IoT-Driven Visual Surveillance: Temporal Masking for Adaptive Motion Compensation in Imaging Technology

Ali Akbar Siddique[1], Wad Ghaban[2], Amer Aljaedi[3], Faisal Saeed[4], Mohammad S. Alshehri[5], Ahmed Alkhayyat[6], Hussain Mobarak Albarakati[7], Member, IEEE

Abstract — *Global security is a matter of critical concern that requires adoption of advanced monitoring technologies. Efficient surveillance systems comprise extensive camera networks across large areas to ensure comprehensive coverage. However, the large volume of data generated by these networks poses challenges for traditional storage and computational resources. This paper presents an innovative video compression technique that focuses on optimizing data management in visual surveillance systems by selectively masking temporal information between frames. This technique introduces a specially designed adaptive masking filter, which hides the undetectable motion in video sequences and enhances video compression. The introduced masking technique uses an adaptive masking parameter 'q' to improve frame prediction or to compensate for the masked temporal activity during decoding and achieves over 30% bit-rate reduction compared to the standard video encoding schemes, such as H.264/AVC. Moreover, the introduced technique also reduces the computational demands while keeping the quality of the output. This can be evidenced by a Peak Signal to Noise Ratio (PSNR) of 33.67 dB and a Structural Similarity Index (SSIM) of 92.7% in a traffic video sequence. The proposed technique holds the potential to be used in efficient IoT-driven video surveillance systems to process video frames efficiently without compromising quality.*

Index Terms — Intra-Frame coding, Temporal Masking, Adaptive Motion Compensation, Video Surveillance, Imaging Technology.

I. INTRODUCTION

Uncompressed video data from cameras isn't suitable for transmission due to its large size [1]. Available channel bandwidth often struggles to sustain the required bitrate for live video streaming, notably in applications needing multilayered streams like surveillance or critical location monitoring. For about a decade, real-time applications for ticketing traffic violations have emerged. It's essential to penalize offenders to encourage caution and prevent accidents [2]. Numerous cameras are strategically placed in various locations prone to violations. Cameras generate tons of data that could surpass storage limits without compression. Wireless channels often lack the capacity for low Compression Ratios (CR) needed. When CR goes up, video quality tends to drop [3]. The Joint Picture Expert Group (JPEG) has a guideline, a quantization table, to assess video quality. Higher CR leads to more distortion like ringing or blurriness, indicating over-compression [4]. For seamless sharing, transmitted information should align with the receiving end's characteristics [5]. An escalation in Compression Ratio (CR) leads to a rise in distortion artifacts like ringing, blockiness, and blurriness within the video content, as highlighted in studies [6]. The aforementioned video artifacts indicate that the video has been excessively compressed, which leads to the degradation of the quality of the video [7]. To facilitate seamless transmission of information, it is important that the characteristics of the data being sent should match the computational capabilities of the receiving system [8]. The compatibility between the transmitted data and the receiving system is crucial for minimizing transmission errors. As explained in the study [9], in video-based applications, a new dimension called time or temporal dimension is added on top of the usual rows and columns. While still images are mostly handled in the spatial domain and don't have any extraneous information, a video series is made up of these still images and has a lot of extraneous information in the time domain across consecutive frames [10]. This natural repetition in the series of frames makes it possible to compress only the temporal activity within these frames, an idea that has been looked into in detail in a number of studies [11-12]. This use of temporal redundancy is the reason for tailored compression methods in video data, which focus on the changes that happen over time in the video stream. Video coding employs two primary compression algorithms: lossy and lossless compression [13]. Lossy compression, responsible for reducing video size, uses Discrete Cosine Transform (DCT) to switch images from spatial to frequency domains, and quantization to eliminate less crucial high-frequency components for frame reconstruction [14-16]. Estimating Motion Vectors (MV) between consecutive frames and capturing their temporal activity is crucial. MV, combined with

A. A. Siddique is with the Department of Telecommunication Engineering, Sir Syed University of Engineering, Karachi, Pakistan (e-mail: asiddiqui@ssuet.edu.pk)

W. Ghaban Applied is with the College, University of Tabuk, Tabuk, 47512, Saudi Arabia (e-mail: Wghaban@ut.edu.sa)

A. Aljaedi is with the College of Computing and Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia (e-mail: aaljaedi@ut.edu.sa)

F. Saeed is with the DAAI Research Group, College of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK (e-mail: faisal.saeed@bcu.ac.uk)

M. S. Alshehri is with the Departments of Computer Science, College of Computer Science and Information Systems, Najran University, Najran 61441 Saudi Arabia (e-mail: msalshehry@nu.edu.sa)

A. Alkhayyat is with the Islamic University, 54001 Najaf, Iraq (e-mail: ahmedalkhayyat85@iunajaf.edu.iq)

Hussain Mobarak Albarakati is with the Computer Engineering Department, College of Computer and Information Systems, Umm Al-Qura University, Makkah 24382, Saudi Arabia (e-mail: hmbarakati@uqu.edu.sa)

residual frame data (difference image), aids in predicting or reconstructing frames during decoding through Motion Compensated Prediction Techniques [17-18].

This paper presents a novel algorithm that extracts motion vectors from consecutive video frames. These vectors capture various motion activities occurring within the frames that are imperceptible to the naked eye in real-time applications. Exploiting this insight, the proposed algorithm masks a negligible amount of temporal activity that contributes minimum to the overall video stream. This strategic application of Motion Masked Compensated Prediction (MMCP) employs a parameter denoted as '*q*' to achieve this masking effect, culminating in a substantial 93.17% Structural Similarity Index (SSIM). Through the utilization of this novel approach, the algorithm adeptly distinguishes and isolates less crucial temporal activity, enhancing the efficiency of video processing and quality assessment in real-time scenarios.

