China Petroleum Processing and Petrochemical Technology

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

IMAGE BLOCK SIZE SPLIT ANALYSIS FOR QUALITY IN ERROR RESILIENCE TRANSMISSION USING HYBRID MULTIPLE DESCRIPTION CODING

D.H.Kitty Smailin* and Dr.Y. Jacob Vetha Raj

*Research Scholar, Reg No. 19123112282009, Dept. of Computer Science, Nesamony Memorial Christian College, Affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-627 012, Tamil Nadu, India, kitty_csa@nmcc.ac.in

Associate Professor, Dept. of Computer Science, Nesamony Memorial Christian College, Affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli 627 012, Tamil Nadu, India. jacobvetharaj@gmail.com

Abstract:

Error-resilient transmission is a distinct requirement for current digital image processing equipment, as it is necessary to compress and store huge digital pictures as compressed images. This also applies to the storage of large digital images. Multiple Description Coding (MDC), which is often used in video transmission, is a technique that sends descriptions through error-prone networks via multiple channels. MDC is currently being developed as an efficient error-resistant lossy coding technique for data transmission through channels that are prone to errors. As videos are compressed, they become easier to work with and require less memory during transmission. The Hybrid Multiple Description Coding (HMDC) method uses a 16x16 image size block split. This methodology uses 8x8, 32x32, and 64x64 image block size splits in the same method, and an evaluation methodology has been used to compare the performance between 8x8, 16x16, 32x32, and 64x64 image block size splits. Using a separate set of frames, the experimental outcomes of the 8x8, 16x16, 32x32, and 64x64 image block size split methods are compared to some of the error resilience strategies that are already in use. This study investigates different image block size splits' effects on several metrics, such as the peak signal-to-noise ratio (PSNR) value, the structural similarity index measurement (SSIM), the compression ratio, and the processing time.

Keywords: video analysis, decoding, encoding, video compression, block split.

1. Introduction

Error-resilient transmission across noisy channels has been a challenging subject, as transmitting raw video requires such a large amount of bandwidth that error-resilient transmission is inevitable. As a result, when compressed video is transmitted over the Internet, significant data loss occurs. HMDC is an error-resistant coding technique used to tackle the packet loss problem in communication networks. However, the possibility of transmission failures still exists when the Internet is overloaded or when signal packets are sent across unreliable, unstable channels. In HMDC, various streams of the video are encoded (also known as descriptions or channels). If not all channels are down at the same time, the video will still be accessible to the receivers. Therefore, HMDC can be used in combination with multi-path transmission to overcome losses and improve

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

the channel's reliability. Fig. 1 shows the structure of the HMDC system. If only one descriptor is received, the image produced will be of poor quality. However, when two or three descriptors are received, a good-quality reconstructed image will be produced. When all four descriptors are available, the best-quality reconstructed image will be produced. Error-resilient transmission is a crucial solution for video transmission because the channel bandwidth is limited and there is a demand for faster transmission.

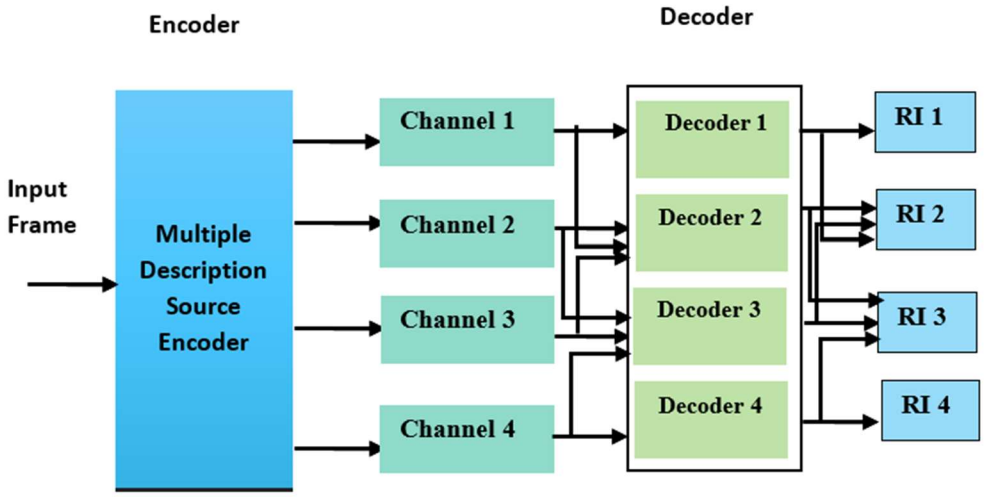


Fig.1 HMDC System Structure

To enhance the visual quality of frames, various enhancement techniques are employed. In this paper, we explore lossy compression using a hybrid method. We utilised an H.264 coder based on hybrid multi-descriptor coding (HMDC) for video compression in this study [1]. To ensure excellent video quality in the reconstructed output, the receiver employs coefficient correlation in the transform domain and leftover pixel correlation in the spatial domain. This processing is carried out in both the spatial and frequency domains, with a kernel size of 16x16. In digital video transmission, ensuring error resilience is of paramount importance due to issues such as network errors, packet loss, and data corruption being prevalent. Current video transmission techniques often struggle to maintain video quality and continuity under adverse network conditions. Therefore, there is a pressing need for an effective solution that can address these challenges. To address these issues, the primary objective of our research is to investigate the effects of different image block size splits on various metrics related to error-resilient transmission and compression of digital images. Our proposed solution for addressing error-resilient video transmission involves implementing a novel Error-Resilience Video Transmission Scheme utilising HMDC. This approach entails splitting the video data into multiple descriptions that can be transmitted via different channels, providing redundancy and error recovery capabilities. By leveraging HMDC, which optimally combines these descriptions for error-prone environments, we aim to significantly improve the robustness of video transmission. This scheme will intelligently adapt to varying network conditions, mitigating the impact of errors, reducing video artifacts, and ultimately ensuring a smoother and more reliable video viewing experience for end-users. The remaining

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

sections of this paper are organised as follows: In Section 2, we review relevant works in the field. Section 3 explains our suggested error-resilient transmission approach, which involves splitting images into smaller pieces before producing DCT (Discrete Cosine Transform) coefficients using a variable kernel size. In Section 4, we present and discuss the results of our study. Finally, Section 5 offers our conclusions.

2. Related Work

The multiple description coding, along with multipath transport, is proficient in compressing video data [2]. When delivering an image to the network, the ideal DCT compression ratio of the X-ray images is selected [3]. Scalable MDC was proposed for the compression of 3D stereoscopic video, generating proficient results for transmission through channels with noise [4]. MDC was adopted [5] for data transmission through the MIMO-OFDM system, with high transmission efficiency and good reconstructed video quality [6]. A novel image compression scheme called Chimaera was proposed, in which the authors developed an image transform intended to compress a particular data type and exemplify the transform's primary feature [7].

Overhead and quality evaluation metrics of MDC are analysed in [8]. A detailed analysis was also conducted, describing where multiple description coding should be used despite single description coding. A detailed review was carried out in [9], describing the role of MDC in error-resilient video delivery systems. Learning for Video Compression [10] The notion of Pixel Motion CNN (PMCNN), which incorporates motion extension and hybrid prediction networks, was proposed in this research. A comparison between layered coding (LC) and MD coding was carried out in [11]. MDC gives equal importance to all sub-streams of data, whereas LC gives varying levels of importance to sub-streams of data. MDC with zero padding was proposed in [12], with 1D zero padding performing better than 2D padding. MDC and non-orthogonal multiple access (NOMA) concepts were integrated for video data compression in [13], with a base station communicating with two nodes using two sub-channels where a single channel is sufficient for reconstruction..

A multi-qubit quantum data compression was proposed in [14], with the minimum utilization of resources being the key advantage. A survey on various data compression strategies and their features with validation metrics is highlighted. Performance Evaluation of Low Bitrate Video Compression Using Adaptive Motion Matching was proposed in [15], with the approach aiming to improve the quality of up-sampled frames while maintaining the originality of down-sampled frames. The proposed down-sampling approach is also provided with a block-wise implementation to reduce the algorithm's computational complexity and storage requirements. Digital Image Compression Using the DCT Algorithm: An Improvement was proposed in [16], which compresses the image but compromises image quality. The algorithm describes how to encode and decode images for JPEG [17].[19] suggested Image Degradation by Zeroing Block

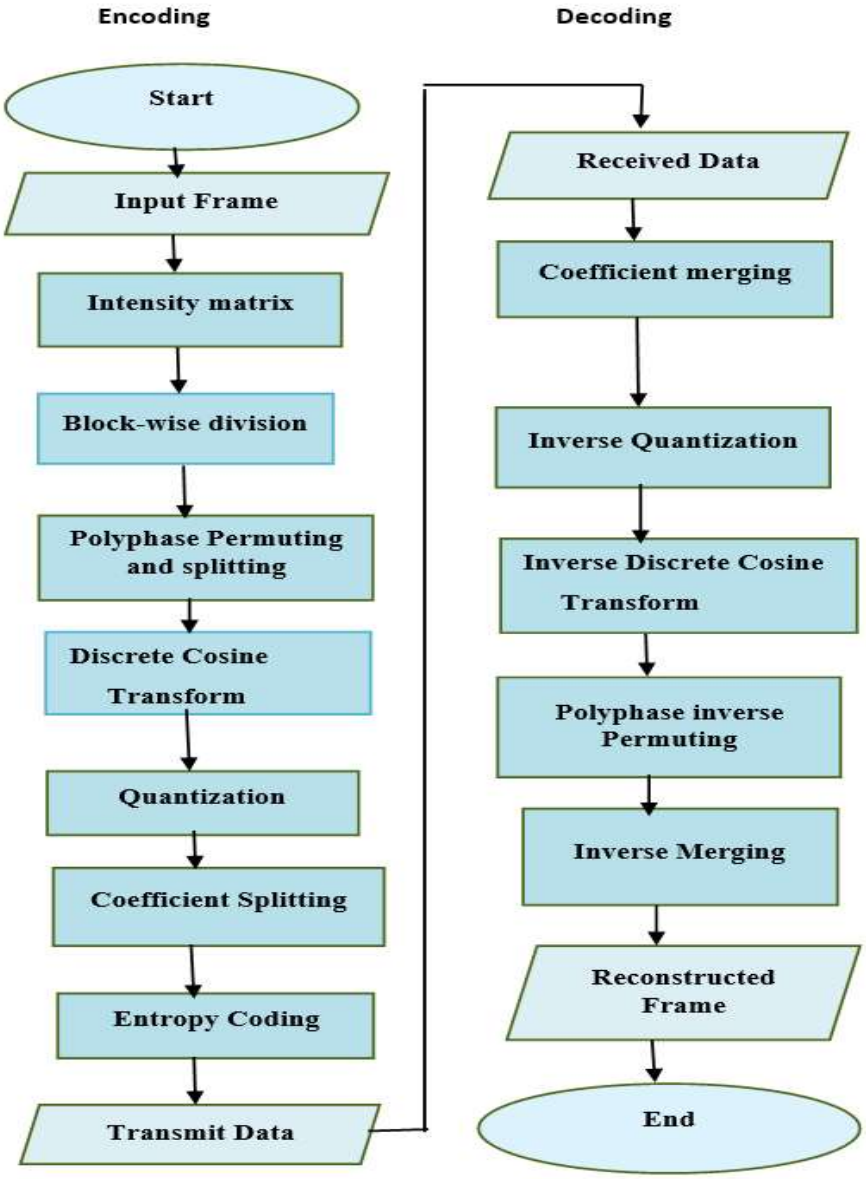
Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

DCT vs. Full DCT Coefficients, and it was revealed that the DCT block size is strongly proportional to the influence of different DCT coefficients on an original image.

