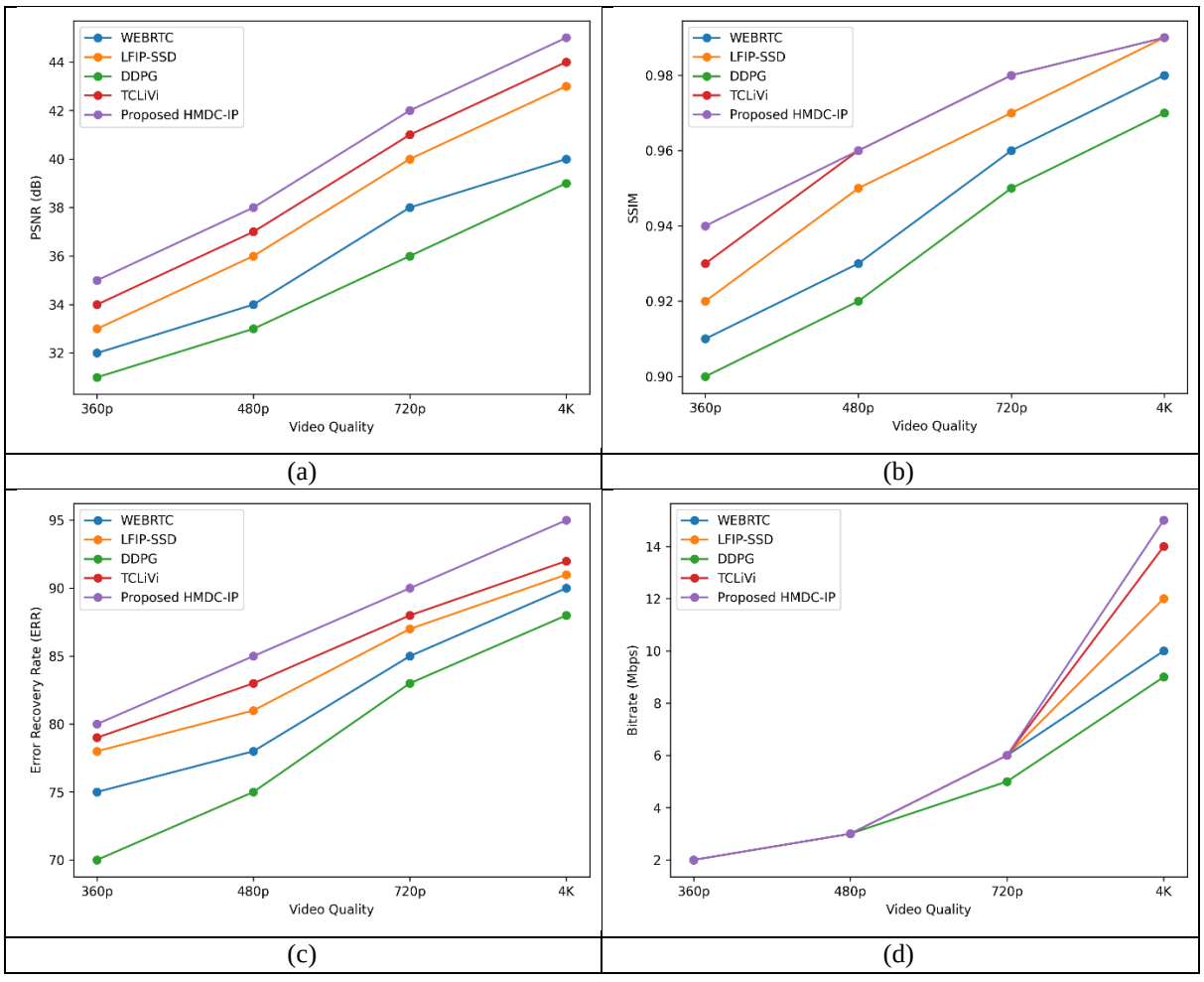


Fig 12: SSIM value for Various Video Frames of 4K Resolution

D. Comparative Analysis

In this section, the efficacy of proposed HMDC-IP is validated with several performance metrics over other video streaming methods. Fig. 13 displays the performance of proposed HMDC-IP over Other Models for various metrics such as PSNR, SSIM, ERR, bitrate, bandwidth utilization, computational efficiency, and latency. For all metrics, the proposed HMDC-IP exposed superior performance and proved its competence in video streaming.