The role of the Internet of Things (IoT) in this particular field is vital in enabling communication between edge devices across the network and also maintaining connectivity. The integration of IoT-based visual surveillance with imaging technology helps in facilitating consumer-focused gadgets that include smartphones, cameras, and other similar imaging devices capable of capturing images and videos in real-time. Fig 1 portrays the emerging technologies within the Consumer Electronics domain. It also signifies that the proposed research relies on an IoT-driven application, aligning directly with the landscape of Consumer Electronics. The proposed technique is important for the advancement of imaging technologies in consumer electronics. It contributes to the current state of the art by tackling the difficulties associated with motion compensation using a novel technique driven by the Internet of Things (IoT).

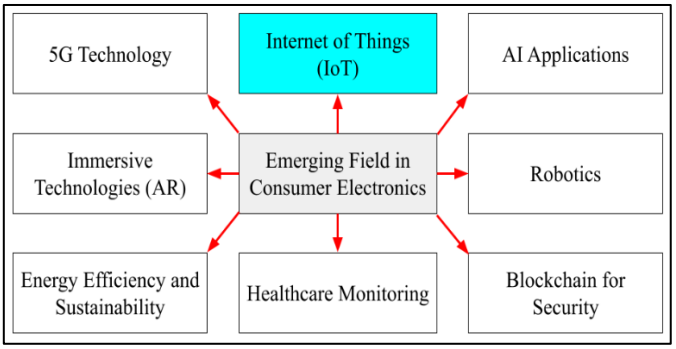

Fig. 1. Emerging Technologies in the Domain of Consumer Electronics

II. Related Work

The process of feature propagation in motion coding is enhanced by repeated encounters, leading to a stronger ability to use long-range temporal correlation. Authors in [19] proposed using hybrid context creation to effectively use the multi-scale context information and enhance the mobility condition. Most hypothesis modules generate different movements and distorted features to extract enough temporal information, enabling diverse inference possibilities from the reference frame. In order to make better use of these hypotheses, authors in [20] included the hypotheses attention module by incorporating the channel-wise squeeze-and-excitation layer and the multi-scale network. Context combination merges weighted hypotheses to create powerful contexts with strong temporal priors. Weighted warped characteristics are combined in the right circumstances to improve compression efficiency [21]. Multi-modes like ConvLSTM-based feature domain prediction, optical flow-conditioned feature domain prediction, and feature propagation can handle static scenes without visible movements to dynamic situations with a moving camera as proposed in [22]. For temporal prediction in spatial block-based representations, authors block the feature space and incorporate dense and sparse post-quantization residual blocks for entropy coding and optional run-length coding on sparse residuals to increase compression. The authors present a unique unsupervised video semantic compression issue that compresses semantics downstream task-agnostically in [23]. They propose a Semantic Mining-then-Compensation (SMC) framework to add semantic coding to the plain video codec to solve this issue. Inspired by current masked image modeling (MIM) approaches, they improve the framework with just unlabeled video data by masking off a part of the compressed video and recreating the masked portions of the original video.

Authors in [24] proposed a reliable underwater image compression method, where an underwater image extreme bit rate compression begins with an autoencoder. After that, a multistep training technique is suggested to progressively learn channel deterioration aspects to strengthen the decoder. The main channel compresses with a low bit rate and great resilience, while the branching path compensates for picture block retransmission using the feedback signal. Experimental findings show that reconstructed images can be identified with a compression ratio of up to 1/768 and an average bit error rate of up to 10^{-1}. It is possible that traditional codecs may not always maintain characteristics that are essential to machine learning algorithms when bandwidth is constrained, which might result in performance that is possibly inferior. An application-driven improvement of programmable commercial codec settings was investigated by the author in [25] for network learning tasks such as image classification. Because they can extract relevant information from vast volumes of complex data, deep learning and AI are ideal for real-time On-board image applications. Authors in [26] present a lossy image reduction approach using a Convolutional Autoencoder (CAE). It can be done on the satellite and can save, reduce, and rebuild camera images. Authors in [27] conduct a comparative analysis of conventional and contemporary lossy image compression methods using the Kodak Dataset. The approaches encompass Autoencoders, Principal Component Analysis (PCA), K-Means, and Discrete Wavelet Transform (DWT).

III. IoT-Enabled Visual Surveillance Framework

Video coding methods play a pivotal role in maximizing the

transmission, storage, and computational resources in surveillance applications inside IoT-driven frameworks. Utilization of well-known video coding standards such as H.264/AVC, H.265/HEVC, or even more recent ones like AV1, in IoT environments tends to compress video data without compromising crucial information. Since IoT devices are not able to cater to the information generated by video sources, processing such as compression becomes vital in such a scenario. The compression process leverages predictive coding intra- and inter-frame prediction, transform coding, and entropy coding to minimize the redundant information present within the consecutive frames of the same scene, this process is performed across the entire video sequence. Motion estimation and compression play a key role in the process of video compression within the visual surveillance domain. Block matching and pixel-based matching approaches are utilized to predict motion in the form of motion vectors between consecutive frames which in turn aids in achieving better Compression Ratio (CR). Predicting and transmitting only the changes (extracted motion vectors) between the frames instead of the entire frame, the bandwidth utilized is reduced, making it feasible for IoT devices with limited or poor connectivity. Fig 2 represents the scenario of integrating a video coding scheme in an IoT-enabled surveillance application.

