Knowledge Engineering and Data Science (KEDS)
Vol 6, No 1, April 2023, pp. 1–14

pISSN 2597-4602
eISSN 2597-4637

Inter-Frame Video Compression based on Adaptive Fuzzy Inference System Compression of Multiple Frame Characteristics

Arief Bramanto Wicaksono Putra [a,1], Rheo Malani [a,2], Bedi Suprapty [a,3,*],
Achmad Fanany Onnilita Gaffar[a,4], Roman Voliansky [b,5]

[a] *The Applied Modern Computing and Robotics System Unit, Politeknik Negeri Samarinda, East Kalimantan, Indonesia*
[b] *Department of Electromechanical Systems Automation and Electric Drives National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, Ukraine*
[1] *ariefbram@gmail.com;* [2] *anaogie@gmail.com;* [3] *bedirheody@gmail.com*;* [4] *onnygaffar212@gmail.com;*
[5] *voliansky@ua.fm*
* *corresponding author*

ARTICLE INFO

Article history:
Received 30 June 2022
Revised 07 March 2023
Accepted 05 April 2023
Published online 30 April 2023

Keywords:
Inter-Frame-based Video Compression
Simple Statistical Features
Features Compression
Adaptive FIS

ABSTRACT

Video compression is used for storage or bandwidth efficiency in clip video information. Video compression involves encoders and decoders. Video compression uses intra-frame, inter-frame, and block-based methods. Video compression compresses nearby frame pairs into one compressed frame using inter-frame compression. This study defines odd and even neighboring frame pairings. Motion estimation, compensation, and frame difference underpin video compression methods. In this study, adaptive FIS (Fuzzy Inference System) compresses and decompresses each odd-even frame pair. First, adaptive FIS trained on all feature pairings of each odd-even frame pair. Video compression-decompression uses the taught adaptive FIS as a codec. The features utilized are "mean", "std (standard deviation)", "mad (mean absolute deviation)", and "mean (std)". This study uses all video frames' average DCT (Discrete Cosine Transform) components as a quality parameter. The adaptive FIS training feature and amount of odd-even frame pairings affect compression ratio variation. The proposed approach achieves CR=25.39% and P=80.13%. "Mean" performs best overall (P=87.15%). "Mean (mad)" has the best compression ratio (CR=24.68%) for storage efficiency. The "std" feature compresses the video without decompression since it has the lowest quality change (Q_dct=10.39%).

I. Introduction

The digital video represents the movement of visual images in the form of encoded digital data. The digital video contains a collection of digital images called frames displayed sequentially. Typically, the videos are stored in an uncompressed format. Video compression is related to the need for storage or bandwidth efficiency for video transmission. Video compression minimizes the number of video data bits through encoding information. The result is fewer video data bits than the original. Lossless compression is usually used for archiving audiovisual files and is usually avoided for use [1]. Lossy compression is a type of compression that eliminates information that is considered not necessary [2]. Lossless compression is a type of compression that removes parts that are considered statistically redundant without losing the importance of information.

There are several approaches to video compression. Inter-frame-based compression [3][4][5] uses at least two frames to compress the current frame. Intra-frame-based compression [6][7][8][9] applies the principles of image compression to each current frame. One commonly used lossless image compression method is Arithmetic Coding (AC). Some of the compression algorithms belonging to this type are Huffman codes [10], the mixture of non-parametric distributions [11], and Integer Wavelet Transform (IWT) [12]. Block-based compression [13][14][15][16] groups video frames into coding blocks to predict, transform, quantize, and encode. Predicting and encoding the first frame of

https://doi.org/10.17977/um018v6i12023p1-14

each block is using the intra-frame-based concept. Both the intra-frame and inter-frame-based compression concepts are then applied to the remaining frames.

Video compression algorithms commonly used in various video compression standards (such as MPEG) typically consist of motion estimation, motion compensation, and Frame Difference [17]. Motion estimation [18][19][20] is finding motion vectors that point to the best macroblock in a frame or field of reference. This process explains the previous and later frames to identify blocks that have not changed and motion vectors are stored instead of blocks. Motion compensation [21][22] describes the transformation of the image from the reference image to the current image. The reference image can be from the previous or later image. Compression efficiency will increase if the image synthesis of the reference image is done accurately. Motion compensation refers to the result of a camera or moving object in a moving frame is the only difference between one frame and another. Based on differences between frames, the frame difference method [23][24][25] mainly focused on the amount of data to be compressed. There are three types of frame difference methods. I-frame is the easiest because it does not require other frames to decode. P-frames tend to use the previous frame as a reference frame to decompress. B-frames use previous frames and subsequently as reference frames to increase data compression ratios.

Image features play an essential role in computer vision and pattern recognition. An image feature is a relevant information for computing needs associated with a particular application. The feature is also considered to represent the uniqueness of an image. Various studies in this field focus on how to build powerful feature extraction methods. Some of them are the Median Robust Extended Local Binary Pattern [26], Local Weighting Pattern [27], dan Gray-Level Dynamic Range Modification Technique [28]. Image features are also widely used in developing various feature-based image compressions algorithms [29][30][31][32].

From the description above, there are several approaches to video compression. One inter-frame-based video compression approach uses at least two frames to compress the frames into a single frame. Various video compression algorithms (besides motion estimation, motion compensation, and frame difference) are a separate area of research. It also applies to the inter-frame-based video compression approach. This study applies an inter-frame-based approach to compressing video by utilizing image features obtained in a certain way. Each pair of adjacent frames (called an odd-even frame pair) compressed into one compressed frame. Compression and video decompression are based on the compressed feature of every odd-even frame pair generated by adaptive FIS. Firstly, adaptive FIS is trained using the features of all odd-even frame couples. The trained adaptive FIS is then used as a codec (encoder-decoder) in the compression-decompression process of the video. The features used are simple statistical features of "mean", "std (standard deviation)", "mad (mean absolute deviation)","mean (std) ", and "mean (mad)". This study assumes that the average DCT component of all video frames is a video quality parameter.