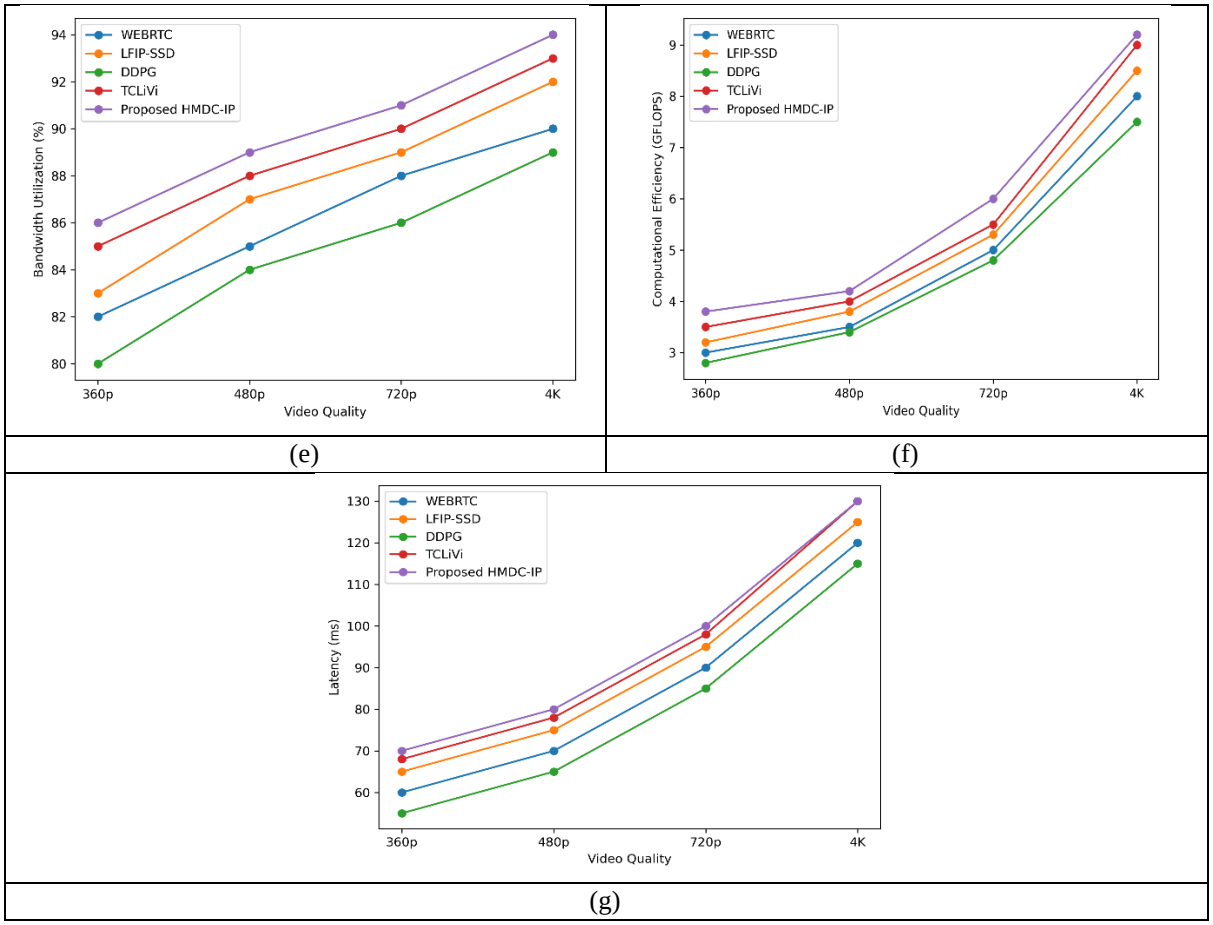


Fig 13: Performance of Proposed HMDC-IP over Other Models for (a) PSNR, (b) SSIM, (c) ERR, (d) Bitrate, (e) Bandwidth Utilization, (f) Computational Efficiency, and (g) Latency

Table VI: Performance of Proposed HMDC-IP over Other Models for Various Metrics concerning 360p Video Quality