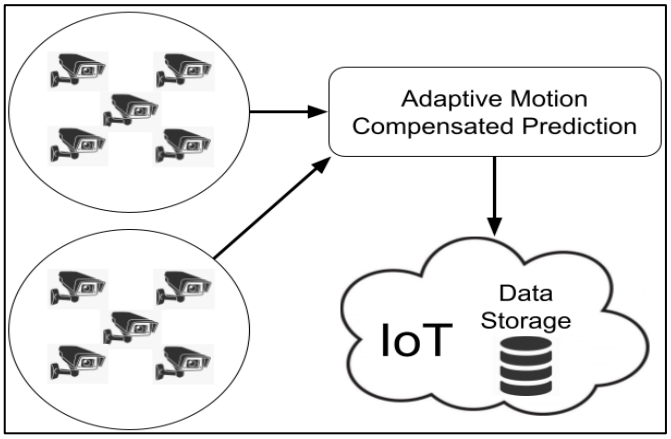

Fig. 2. IoT-Enabled Surveillance Application Integrated with Video Coding in Imaging Technology

IV. Visual Database Correlation and Utilization

Table. 1 outlines the key characteristics of five separate video sequences used. Every sequence is distinguished by a constant resolution of 352×288 pixels, guaranteeing consistency across the visuals. Differences in frames per second (FPS) are present among the sequences, with the 'Sky' and 'Traffic' sequences running at 25 FPS, while 'Foreman,' 'Street,' and 'sampleQCIF' retain a higher speed of 30 FPS. Similarly, the length of these sequences differs ranging from 21.55 MB for 'sampleQCIF' to 105.36 MB for the 'Sky' sequence. The videos strictly conform to the YUV format, indicating their color encoding method. This detailed table clearly illustrates the subtle variations in frame rates, sizes, resolutions, and formats of the essential video sequences that are crucial to the algorithm. The significant variations in file sizes indicate the varying intricacy and information density included within these sequences. The dataset utilized in the proposed work is taken from [28].

TABLE I
Video Dataset Used And Their Charismatics

Video Sequence	Resolution	FPS	Size	Format
Sky	352×288	25	105.36 MB	YUV
Traffic	352×288	25	41.21 MB	YUV
Foreman	352×288	30	43.5 MB	YUV
Street	352×288	30	39.68 MB	YUV
sampleQCIF	352×288	30	21.55 MB	YUV

In an extensive extraction process, a total of 100 frames from each of the 5 video sequences has been retrieved and visualized forming a comprehensive display, a few samples are given in Fig 3. Equation (1) represents the frame extraction process from the YUV video sequences. The variable *i* represents the frame number that is being targeted for extraction. The '*stream*' variable represents all frames and their associated metadata. The variable '*l*' defines the byte size of each frame in the video stream. *w* and *h* represent the width and height of the frame. Equations (2-4) represent the individual *Y*, *U*, and *V* components that make up an entire frame when combined. *Y* component provides information regarding the brightness of a frame depicted as a grayscale image. *U* and *V* components carry color information and represent the color difference between luminance and the actual color itself. The *Y* component is usually subsampled in comparison to these components since human perception is more sensitive to variations in brightness (luminance) than in color. The *U* and *V* components, even at a reduced resolution as shown in (3-4), nonetheless enable precise color reproduction when paired with the *Y* component.

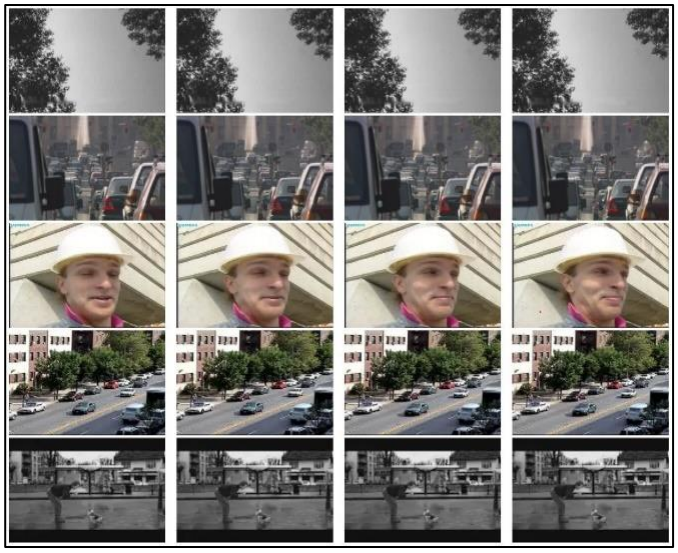

Fig. 3. Row 1: Sky Sequence, Row 2: Traffic Sequence, Row 3: Foreman Sequence, Row 4: Street Sequence, Row 5: sample QCIF Sequence

$$Frame[i] = stream[(i-1) \times l + 1 : i \times l] \quad (1)$$

$$Y_{Component} = reshape(Frame[i][1 : w \times h], w, h)^T \quad (2)$$

$$U_{Comppnent} = resahpe\left(Frame[i][w \times h + 1{:}1.25 \times w \times h], \frac{w}{2}, \frac{h}{2}\right)^{T} \quad (3)$$

$$V_{Component} = reshape\left(Frame[i][1.25 \times w \times h + 1{:}1.5 \times w \times h], \frac{w}{2}, \frac{h}{2}\right) \quad (4)$$

Each video sequence comprises individual frames that, when played at a predetermined pace, constitute a continuous video sequence. The correlation coefficient index for just the first 50 frames of each sequence is shown in Fig 4. Equation (5) is used to find the correlation coefficient index (CCI) for 50 frames of each sequence. CCI_i denoted the correlation coefficient index of consecutive frames in the same video sequence. $cov(img[i], img[i+1])$ is the covariance between the current frame and the incoming frame given in (6), in this equation *i* is the current frame and *j* is the incoming frame. *n* represents the number of pixels in each frame, i_k and j_k are the individual pixel values at the corresponding positions in frames *i* and *j*, respectively. i' and j' denotes the mean of frames *i* and *j*.

$$CCI_i = \frac{cov(img[i], img[i+1])}{\sigma_{img[i]} \cdot \sigma_{img[i+1]}} \quad (5)$$

$$cov_{(i,j)} = \frac{1}{n-1}\sum_{k=1}^{n}(i_k - i') \cdot (j_k - j') \quad (6)$$

