



Intra-frame Based Video Compression Using Deep Convolutional Neural Network (DCNN)

Arief Bramanto Wicaksono Putra^a, Achmad Fanany Onnilita Gaffar^a, Muhammad Taufiq Sumadi^{b,*},
Lisa Setiawati^a

^a The Applied Modern Computing & Robotic Systems Unit, Politeknik Negeri Samarinda, Samarinda, 75131, Indonesia

^b Department of Informatics, Faculty of Science and Technology, Universitas Muhammadiyah Kalimantan Timur, Samarinda, Indonesia
Corresponding author: *sumadi11895@gmail.com

Abstract— In principle, a video codec is built by implementing various algorithms and their development. The next generation of codecs involves more artificial intelligence applications and their development. DCNN (Deep Convolutional Neural Network) is a multi-layer NN concept with a deep learning approach in the field of artificial intelligence development. This study has proposed a DCNN with three hidden layers for intra-frame-based video compression. DCT and fractal methods were used to compare the performance of the proposed method. The training image (obtained from the average of all down-sampled frames) is divided into several square blocks using the square block shift operation until all parts of the image are fulfilled. All pixels in each block act as input data patterns. After the training process, the trained proposed DCNN was then used to construct the feature and sub-feature image obtained through the max function operation in the feature bank and sub-feature bank. These feature and sub-feature images were then a spatial redundancy minimizer with specific manipulation techniques and simultaneously a quantizer without converting the frame's pixels to a bit-stream. The result of this process is a compressed image. Experiments on the entire dataset resulted in AAPR (Average Approximate Performance Ratio) of 147.71%, or an average of 1.5 times better than other methods. For further studies, the performance improvement of the proposed DCNN is performed by modifying its structure so that the output is direct in the form of feature and sub-feature images. Another way is to combine it with the DCT or fractal method to improve the performance of the result.

Keywords— Video codecs; intra frame; video compression; DCNN.

Manuscript received 12 Jul. 2022; revised 15 Aug. 2022; accepted 23 Aug. 2022. Date of publication 30 Sep. 2022.
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

Various needs for the image and video capture process impact increasing data, which inevitably requires methods and techniques to reduce the amount of data sent or stored. The application of these methods and techniques is classified as compression technology. Various video coding standards, over the years, have always had the same core problem, namely how to reduce the size of the video data as much as possible from the original video to the compressed video that is stored or transmitted [1]. Video data contains a high degree of redundancy. The pixels within a frame often repeat or are similar to adjacent pixels, and the correlation between these pixels is known as spatial redundancy. Sequential frames in a video are usually similar, and the correlation between successive frames is referred to as temporal redundancy. In principle, video compression reduces the number of video

data bits by encoding the information by eliminating spatial or temporal redundancy [2].

Video compression reduces data used by encoding digital video content. This reduction is intended to meet smaller storage and lower transmission bandwidth requirements for video content clips. The process of compressing and decompressing video requires a codec (encoder-decoder). An encoder is used to compress video data at a certain target bit rate. Simultaneously, the decoder decompresses the video signal to make it similar to the original [3]. Several video compression standards have developed since the 1990s. AVI (Audio Video Interleave) is one of the oldest video formats made by Microsoft in 1992. AVI files are usually created without compression, resulting in large file sizes. The AVI files are often used when recording before converting them to other formats. The International Telecommunication Union (ITU) and the International Organization for Standardization (ISO) have developed a video compression standard called MPEG (Motion Picture Experts Group). MPEG-1 was the

first MPEG standard finalized in 1992 and widely used for video CDs [4]. Meanwhile, the second generation is MPEG-2, completed in 1995 and widely used for DVD and digital TV broadcasting [5]. MPEG-4 is Advanced Video Coding (AVC / H.264) which was completed in 2003 and is widely used for HDTV and IP-based video services [6]. MPEG-H High-Efficiency Video Coding (HEVC / H.265) was completed in 2013 and is widely used for HDR video applications [7].

There are two main video compression classes: lossy and lossless. Lossy compression permanently eliminates data redundancy, especially for coding of perception based on human color perception. This method allows compressing files to a smaller size or lower bit rate. However, it impacts the quality of the image or video when it is decompressed. In contrast, lossless compression eliminates data redundancy without affecting quality, whereby the process maintains data integrity and can be completely decompressed. Unfortunately, lossless compression does not significantly reduce the number of video data bits [8]. There are several common approaches to video compression. Inter-frame-based video compression eliminates the temporal redundancy of consecutive frames. Several studies that apply inter-frame-based video compression have been conducted in [9]-[13]. Intra-frame-based video compression eliminates spatial redundancy on individual frames [14]. Several studies that apply intra-frame-based video compression have been conducted [14]-[19]. Block-based video compression is a combination of the two approaches. Video frames are grouped into coding blocks to predict, modify, quantize, and encode. The first frame of each block is predicted and coded using an intra-frame-based concept. Next, intra-frame and inter-frame-based concepts are then applied to the remaining frames [20]. Several studies that apply block-based video compression have been conducted in [20]-[23].

In principle, codecs are built by implementing various algorithms and their development. The next generation of codecs involves more artificial intelligence and smart applications. Deep learning, as one of the latest developments in the field of artificial intelligence, has been widely used in various studies on video compression [9], [12], [15], [24]-[27].

Deep Learning is a machine learning method that collects every detail of the learning process by manipulating various compositions of mathematical functions. The goal is to get more abstract data, more multi-level data, and more complex data features. Deep Learning is a sophisticated development of the multi-layer ANN concept [28]. DNN (Deep Neural Network) is a multi-layer neural network with more than three layers, so it is also called Deep NN. DNN's ability to solve problems increases as more layers are used. DNN can consist of various layers used (fully connected, convolution, autoencoder, min/max, dropout, SoftMax, recurrent layer, etc.) [29]. Convolutional Neural Network (CNN) is a type of ANN that consists of a convolutional layer. Each neuron in the convolutional layer is usually connected to only a few input neurons reducing the computational complexity and parameters. Neurons in this layer convolute the matrix on their input. The input connected to neurons in a convolutional layer is sometimes referred to as the neuron's visual field. Due to the inherent spatial dependence between pixels in an image or video, CNN has proven very effective in analyzing image

or video structured data with this convolutional concept. CNN has been widely used in various image and video processing studies [24], [30]-[37].

