Science in Information Technology Letters
Vol. 1., No. 1, May 2020, pp. 9-16
ISSN 2722-4139
http://pubs2.ascee.org/index.php/sitech

Modification of a gray-level dynamic range based on a number of binary bit representation for image compression

Arief Bramanto Wicaksono Putra [a,1,*], Supriadi [a,2], Aji Prasetya Wibawa [b,3], Andri Pranolo [c,4], Achmad Fanany Onnilita Gaffar [a,5]

[a] Department of Information Technology, Politeknik Negeri Samarinda, East Kalimantan, Indonesia
[b] Department of Electrical Engineering, State University of Malang, Indonesia
[c] Informatics Department, Universitas Ahmad Dahlan, Indonesia
[1] ariefbram@gmail.com; [2] supriadi.polnes@gmail.com; [3] aji.prasetya.ft@um.ac.id.com; [4] andri.pranolo@tif.uad.ac.id;
[5] onnygaffar212@gmail.com
* Corresponding Author

ARTICLE INFO

Article history
Received May 3, 2019
Revised June 11, 2019
Accepted November 29, 2019

Keywords
Gray level dynamic range
Image compression
Binary bit
Performance

ABSTRACT

The unique features of an image can be obtained by changing the gray level by modifying the dynamic range of the gray level. The gray-level dynamic range modification technique is one technique to minimize the selected features. Bit rate reduction uses coding information with fewer bits than the original image (image compression). This study using the dynamic level of the gray level of a modified image with the concept of binary bit representation or also called bit manipulation. Using some binary bit representation options used: 4, 5, 6, and 7 of bit can obtain the best compression performance. Measurement of compression ratio and decompression error ratio to a benchmark comparison called compression performance, which is the ultimate achievement of this study. The results of this study show the use of 6-bit binary representation has the best performance, and the resulting image compression does not resize the resolution of the original image only visually looks different.

1. Introduction

A grayscale image is a two-dimensional matrix with image pixel intensity values. For efficiency or transmission, data will be compressed using particular techniques. Mapping the gray level range from the original image to the new image in the output will use the fuzzy function with the MF trapezoid parameter setting technique. The image features obtained are more uniquely, as the new dynamic gray level range minimizes features other than the selected features. Data compression is a reduction in the bit rate, which involves coding information from the original data with fewer bit values. Data compression is divided into two, namely lossy and lossless. Lossless compression is used to identify and remove statistical redundancy. This composition does not omit information. As for lossy compression, unnecessary or less important information will be deleted [1], [2]. This process in data transmission is called source coding. The coding process uses a data source that is before it is saved or sent. Source coding

http://dx.doi.org/10.31763/sitech.v1i1.17

must be distinguished from channel coding for error detection and correction or coding of related lines by mapping data to a signal. In relation to grayscale images, compression is advantageous in many ways such as the cost of transferring large images through low bandwidth channels is reduced, increased transmission efficiency, the space required to store the image is reduced, and compressed images cannot be accessed by unexpected parties and hence provide some level of security [3].

Based on the lossless or lossy compression, there are many methods for the data compression process. Arithmetic coding (AC) is one of the most efficient lossless data compression techniques, and the entropy value approaches the encoded data. Arithmetic coding is different from another coding (Huffman coding) because it encodes data (data string) by making a string of code that represents a fractional value. This technique will also carry out coding error control [4]. Its application is a two-level image compression method based on chain codes and entropy coders [5]. An example of a lossy compression device is that the Markov random field is processed with two-level image compression [6]. The main goal is to estimate the sequence of dependencies between the chain code symbols before entropy coding. For example, the selected order value is low, and the compression results do not meet the minimum requirements. However, if the order selection is high, this can increase the cost and complexity of the encoder, which results in a slight increase in compression due to compression levels. This process occurs if the order value is consistent.

Additionally, block-based lossless image compression is processed with a combination of finite mix models and adaptive arithmetic coding [7]. This method distributes the probabilities of each modeling block by a mixture of nonparametric distributions, which utilize a high correlation between neighboring blocks. Integer Wavelet Transform (IWT) can process lossless rather than the Discrete Wavelet Transform (DWT) [8]. IWT calculations are faster than traditional DWT, and IWT does not require short memory because the computation of transformations is comprehensive. Besides, IWTT has an integer value that can measure low complexity compared to DWT, which has the results of floating-point numbers and is fully reversible with zero practical losses.