Metric	WebRTC	LFIP-SSD	DDPG	TCLiVi	Proposed HMDC-IP
PSNR (dB)	32	33	31	34	35
SSIM	0.91	0.92	0.9	0.93	0.94
Bitrate (Mbps)	2	2	2	2	2
Bandwidth Utilization (%)	82	83	80	85	86
ERR	75	78	70	79	80
Computational Efficiency (GFLOPS)	3	3.2	2.8	3.5	3.8
Latency (ms)	60	65	55	68	70

Table VII: Performance of Proposed HMDC-IP over Other Models for Various Metrics concerning 480p Video Quality

Metric	WebRTC	LFIP-SSD	DDPG	TCLiVi	Proposed HMDC-IP
PSNR (dB)	34	36	33	37	38
SSIM	0.93	0.95	0.92	0.96	0.96

Bitrate (Mbps)	3	3	3	3	3
Bandwidth Utilization (%)	85	87	84	88	89
ERR	78	81	75	83	85
Computational Efficiency (GFLOPS)	3.5	3.8	3.4	4	4.2
Latency (ms)	70	75	65	78	80

Table VIII: Performance of Proposed HMDC-IP over Other Models for Various Metrics concerning 720p Video Quality

Metric	WebRTC	LFIP-SSD	DDPG	TCLiVi	Proposed HMDC-IP
PSNR (dB)	38	40	36	41	42
SSIM	0.96	0.97	0.95	0.98	0.98
Bitrate (Mbps)	6	6	5	6	6
Bandwidth Utilization (%)	88	89	86	90	91
ERR	85	87	83	88	90
Computational Efficiency (GFLOPS)	5	5.3	4.8	5.5	6
Latency (ms)	90	95	85	98	100

Table IX: Performance of Proposed HMDC-IP over Other Models for Various Metrics concerning 4K Video Quality

Metric	WebRTC	LFIP-SSD	DDPG	TCLiVi	Proposed HMDC-IP
PSNR (dB)	40	43	39	44	45
SSIM	0.98	0.99	0.97	0.99	0.99
Bitrate (Mbps)	10	12	9	14	15
Bandwidth Utilization (%)	90	92	89	93	94
ERR	90	91	88	92	95
Computational Efficiency (GFLOPS)	8	8.5	7.5	9	9.2
Latency (ms)	120	125	115	130	130

Table VI summarizes the performance metrics to validate the effectiveness of each approach in handling video quality, including transmission efficiency of various models, such as WebRTC, LFIP-SSD, DDPG, TCLiVi, and the proposed HMDC-IP for 360p video streaming. The proposed HMDC-IP achieves the highest PSNR at 35 dB with an SSIM of 0.94, representing better quality and clarity compared to other models, whose PSNR ranges between 31 and 34 dB, with SSIM ranging between 0.90 and 0.93. All the models maintain a bitrate of 2 Mbps to ensure consistent data transmission. The bandwidth utilization for the proposed model is also the best, at 86%, effectively optimizing available resources. HMDC-IP has an ERR value of 80, outperforming others in the range of 70-79. Additionally, its computational efficiency is higher at 3.8 GFLOPS compared to others, which range from 2.8 to 3.5 GFLOPS. However, the latency for HMDC-IP is higher at 70 ms, remaining comparable to other models. Overall, the HMDC-IP model shows significant gains in video quality and efficiency for 360p streaming.

The evaluation of the HMDC-IP for 480p video quality is presented in Table VII, showing major improvements across metrics compared to other models. HMDC-IP achieves a PSNR of 38 dB, the best among the models for superior video quality. SSIM follows this trend at 0.96, matched only by TCLiVi. All models maintain a bitrate of 3 Mbps, but HMDC-IP leads in bandwidth utilization at 89%. ERR is also best for HMDC-IP at 85,

demonstrating effectiveness in recovering from packet loss, with computational efficiency at 4.2 GFLOPS, higher than others. The latency for HMDC-IP is recorded at 80 ms, slightly higher but competitive.