This study proposes an intra-frame-based video compression using DCNN. Video compression is carried out through the main stages: (a). elimination of redundancy, (b). quantization, (c). entropy coding. DCNN is used to extract the features and sub-features of each frame. These features and sub-features will later function as a spatial redundancy minimizer with specific manipulation techniques and simultaneously as a quantizer without converting the frame's pixels to a bit-stream. The result performance will be compared with other intra-frame-based video compression methods (DCT [38]-[40] and fractal [41]), which are commonly used for image compression.

II. THE PROPOSED METHOD

A. Convolutional Neural Network

CNN is one type of ANN that adopts the concept of image convolution operations. Neurons in the convolution layer convolute the matrix of their input. The convolutional kernel functions as a filter that extracts features from the input image. Kernel size and value can be freely selected as needed. Suppose the input image is 3×3 with a 2×2 convolutional kernel where all kernel values are 1, then the convolution operation using kernel shift is illustrated as in **Error! Reference source not found.**

The convolutional kernel overlaps the input image by starting from the top left corner. It then calculates the product between the numbers in the convolutional kernel and the input image according to their location. It sums all the resulting products to get a pixel value. As an example:

$$\begin{bmatrix} 5 & 5 \\ 15 & 5 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = (5 \times 1) + (15 \times 1) + (5 \times 1) + (5 \times 1) = 30$$

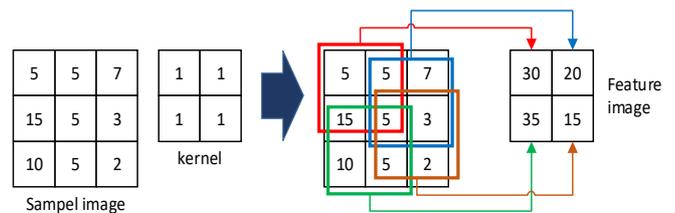


Fig. 1 Illustration of the convolution operation

The (*) symbol is a convolution operator. The kernel is shifted by one pixel (stride 1) to get the next convolution result until all parts of the input image are fulfilled. The concept of image convolution was then adopted by CNN, as shown in 0.

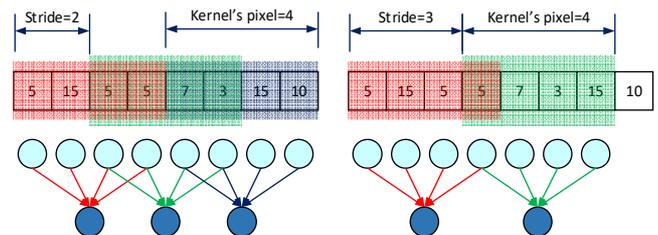


Fig. 2 Adoption of image convolution to CNN

B. The proposed Deep Convolutional Neural Network

In intra-frame-based video compression, each frame's compression is like compressing an image. This study proposes using DCNN to perform convolution operations on each square block of pixels with a specific size. Block shift operation is performed until all parts of the input image are fulfilled. All pixels in a square block act as input data patterns for DCNN. If there is an image of $M \times M$ size, using a square block of $N \times N$ size, there will be a number of $(M - N)^2$ square blocks. It means that there will be $(M - N)^2$ number of input data patterns. The proposed DCNN uses a 2×2 kernel with three hidden layers, as illustrated in **Error! Reference source not found.**. After the training process is complete, DCNN is ready to build the feature image with the size of $(M - N) \times (M - N)$ from the input image.

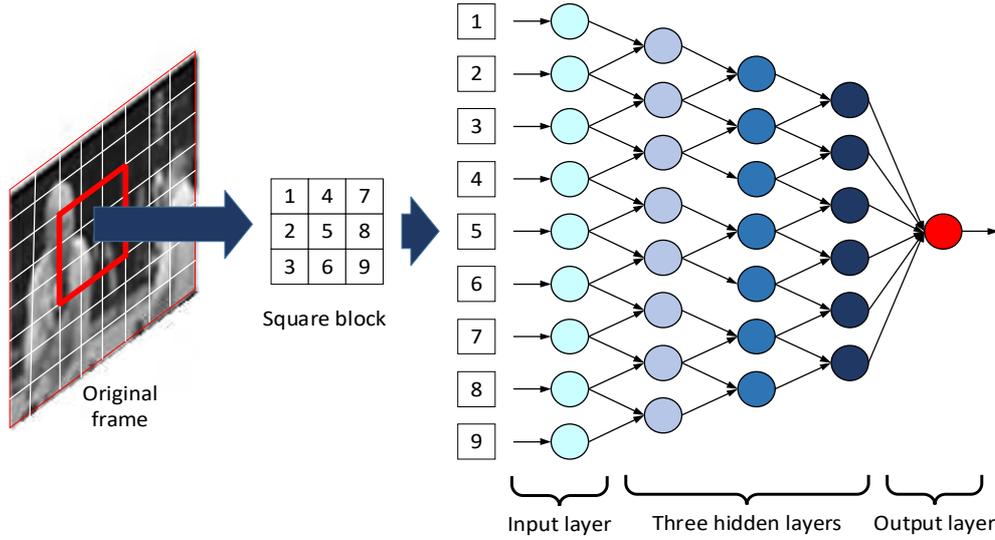


Fig. 3 The proposed DNN input pattern

The composition of the mathematical function of the proposed DCNN is based on Eq. (1). The learning process is performed to reduce network errors by using a gradient descent algorithm. Each layer errors by using a gradient descent algorithm. The backpropagation process is carried out at each gradient point for each layer by applying the chain rule principle. All layer weights are updated using the following formula [42].

In general, the proposed DCNN for intra-frame-based video compression is shown in **Error! Reference source not found.**. Each frame's RGB image is first down sampled to a square size (100×100) using bicubic interpolation. Each component (R, G, B) is used as the input image of the proposed DCNN. The proposed DCNN is trained in such a way that it can produce a feature bank containing eight feature images and a sub-feature bank containing 16 sub-feature images. This study uses a 5×5 square block. If the input image is $M \times M$ size, the feature image will be $(M - 5) \times (M - 5)$, and the sub-feature image will be $\text{floor}(((M - 5)/2) \times ((M - 5)/2))$.