In addition to Arithmetic coding (AC), which also belongs to lossless compression techniques are run-length encoding, entropy and area coding, Huffman encoding [2], [9]–[11]. Included in the lossy coding techniques is predictive coding and transform coding such as Fourier Transform (FT) [12], Discrete Cosines Transform (DCT) [13], Wavelet Transform (IWT, DWT) [14], [15]. In the concept of modeling, more and more implementation of predictive coding techniques in image compression. It is because the advantages of this technique are simple, symmetrical encode/decoder, and flexible. Predictive coding is more focused on the design of mathematical models in the prediction or estimation of each nearest or neighboring pixel value and finding the difference from the expected value and the actual value called residual error or prediction. Polynomial linear-based compression is one of the predictive coding techniques. This technique is based on the distance between the image pixel and the center with the linearization base or first-order Taylor series [16]. The progressive Bit Plane Encoder (BPE) that follows the combined two-dimensional DWT can compress the encoded bitstream [17]. JPEG is the most popular technique in image compression, with a lossy and lossless image compression section. Standard JPEG standard mode is the most popular lossy compression technique than the basic, progressive, and hierarchical method of loss. Although the compression in the original JPEG standard has excellent performance, it needs to be improved. A modified baseline JPEG compression has been studied in [18]. DCT has been improved using the All Phase Bi-orthogonal Transform (APBT) for video compression in [19].

Digital video is a digital image arranged and displayed sequentially in a frame. Codec (encoder-decoder) is very necessary for video coding, both compression, and decompression. The compression of video data uses an encoder, while the signal decompression uses a decoder. Video compression aims to reduce storage or reduce the transmission bandwidth of video clip content [20]. Digital images taken using continuous samples and tone tones require very large storage. For example, a 24-bit color image with 1220 × 1318 pixels requires 925 Kbyte storage.

Compression can reduce the amount of data in digital image representation. The basis of the reduction process is the elimination of redundant data. From a mathematical point of view, this number converts two-dimensional pixel sequences into uncorrelated statistical data sets. There are three types of redundancy in compressed image file sizes, namely encoding redundancy, inter-pixel redundancy, psycho-visual redundancy. The redundancy coding condition is fewer bits to represent the symbol that often occurs. Inter-pixel redundancy has a value that is almost the same as neighboring pixels. Psycho-visual redundancy has the concept of a human visual system that cannot simultaneously distinguish all colors.

In general, coding redundancy is presented when code assigned to a gray level does not usefully gray level probabilities. Therefore it almost always arises when gray level images are represented by binary code. Commonly, the data type of stored image intensity value is an 8-bit unsigned integer. In this case, the pixel intensity value is in the range {0 ... 255}. In this study, the gray-level dynamic range of an image is modified using the concept of binary bit representation. Some binary bit representation options are used to lower the number of binary bit of the original image. The image of the bit number manipulation result is a compressed image. This study aims to obtain the best compression performance from some binary bit representation options used.

2. Method

In this section, brief information on the gray-level dynamic range modification and the binary bit representation for image compression are presented.

2.1. The gray-level dynamic range modification

The gray-level dynamic range of an image can be modified using a particular function. Mathematically expressed by (1):

$$g(x,y) = f\big(I(x,y)\big) \tag{1}$$

It $I(x,y)$ is in the original space then $f()$ transform the original image to the certain new space as $g(x,y)$ defined by $f()$. Commonly, the transformation process is done on the double/float data type of the original image. This is done to expand the range of possibilities in which the transformation process is performed. For image compression, the compression technique used should be capable of producing decompression images that have tiny errors to the original image. That is very difficult on double/float data types of the original image, although there have been many compression techniques that focus on this area. One practical way is compression done on the original data type to reduce errors that arise due to data conversion process. However, this possible way also raises significant decompression errors. Therefore, it is necessary to select the binary bit representation that has the least impact on decompression errors.

2.2. Binary bit representation

Images whose pixel intensity values are represented by using 8-bit binaries have a range of values in {0 ... 255}. The number of binary bits used to represent each pixel intensity value determines the amount of data storage capacity. If any pixel intensity value can be represented by a small number of binary bits then it will reduce its data storage capacity. If using a 4-bit binary representation then the maximum decimal is 1111 = 15. The gray-level dynamic range is transformed from the range {0 ... 255} to {0 ... 15}, similarly with 5-bit, 6-bit, and 7-bit. Mathematically expressed by (2).

$$g(x,y) = uint\left(\big(double\big(I(x,y)\big)/255\big) * B\right) \tag{2}$$

B is maximum decimal of each binary bit representation. For RGB images, (2) applied to each component R, G, and B then the result is rearranged into a compressed image. Simply, the decompression process is done by applying the following (3) to the compressed image.

$$I_d(x,y) = uint\left(\left(double\left(g(x,y)\right)/B\right) * 255\right) \tag{3}$$

Equations (1) and (2) applied to the integer data type of an image. The existence of arithmetic division requires the conversion of data to a double/float type to satisfy mathematical requirements. This is done by a function named *double*. This is different from the image data type conversion. Suppose an 8-bit unsigned integer image will occupy within the range {0 ... 255}. It will fall within the range {0 ... 1} when it converted into double/float data type. In this case, the conversion is done on the data range, not on each data, while (2) and (3) apply to each information. To remain of integer type, the calculation result needs to be converted back to integer type by using a function named *uint* (unsigned integer).