3. Contribution of the proposed work

3.1 Error resilience transmission using HMDC with Varying Block Size

One of the most frequently employed techniques for video coding, particularly for error-resilient transmission, is block-based video coding. In this approach, each video frame is partitioned into coding blocks, which are subsequently predicted, modified, quantized, and entropy-encoded [20, 21]. The size of the kernel used for dividing the input image can vary, including options like 8×8, 16×16, 32×32, and 64×64. We evaluated the results for different kernel sizes, and you can see the encoding/decoding scheme for error-resilient transmission in Fig. 2.



Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

Fig 2: The Scheme of Encoding/Decoding for error resilience transmission

3.2 HMDC encoder architecture

The image is split into 8x8 blocks (micro blocks), 32x32 blocks (mega blocks), and 64x64 blocks (super mega blocks) to improve error resilience, transmission, and processing time while minimising data loss on each channel during image reconstruction. Fig. 3 shows the structure of the HMDC.

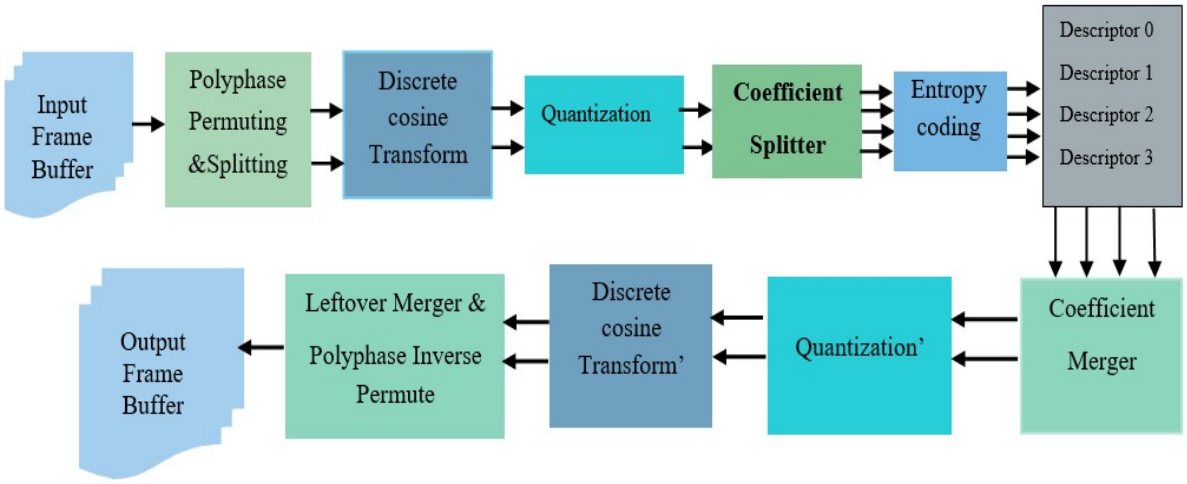


Fig.3 Video Frame Encoder Structure

3.2.1. Image 8 × 8 Block Size Split Codec:

In the encoding loop, the 8x8 block-size split encoder for the image uses a two-level splitting mechanism: 1) polyphase permuting and splitting; and 2) coefficient splitting. The former divides the matrix, while the latter divides the frequency-domain coefficients. After the initial splitting, the encoding route is divided into two, and then into four after the secondary splitting. The leftover domain's 4x4 blocks are separated using polyphase permuting and splitting at the initial level. In each 4x4 block, the remaining data is first polyphase permuted within the block, and then it is divided into two blocks. The four pathways are combined before opposite quantization to recover the entirety of the reference frame's information. Before permutation, the remaining pixels in the 4x4 block are initially designated with values ranging from 0 to 3. To indicate each 4x4 pixel, 0 is placed on the upper-left pixel, 1 on the upper-right pixel, 2 on the lower-left pixel, and 3 on the lower-right pixel. The label-0 pixels are all moved to the upper left 2x2 block, the label-1 pixels to the upper right 2x2 block, the label-2 pixels to the lower left 2x2 block, and the label-3 pixels to the lower right 2x2 block, as seen in the centre of Fig. 4, which displays only one 4x4 block due to the polyphase's permutation. In each mini block, there are four 4x4 blocks, all permuted in the same way.

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

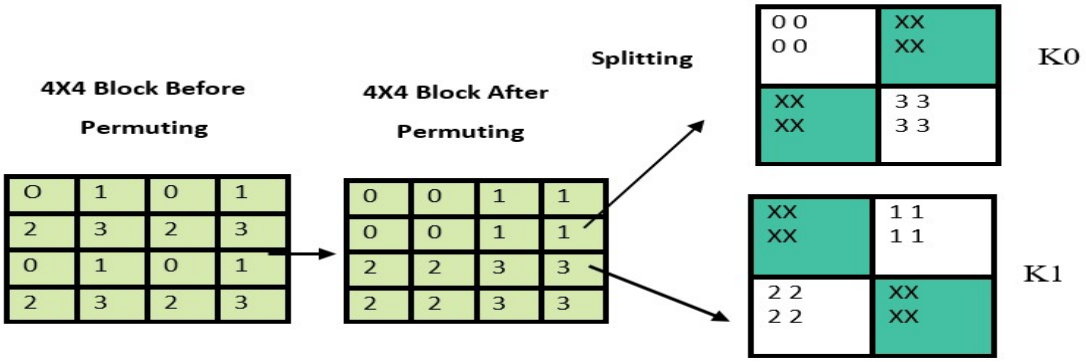


Fig.4 Permuting and splitting of a 4x4 block

After polyphase permutation, the splitting process divides each 4x4 block [22] into two 4x4 blocks known as 'leftover 0' and 'leftover 1.' These blocks are represented by K0 and K1, respectively, and each contains two 2x2 diagonally selected leftover blocks. One 4x4 block includes the upper-left and lower-right 2x2 remaining blocks, while the other 4x4 block contains the upper-right and lower-left blocks. In Fig. 4, for each 4x4 block, two 2x2 blocks labelled 'X' are presented, with all coefficients set to zero. These 'X'-labelled blocks with zero coefficients do not require encoding. The Coefficient Splitter further divides the 4 DCT coefficients of a 2x2 block into three parts: low frequency, median frequency, and high frequency. The low-frequency coefficient is closest to DC, the high-frequency is farthest from DC [23] [27], and the median frequency comprises the others. Using this categorization, the coefficient splitter divides the AC coefficients of each category into two groups, one for each descriptor. The most relevant coefficient and DC are then duplicated.

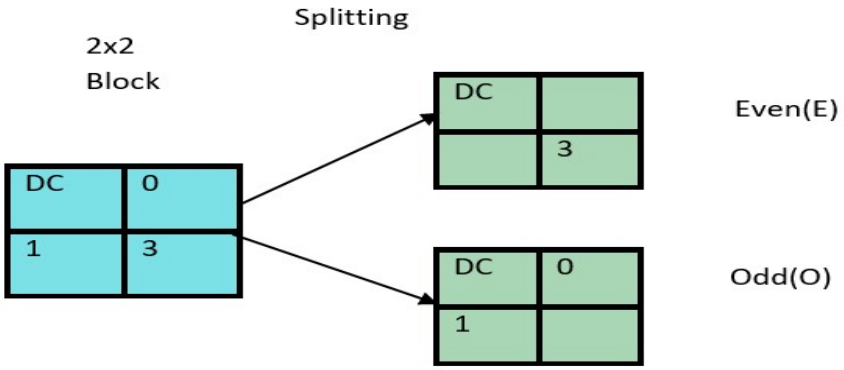


Fig.5 Even and Odd 2X2 blocks generated by Coefficient Splitter

Fig. 5 illustrates the division of a 2x2 block into two 2x2 blocks, each of which contains approximately half of the original coefficients. An E-block is a 2x2 block with an even number of original AC coefficients, whereas an O-block is a block with an odd number of original AC coefficients. Along with replicating DC for both blocks, each ac collection is divided diagonally to produce sensible visual quality. Each descriptor has fewer pairs since it is assigned to every other upper-right to lower-left diagonal of coefficients. Entropy coding such as CABAC or

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

CAVLC is still used for the generated blocks. The resulting S0 and S1 from K0 are respectively called K0S0 and K0S1, and those from K1 are called K1S0 and K1S1, as shown in Fig. 6.