Each feature bank and the sub-feature bank will generate only one feature image and a sub-feature image by applying the max function and the up-sampling operation to the original size. The average of both feature images is used to minimize the spatial redundancy of the original image. Suppose $R_{feat(up)}$ X and $R_{subfeat(up)}$ are feature and sub-

feature images of component R due to the up-sampling operation, respectively. The spatial redundancy removal of the original R component is mathematically expressed by:

$$\begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_{N-1} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & 0 & \dots & 0 \\ 0 & w_{22} & w_{23} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & w_{N-1,N-1} & w_{N-1,N} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_{N-1} \end{bmatrix} \quad (1)$$

$$h_{N-1} = {}^c W_{N-1,N} \cdot X_N + b_N$$

$$H_{N-1} = f(h_{N-1})$$

${}^c W_{N-1,N}$ is a convolution matrix, b_N is the bias vector of a layer, and H_{N-1} is the output layer after being activated by an activation function.

R_{sre} is the gray image of the R component as a result of minimizing its spatial redundancy. The quantization process is expressed by:

$$R_{sre}(\cdot) = R_{ori}(\cdot) * \left(\frac{R_{feat(up)} + R_{subfeat(up)}}{2} \right) \quad (2)$$

$R_{std}(\cdot) = \left(\frac{1}{M-1} \sum_{i=1}^M \left(R_{sre}(i,j) - \frac{1}{M} \sum_{i=1}^M R_{sre}(i,j) \right)^2 \right)^{1/2}$ $R_{comp}(\cdot) = R_{std}(\cdot) + \frac{R_{sre}(\cdot) * (0.9)}{\max(R_{sre}(\cdot)) - \min(R_{sre}(\cdot))}$

$$R_{std}(\cdot) = \left(\frac{1}{M-1} \sum_{i=1}^M \left(R_{sre}(i,j) - \frac{1}{M} \sum_{i=1}^M R_{sre}(i,j) \right)^2 \right)^{1/2} \quad (3)$$

$$R_{comp}(\cdot) = R_{std}(\cdot) + \frac{R_{sre}(\cdot) * (0.9)}{\max(R_{sre}(\cdot)) - \min(R_{sre}(\cdot))}$$

R_{std} is the standard deviation of the R_{msr} , and a coefficient of 0.9 was obtained experimentally. R_{comp} is a quantized gray image of the R component, which is also a compressed image.

C. Training Strategy

All layers in the proposed DCNN use the tangent-sigmoid activation function represented by $\tanh(y) = (e^y - e^{-y}) / (e^y + e^{-y})$, where the derivative is represented by $f'(y) = 1 - (f(y))^2$. The tangent-sigmoid activation function's choice is to ensure the convergence and speed of the training process where all the input from each layer needs to be mapped as close as possible to zero within the range $\{-1 \dots 1\}$. In this case,

the training data sample needs to be normalized by using the following formula:

$$\mu_X = \sum_{i=1}^n X_i \quad \sigma_X = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \mu_X)^2} \quad \hat{X}_i = \frac{X_i - \mu_X}{\sigma_X} \quad (4)$$

X_i is the i th sampled data, n is the number of sampled data, μ_X is the average of sampled data, σ_X is the standard deviation of the sampled data, and \hat{X}_i is the i th sampled of normalized data. The variance between units in a layer must be close to unity to ensure there is no correlation and to ensure the training process's convergence. For this purpose, the

weighting initialization of each layer should be using a random normal distribution with zero means.

DCNN requires network errors and error functions to control its training process. SSE (Sum Squared Error) represents network errors expressed by:

$$SSE = E = \frac{1}{2} \sum_{i=1}^n (e_{(i)})^2 = \frac{1}{2} \sum_{i=1}^n (Y_{(i-1)} - Y_{(i)})^2 \quad (5)$$

$Y_{(i)}$ is the i th net output. For the first net output, $Y_{(1)} = \text{mean}(\hat{X})$ where \hat{X} is the first normalized data input pattern, while n is the number of training data. The training process is stopped if $E \leq \text{target error}$, where the target error is selected as small as possible (close to zero).

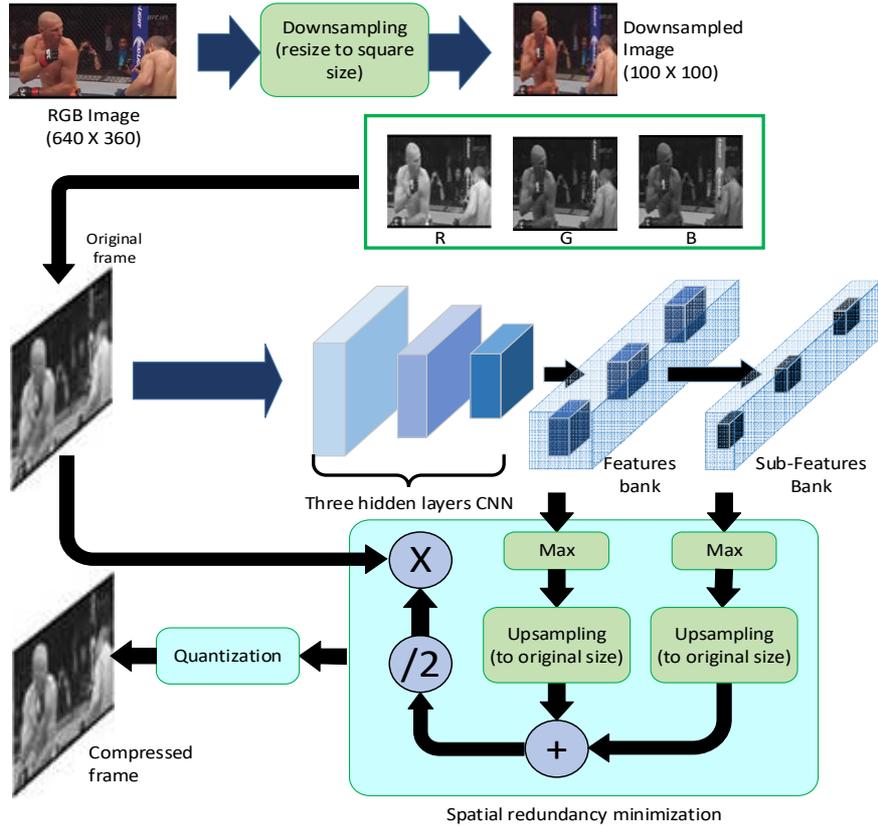


Fig. 4 The proposed DCNN for intra-frame based video compression

For the efficiency of the training process, DCNN requires a reference frame as an input image. If $V(:, :, 3, nf)$ is a video with nf number of frames; in this study, the frame reference is expressed by:

$$F_{ref(c)} = E = \frac{1}{nf} \sum_{i=1}^{nf} V(:, :, c, nf) \quad (6)$$

where c is the number of RGB component (R=1, G=2, B=3).