The fewer number of binary bits used the more significant the compression ratio. The problem will appear when the calculation result is floating value. Suppose $I(x,y) = 77$. Using (2) with B = 15 (4-bit binary) then obtained $g(x,y) = 4.53 \approx 5$. Using (3) then got $I_d(x,y) = 85$. The absolute error generated is 10.38%. If using 5-bit binary (max. decimal 11111=31) then received $g(x,y) = 9.36 \approx 9$. Using (3) then obtained $I_d(x,y) = 74$. The absolute error made is 3.9%. This motivates the use of some binary bit representation options for best compression performance.

2.3. Performance measurement

The compression ratio is considered from the aspect of data size before and after compression. Mathematically expressed by (4).

$$R = \left(1 - \frac{filesize\left(g(x,y)\right)}{filesize\left(I(x,y)\right)}\right) \times 100\% \tag{4}$$

The decompression error stated by *PER* (Peak to Error Ratio) which mathematically expressed by (5).

$$PER = \left(\frac{MAE}{255}\right) \times 100\% \tag{5}$$

MAE (Mean Absolute Error) expressed by (6).

$$MAE = \frac{1}{M \times N \times 3} \sum_{d=1}^{3} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| I(i,j,d) - I_d(i,i,d) \right| \tag{6}$$

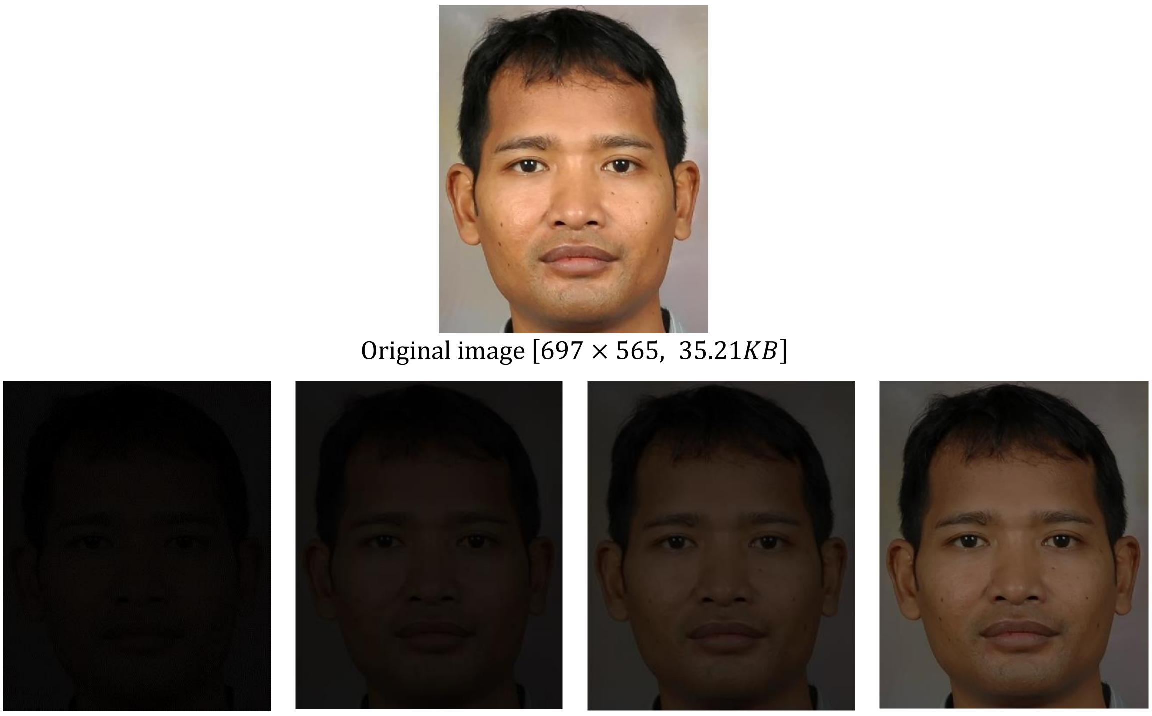
The index constant $M \times N$ is the size of an image.