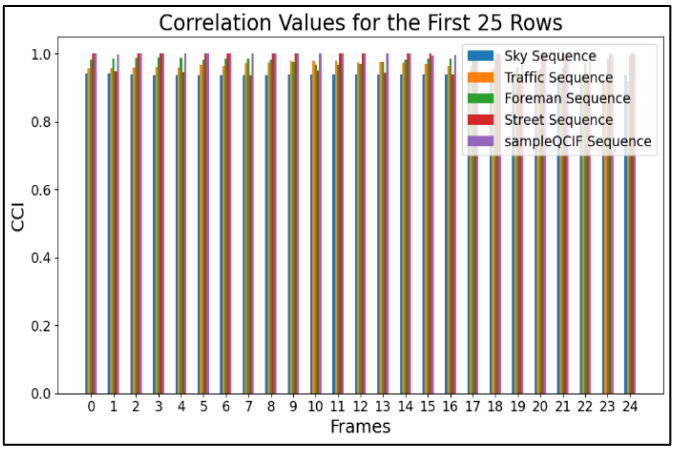

Fig. 4. Correlation Coefficient Index of 5 Video Sequences

V. Temporal Masking For Adaptive Motion Compensation

In the process of video coding, one important step is acquiring motion information from a series of frames that make up a video clip. Complex methods, like block matching algorithms, are often used in this process to find patterns of motion between frames that are next to each other.

A. Block Matching for Dynamic Motion Extraction

Bi-directional motion estimation via block matching algorithms entails the prediction of motion in both the forward and backward directions for every block in successive frames. The method of forward motion estimation with a block-matching algorithm entails determining the motion vector (MV_f) that signifies the displacement between a block in the reference frame I_r and the most suitable matching block in the following frame I_n represented in (7). Here, B_r denotes an 8×8 block in I_r and $B_n(x, y)$ denotes block in I_n at coordinate (x, y). MV_f is determined by finding the block $B_n(x', y')$ in I_n depicting the minimum difference. (x', y') demonstrates the displacement or motion vector. Similarly, for each block in I_r, find the best matching block in the previous frame I_p using the block matching algorithm. This process identifies the motion vector MV_b that represents the backward displacement. The search area for the proposed algorithm is 16 pixels. The miniature red arrows depicted in Fig 5 illustrate the directional flow of the macroblocks movement between the successive frames.

$$MV_{f,b} = argmin_{(x',y')} \sum_{(x,y)} \|B_r(x, y) - B_n(x + x', y + y')\|^2 \quad (7)$$

B. Proposed Adaptive Temporal Masking Strategy

Given that most of the information between consecutive frames is redundant. To optimize use, it is best to code just the changes within the frame, rather than the entire frame itself. In the context of consecutive frames within a video sequence, there is minimal temporal activity or motion between two adjacent frames. The human eye is unable to detect this degree of motion, particularly when the video is played in real-time. Temporal masking exploits the phenomenon of reduced motion activity between successive frames and conceals the activity that is imperceptible to the human eye. This method aims to preserve the integrity of crucial temporal activity while deliberately concealing low-motion activity. Utilizing statistical motion analysis within the frame given in (8), the standard deviation (σ) of motion vector magnitude is computed given in (9). N represents the total number of motion vectors while dx_i and dy_i are the motion vectors along *x* and *y* axis respectively.

$$MV_{(x,y)} = \sum_{i=1}^{N}(dx_i, dy_i) \quad (8)$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(dx_i^2 + dy_i^2)} \quad (9)$$

Temporal masking uses adaptive thresholding to determine a threshold depending on temporal data. This approach adjusts the threshold for video sequence motion intensity. In adaptive thresholding, a statistical feature like standard deviation is used to dynamically alter the threshold to differentiate significant changes from minor variations. This allows exact motion data separation and selective treatment. Equation (10) represents the Adaptive Threshold (q_a) where α is the sensitivity of the thresholding factor α parameter governs the reaction of q_a and its variation for the motion vector magnitudes. An increased value of α results in a more rigorous threshold. Establishing a higher threshold value selectively eliminates less motion activity.

$$q_a = \alpha \times \sigma \tag{10}$$

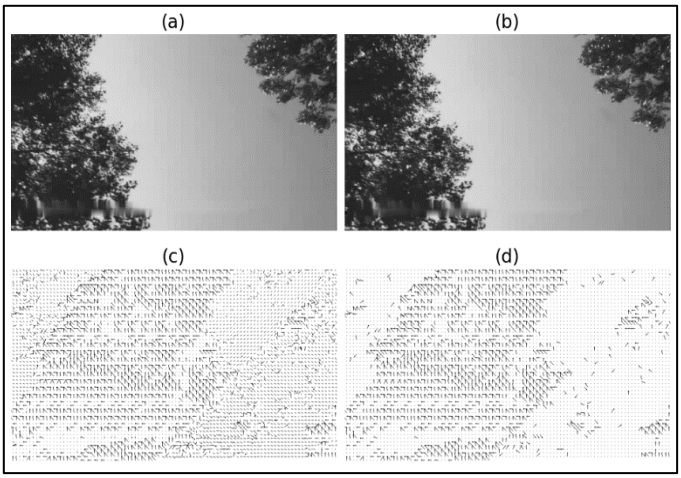

Fig. 5. (a) Current Frame. (b) Next Frame. (c) Extracted MV from the current frame and the next frame. (d) Masked MV(x', y') at $\alpha = 0.5$

Equation (11) depicts the filter that integrates q_a to mask the less motion activity. $MV_{(x',y')}$ represents the Masked Motion Vectors, this condition assesses each motion vector (dx_i, dy_i) and adjusts it to (0, 0) if both components are within the adaptive threshold range. Otherwise, it keeps the original motion vector. Fig 5 depicts the order of frames, including the present frame Fig 5(a), and the next frame Fig 5(b). The MV extracted between these frames is shown in Fig 5(c), illustrating the directional information of pixel displacements. Fig 5(d) displays the masked motion vectors $MV_{(x',y')}$, which shows the motion information that has been filtered or adjusted using the proposed masking technique.