Table VIII presents the evaluation of 720p video quality, with HMDC-IP continuing to stand out. It achieves the highest PSNR at 42 dB and SSIM at 0.98, underscoring its ability to deliver excellent video quality. While all models maintain a bitrate of 6 Mbps, HMDC-IP leads in bandwidth utilization at 91%. The ERR reaches 90, indicating good video integrity even during network fluctuations. Computational efficiency is strong at 6 GFLOPS, demonstrating optimal performance without excessive resource demand. The latency increases to 100 ms, consistent with higher resolutions.

Finally, Table IX presents the performance of HMDC-IP for 4K video quality, showing further improvements across all metrics. The peak PSNR is 45 dB with an SSIM of 0.99, marking it as the top performer in video quality. Bandwidth utilization reaches 94%, with a bitrate increase to 15 Mbps, showing efficiency in resource management. ERR improves to 95, highlighting strong error resilience. Computational efficiency reaches 9.2 GFLOPS, outperforming other models. Latency stabilizes at 130 ms, suggesting that while the model excels in quality and efficiency, careful management is required to ensure responsiveness in high-resolution applications. Thus, HMDC-IP demonstrates consistent superior performance metrics across various video qualities, making it highly effective for modern video streaming applications.

E. Discussion

The proposed HMDC-IP model showcases significant advancements in video streaming quality across various resolutions, outperforming existing models in metrics such as PSNR, SSIM, bandwidth utilization, and ERR. By using a hybrid approach that MDC with an Image Pyramid framework, the proposed HMDC-IP effectively mitigates the adverse effects of packet loss while maintaining video quality and bandwidth usage. The use of arithmetic encoding and decoding, further enhances the system's adaptability under varying network conditions, making it a robust solution for real-time video applications, including live streaming, video conferencing, and multimedia content delivery. In transmission, one or more loss of descriptors is evaluated for various video qualities. For one descriptor loss, the proposed HMDC-IP model exposed better PSNR, SSIM, and ERR scores than 2 or more descriptor loss. On the other hand, when all descriptors are received, the model exhibits high PSNR, SSIM equals to one and better ERR.

However, despite its notable advantages, the proposed HMDC-IP model is not without limitations. The latency associated with higher resolutions, particularly in 4K video streaming, remains a concern, potentially impacting user experience (QoE) in latency-sensitive applications. Additionally, while the computational efficiency is commendable, the model requires substantial resources, which could limit its applicability in environments with constrained computational power. Future work could focus on addressing these limitations by optimizing processing efficiency and reducing latency, thereby enhancing the practicality of proposed HMDC-IP in diverse real-world scenarios.

V.CONCLUSION

This research introduced HMDC-IP, a hybrid error-resilient video streaming model that employed hybrid MDC and an Image Pyramid framework to improve video transmission performance in fluctuating network conditions. The MDC technique divided the video into multiple independently decodable descriptions, enabling partial recovery when packet loss occurred. Concurrently, the Image Pyramid method generated multiple resolution layers, allowing flexible reconstruction at reduced quality if higher-resolution data was lost. To optimize system performance, arithmetic coding was employed to ensure efficient use of bandwidth and computational resources. Simulations were conducted under various network conditions, evaluating key performance indicators such as PSNR, SSIM, and ERR. Future work could explore the integration of ML techniques to optimize the HMDC-IP model for real-time decision-making in dynamic network conditions. Additionally, developing adaptive algorithms that minimize latency while maintaining video quality across all resolutions would enhance user experience (QoE). Investigating the application of the model in different video content types such as fast motion like sports, slow motion like natural documentaries, talking heads like interviews, and specifically, medical videos and network scenarios could provide insights into its scalability and robustness. Furthermore, incorporating edge computing solutions could reduce the computational load on devices, making the system more efficient.

REFERENCES

[1] Wang, M. H., Hsieh, T. S., Tseng, Y. Y., & Chi, P. W. (2023). An SDN-Driven Reliable Transmission Architecture for Enhancing Real-Time Video Streaming Quality. IEEE MultiMedia.

[2] Taha, M., Canovas, A., Lloret, J., & Ali, A. (2021). A QoE adaptive management system for high definition video streaming over wireless networks. Telecommunication Systems, 77(1), 63-81.

[3] Maheswari, K., & Nimmagadda, P. (2023). Error resilient wireless video transmission via parallel processing using puncturing rule enabled coding and decoding. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 6, 100324.