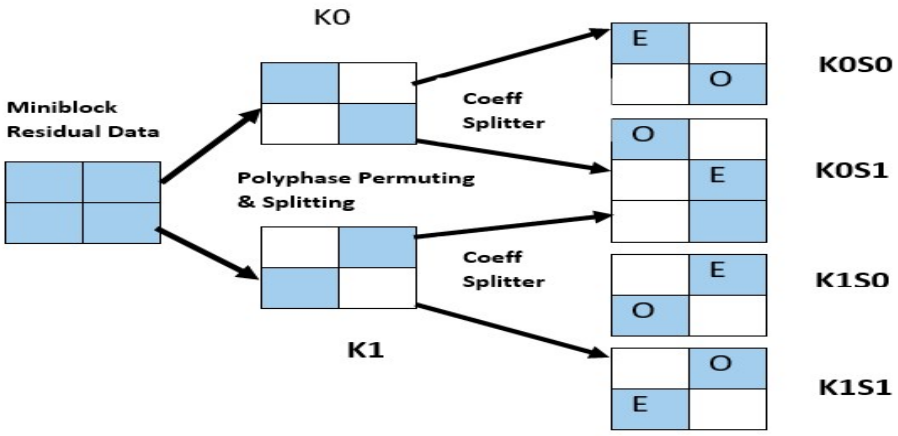


Fig. 6. Mini block pattern after two-level splitting

To more accurately estimate the missing description, even and odd blocks are alternately allocated in the resulting microblock diagonally. Additionally, since O-blocks have one more coefficient than E-blocks, this arrangement balances the difference in consistency and bit rate between the two types of blocks.As a result of the two-level splitting, each leftover microblock is divided into four microblocks, one for each descriptor. It should be noted that each 2x2 block is divided into two 2x2 blocks, each containing almost half of the original coefficients after the coefficient splitter is applied. In each resulting 2x2 block, nearly half of the coefficients are zero because those allocated to one block will have their values fixed to zero in the other block. The coding efficiency is affected, but for the resulting blocks, entropy codings such as CABAC or CAVLC [29] are still valid.

A two-level merging is used to create the full details of the reference frame. The coefficient merger is used to combine K0S0 and K0S1 in the frequency domain and to merge K1S0 and K1S1. The merger is carried out by applying ac [23] coefficients to the same even and odd block positions by selecting one of the two duplicated DC coefficients. K0 and K1 are obtained after the inverse transformation, and then the second level merging is implemented based on 4x4 blocks. First, by discarding the all-zero 2x2 blocks, the leftover merger is performed, and then polyphase inverse permuting is done to recreate the initial 4x4 blocks. This process is applied to all four 4x4 blocks for each microblock.

3.2.2. Image 16x16 Block Size Split Codec:

In the encoding loop, the Image 16x16 block size split (Macroblock) encoder [1] uses a two-level splitting technique. The working principle of the image 16x16 block size split is similar to the image 8x8 block size split. After the initial-level splitting, the encoding route is divided into two and then four after the secondary-level splitting. The difference between the image 8x8 block size split and the image 16x16 block size split is the block size split. Image 8x8 block size split [26] is split into 8x8 block size, and image 16x16 block size split is divided into 16x16 block size.

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

3.2.3. Image 32x32 Block Size Split Codec:

The image 32x32 block size split (Mega block) encoder optimizes the reconstructed image's error resilience, transmission, and processing time while minimising data loss on each channel. The working principle of the image 32x32 block size split is similar to the image 8x8 block size split. In the encoding loop of the image 32x32 block split encoder, there are two levels of splitting. As illustrated in Fig. 7, the leftover data is first permuted within the block in polyphase in each 16x16 block before being separated into two blocks. As seen in the middle of Fig. 6, the polyphase permutation rearranges all of the pixels.

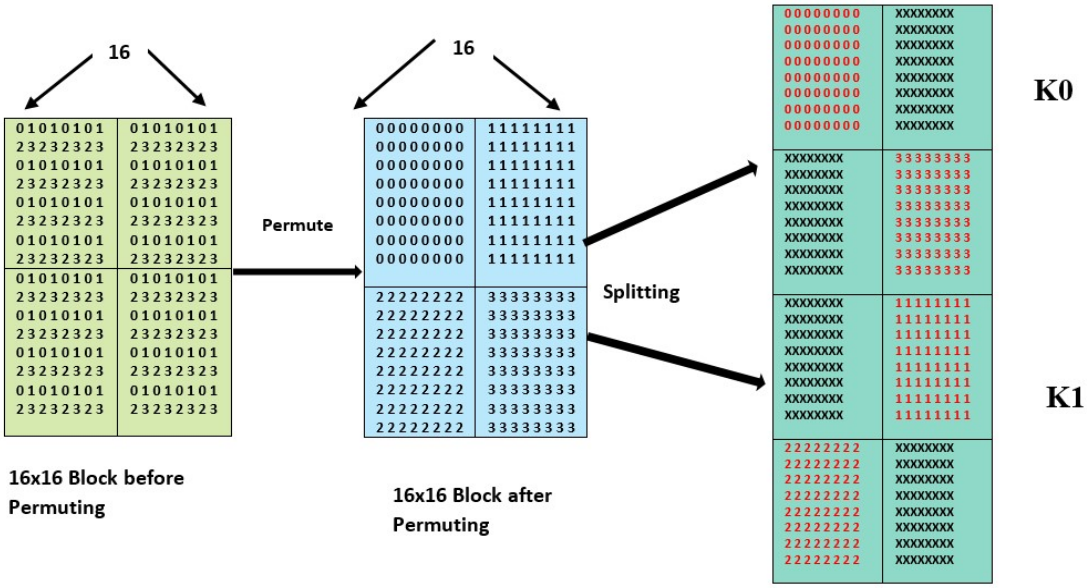


Fig.7. Permuting and splitting of a 16X16 block

Fig. 8 indicates how an 8x8 block is divided into two 8x8 blocks, each of which contains approximately half the original coefficients. In an alternate diagonal layout, the E-blocks and O-blocks are each allotted to two descriptors.

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

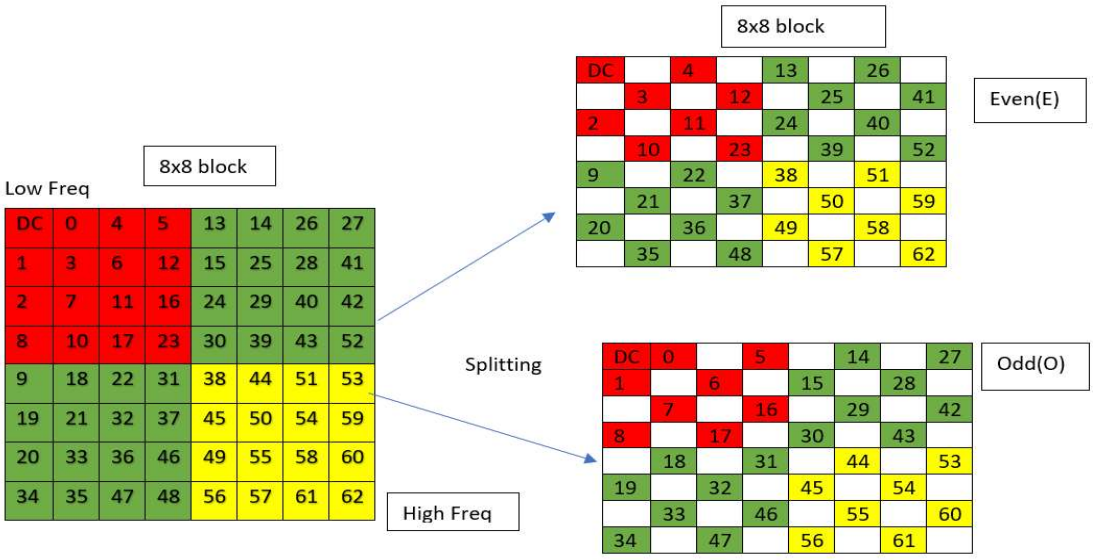


Fig. 8. Even and Odd 8x8 blocks generated by Coefficient Splitter

In Fig. 9, K0 and K1 are the megablocks obtained from the initial level splitting. The white, colourless blocks in K0 and K1 are all-zero 8x8 blocks. The Coefficient Splitter second-level splitting is applied to K0 and K1 to divide each non-zero 8x8 block into E-blocks and O-blocks, which are then allocated to two different megablocks, S0 and S1. After inverse transformation, K0 and K1 are obtained, and the second-level merging is performed using 16x16 blocks. For each megablock, the four 16x16 blocks are processed in the same manner.

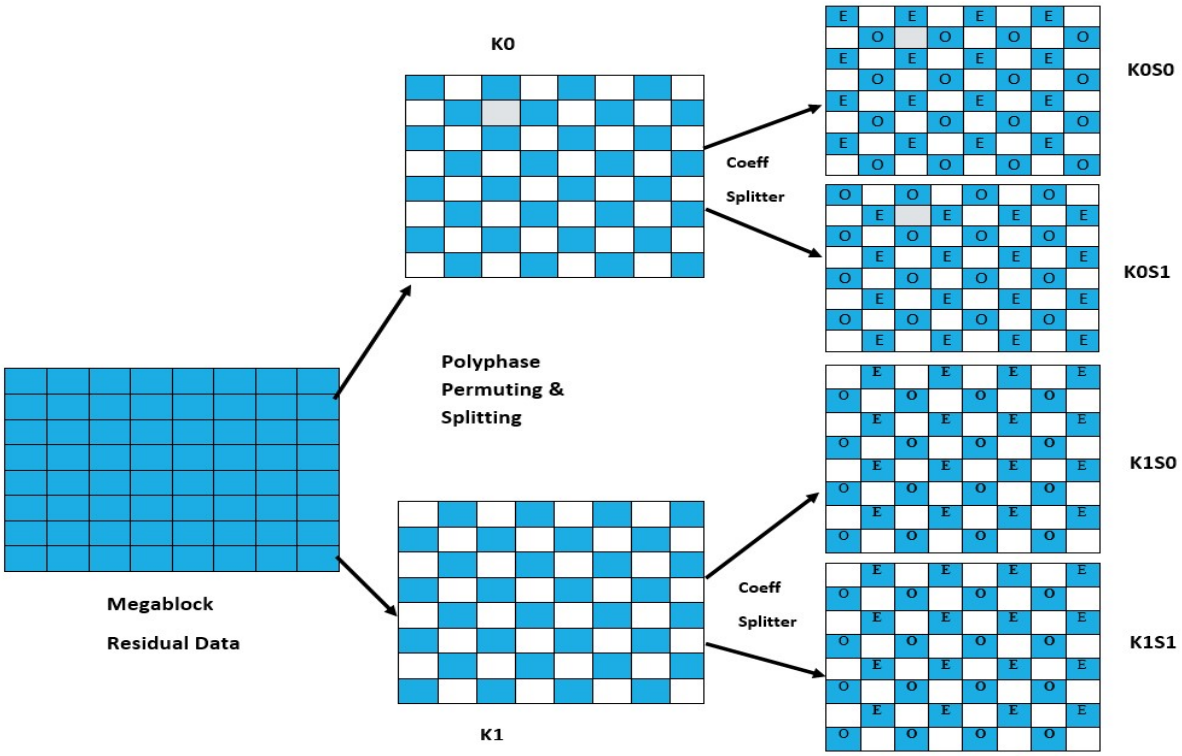


Fig.9 Mega block pattern after two level splitting

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

3.2.4. Image 64 ×64 Block Size Split Codec:

The Image 64x64 block size split (Super Mega block) encoder utilises a two-level dividing procedure in the encoding phase. Its working principle is similar to that of the 8x8 block size split. After the first-level splitting, the encoding pathway is divided into two and then into four after the secondary-level splitting, as illustrated in Fig. 10. The difference between the image 8x8 block size split and the image 64x64 block size split is the block size split. While the image 8x8 block size split divides the image into 8x8 blocks, the image 64x64 block size split divides it into 64x64 blocks.