$$MV_{(x',y')} = \begin{cases} 0 & if -q_a < dx_i < q_a \\ 0 & if -q_a < dy_i < q_a \\ (dx_i, dy_i) & Otherwise \end{cases} \tag{11}$$

C. Adaptive Motion Compensated Prediction

The process starts by obtaining extracted frames in real time depicted in Fig 6. The initial frame, or the first frame of the sequence, will remain unchanged. This frame will act as a reference frame $\bar{f}(x, y, t)$ for the subsequent incoming frame $f(x, y, t)$. Residual frame $e(x, y, t)$ is obtained by taking the difference between the reference frame and the targeted frame. Spatial and temporal redundancy aids the compression process by encoding only the motion features. At the same time, motion is extracted by identifying areas of similarity between the two frames. This is the vital information that assists in generating a prediction of a frame at the decoder end. The proposed algorithm masks the temporal activity based on the amount of motion that it possesses by adaptively selecting appropriate q_a parameter. This innovative approach enriches the encoding process by effectively masking regions with minimal motion, optimizing the utilization of available data. This is followed by transform coding, usually using Discrete Cosine Transform (DCT) given in (12-13), on the residual frame and occasionally the prediction. Each frame of an entire video sequence is present in the spatial domain, it is necessary to transform it into the frequency domain which can be done using DCT. The DCT translates spatial data into frequency components, concentrating signal energy in fewer coefficients. Quantization reduces data by approximating or zeroing less essential DCT coefficients given in (14). Lastly, the entropy coding method gives shorter codes to more common patterns to compress the remaining data, resulting in a highly compressed video with maintained quality.

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) Cos\left[\frac{(2x+1)i\pi}{2N}\right] Cos\left[\frac{(2y+1)j\pi}{2M}\right] \tag{12}$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & if\ u = 0 \\ 1 & if\ u > 0 \end{cases} \tag{13}$$

$$Q(i,j) = round\left(\frac{C(i,j)}{Q_i \times Q_j}\right) \tag{14}$$

The essence of video compensation lies in expressing the prediction of a current frame by leveraging the information from the previously encoded frame given in (15). $\hat{f}_t(x, y, t)$ represents the predicted frame at the decoder end that utilizes the encoded residual frame $\bar{e}_t(x, y, t)$ information and the information of the Masked Motion Compensated Prediction $\bar{f}_t(x, y, t)$ to reconstruct as depicted in (16).

$$\bar{f}_t(x,y,t) = \sum_{i=-p}^{p} \sum_{j=-r}^{r} f_{t-1}(x + dx, y + dy, t) \tag{15}$$

$$\hat{f}_t(x,y,t) = \bar{f}_t(x,y,t) + \bar{e}_t(x,y,t) \tag{16}$$

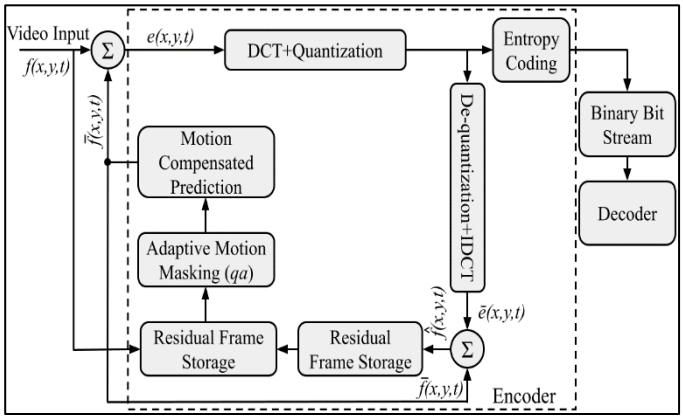

Fig. 6. Motion Compensated Prediction Integrated with Adaptive Temporal Masking [12]

In the final stages of the compression process, entropy coding is applied. The process of entropy coding is important as it transforms the encoded information into a binary bitstream and prepares it for transmission where the decoder decodes this information. In the field of video coding Arithmetic coding is the most widely used entropy coding technique [29]. Usually, Arithmetic coding achieves a better compression ratio as compared to other schemes like Huffman coding [30]. This particular feature is beneficial in applications like compression where reducing the size of the information is crucial without losing the overall quality of the content. It adequately adapts to

the data that needs compression by assigning shorter codes to more frequent symbols in the data which could lead to much better performance. Arithmetic coding can also work with context modeling methods, which cater to the probability of the specific symbol appearing in the data stream. This really helps in video compression applications where most of the information is redundant between the consecutive frames. Even a frame contains a correlation between the pixels and the redundancy is maintained through the compression process. The compression process requires two distinct processes which are motion extraction and masking, and motion compensation. Both of these steps occur at different stages of the video coding process. During the process of encoding, motion extraction, and masking is performed which identifies and isolates the temporal activity between the consecutive frames and applies the proposed adaptive temporal masking procedure. Conversely, the process of motion compensation is performed at the decoder end where the previously encoded frame is reconstructed as accurately as possible.

VI. Results

The assessment of the proposed algorithm's performance is based on three fundamental metrics: Mean Squared Error (MSE) in (17), Peak Signal-to-Noise Ratio (PSNR) in (18), and Structural Similarity Index (SSIM) in (19). MSE measures the average of the squared differences between the original and reconstructed frames, serving as a numerical indicator of the precision of the reconstruction. A large value of PSNR corresponds to high video quality. PSNR is calculated by comparing the maximum potential power of the input signal to the power of the distorting noise. Whereas the Structural Similarity Index (SSIM) assesses how closely the structural details match between the input frames and the reconstructed frames. These two parameters can thoroughly evaluate the performance of the introduced technique to ensure that the reconstructed frames are minimally distorted, precise, and retain the structural information.

$$MSE = \frac{1}{N \times M} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[f(x,y,t) - \bar{f}(x,y,t) \right]^2 \quad (17)$$

$$PSNR = 10 \log_{10} \left(\frac{(Max\ Pixel\ Value)^2}{MSE} \right) \quad (18)$$

$$SSIM(a,b) = \frac{(2\mu_a\mu_b + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)} \quad (19)$$

In order to assess the quality of video compression, it is important to understand the importance of these metrics. A figure can be analyzed by comparing these metrics at different alpha values, i.e., 0.25, 0.5, and 1 with the standard H.264 encoding across various bitrates from 250 to 2000 kbps.