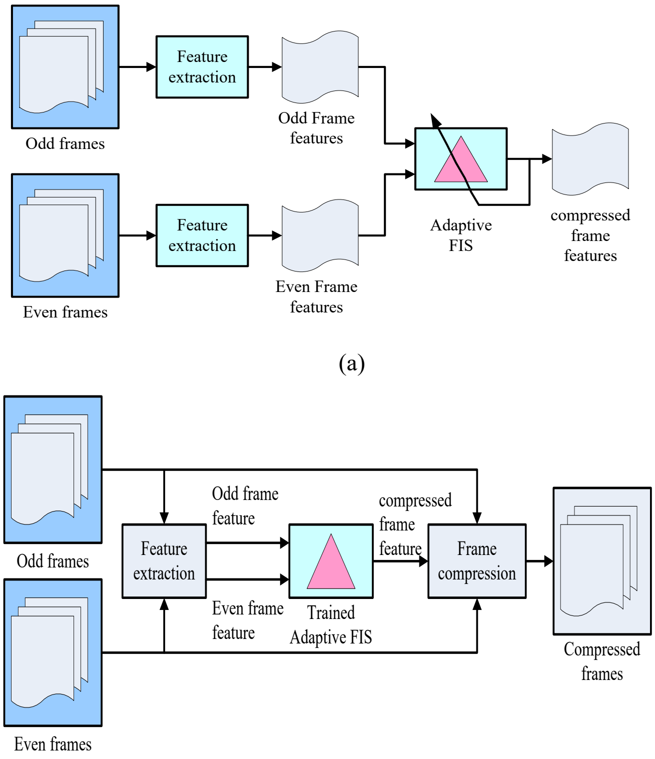
II. Methods

This section presents details of the adaptive FIS training phase to generate compressed features and their use in inter-frame-based video compression. The performance of compression-decompression results is measured by several parameters selected.

A. General Concept

Various video compression algorithms are generally based on motion estimation, compensation, and frame differences. Motion estimation and compensation are commonly used simultaneously. The motion estimation produces a motion vector. This motion vector is used to compensate each frame in a certain way to produce a compressed frame. Video compression uses the difference frame approach by determining the threshold value obtained from calculating frame differences from each pair of adjacent frames. Each pair's first or second frame will be eliminated if the difference between frames is equal to or less than the threshold value. The remaining frame is considered as the result of video compression after being reconstructed in a certain way.

This study proposes using an inter-frame-based approach for video compression, combining each adjacent frame into one compressed one through its compression feature. In general, the proposed method is shown in Figure 1.



(a)

(b)

Fig. 1. The proposed method (a). training stage (b). compression stage

Each pair of adjacent frames is assumed as an odd-even frame pair. One video file has two sets of frames, odd and even. The proposed method consists of two stages: training and compression. The training stage is used to build rule-based adaptively. Input training is a set of feature pairs of all odd-even frame pairs. The training target is a set of the average feature pairs considered compressed features. In the compression phase, the trained adaptive FIS applied to each odd-even frame pair to produce the compressed feature. This compressed feature is then used to compress the odd-even frame pair into a single compressed frame in a certain way.

B. Training Stage

A video file consists of multiple frames. Each frame is presented as an RGB image. Suppose a video file contains a number of L frames where each frame is an RGB image size of M×N, then mathematically, each frame is declared as in (1).

$$F_n = I_n(i, j, d) \tag{1}$$

where $n = 1 \dots L$, $i = 1 \dots M$, $j = 1 \dots N$, and d is the index of the RGB component ($R = 1, G = 2, B = 3$). Each adjacent frame assumed an odd-even frame pair. There will be two sets of frames, odd (F_{odd}) and even (F_{even}). The odd and even frame pair is denoted by F_{2k-1} and F_{2k}, respectively, where $k = 1 \dots\ L/2$.

1) Feature Extraction

Feature extraction was performed on each component R, G, and B. Each frame will have a combination of the three component features. This study uses some simple statistical measures as a feature: mean, std (standard deviation), mad (mean absolute deviation), mean of std, and mean of mad expressed as in (2) to (6).

$$f_{mean}(d) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} I(i,j,d) \tag{2}$$

$$f_{mean} = [f_{mean}(1) \quad f_{mean}(2) \quad f_{mean}(3)]$$

$$f_{std}(1,j,d) = \left(\frac{1}{M-1} \sum_{i=1}^{M} \left(I(i,j,d) - \frac{1}{M} \sum_{i=1}^{M} I(i,j,d) \right)^2 \right)^{1/2}$$

$$f_{std}(d) = \left(\frac{1}{N-1} \sum_{j=1}^{N} \left(f_{std}(1,j,d) - \frac{1}{N} \sum_{j=1}^{N} f_{std}(1,j,d) \right)^2 \right)^{1/2} \tag{3}$$

$$f_{std} = [f_{std}(1) \quad f_{std}(2) \quad f_{std}(3)]$$

$$f_{mad}(1,j,d) = \frac{1}{M} \sum_{i=1}^{M} \left| I(i,j,d) - \frac{1}{M} \sum_{i=1}^{M} I(i,j,d) \right|$$

$$f_{mad}(d) = \frac{1}{N} \sum_{j=1}^{N} \left| f_{mad}(1,j,d) - \frac{1}{N} \sum_{j=1}^{N} f_{mad}(1,j,d) \right| \tag{4}$$

$$f_{mad} = [f_{mad}(1) \quad f_{mad}(2) \quad f_{mad}(3)]$$

$$f_{mean(std)}(d) = \frac{1}{N} \sum_{j=1}^{N} f_{std}(1,j,d) \tag{5}$$

$$f_{mean(std)} = [f_{mean(std)}(1) \quad f_{mean(std)}(2) \quad f_{mean(std)}(3)]$$

$$f_{mean(mad)}(d) = \frac{1}{N} \sum_{j=1}^{N} f_{mad}(1,j,d) \tag{6}$$

$$f_{mean(mad)} = [f_{mean(mad)}(1) \quad f_{mean(mad)}(2) \quad f_{mean(mad)}(3)]$$

2) *FIS Concept*

FIS is a Fuzzy Logic based inference system where the reasoning process adopts human reasoning abilities. The inference system consists of fuzzification, implication, aggregation, and defuzzification. Fuzzification is the stage that maps each FIS input into a fuzzy input number by using a fuzzy set constructed from a particular membership function. The implication is the stage that maps each pair of fuzzy input numbers into a fuzzy output set by applying each rule-based. If there are N rules, there will be N output fuzzy set of the implication result for each input fuzzy number pair. Aggregation is a phase that aggregates all implication results into a set of fuzzy outputs. Defuzzification converts the fuzzy set of outputs into one crisp number.

A fuzzy set represents the linguistic values in the universe of discourse by using fuzzy MF (Membership Function). There are several types of fuzzy MF. One of the most commonly used is Triangular MF. A fuzzy set constructed by Triangular MF with three linguistic values (Low, Medium, High) shown in Figure 2.