[4] Ghasemkhani, B., Yilmaz, R., & Kut, R. A. A Novel Construction Proposal of Error Protection Code for Video Transmission Over Wireless Broadband Networks.

[5] Hari, S. K. S., Sullivan, M. B., Tsai, T., & Keckler, S. W. (2021). Making convolutions resilient via algorithm-based error detection techniques. IEEE Transactions on Dependable and Secure Computing, 19(4), 2546-2558.

[6] Kesavan, S., Saravana Kumar, E., Kumar, A., & Vengatesan, K. (2021). An investigation on adaptive HTTP media streaming Quality-of-Experience (QoE) and agility using cloud media services. International Journal of Computers and Applications, 43(5), 431-444.

[7] Mandal, S., Chakrabarti, A., & Bodapati, S. (2021). Clustered error resilient SRAM-based reconfigurable computing platform. IEEE Transactions on Aerospace and Electronic Systems, 57(3), 1768-1779.

[8] Farahani, S. S., Reshadinezhad, M. R., & Fatemieh, S. E. (2024). New design for error-resilient approximate multipliers used in image processing in CNTFET technology. The Journal of Supercomputing, 80(3), 3694-3712.

[9] Bari, A. M., Siraj, M. T., Paul, S. K., & Khan, S. A. (2022). A Hybrid Multi-Criteria Decision-Making approach for analysing operational hazards in heavy fuel oil-based power plants. Decision Analytics Journal, 3, 100069.

[10] Lee, R., Venieris, S. I., & Lane, N. D. (2021). Deep neural network–based enhancement for image and video streaming systems: A survey and future directions. ACM Computing Surveys (CSUR), 54(8), 1-30.

[11] Hu, S., Fan, G., Zhou, J., Fan, J., Gan, M., & Chen, C. P. (2024). Hybrid network via key feature fusion for image restoration. Engineering Applications of Artificial Intelligence, 137, 109236.

[12] Peña-Ancavil, E., Estevez, C., Sanhueza, A., & Orchard, M. (2023). Adaptive Scalable Video Streaming (ASViS): An Advanced ABR Transmission Protocol for Optimal Video Quality. Electronics, 12(21), 4542.

[13] Liang, Z., Liu, J., Dasari, M., & Wang, F. (2024). Fumos: Neural Compression and Progressive Refinement for Continuous Point Cloud Video Streaming. IEEE Transactions on Visualization and Computer Graphics.

[14] Shahid, M. A., Islam, N., Alam, M. M., Mazliham, M. S., & Musa, S. (2021). Towards Resilient Method: An exhaustive survey of fault tolerance methods in the cloud computing environment. Computer Science Review, 40, 100398.

[15] Sreekala, K., Raj, N. N., Gupta, S., Anitha, G., Nanda, A. K., & Chaturvedi, A. (2023). Deep convolutional neural network with Kalman filter based objected tracking and detection in underwater communications. Wireless Networks, 1-18.

[16] Li, J., Li, B., & Lu, Y. (2022, October). Hybrid spatial-temporal entropy modelling for neural video compression. In Proceedings of the 30th ACM International Conference on Multimedia (pp. 1503-1511).

[17] Alhilal, A., Braud, T., Han, B., & Hui, P. (2022, April). Nebula: Reliable low-latency video transmission for mobile cloud gaming. In Proceedings of the ACM Web Conference 2022 (pp. 3407-3417).

[18] Tung, T. Y., & Gündüz, D. (2022). DeepWiVe: Deep-learning-aided wireless video transmission. IEEE Journal on Selected Areas in Communications, 40(9), 2570-2583.

[19] Zhang, Y., & Gu, B. (2021). Error-resilient coding by convolutional neural networks for underwater video transmission. Journal of the Franklin Institute, 358(17), 9307-9324.

[20] Minallah, N., Ullah, K., Frnda, J., Cengiz, K., & Awais Javed, M. (2021). Transmitter diversity gain technique aided irregular channel coding for mobile video transmission. Entropy, 23(2), 235.

[21] Hu, X., Pan, Y., Wang, Y., Zhang, L., & Shirmohammadi, S. (2021). Multiple description coding for best-effort delivery of light field video using GNN-based compression. IEEE Transactions on Multimedia, 25, 690-705.