Fig. 7a compares PSNR values for various α levels at different bit rates with the standard H.264 encoding. It demonstrates a steady trend for all α levels: the PSNR values rise in tandem with the bit rate. This occurrence is consistent with predicted behavior as greater bit rates enable the allocation of more data, which enhances the quality of the frame during compression and reconstruction. In contrast to α values of 0.5 and 0.25, α at 1 notably consistently exhibits the greatest PSNR values, demonstrating its efficacy in maintaining image quality during encoding. In this way, compression is performed based on the amount of motion present between the two frames. Fig .7b illustrates the SSIM values for the same α values and bit rates. Similar to the PSNR trend, the SSIM values also exhibit an increasing trend with rising bit rates for all α values. Although a higher amount of α will make q smaller which will have minimum effect on the masking filter which in turn generates a frame that correlates with the standard encoding scheme like H.264. Reiterating the importance of retaining image quality in video compression circumstances, the SSIM measure shows that greater bit rates and higher α values result in better image retention and similarity. Table 2 demonstrates the performance of the standard H.264/AVC encoding scheme.

TABLE II
PSNR Vs SSIM (H.264/AVC Standard)

Video Sequence	PSNR	SSIM
Sky	32.07	0.903
Traffic	36.80	0.933
Foreman	35.01	0.924
Street	35.32	0.937
sampleQCIF	37.83	0.93

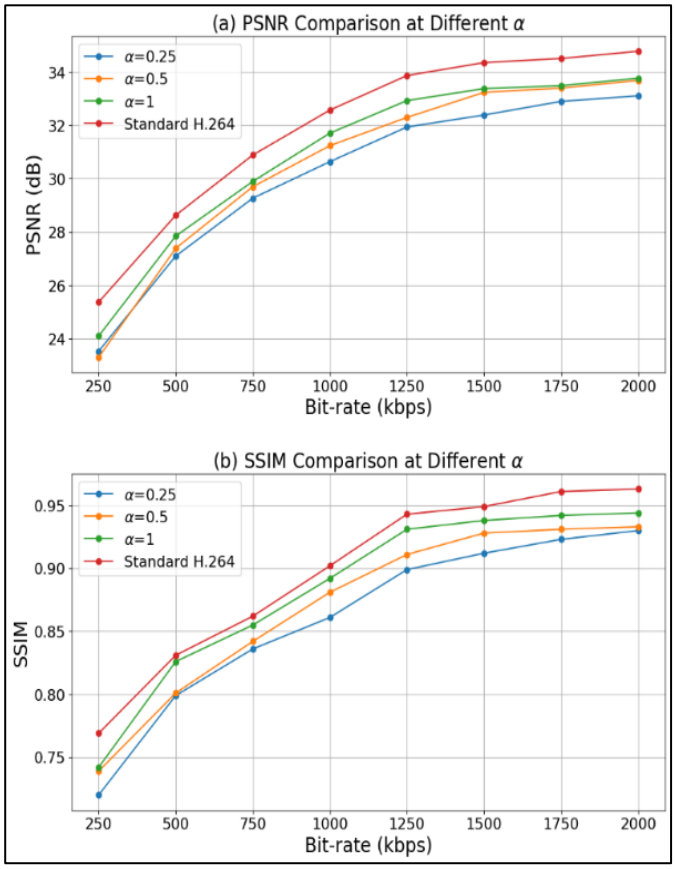

Fig. 7. Sky Sequence (a) PSNR at different α. (b) SSIM at different α

The comparison between the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) for five video sequences at different alpha (α) values is shown in Table 4. In all video sequences, greater alpha values are often correlated with higher PSNR and SSIM values. 'Sky' at alpha of

0.25 displays the lowest PSNR of 30.67 dB and SSIM of 0.858 among the sequences, indicating relatively lower image quality and similarity at lower alpha levels. In contrast, 'sampleQCIF' at α = 1 records the highest PSNR of 37.11 dB and SSIM of 0.921, suggesting better image fidelity and similarity. To achieve the best quality to compression ratio, the range of α is selected to be from 0.4 to 0.6 which can be verified from table 3 as well.

The in-depth analysis of PSNR and SSIM at various alpha values, and bitrate reduction demonstrates a discernible pattern. When alpha is set to 0.25, the video sequences demonstrate decreased PSNR and SSIM values in comparison to higher alpha levels, indicating a decline in image quality and similarity after reconstruction. As an example, the image 'Sky' has a Peak Signal-to-Noise Ratio (PSNR) of 30.67 dB and a Structural Similarity Index (SSIM) of 0.89, indicating reduced accuracy and resemblance. Moreover, for this specific alpha level, the range of bitrate savings varies from 27.83% for the video 'Traffic' to 34.98% for the video 'Street'. The association between decreased alpha values diminished PSNR/SSIM, and bitrate reductions underscore the compromise between image quality and compression effectiveness in video encoding. The Adaptive temporal masking function demonstrates the ability to decrease the amount of information without significantly compromising quality. Increasing the alpha values often results in improved images, as seen by higher PSNR/SSIM scores. However, this comes at the expense of reduced bitrate savings. On the other hand, decreasing the alpha values allows for more compression, but at the sacrifice of image fidelity and similarity. It is necessary to keep checking the compression ratio because if the compression is extensive, a phenomenon called frame-skipping happens in which the information is compressed to an extent that it would seem the reference frame is still for a few seconds and the frame information is lost. The proposed adaptive technique helps cater to this problem by keeping the α to a level that avoids frame skipping.