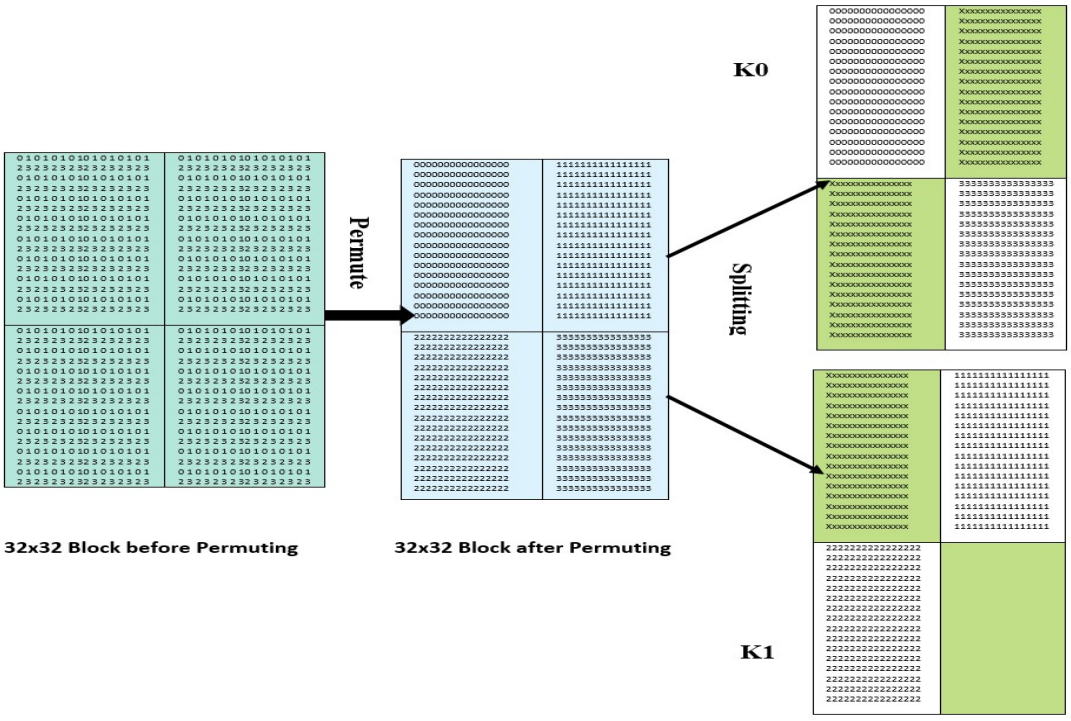


Fig. 10. Permuting and Splitting of a 32x32 block

In Fig. 11, a 16x16 block is split into two 16x16 blocks, each containing approximately half of the original coefficients. Almost half of the coefficients in each 16x16 block are zero because the coefficients assigned to one block are fixed to zero in the other block.

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

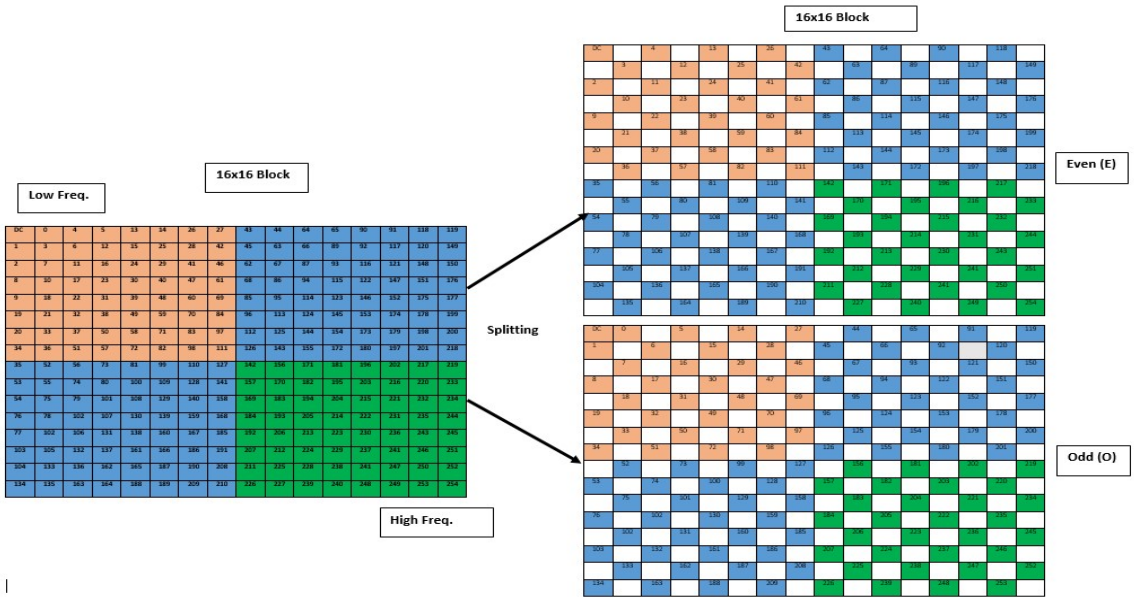


Fig. 11. Even and Odd 16x16 blocks generated by Coefficient Splitter

3.3 HMDC Decoder Architecture:

The structure of the HMDC is illustrated in Fig. 12. The decoder architecture for the Image 8x8 block size split consists of four descriptors, namely K0S0, K0S1, K1S0, and K1S1. These descriptors are first entropy-decoded separately. Next, on the encoder side, the "Coefficient Merger" and "Leftover Merge and Polyphase Inverse Permuting" processes are performed similarly [20].

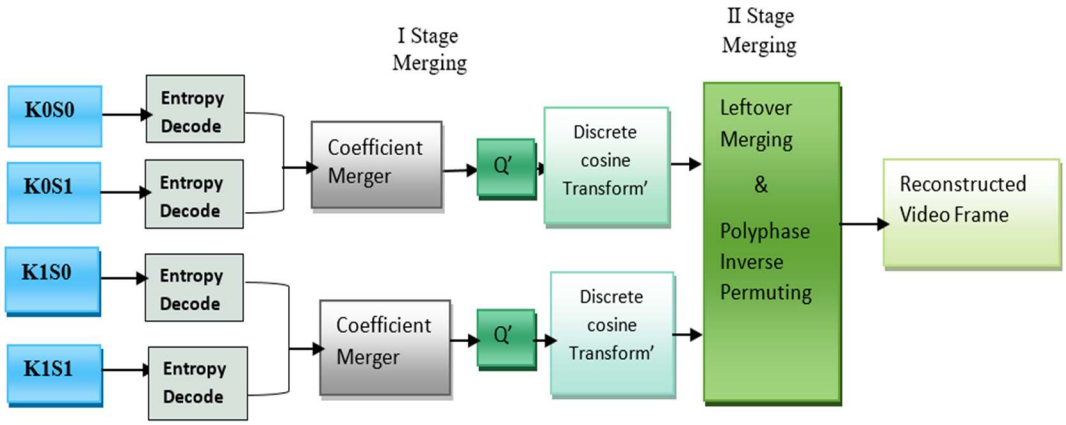


Fig.12 HMDC Decoder Structure

The lost description method of frequency estimation is used to rebuild lost data. Frequency estimation is performed through the blocks in the leftover domain's counterpart by the AC coefficient estimate. The two leftover 4x4 blocks are named K0 and K1, generated as the two leftover areas from the initial level splitting. Two adjacent 2x2 blocks in separate leftover areas have a one-pixel gap in the 4x4 block of the original image due to polyphase permutation. It should be efficient to estimate the lost description by AC prediction from the neighbouring blocks in the

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

leftover domain's counterpart. The frequency estimate is used in four instances of one descriptor failure in Fig. 13. For instance, in the case of K1S1 failure, K0S1 is used to predict AC. Additionally, since E-blocks and O-blocks include complementary coefficients, the estimate is consistent with the idea that E-blocks are anticipated from E-blocks and O-blocks are predicted from O-blocks. For the same instance where K1S1 is lost, the lost O-blocks in K1S1 are forecasted from the left neighbouring O-blocks in K0S1, but the lost E-blocks in K1S1 are forecasted from the right neighbouring E-block in K0S1. The predictions for the four situations, as seen in Fig. 13, are all made in a horizontal orientation. The lost, leftover domain can be restored after the forecasting for the lost descriptor. Four out of the six possible instances demonstrate that two descriptors are missing. They employ frequency estimation, and the descriptors used for estimating come from a duplicate of the leftover area as well. In the image's 32x32 block size split decoder's architecture, the four descriptors are K0S0, K0S1, K1S0, and K1S1. Entropy is used to independently decode these descriptors, after which the encoder side processes "Coefficient Merger," "Leftover Merge," and "Polyphase Inverse Permuting" in a similar manner. If the decoder does not receive all of the descriptors, the data is reconstructed using the lost description method of frequency estimation.