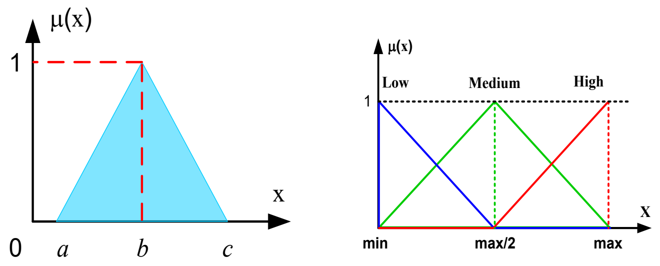


Fig. 2. A fuzzy set by triangular MF

Mathematically, the Triangular MF is expressed as in (7).

$$\mu(x)=\begin{cases}0 & if\ x\le a;x\ge c\\(x-a)/(b-a) & if\ a\le x\le b\\(c-x)/(c-b) & if\ b\le x\le c\end{cases}\tag{7}$$

There are various inference methods. One of the most commonly used is Mamdani, as shown in Figure 3. A fuzzy operator is a logical operation used to obtain a single truth value as the output of each rule based on two or more fuzzy input numbers. The Mamdani method usually uses the AND fuzzy operator (min). The implication method generates a fuzzy output set for each rule based on a single truth value. The Mamdani method typically uses an AND (min) implication method that trims the fuzzy output set based on a single truth value. The process of applying one rule using the Mamdani method is illustrated in Figure 4.

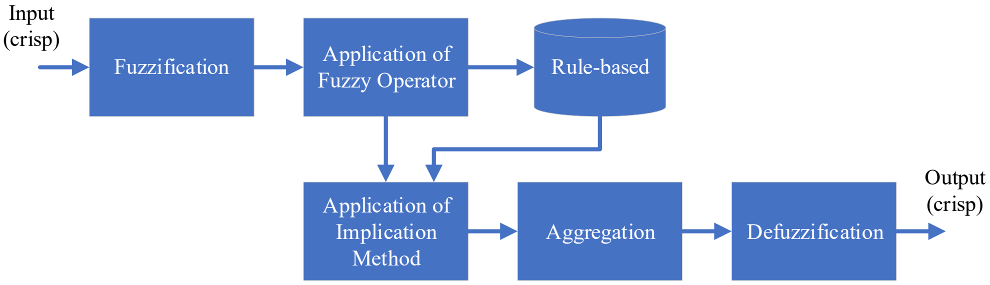


Fig. 3. The mamdani's fuzzy inference system

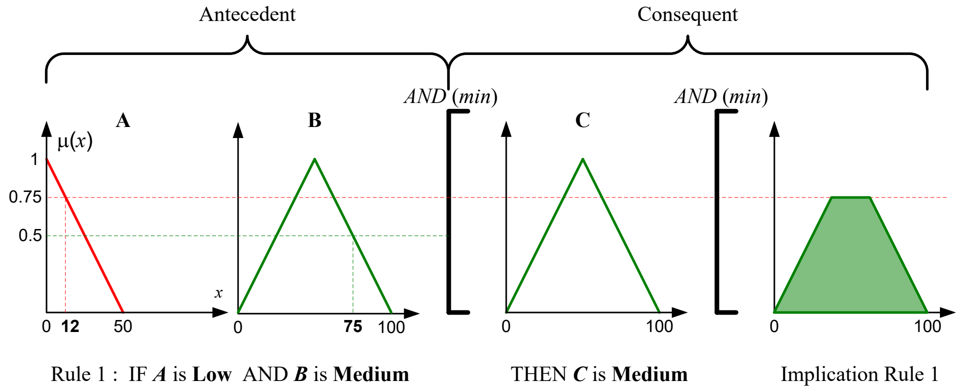


Fig. 4. The illustration of the rule application using mamdani method

The inference process is performed by implementing the rule-based for each pair of fuzzy input numbers. Rule-based is a collection of rules that are made to make decisions. A particular method aggregates all the output fuzzy sets of the implication result. The Mamdani method usually uses OR (max) aggregation method. The aggregated output fuzzy set is then defuzzified using a particular method to obtain a single crisp number. The commonly used method is COA (Center Of Area), mathematically expressed as in (8).

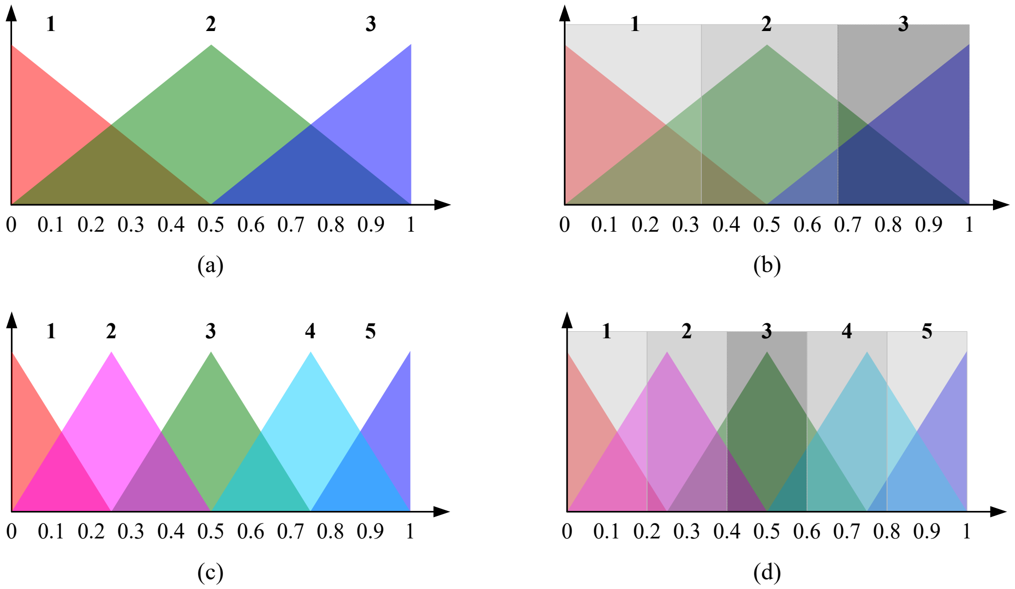
$$x_c = \left(\sum_{i=1}^{n} x_i . \mu(x_i)\right) / \sum_{i=1}^{n} \mu(x_i) \tag{8}$$

where x_c, x_i and $\mu(x_i)$ are the centroid value, the *i*th *x* value, and degree of membership of x_i, respectively.

3) Adaptive FIS

If c is the number of crisp input, l is the number of linguistic value, the maximum number of the rule is r=(l)^c. The accuracy of the result depends on the created fuzzy set and the number of the rules in rule-based. Adaptive FIS is used primarily to obtain the accuracy of the results with the specified target error. The term "adaptive" refers to the linguistic value adaptation, the parameters of the fuzzy MF and others during the FIS training process.

In this study, the adaptive FIS is used to renew the number of the linguistic value. The adaptation results are used to build new rule-based based on a crisp input data set. The adaptation is performed by partitioning the universe of discourse starting from the smallest amount (3 linguistic values). Each partition is labeled sequentially. Each crisp input is labeled according to the partition label number of its position. This process is called labeling. Suppose a fuzzy set with three linguistic values, as shown in Figure 5 (a). The universe of discourse is partitioned into three parts with the labeling results as shown in Figure 5 (b). If necessary, the partition process continues, as shown in Figure 5 (c) and Figure 5 (d).