[22] Wang, F., Chen, J., Zeng, H., & Cai, C. (2022). Spatial-frequency HEVC multiple description video coding with adaptive perceptual redundancy allocation. Journal of Visual Communication and Image Representation, 88, 103614.
[23] Li, B., Zhang, Y., & Feng, Q. (2022). Video Error-Resilience Encoding and Decoding Based on Wyner-Ziv Framework for Underwater Transmission. Wireless Communications and Mobile Computing, 2022(1), 2697877.
[24] Alaya, B., & Sellami, L. (2022). Multilayer video encoding for QoS managing of video streaming in VANET environment. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 18(3), 1-19.
[25] Zhao, L., Zhang, J., Bai, H., Wang, A., & Zhao, Y. (2022). LMDC: Learning a multiple description codec for deep learning-based image compression. Multimedia Tools and Applications, 81(10), 13889-13910.
[26] Wang, Y., Reibman, A. R., & Lin, S. (2004). Multiple description coding for video delivery. Proceedings of the IEEE, 93(1), 57-70.
[27] Huang, Y. W., An, J., Huang, H., Li, X., Hsiang, S. T., Zhang, K., Gao, H., Ma, J., & Chubach, O. (2021). Block partitioning structure in the VVC standard. IEEE Transactions on Circuits and Systems for Video Technology, 31(10), 3818-3833.
[28] Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., & Ogden, J. M. (1984). Pyramid methods in image processing. RCA engineer, 29(6), 33-41.
[29] Saha, S., & Gokhale, T. (2024). Improving Shift Invariance in Convolutional Neural Networks with Translation Invariant Polyphase Sampling. arXiv preprint arXiv:2404.07410.
[30] Rojas-Gomez, R. A., Lim, T. Y., Schwing, A., Do, M., & Yeh, R. A. (2022). Learnable polyphase sampling for shift invariant and equivariant convolutional networks. Advances in Neural Information Processing Systems, 35, 35755-35768.
[31] Tu, C., & Tran, T. D. (2002). Context-based entropy coding of block transform coefficients for image compression. IEEE Transactions on Image Processing, 11(11), 1271-1283.
[32] Afandi, T. M. K., Fandiantoro, D. H., & Purnama, I. K. E. (2021, July). Medical images compression and encryption using DCT, arithmetic encoding and chaos-based encryption. In 2021 international seminar on intelligent technology and its applications (ISITIA) (pp. 1-5). IEEE.
[33] Zaibi, S., Zribi, A., Pyndiah, R., & Aloui, N. (2012). Joint source/channel iterative arithmetic decoding with JPEG 2000 image transmission application. EURASIP Journal on Advances in Signal Processing, 2012, 1-13.
[34] De Fré, M., van der Hooft, J., Wauters, T., & De Turck, F. (2024, April). Scalable MDC-Based Volumetric Video Delivery for Real-Time One-to-Many WebRTC Conferencing. In Proceedings of the 15th ACM Multimedia Systems Conference (pp. 121-131).
[35] Pang, Y., Wang, T., Anwer, R. M., Khan, F. S., & Shao, L. (2019). Efficient featurized image pyramid network for single shot detector. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 7336-7344).
[36] Miao, J., Bai, S., Mumtaz, S., Zhang, Q., & Mu, J. (2024). Utility-Oriented Optimization for Video Streaming in UAV-Aided MEC Network: A DRL Approach. IEEE Transactions on Green Communications and Networking.
[37] Cui, L., Su, D., Yang, S., Wang, Z., & Ming, Z. (2020). TCLiVi: Transmission control in live video streaming based on deep reinforcement learning. IEEE Transactions on Multimedia, 23, 651-663.
[38] López-Paniagua, I., Rodríguez-Martín, J., Sánchez-Orgaz, S., & Roncal-Casano, J. J. (2020). Step by step derivation of the optimum multistage compression ratio and an application case. Entropy, 22(6), 678.
[39] Al-Najjar, Y., & Chen, D. (2012). Comparison of image quality assessment: PSNR, HVS, SSIM, UIQI. International Journal of Scientific and Engineering Research, 3(8), 1-5.
[40] Hsiao, C. W., & Tsai, W. J. (2009). Hybrid multiple description coding based on H. 264. IEEE Transactions on Circuits and Systems for Video Technology, 20(1), 76-87.