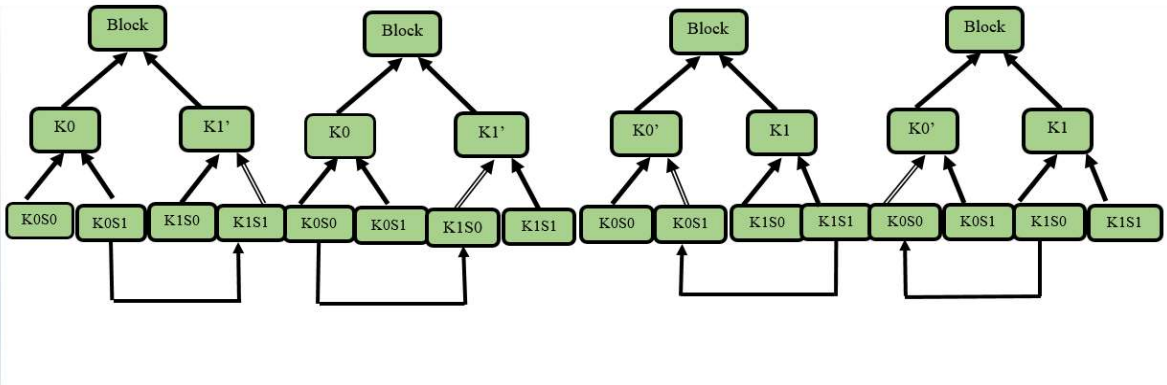


Fig.13 Ac -prediction for two missing Channels

The names of the four descriptors for the Image 64x64 block size splitting decoder design are K0S0, K0S1, K1S0, and K1S1. These descriptors are independently entropy-decoded. Afterward, the encoder performs "Coefficient Merge" and "Leftover Merge and Polyphase Inverse Permuting" in a similar manner. If the decoder does not receive all of the descriptors intact, the missing data is recreated using the frequency estimate method. In the reconstruction phase, a Gaussian filter [30] was used to improve the image quality. The filter's probability density function can be described using the following generic form:

$$f(a) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{a-\mu}{\sigma}\right)^2} \quad (1)$$

In this case, 'a' stands for the grey level, 'μ' stands for the mean value, and 'σ' stands for the standard deviation. After applying the Gaussian filter, the image is enhanced using the bicubic interpolation method. This paper only analyses colour images. To achieve better quality

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

improvement in the reconstructed frame, we use the bicubic interpolation method. We use the proposed method of image block size splitting for error-resilient transmission.

3.5 Performance Evaluation Metrics

The algorithms were developed in Matlab 2020a and validated using performance metrics. We used five standard test frames with 720x1280 dimensions to evaluate the proposed frame's efficiency.

3.5.1 Frame Compression Ratio

The compression ratio (CR) is defined as the ratio of the original frame size to the compressed frame size.

$$\text{CR} = \frac{\text{Original frame Size}}{\text{Compressed frame Size}} \quad (2)$$

3.5.2 Peak-to-Signal Noise Ratio

The PSNR value [31] is the most widely used objective video quality measurement. The PSNR block compares two pictures and evaluates the PSNR. The PSNR is calculated using Equation 3.

$$PSNR\ Value\ of\ frame = 10 * log_{10}\left({255^2}/{MSE}\right) \quad (3)$$

3.5.3 The Structural Similarity Index Measurement

The method for calculating the similarity between two images is called the SSIM [31] [32]. The SSIM value is a decimal number between -1 and 1, with 1 indicating that the two frames are structurally identical. The SSIM is calculated using Equation 4:

$$Frame\ SSIM(x,y) = \frac{(2\mu_x\,\mu_y + p_1)(2\sigma_X\sigma_y + p_2)}{(\mu^2{}_x + \mu^2{}_y + p_1)(\sigma^2{}_{x+}\sigma^2{}_y + p_2)} \quad (4)$$

Where 'x' and 'y' represent windows in the original and denoised image, respectively. 'σ' and 'μ' denote the standard deviation and mean of 'x' and 'y', and 'p1' and 'p2' are constants.

3.5.4 Compression Percentage

The compression percentage expresses the difference between the original image size of a file and its final size after compression. It is defined as follows:

$$Compression\ percentage = 100 * \frac{original\ frame.\ lengt - compressed\ frame.\ length}{original\ frame\ .leng} \quad (5)$$

3.5.5 PSNR Percentage

The PSNR is a commonly used measure of image quality, with values typically ranging from 30 to 50 dB, with higher values indicating better quality as long as the bit depth is 8 bits. The PSNR percentage between two images is defined as follows:

$$\text{PSNR Percentage} = (\text{(PSNR of reconstructed image)} - \text{(PSNR of the original image))} / \text{(PSNR of the original image)} * 100\%$$

This measure indicates the percentage decrease in PSNR between the original and reconstructed images.

Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

$$\text{PSNR Percentage} = (\text{PSNR Value}/50) * 100 \quad (6)$$

4. Experimental Results and Discussion

This section presents the results of both objective and subjective experiments to evaluate the performance of the proposed scheme. Five standard video frames with dimensions of 720x1280 pixels are used to assess the quality of the reconstructed frames using the proposed distributed framework. The visual quality of the reconstructed frames is evaluated using PSNR and structural similarity index measurements. The frames used in this experiment are colour images with a size of 27,64,800 bytes. The HMDC method is used for performance evaluation, and five test sequences are used: foreman, coastguard, car phone, mobile, and news. The performance of the proposed method in terms of compressed file size, compression ratio, PSNR value, SSIM, encoding time, and decoding time is shown in Tables 1, 2, 3, and 4. The proposed image block size split method for error resilience transmission using HMDC methods is simulated in Matlab 2021a. In this paper, a comparative analysis of error and resilience transmission is done using the DCT method. DCT provides a higher compression ratio and better quality of the reconstructed frame while being computationally efficient compared to other techniques. The performance of various error-resilience transmission algorithms is evaluated on different types of frames. Based on the results presented in Tables 1, 2, 3, and 4, it can be concluded that the image block size split method provides better PSNR and structural similarity index measurement, which compares two frames based on brightness, contrast, and structure. The effectiveness of the suggested HMDC techniques in an ideal channel environment is also investigated in this section. In situations where some descriptors are presumed to be received with no information lost, while others are lost entirely, side rebuilding is used. Table 1 shows the image 8x8 block size split based on HMDC performance metrics.

Table 1: image 8x8 block size split-based HMDC

Image Name	Compressed File Size	Compression Ratio	SSIM (colour Image)	Processing Time (seconds)
Foreman	**2278658**	1.2	0.97774	139
Mobile	**2183972**	1.3	0.97634	138
Coastguard	**2182300**	1.3	0.97642	135
Traffic	**2198082**	1.25	0.97641	137
Ship	**2197573**	1.26	0.96693	136

Encoding time and decoding time vary depending on the frame size split, frame block size, network speed, and software. The 8x8 block size image split takes more time to encode and decode the frame, and the 64x64 block size image split takes less time to encode and decode the frame.

Table 2: image 16× 16 block size split-based HMDC

Image Name	Compressed File Size	Compression Ratio	SSIM (colour image)	Processing Time (seconds)
Foreman	825963	3.35	0.96488	35
Mobile	839834	4.04	0.96404	39
Coastguard	826367	3.35	0.96416	40
Traffic	806057	3.43	0.96342	37
Ship	825973	3.35	0.95632	31

Table 2 shows the image 16x16 block size split, and Table 3 shows the image 32x32 block size split values.

Table 3: image 32 ×32 block size split-based HMDC

Image Name	Compressed File Size	Compression Ratio	SSIM (colour Image)	Processing Time (seconds)
Foreman	180727	15.30	**0.88519**	**9**
Mobile	180031	15.36	**0.88589**	**10**
Coastguard	206801	13.36	**0.88678**	**11**
Traffic	181457	14.24	**0.88474**	**8**
Ship	203936	13.55	**0.86529**	**10**

Table 4: image 64 ×64 block size split-based HMDC

Image Name	Compressed File Size	Compression Ratio	SSIM (colour Image)	Processing Time (seconds)
Foreman	134821	20.51	**0.71913**	4.1
Mobile	123841	23.34	**0.74844**	4.5
Coastguard	134616	20.54	**0.75715**	4
Traffic	122700	22.53	**0.74699**	4.4
Ship	125424	22.04	**0.74715**	4.2

Table 4 presents the values obtained using the image block size split of 64x64. The encoding and decoding times are notably lower in the 64x64 method compared to the 8x8 method. In Figure 14, the compressed file size of the image is illustrated. It can be observed that the image block size split of 8x8 results in a larger compressed file size, whereas the 64x64 block size split method produces a smaller compressed file size.

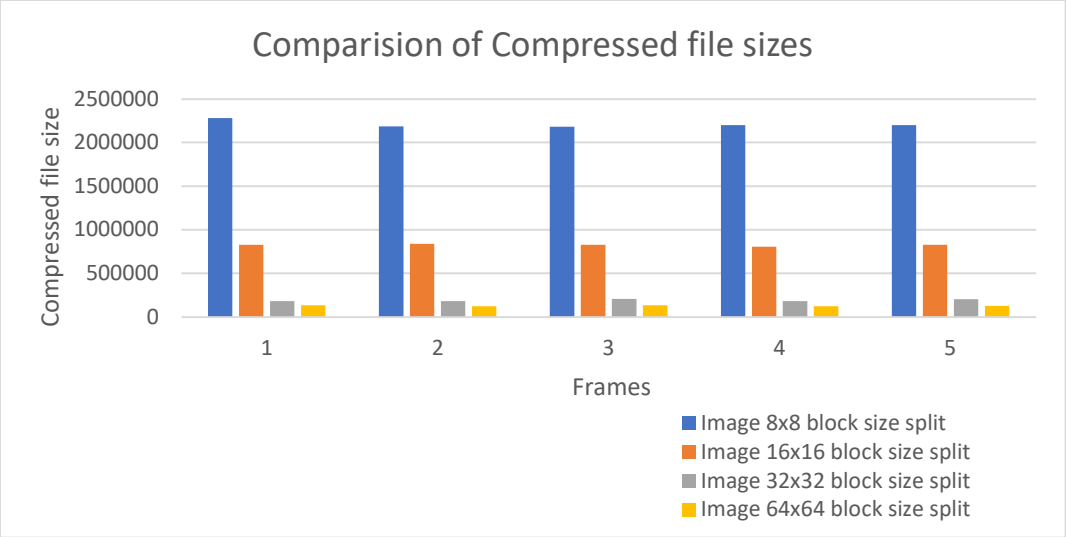


Fig.14 Compressed file size Vs Image block size split for different Frames

Figure 15 displays the compression ratio of the frames. The image block size split of 32x32 and 64x64 results in high compression ratios, whereas the block size split of 16x16 yields a medium compression ratio, and the block size split of 8x8 gives a low compression ratio.