(a)

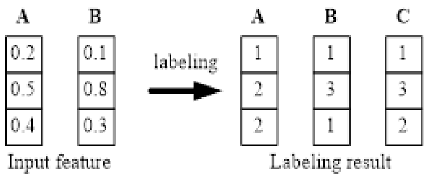
(b)

(c)

(d)

Fig. 5. The universe of discourse partition process (a). A fuzzy set with three linguistic values (b). The three partitions results of its universe of discourse (c). A fuzzy set with five linguistic values (d). The five partitions results of its universe of discourse

From Figure 5 (b), we can obtain the result of partitioning the universe of discourse by three linguistic values are: label 1:{0 ... 0.33}, label 2:{0.34 ... 0.66}, label 3:{0.67 ... 1}. Suppose there is a pair of crisp input features A and B. The labeling result by using three linguistic values is shown in Figure 6 (a) where $(C) = round\left(\left(label(A) + label(B)\right)/2\right)$. The Rule-based generated from labeling results is shown in Figure 6 (b).



(a)

IF *A* is 1 AND *B* is 1 THEN *C* is 1

IF *A* is 2 AND *B* is 3 THEN *C* is 3

IF *A* is 2 AND *B* is 1 THEN *C* is 2

(b)

Fig. 6. The rule-based generation (a). An example of a labeling process (b). The result

The percent of decompression errors determines the termination of the adaptive FIS training process to the original. The training process continues if $|F_{decomp.} - F_{original}| > target\ error$, and vice versa.

If $f(A)$ and $f(B)$ are each set of odd $\{F_1, F_3, \dots F_{L-1}\}$ and even $\{F_2, F_4, \dots F_L\}$ frame features, the features compression by using FIS mathematically expressed as in (9).

$$Fuzz\big(f(A), f(B)\big) = f(C) \tag{9}$$

The $f(C)$ is the compressed feature set. The features decompression is mathematically expressed as in (10).

$$Fuzz\big(f(C)\big) = [f(A') \quad f(B')] \tag{10}$$

$f(A')$ and $f(B')$ are the set of the decompressed odd and even frame features, respectively.

Training error of adaptive FIS stated by using MAPE (Mean Absolute Percentage Error) which is mathematically expressed as in (11).

$$\begin{gathered} MAPE(i) = \left(\left(\frac{|f(A_i) - f(A_i')|}{f(A_i)} + \frac{|f(B_i) - f(B_i')|}{f(B_i)}\right)/2\right) * 100 \\ MAPE = \frac{2}{L}\sum_{i=1}^{L/2} APE(i) \end{gathered} \tag{11}$$

In general, the adaptive FIS training process is shown in Figure 7.

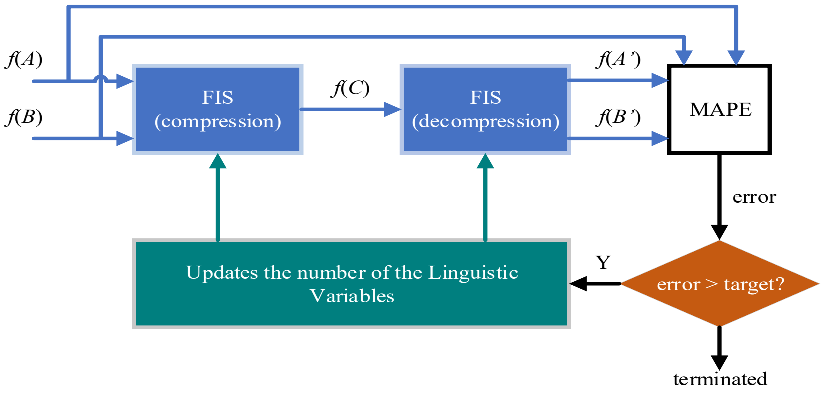


Fig. 7. Adaptive FIS training process

C. Compression and Decompression Stage

The trained adaptive FIS consists of two parts: FIS compression and decompression, as stated by Eq. (7) and (8). The trained adaptive FIS, which contains trained rule-based, is used for video compression and decompression. The compressed frame is generated as in (12).

$$F_{comp(k)}(d) = \left(\frac{F_{2k-1}(d)}{f\big(A_k(d)\big)} + \frac{F_{2k}(d)}{f\big(B_k(d)\big)}\right)/f\big(C_k(d)\big) \tag{12}$$

Where $k = 1 \ldots L/2, L$ is the number of frame, and d is the index of RGB component. Finally, the compressed video file is obtained from the reconstruction process of all compressed frames.

Referring to Eq. (10), the decompression of the compressed video by using FIS mathematically expressed as in (13).

$$Fuzz\big(f(C')\big) = [f(A') \quad f(B')] \tag{13}$$

$f(A')$ and $f(B')$ are the set of the decompressed odd and even frame features from compressed video, respectively. The set of compressed feature $f(C')$ obtained from the set of the compressed frame $\{F_{comp(1)} \ldots F_{comp(L/2)}\}$, mathematically expressed as in (14)

$$f\big(C'_k(d)\big) = \frac{2}{\frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N} F_{comp(k)}(i,j,d)} \tag{14}$$

The decompressed frame generated as in (15)

$$F'_{2k-1}(d) = F_{comp(k)}(d) * f\big(C'_k(d)\big) * f\big(A'_k(d)\big)/2$$
$$F'_{2k}(d) = F_{comp(k)}(d) * f\big(C'_k(d)\big) * f\big(B'_k(d)\big)/2 \tag{15}$$

All the decompressed frames are reconstructed into a decompressed video file.

D. Performance Parameter

There are a variety of measurable parameter standards for assessing video compression performance. This study uses some of the following performance parameters.

1) Compression Ratio (CR)

CR states the degree of compression achieved. Commonly, a reasonably small CR is expected to increase the efficiency of storage and video data transmission requirements. If $size\ of\ Vid_{(ori)}$ is the original video file and $size\ of\ Vid_{(comp)}$ is the compressed video file, CR is expressed as in (16).

$$CR = \left(\frac{size\ of\ Vid_{(comp)}}{size\ of\ Vid_{(ori)}}\right) \times 100\% \tag{16}$$

2) Quality of compression result

Typically, video compression affects the decrease in video quality. The smaller CR indicates the less quality of the compression result. It means the compression and quality ratio are important factors to be considered. Two types of quality will be measured related to video compression: the quality of compression and decompression.