D. Dataset

This study uses the selected MMA (Mix Martial Arts) video clips downloaded from YouTube (mp4) as a dataset. This video clip is converted into the uncompressed AVI format using a commonly used video converter application. The dataset specifications used are shown in 0

E. Performance Measurement

1) *PSNR (Peak Signal-to-Noise Ratio)*: PSNR is the ratio between the maximum possible power of the signal and the noise's destructive power, which affects the representation's accuracy. In image or video compression, noise is an error that arises from the compression process. This noise is usually expressed in MSE (Mean Squared Error), which is the difference between the compression result and the original one. Hence, the PSNR is considered an estimate of the human perception of the reconstructed compression output quality. If V and V_{com} are original and compressed images, with $M \times N$ spatial resolution in pixels and nf number of frames; the PSNR is denoted by:

$$MSE(c, k) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (V(i, j, c, k) - V_{com}(i, j, c, k))^2$$

$$PSNR(c, k) = 10 * \log_{10} \left(\frac{MAX^2}{MSE(c, k)} \right) \text{ dB} \quad (7)$$

$$\overline{PSNR} = \frac{1}{3 \times nf} \sum_{c=1}^3 \sum_{k=1}^{nf} PSNR(c, k)$$

MAX is the maximum pixel value in image (for 8-bit image of 255), c is the number of RGB component, and $k = 1 \dots nf$. \overline{PSNR} is the average PSNR value across frames, the PSNR value of a compressed video.

Typical values for PSNR in the lossy image and video compression are between 30 and 50 dB for an 8-bit bit depth, where higher is better. As for 16-bit data, it is usually between 60 and 80 dB [43].

TABLE I
VIDEO CLIP DATASET (FRAME RATE: 25 FPS, SPATIAL RESOLUTION: 640 x 360, LENGTH: 00:00:15)

No.	Video clip	Size (bytes)	Total bit rate (kbps)
1	MMA Best Kick 1	10,229,778	5,426
2	MMA Best Kick 2	9,090,980	4,822
3	MMA Best Kick 3	9,268,436	4,916
4	MMA Best Kick 4	10,068,010	5,312
5	MMA Best Kick 5	9,180,776	4,870
6	MMA Elbow KO 1	8,996,248	4,797
7	MMA Elbow KO 2	8,351,834	4,418
8	MMA Elbow KO 3	8,484,404	4,525
9	MMA Elbow KO 4	9,888,212	5,231
10	MMA Elbow KO 5	7,713,524	4,113

2) *SSIM (Structural Similarity Index Measurement)*: SSIM is a metric used to measure the similarity between two images. The SSIM index is a full reference quality metric [44] which states that image quality predictions are based on uncompressed or distorted initial images as a reference. If $X = V(c, k)$ and $Y = V_{com}(c, k)$, then SSIM index stated by:

$$SSIM(X, Y) = lum(X, Y) * con(X, Y) * str(X, Y)$$

$$lum(X, Y) = \frac{2\mu_X\mu_Y + c_1}{\mu_X^2 + \mu_Y^2 + c_1}$$

$$con(X, Y) = \frac{2\sigma_X\sigma_Y + c_2}{\sigma_X^2 + \sigma_Y^2 + c_2}$$

$$str(X, Y) = \frac{\sigma_{XY} + c_3}{\sigma_X\sigma_Y + c_3} \quad (8)$$

$$\overline{SSIM} = \frac{1}{3 \times nf} \sum_{c=1}^3 \sum_{k=1}^{nf} SSIM(X(c, k), Y(c, k))$$

The SSIM index of an image against itself is 1. The functions of lum , con , and str are luminance, contrast, and structure, respectively. The variables of μ_X , μ_Y , σ_X^2 , σ_Y^2 , and σ_{XY} are the average of X , the average of Y , the variance of X , the variance of Y , and the covariance of X and Y , respectively. The variables of σ_X and σ_Y are the deviation standard of X and Y . The constants of c_1 , c_2 and c_3 are the stabilizer for the division with a weak denominator, where $c_1 = (0.01 * L)^2$, $c_2 = (0.03 * L)^2$, and $c_3 = c_2/2$. Commonly, $L = 255$ for an 8-bit depth image. \overline{SSIM} is the average SSIM value across frames, which is the SSIM value of a compressed video.

3) *CR (Compression Ratio) and SS (Space Saving)*: The video compression ratio (CR) is defined as the ratio between

the original video size and the compressed video size. In contrast, space-saving (SS) is defined as a reduction in size relative to the original size. Those metrics denoted as [45]:

$$CR = \frac{\text{original video size}}{\text{compressed video size}} \times 100\% \quad (9)$$

$$SS = \left(1 - \frac{\text{compressed video size}}{\text{original video size}} \right) \times 100\%$$

4) *DCT (Discrete Cosine Transform)*: Commonly, video compression affects the decrease in video quality. The smaller the PSNR and SSIM, the lower the quality of the compression results. The gray image in the form of a series is considered a discrete signal. Signal energy is one of the essential characteristics of a signal, like a feature. DCT is a signal transformation method with better energy compaction properties that present the main energy components in sequence with only a few transformation coefficients. Suppose there is a discrete signal $x(n)$ of length N . The transformation of signal $x(n)$ using DCT is mathematically expressed by:

$$x_{dct}(k) = w(k) \sum_{n=1}^N x(n) * \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right) \quad k = 1 \dots N \quad (10)$$

$$w(k) = \begin{cases} 1/\sqrt{N} & k = 1 \\ \sqrt{2/N} & 2 \leq k \leq N \end{cases}$$

$x_{dct}(k)$ is the DCT coefficients of $x(n)$.