TABLE III
PSNR VS SSIM AT DIFFERENT α LEVELS

Video Sequence	PSNR (dB) at α			SSIM at α		
	α=1	α=0.5	α=0.25	α=1	α=0.5	α=0.25
Sky	31.99	31.23	30.67	0.89	0.863	0.858
Traffic	36.41	34.40	33.68	0.927	0.907	0.862
Foreman	34.26	32.15	30.40	0.912	0.882	0.851
Street	34.98	31.39	29.45	0.929	0.879	0.843
sampleQCIF	37.11	34.9	31.41	0.921	0.909	0.877

TABLE IV
PSNR VS SSIM AT α = 0.25 AND NUMBER OF BITRATE SAVED

Video Sequence	PSNR (dB) at α = 0.25	SSIM at α = 0.25	Bitrate Saved (%)
Sky	30.67	0.89	33.23
Traffic	33.68	0.927	31.26
Foreman	30.40	0.912	30.68
Street	29.45	0.929	34.98
sampleQCIF	31.41	0.921	27.83

VII. CONCLUSION

This paper presented an adaptive temporal masking technique for video surveillance systems that compresses the video frames while maintaining their quality. The introduced technique utilized an 'alpha' parameter that dynamically adjusts the 'q' parameter to optimize the bitrate for video encoding. The proposed technique reduced the temporal activity in vide frames to a level, which is undetectable by the human eye. This results in significant bit-rate reductions while preserving the video quality. Thie technique enables extended storage of high-resolution videos, which are extremely important for comprehensive post-event analysis of videos in surveillance applications. Moreover, the algorithm can adapt to varying conditions in surveillance, e.g., different levels of motion complexity or limited bandwidth. This property makes it a suitable choice for video surveillance applications in IoT environments. The proposed technique achieved over 30% bit rate reduction compared to the standard video encoding schemes, such as H.264/AVC and preserved the quality of the original videos, which is evident by the results of PSNR of 33.67 dB and SSIM of 92.7%. The proposed adaptive temporal masking approach represented an advancement in maximizing the bandwidth and reducing the computational overhead.

ACKNOWLEDGMENT

This research is supported by Najran University.

REFERENCES

[1] L. Guo, et al. "Dvc: An end-to-end deep video compression framework." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[2] L. Guo, et al. "An end-to-end learning framework for video compression." *IEEE transactions on pattern analysis and machine intelligence* (2020).

[3] R. Yang, et al. "Learning for video compression with hierarchical quality and recurrent enhancement." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[4] E. Agustsson, et al. "Scale-space flow for end-to-end optimized video compression." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[5] M. Siwei, et al. "Image and video compression with neural networks: A review." *IEEE Transactions on Circuits and Systems for Video Technology* 30.6 (2019): 1683-1698.

[6] T. Shanableh, "Feature extraction and machine learning solutions for detecting motion vector data embedding in HEVC videos." *Multimedia Tools and Applications* (2020): 1-20.

[7] H. Chen, et al. "Real-time action feature extraction via fast PCA-Flow." *Concurrency and Computation: Practice and Experience* 33.11 (2021): e5507.

[8] Y. Yuan, et al. "Key frame extraction based on global motion statistics for team-sport videos." *Multimedia Systems* (2021): 1-15.

[9] P. Sykora, et al. "Comparison of Neural Networks with Feature Extraction Methods for Depth Map Classification." *Advances in Military Technology* 15.1 (2020).

[10] X. Xuguang, C. Feng, and S. He, "A method for the micro-motion signal separation and micro-Doppler extraction for the space precession target." *IEEE Access* 8 (2020): 130392-130404.

[11] S. Rana, K. Rohit, and A. Sur. "Motion vector based video steganography using homogeneous block selection." *Multimedia Tools and Applications* 79.9 (2020): 5881-5896.

[12] A. A. Siddique, M. T. Qadr, and Z. Mohy-Ud-Din. "Masking of temporal activity for video quality control, measurement and assessment." *Measurement and Control* 53, no. 9-10 (2020): 1817-1824.