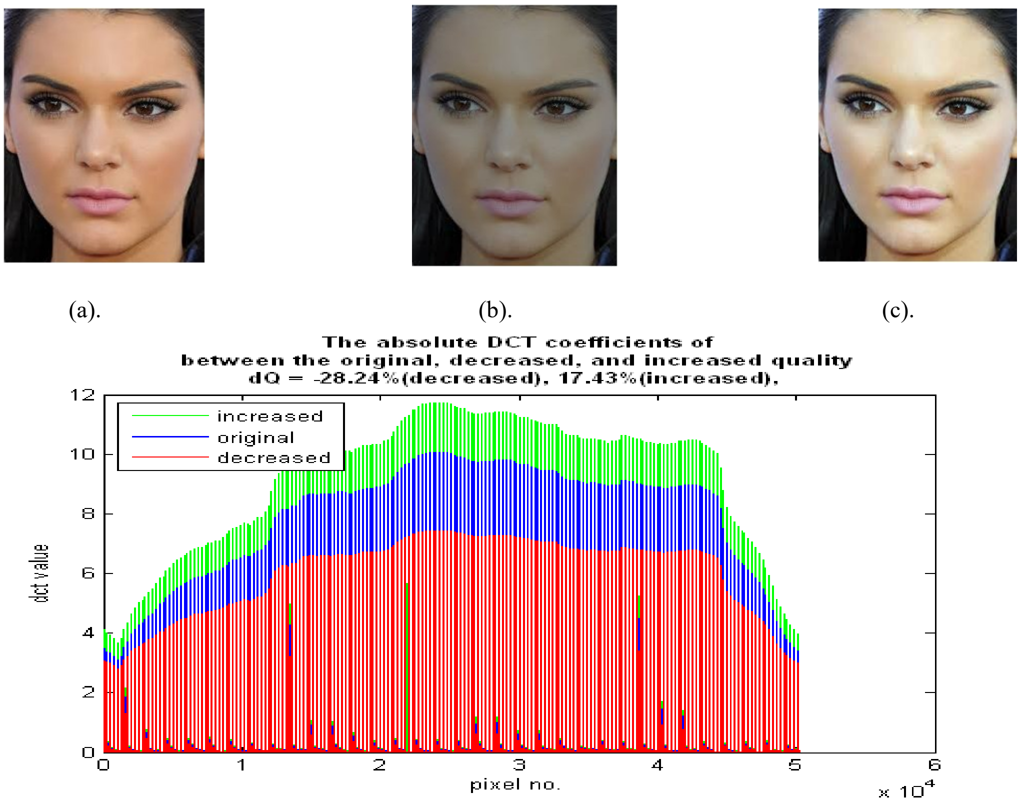
A gray image in the form of a series is considered as a discrete signal. The signal energy is one of the important characteristics of a signal, such as a feature. DCT is one signal transformation method with a better energy compaction property, presenting the major energy components in sequence with only a few transformation coefficients. Suppose a discrete signal $x(n)$ of length N is mathematically expressed as in (17).

$$x_{dct}(k) = w(k)\sum_{n=1}^{N} x(n) * cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right) \quad k = 1 \dots N$$
$$w(k) = \begin{cases} 1/\sqrt{N} & k = 1 \\ \sqrt{2/N} & 2 \le k \le N \end{cases} \tag{17}$$

The variable of $x_{dct}(k)$ is the DCT coefficients of $x(n)$. The DCT coefficients of a frame are the average DCT coefficients of components R, G, and B. The average DCT coefficients of the whole frame considered as the energy feature of a video. Video manipulation with a variety of purposes will impact the average change of absolute DCT coefficients. In this study, this change assumed as a video quality change. If X_{dct} and Y_{dct} the average DCT coefficients of the original and compressed video file, respectively, are the percent change in video quality of compression result expressed as in (18).

$$\bar{X}_{dct(abs)} = \frac{1}{N}\sum_{k=1}^{N}|X_{dct}(k)| \quad \overline{XY}_{dct(abs)} = \frac{1}{N}\sum_{k=1}^{N}|Y_{dct}(k)| - |X_{dct}(k)|$$
$$\Delta Q_{dct} = \left(\overline{XY}_{dct(abs)}/\bar{X}_{dct(abs)}\right) \times 100\% \tag{18}$$

$\bar{X}_{dct(abs)}$ is the average absolute value of X_{dct} and $\overline{XY}_{dct(abs)}$ is the average of the difference between the absolute value of Y_{dct} and X_{dct}. Whereas ΔQ_{dct} is percent change of the absolute DCT coefficients between X_{dct} and Y_{dct}. If the value is positive, then it is considered to have an improvement in quality and vice versa. The illustration is shown in Figure 8.



(a). (b). (c).

(e). The absolute DCT coefficients difference between (a) original, (b) decreased, and (c) increased quality

Fig. 8. The illustration of image quality change by DCT coefficients difference ratio

The set of DCT coefficients is also used to measure the percent of the decompression video quality change. If XX_{dct} and XX'_{dct} are the set of DCT coefficients per frame of the original and decompressed video, the percent change in the quality of decompression results is the average of ΔQ of the entire frames.

Another performance parameter also used in this study to measure the quality of video decompression results is PSNR (Peak Signal to Noise Ratio). PSNR provides a quantitative measure of the distortion that occurs during the decompression process. If F and F 'are the set of frames of the original video and the decompressed video, then PSNR is expressed as in (19).

$$\begin{aligned} MSE(n) &= \frac{1}{M \times N \times 3} \sum_{d=1}^{3} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(F_n(i,j,d) - F'_n(i,j,d)\right)^2 \\ PSNR &= \frac{1}{L} \sum_{n=1}^{L} 10 * log_{10}\left(255^2 / MSE(n)\right) \end{aligned} \tag{19}$$

MSE is the cumulative square error between the compressed and the original frame. It means PSNR in Eq. (19) states the average PSNR of all frames. If the assumed perfect decompression has a PSNR=60dB equivalent to MMSE=0.065, then the PSNR performance difference is expressed as in (20).

$$\Delta PSNR = \frac{60 - PSNR}{60} \times 100\% \tag{20}$$

3) *Total Performance (P)*

The smaller CR indicates more and more video parts are successfully compressed. The smaller $\Delta Q_{dct(comp)}$ indicates that the compressed video quality is getting better. It means the compression performance can be expressed as in (21).

$$P_{comp} = \left(1 - \left(\frac{CR + \Delta Q_{dct(comp)}}{2}\right)\right) \times 100\% \tag{21}$$

In the same way, the performance of decompression can be expressed as in (22).

$$P_{decomp} = \left(1 - \left(\frac{\Delta PSNR + \Delta Q_{dct(comp)}}{2}\right)\right) \times 100\% \tag{22}$$

Thus, total performance can be expressed as in (23).