The DCT coefficient of a frame is the average DCT coefficient of the R, G, and B components. The average DCT coefficient of the entire frame is considered a feature of video energy. Video manipulation with various purposes will impact the change in the average of the absolute DCT coefficient. In this study, these changes are assumed to be changes in video quality. Suppose X_{dct} and Y_{dct} are the average DCT coefficients of the original and compressed video file, respectively. The percentage change in the quality of the compressed video relative to the original is represented by:

$$\bar{X}_{dct(abs)} = \frac{1}{N} \sum_{k=1}^N |X_{dct}(k)| \quad \bar{Y}_{dct(abs)} = \frac{1}{N} \sum_{k=1}^N |Y_{dct}(k)| \quad (11)$$

$$\Delta Q_{dct} = (\bar{Y}_{dct(abs)} / \bar{X}_{dct(abs)}) \times 100\%$$

$\bar{X}_{dct(abs)}$ is the average absolute value of X_{dct} , and $\bar{Y}_{dct(abs)}$ is the average of the difference between the absolute value of Y_{dct} and X_{dct} . Whereas ΔQ_{dct} is the percent change of the absolute DCT coefficients between Y_{dct} and X_{dct} . If the value is positive, then it is considered to have an improvement in quality and vice versa. The illustration is shown in 0.

5) *APR (Approximate Performance Ratio) and AAPR (Average APR)*: APR is used to measure the performance of the proposed method with other methods for specific performance metrics. Suppose P_i and Q_i^k are the i th performance of the proposed method and the k th other method, respectively. Mathematically, the APR and MAPR are expressed by:

$$APR_i^k = \frac{P_i}{Q_i^k} \times 100 \quad AAPR = \frac{1}{K \times L} \sum_{k=1}^K \sum_{i=1}^L APR_i^k \quad (12)$$

K and L are the number of other methods and performance metrics used. The proposed DCNN training process uses MATLAB programming. The reference frame for each dataset generated by using Eq. (6) uses as an input image. Then, the trained proposed DCNN is used to compress the entire dataset.

III. RESULT AND DISCUSSION

The proposed DCNN training process uses MATLAB programming. The frame of reference generated using Eq. (4) for each dataset, as shown in 0 used as an input image. Furthermore, each frame's compression would be performed using trained DCNN. This section's discussion uses the "MMA Elbow KO 2" video clip file. An example of each stage's results, as shown in 0, as illustrated in 0 using 200th frame. The results of comparing all methods for the 200th

frame are shown in 0. The comparison of all methods for PSNR and SSIM of the entire frame is shown in 0.

Referring to 0, the result of spatial redundancy elimination has a smaller correlation between pixels than the original (46.33% decrease). Meanwhile, the quantization process produces a smaller file size than the original (35.62% decrease).

Referring to 0, the PSNR value of the proposed DCNN is greater than the other methods. In general, the SSIM values for all methods were almost the same. Although the SSIM value of the Fractal method is greater than the proposed DCNN (it is in line with the fractal method's ΔQ_{dct} value, which is greater than the proposed method), the PSNR value is less than the required PSNR value for 8-bit images (between 30 and 50 dB). It proves that the proposed DCNN is still much better than other methods. It is also proven by the ΔQ_{dct} value of the proposed DCNN, which is smaller than other methods.

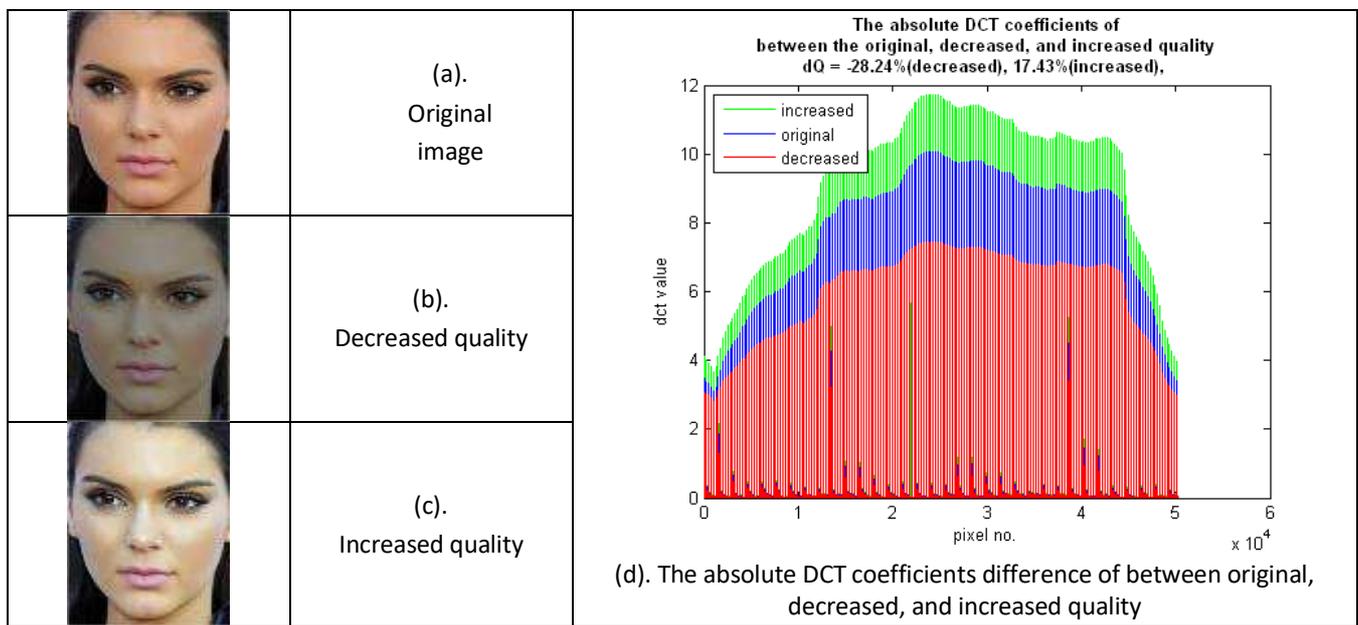
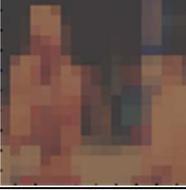