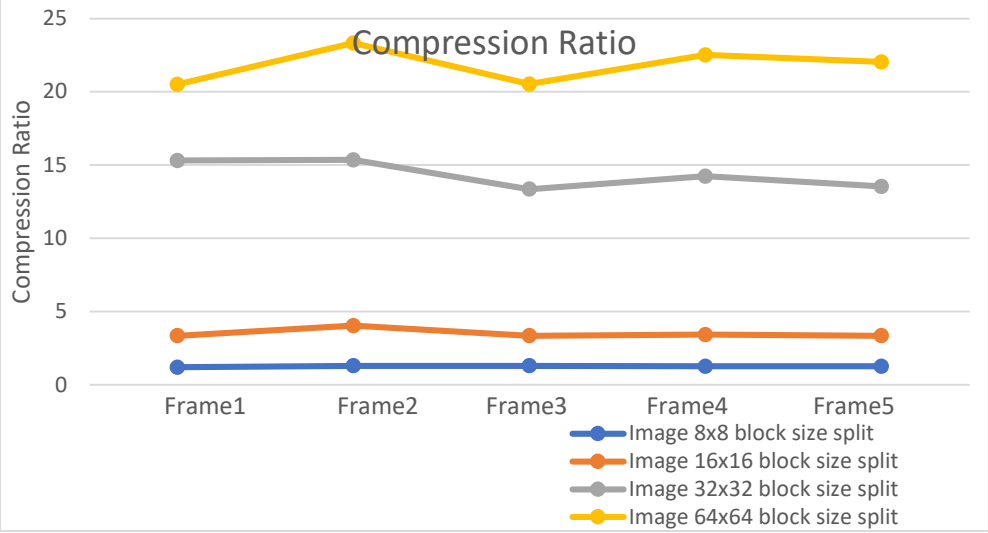


Fig.15. Compression Ratio Vs Image block size split for different frames

The image 8x8 block size split gives a low compression ratio, while the image 64x64 block size split gives a high compression ratio, and the image 16x16 block size split and the image 32x32 block size split produce a medium compression ratio. The PSNR value and the SSIM value are higher in the 8x8 block size split and lower in the image 64x64 block size split. Fig. 16 shows the SSIM value. The PSNR value and the SSIM value are medium in the image 16x16 block size split and the image 32x32 block size split.

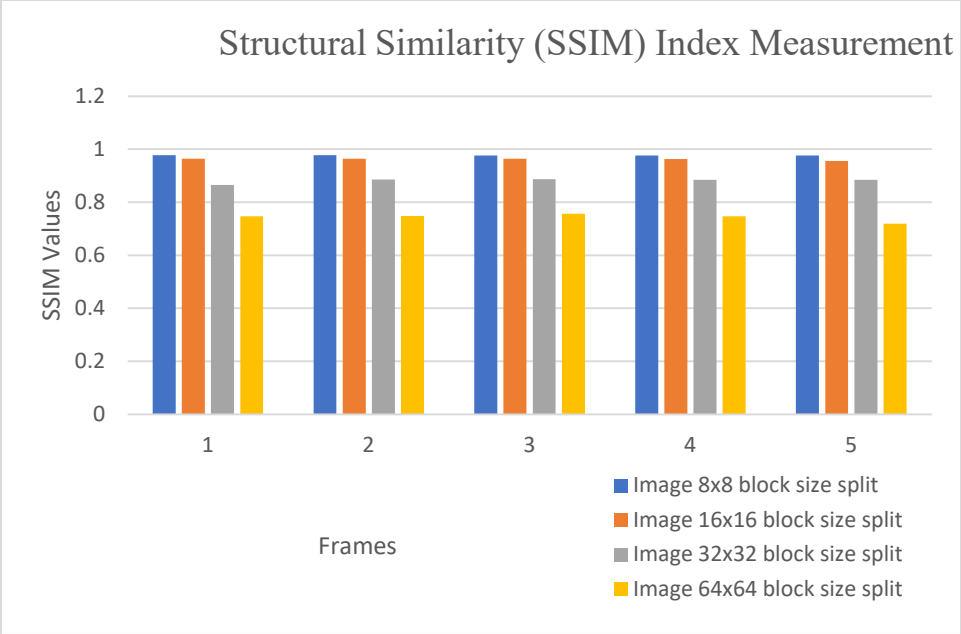


Fig.16. SSIM Value Vs image block size split for different Frames

Fig. 17 shows the processing time of the frames. The image 32x32 block size split and the image 64x64 block size split take less compression time. However, the image 8x8 block size split takes a longer compression time, and the image 16x16 block size split takes a medium compression time.

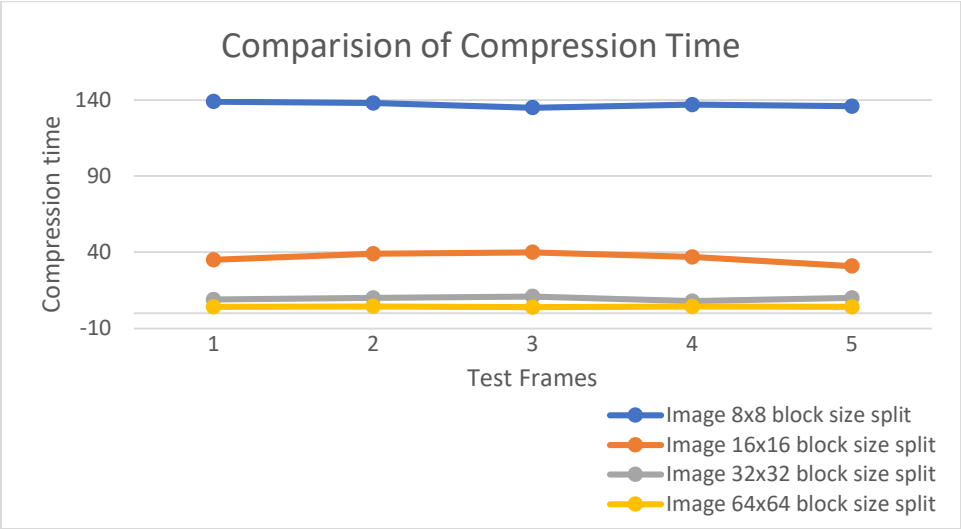


Fig.17 Processing Time Vs image block size split for Different Frames

Fig. 18 illustrates the average processing time of the frames. The average processing time of frames can depend on the size of the image block split. The average processing time decreases when the images are split into 32x32 and 64x64 block sizes. However, the image 8x8 block size split takes a long time to compress, while the image 16x16 block size split takes a medium amount of compression time.

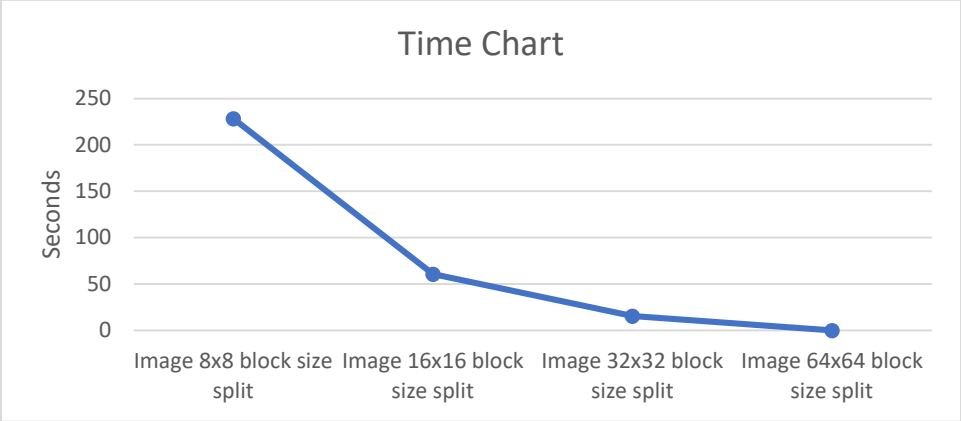


Fig.18 Average Processing Time Vs image block size split for different Frames

The average compression ratio percentage and the average PSNR percentage value are also displayed in Fig. 19. A low compression ratio percentage is produced by the image 8x8 block size split, a high compression ratio percentage is produced by the image 64x64 block size split, and a medium compression ratio percentage is produced by the image 16x16 block size split and the image 32x32 block size split. The PSNR percentage value is higher in the image 8x8 block size split and lower in the image 64x64 block size split. The PSNR percentage values are medium in the image 16x16 block size split and 32x32 block size split.