$$P = \left(P_{comp} + P_{decomp}\right)/2 \tag{23}$$

E. *Sample Video Files*

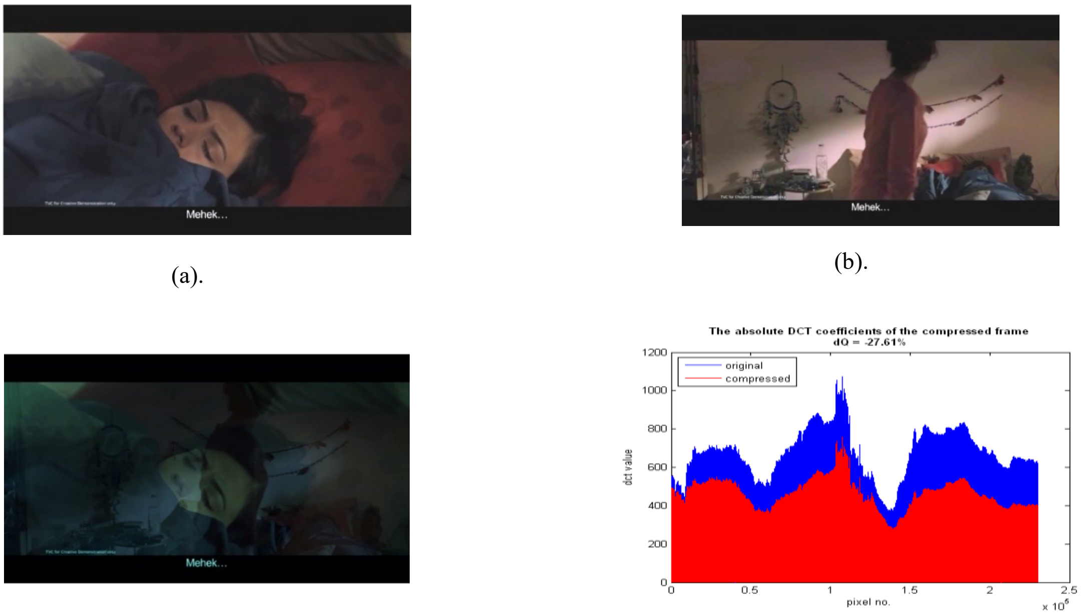
This study has used four video file samples. Table 1 shows the specifications of the entire sample. These samples have been obtained from partitioning a long video file into several video files with random sampling positions. Partition aims to get video files with an even number of frames.

Table 1. Specification of sample video file (frame rate 25 Fps, spatial resolution: 640×360)

No.	Video File	Length	Number of frame	File size (bytes)
1	vidfile1.avi	00:00:02	50	823,808
2	vidfile2.avi	00:00:03	90	1,468,416
3	vidfile3.avi	00:00:06	150	2,600,960
4	vidfile4.avi	00:00:08	200	4,418,048

III. Result and Discussion

In this section, we present sample compression frames 21 and 22 of sample 1 from Table 1 using the "mean" feature. The adaptive FIS training has used the $target\ error = 2\%$. Figure 9 illustrates the compression result using adaptive FIS $\Delta Q_{dct} = -27.61\%$. Figure 10 illustrates the decompression results of Figure 9. Decompression frames 21 and 22 each produce $\Delta Q_{dct} = -2.40\%, PSNR = 42.03dB$ and $\Delta Q_{dct} = -2.78\%, PSNR = 36.35dB$.



(a).

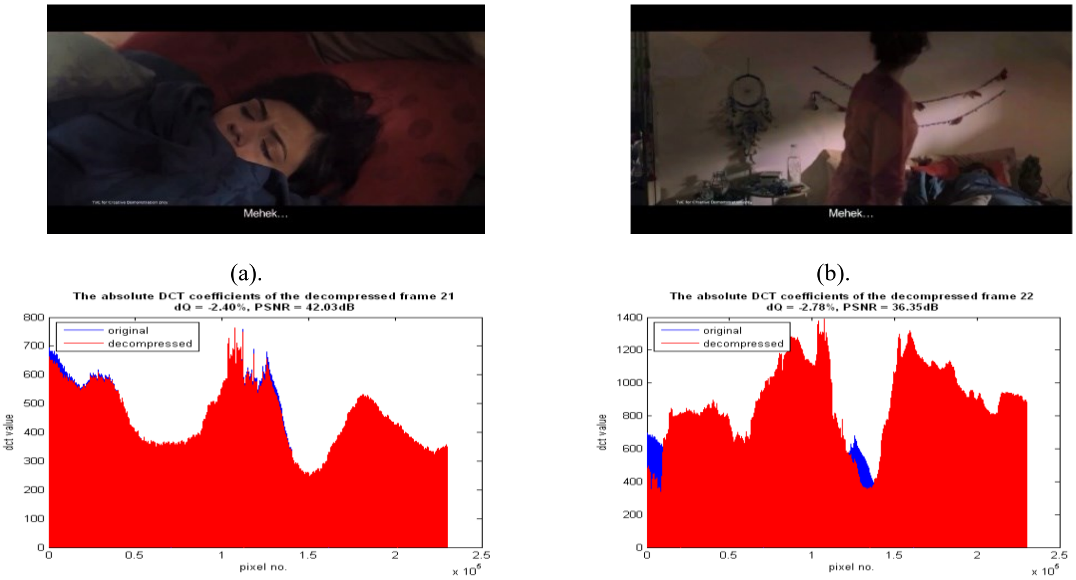
(b).

(c).

(d).

Fig. 9. The frame compression (a) frame 21 (b) frame 22 (c) compressed frame (d). The absolute DCT coefficients, ΔQ_dct=-27.61%.

The result of compression-decompression of all sample video files from Table 1 has presented in Table 2. Figure 11 shows the CR set curve between samples. If the CR set curve between samples has a good enough correlation visually, then the compression result is only dominant depending on the feature type. Figure 11 shows the CR set curve between samples uncorrelated visually. It has been proven that the compression result depends on the variation of the compressed features used for the compression process. The variation of the compressed features depends on the feature types and the number of the odd-even frame pair for adaptive FIS training.



(a).

(b).

(c).

(d).

Fig. 10. The frame decompression (a) decompressed frame 21 (b) decompressed frame 22 (c). The absolute DCT coefficients of decompressed frame 21(d). The absolute DCT coefficients of decompressed frame 22.