Fig. 5 The illustration of image quality change by DCT coefficients difference ratio

	<p>Original image ($640 \times 360 \times 3$) correlation coefficient between pixels = 0.5124 size = 18.251 bytes</p>	
		
Down-sampled image ($100 \times 100 \times 3$)	Feature image ($96 \times 96 \times 3$)	Sub-feature image ($19 \times 19 \times 3$)

Up-sampled feature image (640 × 360 × 3)	Up-sampled sub-feature image (640 × 360 × 3)	Spatial redundancy elimination correlation coefficient between pixels = 0.2750
	Quantization result as compressed image size = 11.750 bytes	

Fig. 6 The illustration of each stage's results using 200th frame

Original frame		
Proposed DCNN <i>MSE</i> = 11.69, <i>PSNR</i> = 39.67dB <i>SSIM</i> = 0.9991	DCT <i>MSE</i> = 14.48 <i>PSNR</i> = 36.64 dB <i>SSIM</i> = 0.9988	Fractal <i>MSE</i> = 165.19 <i>PSNR</i> = 25.88 dB <i>SSIM</i> = 0.9999
$\Delta Q_{dct} = -8.44$	$\Delta Q_{dct} = -14.85$	$\Delta Q_{dct} = -33.51$

Fig. 7 The comparison of all methods for the 200th frame

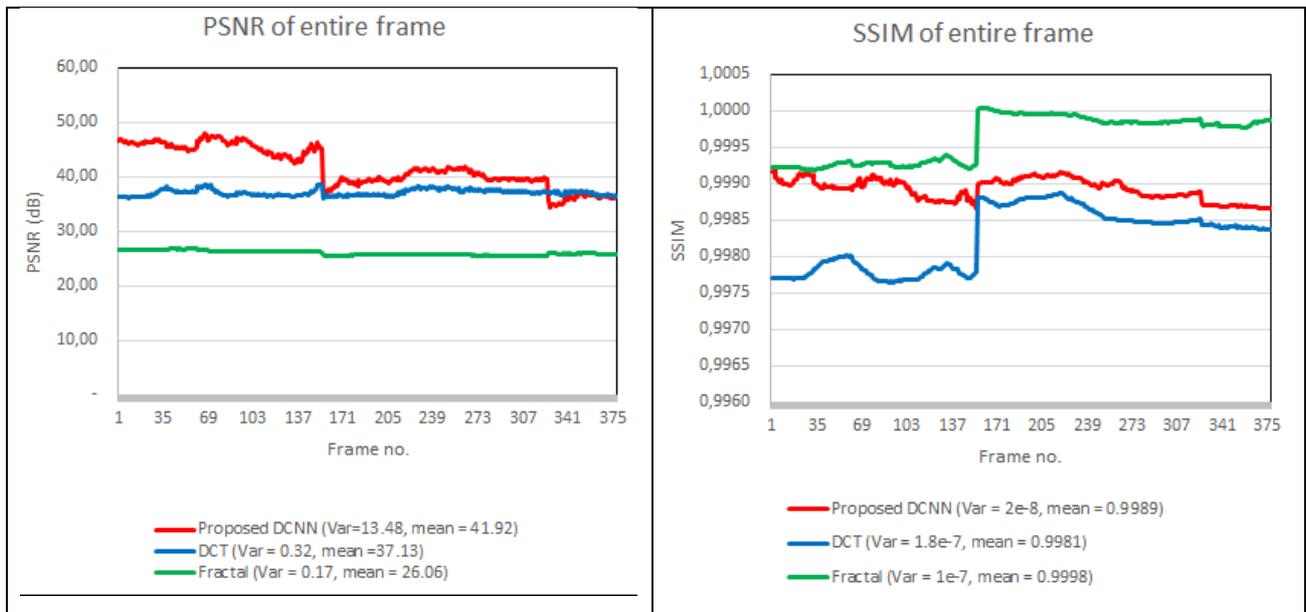


Fig. 8 PSNR and SSIM of the entire frame

Referring to 0, the mean PSNR value of the proposed method is greater than that of other methods. It means that the proposed method produces better compression quality than other methods. Also, the variance value of the proposed method's PSNR is greater than the other methods. It showed the adaptability of the proposed method to improve the quality

of the compression results better than other methods. It was in line with the variance value of the proposed method's SSIM, which is smaller than the other methods.

A summary of the performance comparison of all methods for the entire dataset is shown in TABLE I. and 0 From both tables, it was obtained AAPR = 147.71%.

TABLE II
THE *SS* AND *CR* COMPARISON OF ALL METHODS

No.	Video clip dataset	<i>SS</i> (%)			<i>CR</i> (%)		
		<i>DCNN</i>	<i>DCT</i>	Fractal	<i>DCNN</i>	<i>DCT</i>	Fractal
1	MMA Best Kick 1	37.08	15.30	31.61	158.92	118.06	146.21
2	MMA Best Kick 2	30.30	10.51	26.67	143.54	111.74	136.36
3	MMA Best Kick 3	31.84	11.76	27.40	146.72	113.33	137.74
4	MMA Best Kick 4	32.53	12.03	27.90	148.22	113.67	138.69
5	MMA Best Kick 5	31.42	14.11	27.84	145.81	116.42	138.57
6	MMA Elbow KO 1	26.14	10.63	26.44	135.38	111.89	135.94
7	MMA Elbow KO 2	26.75	10.52	25.02	136.52	111.76	133.37
8	MMA Elbow KO 3	23.18	8.81	24.77	130.18	109.66	132.93
9	MMA Elbow KO 4	31.41	10.8	27.30	145.79	112.11	137.55
10	MMA Elbow KO 5	23.64	8.44	22.90	130.96	109.22	129.70
Mean		29.43	11.29	26.79	142.20	112.79	136.71
<i>APR</i>			260,67	109,88		126,08	104,02