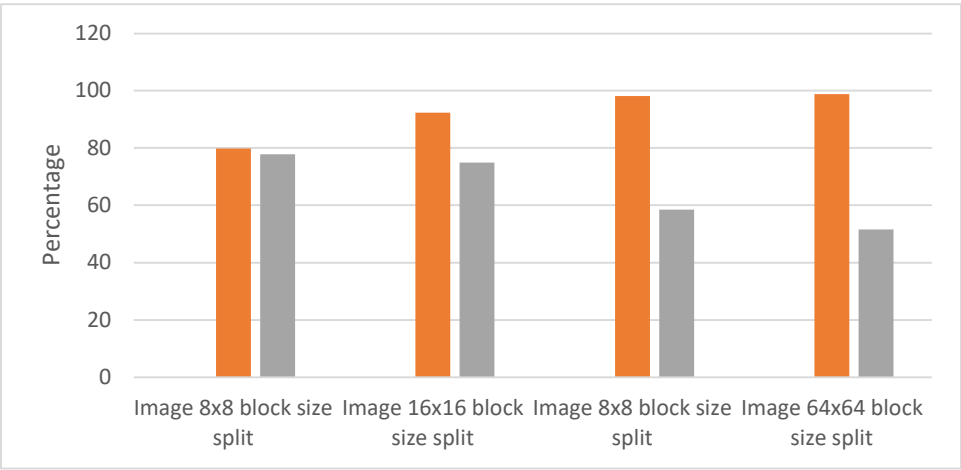
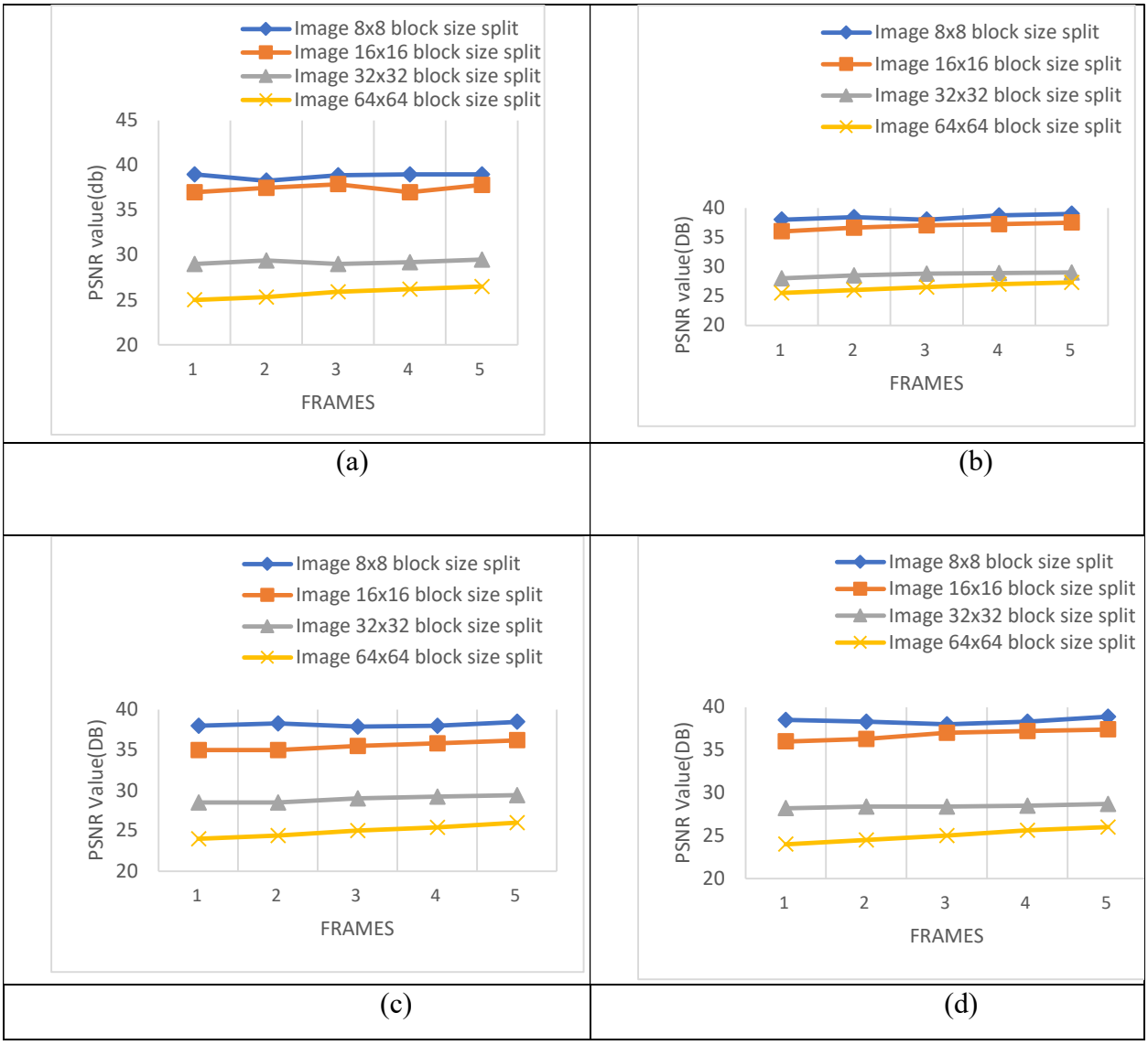


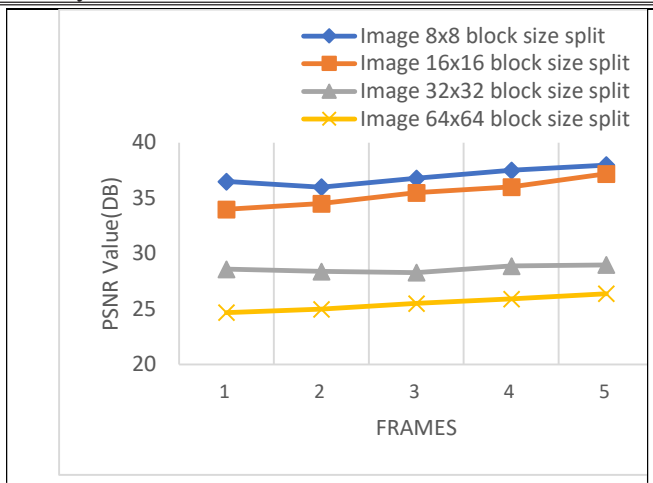
Fig.19 Average Compression Ratio & PSNR Value Vs image block size split for different Frames

The average PSNR of the nine-channel loss scenarios is provided in Fig. 20, which displays the results of five test sequences. The Image 8x8 Block Size Split approach was found to produce the best estimation for the Foreman and News sequences, whereas the Image 32x32 Block Size Split and Image 64x64 Block Size Split sequences performed the poorest. The outcomes show that coefficient duplication is preferable to use in poor channel circumstances. One missing channel:

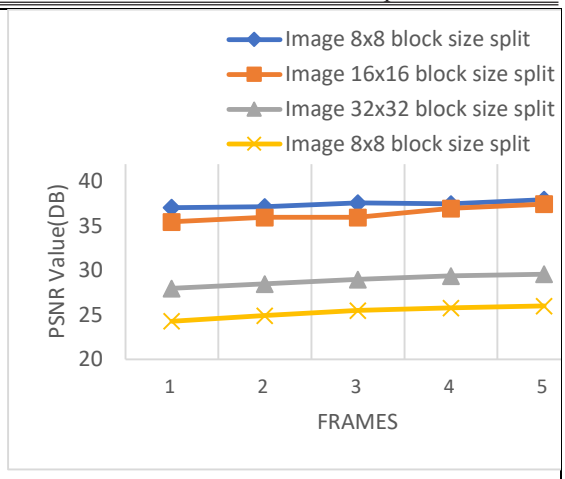
Catalyst Research Volume 23, Issue 2, November 2023 Pp. 3194-3216

The performance results of one-descriptor loss shown in Fig. 20(a–c) show the reconstructed PSNR of three sequences. The depicted PSNR is the average of the four probable scenarios of one-descriptor loss. In Fig. 20(a), (b), and (c), it can be shown that the image 8x8 block size split outperformed the others. Two Missing Channels: Figure 20(d-f) shows the effects of a two-channel loss. The reconstructed PSNR of three sequences is shown in Figure 20(d–f). The PSNR displayed is the average of the six probable two-channel loss instances. As shown in Fig. 4, frequency estimation of lost descriptions was applied for two descriptor losses, and the results demonstrate their effects. In Fig. 20(d)–(f), the image 8x8 block size split technique performed better than all others for all sequences. Frequency estimation of lost description was used for two channel losses, and the results show their effects. Three missing channels: The performance results of three-channel loss for three different sequences are shown in Fig. 20(g-i). The PSNR displayed is the average of four different three-channel loss scenarios. When more information is lost, the consequences of estimating the lost description rise, demonstrating the superiority of the estimation methodologies utilised in the hybrid method.

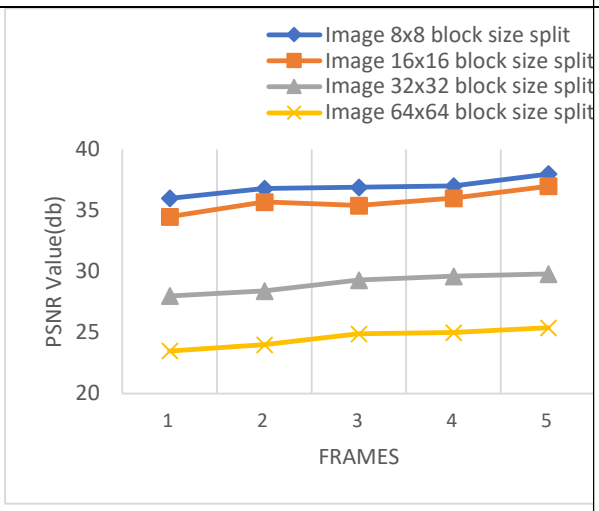




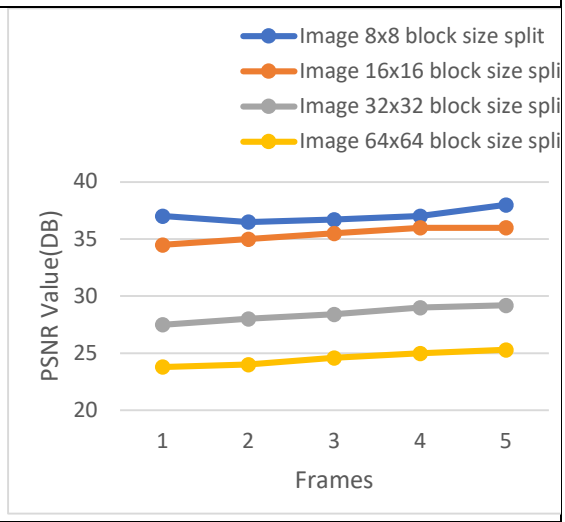
(e)



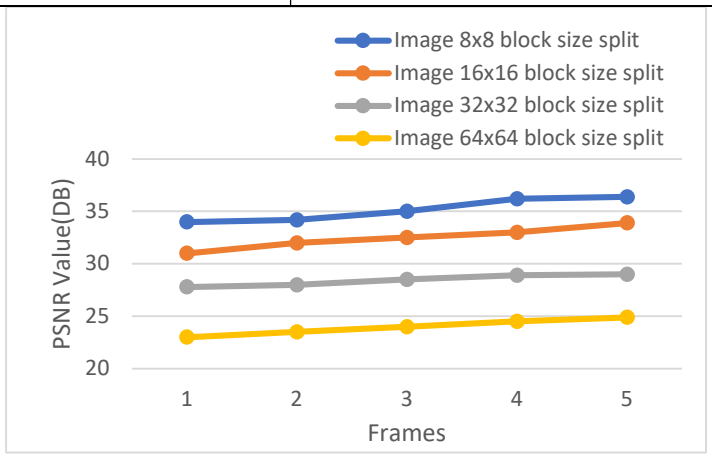
(f)



(g)



(h)



(i)

Fig.20 PSNR of one, two, and three-channel loss.

a) Foreman - one channel loss. b) Mobile - one channel loss. c) Coastguard - one channel loss. d) Foreman – Two-channel loss. e) Mobile – Two-channel loss. f) Coastguard – Two-channel loss. g) Foreman – Three-channel loss. h) Mobile – Three-channel loss. i) Coastguard – Three-channel loss.