Table 2. The results of the compression-decompression process

No.	Sample Video File	Feature types	MAPE (train)	Compression			Decompression	
				File size	CR	ΔQdct (decreased)	ΔQ_{dct} (decreased)	PSNR (dB)
1	vidfile1.avi (50 frames) (823,808 bytes)	mean	1.62%	193,024	23.43%	26.67%	0.62%	56.03
		std	1.68%	214,528	26.04%	13.61%	10.37%	34.64
		mad	1.99%	215,040	26.10%	12.07%	12.05%	33.38
		mean(std)	1.49%	197,632	23.99%	20.32%	3.02%	45.95
		mean(mad)	1.70%	195,072	23.68%	23.91%	5.84%	40.01
2	vidfile2.avi (90 frames) (1,468,416 bytes)	mean	1.98%	405,504	27.62%	8.18%	0.73%	53.29
		std	1.86%	427,008	29.08%	1.66%	9.54%	33.96
		mad	1.91%	425,472	28.97%	6.53%	11.37%	32.39
		mean(std)	1.68%	401,920	27.37%	6.62%	3.03%	44.18
		mean(mad)	1.97%	402,944	27.44%	6.97%	5.61%	38.81
3	vidfile3.avi (150 frames) (2,600,960 bytes)	mean	1.80%	391,168	22.87%	15.70%	0.65%	49.05
		std	1.72%	398,080	23.21%	14.09%	11.46%	27.36
		mad	1.37%	397,056	23.21%	13.87%	12.15%	26.83
		mean(std)	1.63%	379,904	22.39%	16.83%	4.64%	35.74
		mean(mad)	1.83%	376,064	22.12%	18.20%	6.23%	33.06
4	vidfile4.avi (200 frames) (4,418,048 bytes)	mean	1.95%	576,000	26.39%	10.27%	0.84%	49.52
		std	2.20%	581,632	26.62%	12.18%	12.43%	28.29
		mad	2.10%	574,208	26.27%	10.37%	12.97%	27.90
		mean(std)	1.67%	559,104	25.64%	12.05%	4.51%	37.73
		mean(mad)	1.86%	555,520	25.48%	13.17%	6.06%	35.05

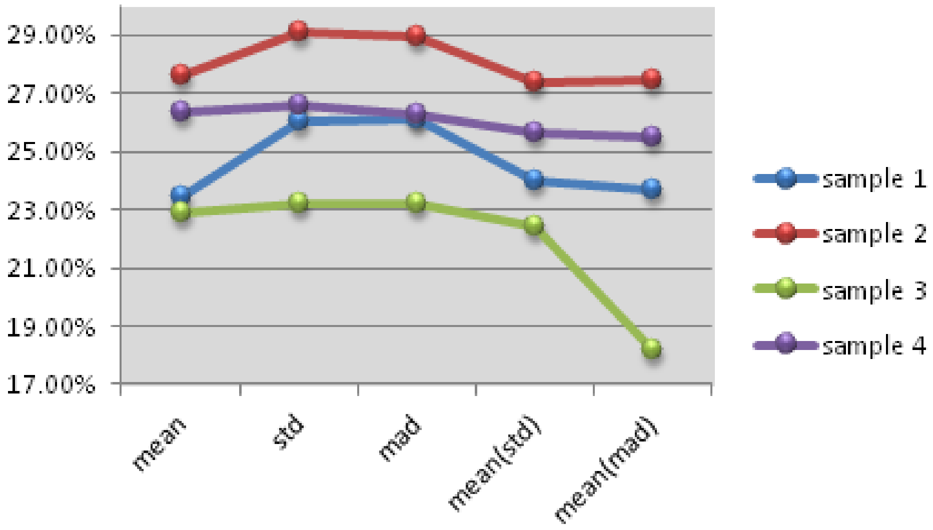


Fig. 11. The comparison of CR between samples

The average compression-decompression result based on the feature type has presented in Table 3. Overall, the proposed method yields an average $CR = 25.39\%$ with total performance $P = 80.13\%$. Of the five feature types selected, the "average" feature has resulted in the best video decompression with $CR = 25.08\%$ and $P = 87.15\%$. Table 3 has also presented various compression performance options by feature type as needed. The "*mean* (*mad*)" feature has resulted in the best compression ratio ($CR = 24.68\%$) making it suitable for storage efficiency requirements. The "*std*" feature has resulted in the smallest quality change in the compression process ($\Delta Q_{dct} = 10.39\%$), making it suitable for video transmission needs without decompression. Its best compression performance has also indicated it $\left(P_{comp} = 81.69\%\right)$.

Table 3. The average of the compression-decompression results by feature type

No.	Feature types	**Compression**		P_{comp}	**Decompression**		P_{decomp}	P
		CR	**ΔQdct (decreased)**		**ΔQdct (decreased)**	$\Delta PSNR$		
1	25.08%	15.21%	79.86%	0.71%	10.40%	94.45%	**87.15%**	25.08%
2	26.24%	10.39%	81.69%	10.95%	46.45%	71.30%	76.50%	26.24%
3	26.14%	10.71%	81.58%	12.14%	48.06%	69.90%	75.74%	26.14%
4	24.85%	13.95%	80.60%	3.80%	29.48%	83.36%	81.98%	24.85%
5	24.68%	15.56%	79.88%	5.93%	36.67%	78.70%	79.29%	24.68%
Average		**25.39%**	**13.16%**	**80.72%**	**6.71%**	**34.21%**	**79.54%**	**80.13%**

IV. Conclusion

In general, the results of this study have proven that inter-frame-based video compression can be performed by applying feature compression techniques from all even frame pairs using adaptive FIS. The resulting compression ratio depends on the type of feature used and the number of odd-even frame pairs used for the adaptive FIS training. The proposed method yields an average $CR = 25.39\%$ with total performance $P = 80.13\%$. Of the five selected feature types, the "mean" feature has produced the best video decompression with $CR = 25.08\%$ and $P = 87.15\%$. The results of this study have also presented various compression performance options by feature type as needed. The "mean (mad)" feature answers the need for storage efficiency. The "std" feature is suitable for video transmission needs without a decompression process. Further studies focused on increasing the variety of uniqueness of compressed features through the use of other features and frame selection techniques for adaptive FIS training. The expected impact is an increase in inter-frame-based video compression performance.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission information are available at http://journal2.um.ac.id/index.php/keds.

Publisher’s Note: Department of Electrical Engineering and Informatics - Universitas Negeri Malang remains neutral with regard to jurisdictional claims and institutional affiliations.