TABLE III
THE *PSNR*, *SSIM* AND ΔQ_{dct} COMPARISON OF ALL METHODS

No.	Video clip dataset	<i>PSNR</i> (dB)			<i>SSIM</i>			ΔQ_{dct}		
		<i>DCNN</i>	<i>DCT</i>	Fractal	<i>DCNN</i>	<i>DCT</i>	Fractal	<i>DCNN</i>	<i>DCT</i>	Fractal
1	MMA Best Kick 1	37.54	35.04	26.17	0.9989	0.9977	0.9999	-5.17	-5.60	-28.97
2	MMA Best Kick 2	38.24	36.35	26.02	0.9985	0.9979	0.9997	-9.19	-8.76	-33.42
3	MMA Best Kick 3	37.27	35.24	25.70	0.9987	0.9978	0.9999	-1.56	-5.51	-28.62
4	MMA Best Kick 4	37.05	35.12	25.68	0.9986	0.9977	0.9999	-4.09	-6.07	-30.38
5	MMA Best Kick 5	41.60	36.34	25.95	0.9992	0.9980	0.9997	-1.11	-5.23	-51.80
6	MMA Elbow KO 1	27.28	36.69	25.76	0.9988	0.9982	0.9998	-21.18	-25.18	-29.98
7	MMA Elbow KO 2	41.92	37.13	26.06	0.9991	0.9983	0.9996	-11.35	-12.40	-28.90
8	MMA Elbow KO 3	36.55	37.29	26.18	0.9990	0.9986	0.9999	-19.43	-12.34	-30.98
9	MMA Elbow KO 4	36.68	35.10	25.54	0.9989	0.9983	0.9999	-17.23	-12.02	-29.85
10	MMA Elbow KO 5	37.84	36.87	26.13	0.9992	0.9984	0.9997	-9.25	-12.43	-29.58
Mean		37.20	36.12	25.92	0.9989	0.9981	0.9998	-9.96	-10.55	-32.25
<i>APR</i>			102,99	143,51		100,08	99,91		106,01	323,91

IV. CONCLUSION

This study has proposed a DCNN with three hidden layers for intra-frame-based video compression. DCT and fractal methods were used to compare the performance of the proposed method. The reference frame, the average of all frames, is used as the training input image after the down-sampling process. The training image is divided into several square blocks using the square blocks shift operation until all parts of the image are fulfilled. All pixels in each block act as an input data pattern. The number of square blocks of the training image is the number of training data for the proposed DCNN.

The trained proposed DCNN was then used to construct the feature and sub-feature image obtained through the max function operation in the feature bank and sub-feature bank. Minimizing spatial redundancy and quantizing the original image uses feature images and sub-features to produce a compressed image. Experiments on the entire dataset resulted in an AAPR (Average Approximate Performance Ratio) of 147.71%. For further studies, the performance improvement of the proposed DCNN is performed by modifying its structure so that the output is direct in the form of feature and sub-feature images. Another way is to combine it with the DCT or fractal method to improve the performance of the result.

ACKNOWLEDGMENT

The authors are grateful to the Applied Modern Computing & Robotic Systems Unit Politeknik Negeri Samarinda, East Kalimantan, Indonesia

CONFLICT OF INTEREST

The authors declare no conflict of interest

AUTHOR CONTRIBUTION

Arief Bramanto WP and Achmad FO Gaffar: Supervision, conceptualization, writing-original, writing-review and formal analysis. Muhammad Taufiq Sumadi and Lisa Setiawati: draft preparation, coding, writing review. data acquisition and validation.

REFERENCES

- [1] A. Punchihewa, "Video Compression: Challenges and Opportunities," in *Project - 22 - Image and Video Coding and Compression*. vol. 2019, ed, 2019, pp. 24-28.
- [2] R. E. Childers and U. o. C. A. D. o. C. Science, *A Study of Rate Control for H.265/HEVC Video Compression*: University of Central Arkansas, Department of Computer Science, 2020.
- [3] A. B. W. Putra, A. F. O. Gaffar, A. Wajiansyah, and I. H. Qasim, "Feature-Based Video Frame Compression Using Adaptive Fuzzy Inference System," in *2018 International Symposium on Advanced Intelligent Informatics (SAIN)*, 2018, pp. 49-55.
- [4] ISO/EIC, "Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 3: Audio," vol. ISO/EIC 11172-3:1993 ICS : 35.040.40 ed: Technical Committee : ISO/IEC JTC 1/SC 29, 1993, p. 150.
- [5] ISO/IEC, "Information technology — Generic coding of moving pictures and associated audio information — Part 2: Video," vol. ISO/IEC 13818-2:2013 ICS : 35.040.40, ed: Technical Committee : ISO/IEC JTC 1/SC 29, 2013, p. 225.
- [6] ISO/IEC, "Information technology — Coding of audio-visual objects — Part 10: Advanced video coding," vol. ISO/IEC 14496-10:2020