Conclusion:

Our research presents a robust error-resilient transmission model for video data, utilising HMDC with varying block sizes for image splitting. The employment of the DCT for transform domain conversion and a frame size of 720 × 1280 have been explored. The findings reveal that the 8x8 block size split yields a reconstructed image of high quality, as evidenced by its high PSNR and SSIM scores. Conversely, the 64x64 block size split produces lower-quality reconstructed images but boasts faster computation times. The 32x32 block size split strikes a balance, offering moderate compression times, medium-quality reconstructed images, and moderate compression ratios, albeit lower than the 64x64 block size split. While the 8x8 block size split demands more execution time and provides a lower compression ratio, it remains a viable option for certain applications. Notably, an optimal block size of 16 × 16 emerges as a recommendation for real-time applications, offering a balanced compromise suitable for regular applications while preserving crucial information. It is imperative to recognise that the compression time, compression ratio, and PSNR value exhibit exponential changes for each block size split. This research provides valuable insights for researchers in the field of data transfer through networks, shedding light on the trade-offs and considerations when selecting block sizes in HMDC-based error-resilient video transmission. As technology advances, these findings can contribute to more efficient and effective video transmission solutions.

Declaration

The authors declare that they have no known completing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Conflicts of interest

The authors declare that they have no conflict of interest.

Author Contributions

All authors equally contributed to preparing, reviewing, editing, and implementing the manuscript.

REFERENCES

[1] C. W. Hsiao and W. J. Tsai, “Hybrid multiple description coding based on H.264,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 1, pp. 76–87, 2010, doi: 10.1109/TCSVT.2009.2026973.

[2] Y. Wang, A. R. Reibman, and S. Lin, “Multiple description coding for video delivery,” *Proc. IEEE*, vol. 93, no. 1, pp. 57–69, 2005, doi: 10.1109/JPROC.2004.839618.

[3] K. Dimililer, “DCT-based medical image compression using machine learning,” *Signal, Image Video Process.*, vol. 16, no. 1, pp. 55–62, 2022, doi: 10.1007/s11760-021-01951-0.

[4] H. A. Karim, C. T. E. R. Hewage, S. Worrall, and A. M. Kondoz, “Scalable multiple description video coding for stereoscopic 3D,” *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 745–752, 2008, doi: 10.1109/TCE.2008.4560156.

[5] H. Zheng, C. Ru, L. Yu, and C. W. Chen, “Robust video transmission over MIMO-OFDM system

using MDC and space time codes," *2006 IEEE Int. Conf. Multimed. Expo, ICME 2006 - Proc.*, vol. 2006, pp. 633–636, 2006, doi: 10.1109/ICME.2006.262488.

[6] W. Khalaf, A. S. Mohammad, and D. Zaghar, "Chimera: A new efficient transform for high quality lossy image compression," *Symmetry (Basel).*, vol. 12, no. 3, 2020, doi: 10.3390/sym12030378.

[7] P. Acelas, P. Arce, J. C. Guerri, and W. Castellanos, "Evaluation of the MDC and FEC over the quality of service and quality of experience for video distribution in ad hoc networks," *Multimed. Tools Appl.*, vol. 68, no. 3, pp. 969–989, 2014, doi: 10.1007/s11042-012-1111-3.

[8] M. Kazemi, S. Shirmohammadi, and K. H. Sadeghi, "A review of multiple description coding techniques for error-resilient video delivery," *Multimed. Syst.*, vol. 20, no. 3, pp. 283–309, 2014, doi: 10.1007/s00530-013-0319-z.

[9] Z. Chen, T. He, X. Jin, and F. Wu, "Learning for Video Compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 566–576, 2020, doi: 10.1109/TCSVT.2019.2892608.

[10] Y. C. Lee, J. Kim, Y. Altunbasak, and R. M. Mersereau, "Layered coded vs. multiple description coded video over error-prone networks," *Signal Process. Image Commun.*, vol. 18, no. 5, pp. 337–356, 2003, doi: 10.1016/S0923-5965(02)00138-8.

[11] D. Wang, N. Canagarajah, D. Redmill, and D. Bull, "Multiple description video coding based on zero padding," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2, pp. 1–4, 2004, doi: 10.1109/iscas.2004.1329244.

[12] T. Tang, A. Wang, S. Muhaidat, S. Li, M. Li, and J. Liang, "MDC-NOMA: Multiple Description Coding-Based Nonorthogonal Multiple Access for Image Transmission," *IEEE Syst. J.*, pp. 1–10, 2020, doi: 10.1109/JSYST.2020.3010791.

[13] C. Fan, B. Lu, X. Feng, W. Gao, and C. Wang, "Efficient multi-qubit quantum data compression," *Quantum Eng.*, vol. 3, no. 2, pp. 1–8, 2021, doi: 10.1002/que2.67.

[14] U. Jayasankar, V. Thirumal, and D. Ponnurangam, "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 33, no. 2, pp. 119–140, 2021, doi: 10.1016/j.jksuci.2018.05.006.

[15] R. V. Vadhana, M. R. Revathi, R. Venkadesh, and T. R. Anns, "Performance Evaluation of Low Bitrate Video Compression using Adaptive Motion Matching," vol. 4, no. 1, pp. 21–27, 2017.

[16] P. Rajasekar and L. N. B. Srinivas, "Digital Image Compression using DCT Algorithm: An Improvement," *Int. J. Sci. Eng. Res.*, vol. 10, no. 2, pp. 1291–1295, 2019.

[17] W. Xiao, N. Wan, A. Hong, and X. Chen, "A Fast JPEG Image Compression Algorithm Based on DCT," *Proc. - 2020 IEEE Int. Conf. Smart Cloud, SmartCloud 2020*, pp. 106–110, 2020, doi: 10.1109/SmartCloud49737.2020.00028.

[18] Z. Liu, H. Chen, and S. Sun, "Video error-resilience research based on error-resilient screen content," *Appl. Sci.*, vol. 10, no. 14, 2020, doi: 10.3390/app10144923.

[19] S. Ying and R. Yin, "Image Degradation by Zeroing Block DCT vs. Full DCT Coefficients," *ACM Int. Conf. Proceeding Ser.*, no. 3, pp. 5–8, 2020, doi: 10.1145/3417473.3417484.

[20] M. Gupta, "Analysis Of Image Compression Algorithm Using DCT Maneesha Gupta *, Dr . Amit Kumar Garg **," vol. 2, no. 1, pp. 515–521, 2012.

[21] S. Petersson and H. Grahn, "Improving image quality by SSIM based increase of run-length zeros in GPGPU JPEG encoding," *Conf. Rec. - Asilomar Conf. Signals, Syst. Comput.*, vol. 2015-April, pp. 1714–1718, 2015, doi: 10.1109/ACSSC.2014.7094760.

[22] M. Dong, H. Zeng, J. Chen, C. Cai, and K. K. Ma, "Multiple description video coding based on adaptive data reuse," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 378–385, 2016, doi:

10.1016/j.jvcir.2016.03.016.
[23] M. P. Singh and V. Pandey, "Block wise image compression & Reduced Blocks," *Int. J. Sci. Res. Publ.*, vol. 5, no. 3, pp. 1–10, 2015, [Online]. Available: https://www.researchgate.net/publication/279979930.
[24] S. Singh and S. Assistant, "Efficient Image Compression using all the Coefficients of 16x16 DCT Image Sub-Block," vol. 4, no. 8, pp. 4–8, 2015.
[25] S. Devi, "Image Compression Using Discrete Cosine Transform (DCT) & Discrete Wavelet Transform (DWT) Techniques," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. V, no. X, pp. 1689–1696, 2017, doi: 10.22214/ijraset.2017.10246.
[26] M. M. Sathik, K. S. Kannan, and Y. J. V. Raj, "H YBRID JPEG C OMPRESSION U SING E DGE B ASED S EGMENTATION," vol. 2, no. 1, pp. 165–176, 2011.
[27] L. Zhao, H. Bai, A. Wang, and Y. Zhao, "Multiple Description Convolutional Neural Networks for Image Compression," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–13, 2018, doi: 10.1109/TCSVT.2018.2867067.
[28] R. A.M, K. W.M, E. M. A, and W. Ahmed, "Jpeg Image Compression Using Discrete Cosine Transform - A Survey," *Int. J. Comput. Sci. Eng. Surv.*, vol. 5, no. 2, pp. 39–47, 2014, doi: 10.5121/ijcses.2014.5204.
[29] G. C. Harisha, N. Murugendrappa, B. S. Manjushree, C. M. Kunjana, L. S. C, and B. L. Mahima, "I MPLEMENTATION OF C ONTEXT A DAPTIVE V ARIABLE L ENGTH C ODING AND D ECODING A LGORITHM FOR H . 264 VIDEO CODEC USING M ATLAB," vol. 11, no. 3, pp. 113–121, 2018.
[30] Z. J. Min, "Analysis and Comparison on Image Restoration Algorithms Using MATLAB," *Int. J. Eng. Res. Technol.*, vol. 2, no. 12, pp. 1350–1360, 2013.
[31] Y. Y. . Al-najjar and D. C. . Soong, "Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI," *Int. J. Sci. Eng. Res.*, vol. 3, no. 8, pp. 1–5, 2012, [Online]. Available: http://www.ijser.org/researchpaper%5CComparison-of-Image-Quality-Assessment-PSNR-HVS-SSIM-UIQI.pdf.
[32] R. Mishra, N. Mittal, and S. K. Khatri, "Digital Image Restoration using Image Filtering Techniques," *2019 Int. Conf. Autom. Comput. Technol. Manag. ICACTM 2019*, pp. 268–272, 2019, doi: 10.1109/ICACTM.2019.8776813.