References

[1] R. Wright, “Preserving Moving Pictures and Sound: DPC Technology Watch Report 12-01,” Mar. 2012.
[2] J. H. Pujar and L. M. Kadlaskar, “A new lossless method of image compression and decompression using Huffman coding techniques.,” *J. Theor. Appl. Inf. Technol.*, vol. 15, 2010.
[3] E. B. Tashmanov and R. A. Raxmonberdiev, “The Interframe Image Processing in a Video Codec Based on the Wavelet Transformation,” *Int. J. Res. Eng. Technol.*, vol. 06, no. 05, pp. 1–2, 2017.
[4] S. Zhu, S. Zhang, and C. Ran, “An Improved Inter-Frame Prediction Algorithm for Video Coding Based on Fractal and H.264,” *IEEE Access*, vol. 5, pp. 18715–18724, 2017.
[5] Q. Zhang, M. Chen, X. Huang, N. Li, and Y. Gan, “Low-complexity depth map compression in HEVC-based 3D video coding,” *EURASIP J. Image Video Process.*, vol. 2015, no. 1, p. 2, Dec. 2015.
[6] B. Li, J. Han, and Y. Xu, “Co-located Reference Frame Interpolation Using Optical Flow Estimation for Video Compression,” in *2018 Data Compression Conference*, Mar. 2018, pp. 13–22.
[7] N. Manjanaik, B. D. Parameshachari, S. N. Hanumanthappa, and R. Banu, “Intra Frame Coding In Advanced Video Coding Standard (H.264) to Obtain Consistent PSNR and Reduce Bit Rate for Diagonal Down Left Mode Using Gaussian Pulse,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 225, p. 012209, Aug. 2017.
[8] K. S. Reddy, B. Srikanth, and C. L. Reddy, “Design and analysis of video compression technique using HEVC intra-frame coding,” *IJESRT (International J. Eng. Sci. Res. Technol. vol. 06, pp. 477-482,* 2017.

[9] F. Sampaio, B. Zatt, M. Shafique, L. Agostini, J. Henkel, and S. Bampi, “Content-adaptive reference frame compression based on intra-frame prediction for multiview video coding,” in *2013 IEEE International Conference on Image Processing*, Sep. 2013, pp. 1831–1835.

[10] A. Habib and D. Chowdhury, “An Efficient Compression Technique Using Arithmetic Coding,” *J. Sci. Res. Reports*, vol. 4, no. 1, pp. 60–67, 2015.

[11] A. Masmoudi, W. Puech, and A. Masmoudi, “An improved lossless image compression based arithmetic coding using mixture of non-parametric distributions,” *Multimed. Tools Appl*., vol. 74, no. 23, pp. 10605–10619, Dec. 2015.

[12] H. S. Shivaputra and V. L. Sheshadri, “An Efficient Lossless Medical Image Compression Technique for Telemedicine Applications,” *Comput. Appl. An Int. J*., vol. 2, no. 1, pp. 63–69, 2015.

[13] M. Ebrahim and W. C. Chai, “Multi-phase joint reconstruction framework for multi-view video compression using block-based compressive sensing,” in *2015 Visual Communications and Image Processing (VCIP)*, Dec. 2015, pp. 1–4.

[14] F. Kamisli, “Block-Based Spatial Prediction and Transforms Based on 2D Markov Processes for Image and Video Compression,” *IEEE Trans. Image Process*., vol. 24, no. 4, pp. 1247–1260, Apr. 2015.

[15] P. Patil, S. B. Patil, H. Shelke, and A. Sankhe, “Analysis of Video Compression using DCT,” *Imp. J. Interdiscip. Res*., vol. 3, 2017.

[16] W. Yu, D. Hu, N. Tian, and Z. Zhou, “A novel search method based on artificial bee colony algorithm for block motion estimation,” *EURASIP J. Image Video Process*., vol. 2017, no. 1, p. 66, Dec. 2017.

[17] S. Mahesh, “A Survey: Various Techniques of Video Compression,” *Int. J. Eng. Trends Technol*., vol. 7, no. 1, pp. 10–12, Jan. 2014.

[18] P. K. Charles and K. S. Rao, “A novel search technique of motion estimation for video compression,” *Glob. J. Comput. Sci. Technol*., vol. 17, no. F2, pp. 1–5, 2017.

[19] D. García-Lucas, G. Cebrián-Márquez, A. J. Díaz-Honrubia, and P. Cuenca, “Acceleration of the integer motion estimation in JEM through pre-analysis techniques,” *J. Supercomput*., vol. 75, no. 3, pp. 1203–1214, Mar. 2019.

[20] G. Pastuszak and M. Jakubowski, “Optimization of the Adaptive Computationally-Scalable Motion Estimation and Compensation for the Hardware H.264/AVC Encoder,” *J. Signal Process. Syst*., vol. 82, no. 3, pp. 391–402, Mar. 2016.

[21] P. Jadhav and G. K. Siddesh, “Codec with Neuro-Fuzzy Motion Compensation and Multi-scale Wavelets for Quality Video Frames,” in *Advances in Intelligent Systems and Computing*, 2018, pp. 599–606.

[22] M. A. Shaikh and S. S. Badnerkar, “Video Compression Algorithm Using Motion Compensation Technique: A Survey,” *Int. J*., vol. 2, no. 3, pp. 625–629, 2014.

[23] M. Al-Ani and T. A. Hammouri, “Video compression algorithm based on frame difference approaches,” *Int. J. Soft Comput*., vol. 2, no. 4, p. 67, 2011.

[24] S. Y. Kahu and K. M. Bhurchandi, “A Low-Complexity, Sequential Video Compression Scheme Using Frame Differential Directional Filter Bank Decomposition in CIE La * b * Color Space,” *IEEE Access*, vol. 5, pp. 14914–14929, 2017.

[25] S. M. Tirupathamma, “Key frame based video summarization using frame difference,” *Int. J. Innov. Comput. Sci. Eng*., vol. 4, no. 03, pp. 160–165, 2017.

[26] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, “Median Robust Extended Local Binary Pattern for Texture Classification,” *IEEE Trans. Image Process*., vol. 25, no. 3, pp. 1368–1381, Mar. 2016.

[27] A. Fanany, A. Bramanto, A. Wajiansyah, and S. S., “Texture Feature Extraction based on Local Weighting Pattern (LWP) using Fuzzy Logic Approach,” *Int. J. Comput. Appl*., vol. 179, no. 28, pp. 1–8, 2018.

[28] A. B. W. Putra, R. Malani, and M. Mulyanto, “A Gray-Level Dynamic Range Modification Technique for Image Feature Extraction Using Fuzzy Membership Function,” Indones. *J. Artif. Intell. Data Min*., vol. 1, no. 1, p. 6, Mar. 2018.

[29] D. J. Ashpin Pabi, P. Aruna, and N. Puviarasan, “Color Image Compression Based on Feature Extraction,” in *Advances in Intelligent Systems and Computing*, 2018, pp. 375–385.

[30] K. Jaferzadeh, I. Moon, and S. Gholami, “Enhancing fractal image compression speed using local features for reducing search space,” *Pattern Anal. Appl*., vol. 20, no. 4, pp. 1119–1128, Nov. 2017.

[31] E. Morales-Cruz and J. J. Garcia-Hernandez, “Detection of JPEG compression on bitmap image based on phase spectrum statistical features,” in *2017 International Conference on Information Society (i-Society)*, Jul. 2017, pp. 111–115.

[32] Z. Wang, Z. Lin, L. Xu, Y. Zhao, and J. Xin, “Batch images compression algorithm based on the common features,” in 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Oct. 2017, pp. 1–6.