- ICS : 35.040.40, ed: Technical Committee : ISO/IEC JTC 1/SC 29, 2020, p. 859.
- [7] ISO/IEC, "Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding," vol. ISO/IEC 23008-2:2020 ICS : 35.040.40, ed: Technical Committee : ISO/IEC JTC 1/SC 29, 2020, p. 889.
- [8] S. Akramullah, *Digital video concepts, methods, and metrics: quality, compression, performance, and power trade-off analysis*: Springer Nature, 2014.
- [9] A. Djelouah, J. Campos, S. Schaub-Meyer, and C. Schroers, "Neural inter-frame compression for video coding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6421-6429.
- [10] E.B.Tashmanov and R. A. Raxmonberdiyev, "The Inter Frame Image Processing In a Video Codec Based On The Wavelet Transformation," *IJRET: International Journal of Research in Engineering and Technology*, vol. 06, 2017.
- [11] M. Z. Islam, M. E. H. Eimon, B. Ahmed, and M. A. M. Hasan, "Classification Based Inter-Frame Prediction in Video Compression," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, 2019, pp. 404-408.
- [12] L. Sinapayan and T. Ikegami, "Video Compression with a Predictive Neural Network," in *The 31st Annual Conference of the Japanese Society for Artificial Intelligence*, Tokyo, 2017, pp. 3M22-3M22.
- [13] 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.
- [14] O. Rippel, S. Nair, C. Lew, S. Branson, A. G. Anderson, and L. Bourdev, "Learned video compression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3454-3463.
- [15] F. Brand, J. Seiler, and A. Kaup, "Intra frame prediction for video coding using a conditional autoencoder approach," in *2019 Picture Coding Symposium (PCS)*, Ningbo, China, 2019, pp. 1-5.
- [16] N. Manjanaik, B. Parameshachari, S. 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," in *IOP Conference Series: Materials Science and Engineering*, 2017, p. 012209.
- [17] K. S. Reddy, B. Srikanth, and C. L. Reddy, "Design and Analysis of Video Compression Technique using HEVC Intra-frame Coding," *IJESRT (International Journal of Engineering Sciences & Research Technology)*, vol. 06, pp. 477-482, 2017.
- [18] B. Li, J. Han, and Y. Xu, "Co-located Reference Frame Interpolation Using Optical Flow Estimation for Video Compression," in *2018 Data Compression Conference*, Snowbird, UT, USA, 2018, pp. 13-22.
- [19] 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*, 2013, pp. 1831-1835.
- [20] F. Kamisli, "Block-based spatial prediction and transforms based on 2D Markov processes for image and video compression," *IEEE Transactions on Image Processing*, vol. 24, pp. 1247-1260, 2015.
- [21] P. K. Charles and K. Habibulla Khan, "A novel search technique of motion estimation for video compression," *Global Journal of Computer Science and Technology*, vol. 17, pp. 1-5, 2017.
- [22] 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)*, 2015, pp. 1-4.
- [23] J. Lin, D. Liu, H. Li, and F. Wu, "M-LVC: Multiple frames prediction for learned video compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3546-3554.
- [24] A. Jacob, V. Pawar, V. Vishwakarma, and A. Mane, "Deep Learning Approach to Video Compression," in *2019 IEEE Bombay Section Signature Conference (IBSSC)*, 2019, pp. 1-5.
- [25] R. Birman, Y. Segal, and O. Hadar, "Overview of research in the field of video compression using deep neural networks," *Multimedia Tools and Applications*, vol. 79, pp. 11699-11722, 2020.
- [26] P. Dhungel, P. Tandan, S. Bhusal, S. Neupane, and S. Shakya, "Video Compression for Surveillance Application using Deep Neural Network," *Journal of Artificial Intelligence and Capsule Networks*, vol. 2, pp. 131-145, 2020.
- [27] J. Lee, K. Kong, G. Bae, and W.-J. Song, "BlockNet: A deep neural network for block-based motion estimation using representative matching," *Symmetry*, vol. 12, p. 840, 2020.

- [28] R. Society and R. S. Staff, *Machine Learning: The Power and Promise of Computers That Learn by Example*. Great Britain: Royal Society, 2017.
- [29] S. Ginanjar, A. Wibowo, and E. Sarwoko, "The best architecture selection with deep neural network (DNN) method for breast cancer classification using MicroRNA data," in *Journal of Physics: Conference Series, The 9th International Seminar on New Paradigm and Innovation of Natural Sciences and its Application*, Central Java, 2020, p. 012106.
- [30] S. Emmot, "Characterizing Video Compression Using Convolutional Neural Networks," Independent thesis Advanced level (professional degree), Computer Science and Engineering, master's level, Luleå University of Technology, Sweden, 2020.
- [31] X. Lei, H. Pan, and X. Huang, "A dilated CNN model for image classification," *IEEE Access*, vol. 7, pp. 124087-124095, 2019.
- [32] N. Krishnaraj, M. Elhoseny, M. Thenmozhi, M. M. Selim, and K. Shankar, "Deep learning model for real-time image compression in Internet of Underwater Things (IoUT)," *Journal of Real-Time Image Processing*, vol. 17, pp. 2097-2111, 2020.
- [33] D. Im, D. Han, S. Choi, S. Kang, and H.-J. Yoo, "DT-CNN: Dilated and transposed convolution neural network accelerator for real-time image segmentation on mobile devices," in *2019 IEEE international symposium on circuits and systems (ISCAS)*, 2019, pp. 1-5.
- [34] J. H. Park, J. H. Kim, and S. I. Cho, "The analysis of CNN structure for image denoising," in *2018 International SoC Design Conference (ISOCC)*, 2018, pp. 220-221.
- [35] Z. Chen, Y. Li, F. Liu, Z. Liu, X. Pan, W. Sun, Y. Wang, Y. Zhou, H. Zhu, and S. Liu, "CNN-optimized image compression with uncertainty based resource allocation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2559-2562.
- [36] J. Yang and J. Li, "Application of deep convolution neural network," in *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2017, pp. 229-232.
- [37] L. Cavigelli, P. Hager, and L. Benini, "CAS-CNN: A deep convolutional neural network for image compression artifact suppression," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 752-759.
- [38] P. Kapoor and S. Patyal, "DCT Image Compression for Color Images," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, pp. 3247-3252, 2014.
- [39] F. Alfiah, A. Setiadi, Saepudin, A. Supriadi, and I. Maulana, "DCT Methods on Compression RGB and Grayscale image," *International Journal of Computer Technique*, vol. 04, pp. 24-29, 2017.
- [40] X. Zhou, Y. Bai, and C. Wang, "Image compression based on discrete cosine transform and multistage vector quantization," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, pp. 347-356, 2015.
- [41] D. Sandhya and V. Rathod, "Fractal based image compression techniques," *Int J Comput Appl*, vol. 178, pp. 11-18, 2017.
- [42] N. Buduma and N. Locascio, *Fundamentals of Deep Learning: Designing Next-generation Machine Intelligence Algorithms* vol. 1st Edition. USA: O'Reilly Media, 2017.
- [43] Y. Li, F. Qi, and Y. Wan, "Improvements on bicubic image interpolation," in *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2019, pp. 1316-1320.
- [44] O. Ieremeiev, V. Lukin, K. Okarma, and K. Egiazarian, "Full-reference quality metric based on neural network to assess the visual quality of remote sensing images," *Remote Sensing*, vol. 12, p. 2349, 2020.
- [45] O. F. Mohammad, M. S. M. Rahim, S. R. M. Zeebaree, and F. Ahmed, "A survey and analysis of the image encryption methods," *International Journal of Applied Engineering Research*, vol. 12, pp. 13265-13280, 2017.