[13] S. Zhang, et al. "A video deblurring algorithm based on motion vector and an encorder-decoder network." *IEEE Access* 7 (2019): 86778-86788.
[14] Z. Wang, and Y. Zhu. "Video Key Frame Monitoring Algorithm and Virtual Reality Display Based on Motion Vector." *IEEE Access* 8 (2020): 159027-159038.
[15] A. A. Siddique, M. T. Qadri, N. A. Siddiqui, and Z. Mohy-ud-Din. "Temporal Masking with Luma Adjusted Interframe Coding for Underwater Exploration Using Acoustic Channel." *Wireless Personal Communications* 116 (2021): 1493-1506.
[16] V. K. Ghassab, R. Gonsalves, S. Mathur, and N. Bouguila. "Optimizing Video Compression With CNN-Based Autoencoders With Chroma Subsampling." *SMPTE Motion Imaging Journal* 132, no. 3 (2023): 18-26.
[17] B. Patel, "Dual autoencoder-based framework for image compression and decompression." In *Fifteenth International Conference on Machine Vision* (ICMV 2022), vol. 12701, pp. 549-557. SPIE, 2023.
[18] F. Galpin, M. Balcilar, F. Lefebvre, F. Racapé, and H. Pierre, "Entropy Coding Improvement for Low-complexity Compressive Auto-encoders." *arXiv preprint arXiv*: 2303.05962 (2023).
[19] W. Hamidouche, F. Pescador, T. Biatek, and E. François, "Editorial Real-Time Implementation of VVC Standard for Consumer Electronic Devices." *IEEE Transactions on Consumer Electronics* 68, no. 2 (2022): 93-95.
[20] Schimpf, Michael G., Nam Lign, and Ying Liu. "Compressing of Medium-to Low-Rate Transform Residuals With Semi-Extreme Sparse Coding as an Alternate Transform in Video Coding." *IEEE Transactions on Consumer Electronics* 69, no. 3 (2023): 271-286.
[21] A. A. Siddique, S. M. U. Talha, M. U. Khan, A. Israr, U. Jilani, and V. Uddin, "Efficient Online Lecture Platform: Design and Implementation of Optimized Temporal Masking Technique for Compressed Video Streaming." *Wireless Personal Communications* (2023): 1-18.
[22] D. E-Jabeen, T. Khan, R. Iftikhar, A. A. Siddique, and S. Asghar. "An Algorithm to Reduce Compression Ratio in Multimedia Applications." *Computers, Materials & Continua* 75, no. 1 (2023).
[23] B. Liu, C. Yu, R. C. Machineni, S. Liu, and K. Hun-Seok, "MMVC: Learned Multi-Mode Video Compression with Block-based Prediction Mode Selection and Density-Adaptive Entropy Coding." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18487-18496. 2023.
[24] J. Liu, F. Yuan, C. Xue, Z. Jia, and E. Cheng, "An Efficient and Robust Underwater Image Compression Scheme Based on Autoencoder." *IEEE Journal of Oceanic Engineering* (2023).
[25] A. Singhadia, M. Mamillapalli, and I. Chakrabarti, "Hardware-efficient 2D-DCT/IDCT architecture for portable HEVC-compliant devices." *IEEE Transactions on Consumer Electronics* 66, no. 3 (2020): 203-212.
[26] G. Guerrisi, F. D. Frate, and G. Schiavon, "Convolutional Autoencoder Algorithm for On-Board Image Compression." In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, pp. 151-154. IEEE, 2022.
[27] A. Thakker, N. Namboodiri, R. Mody, R. Tasgaonkar, and M. Kambli. "Lossy Image Compression-A Comparison Between Wavelet Transform, Principal Component Analysis, K-Means and Autoencoders." In *2022 5th International Conference on Advances in Science and Technology* (ICAST), pp. 569-576. IEEE, 2022.
[28] K. Seshadrinathan, R. Soundararajan, A. C. Bovik and L. K. Cormack, "Study of Subjective and Objective Quality Assessment of Video", IEEE Transactions on Image Processing, vol.19, no.6, pp.1427-1441, June 2010.
[29] Pastuszak, Grzegorz. "Optimization of the Generative Multi-Symbol Architecture of the Binary Arithmetic Coder for UHDTV Video Encoders." *Electronics* 12, no. 22 (2023): 4643.
[30] Zhu, Xia, Jing Zhang, and Hongbo Zhu. "Research of Data Compression Using Huffman Coding and Arithmetic Coding." In *International Conference on Computer Engineering and Networks*, pp. 954-961. Singapore: Springer Nature Singapore, 2022.

Ali Akbar Siddique obtained his bachelor's degree from Sir Syed University in March 2009 and his major was Electronics Engineering. He completed his Masters in Control and Automation from Usman Institute of Technology (UIT) with a 3.81 CGPA and was a Gold Medalist in his respective discipline. He completed his Ph.D. in the field of Electronic Engineering and his specialization is in Signal, Image, and Video processing.

Wad Ghaban is an assistant professor in the applied college at the University of Tabuk, Saudi Arabia. Wad received her BSc in computer science from King Abdul-Aziz University in Jeddah with honors degree. Then, she received her MSc. in advanced computer science with distinction in University of Birmingham by 2015. Later, she got her PhD from University of Birmingham by 2020. During her study, Wad worked on several projects related to Human computer interaction, survival analysis, online learning, Natural language processing and sentiment analysis.

Amer Aljaedi received his Ph.D. degree in security engineering from the Computer Science Department at Colorado University, Colorado Springs, USA, in 2018. He received his M.Sc. degree in information systems security from Concordia University of Edmonton, Canada, in 2011, and the B.Sc. degree from King Saud University, Saudi Arabia, in 2007. He is currently an Associate Professor at the College of Computing and Information Technology, University of Tabuk. Before that, he was a senior research member with the Cybersecurity Laboratory at Colorado University, and he received multiple research awards from UCCS, UT, and SACM for his outstanding research papers.

Faisal Saeed is a Senior Lecturer in the Computing and Data Science Department at the School of Computing and Digital Technology, Birmingham City University (BCU), UK. He is leading the smart health lab at Data Analytics and AI Research Group at BCU. Faisal received his BSc in Computers (Information Technology) from Cairo University, Egypt, MSc in Information Technology Management and PhD in Computer Science from UTM, Malaysia in 2010 and 2013 respectively.

Mohammed S. Alshehri received the B.S. degree in Computer Science from the King Khalid University, Abha, Saudi Arabia, in 2010, the M.S. degree in Computer Science from the University of Colorado Denver, Denver, USA, in 2014, and the Ph.D. degree in Computer Science with concentration on Information Security from the University of Arkansas, Fayetteville, USA, in 2021. Mohammed's areas of interest are Cybersecurity, Computer Networks, Blockchain, Machine Learning, and Deep Learning.

Ahmed Alkhayyat received the B.Sc. degree in electrical engineering from AL KUFA University, Najaf, Iraq, in 2007 and the M.Sc. degree from the Dehradun Institute of Technology, Dehradun, India, in 2010. He contributed in organizing several IEEE conferences, workshop, and special sessions. He is currently a dean of international relationship and manager of the world ranking in the Islamic University, Najaf, Iraq.

Hussain Mobarak Albarakati is with Department of Computer Engineering, College of Computer and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia., He is a Senior Professor of the university where teaching courses related to AI and embedded systems. In addition, he is a senior AI researcher related to remote sensing and medical. He published more than 50 research articles and also a reviewer for several good journals.