In this study, compression performance defined as the ratio of decompression error to its compression ratio can be seen in (7).

$$P = \left(1 - \frac{PER}{R}\right) \times 100\% \tag{7}$$

3. Results and Discussion

In this study, the representation of 4-bit, 5-bit, 6-bit, and 7-bit binary bits are used to compress some of the test images used. The result of the compression of the original image 1 by using those binary bit representations is shown in Fig. 1. The decompression results are shown in Fig. 2.



Original image [$697 \times 565,\ 35.21KB$]

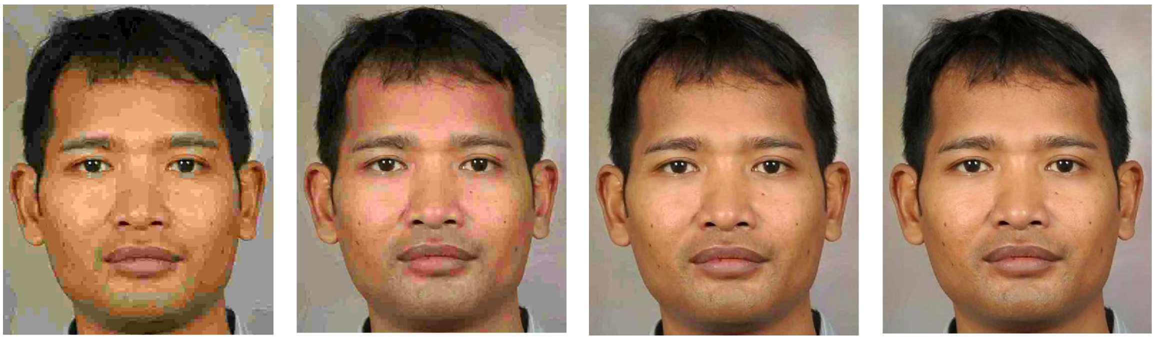
4-bit compressed image [$9.05KB, 74.30\%$]

5-bit compressed image [$11.54KB, 67.24\%$]

6-bit compressed image [$16.53KB, 53.05\%$]

7-bit compressed image [$27.33KB, 22.38\%$]

Fig. 1. The original image 1 and its various compressed results



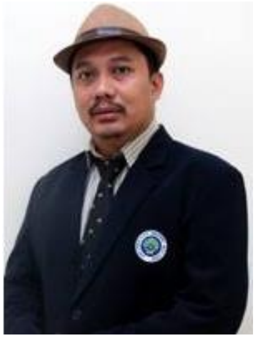
4-bit decompressed image [$PER = 3.80\%$]

5-bit decompressed image [$PER = 2.53\%$]

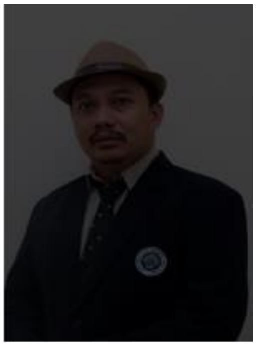
6-bit decompressed image [$PER = 1.38\%$]

7-bit decompressed image [$PER = 0.90\%$]

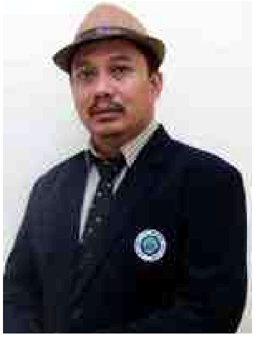
Fig. 2. The results of various decompressed images



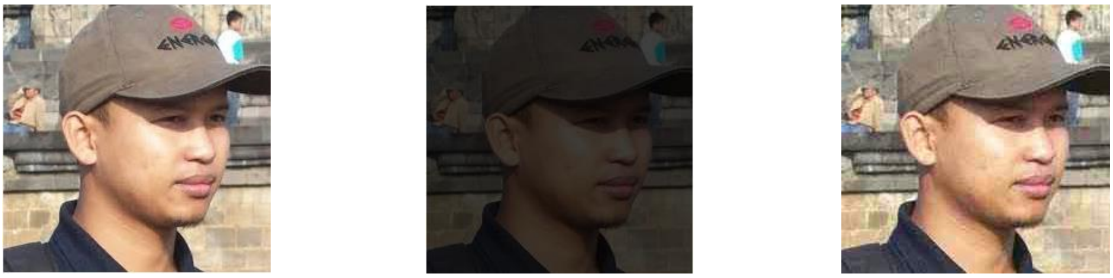
Original image [$225 \times 150,\ 4.74KB$]



compressed image [$2.68KB,\ 43.50\%$]



decompressed image [$PER = 1.54\%$]



Original image $[225 \times 225, 9.46KB]$ compressed image $[5.22KB,\ 45.00\%]$ decompressed image $[PER = 2.09\%]$

Fig. 3. The results of various compressed & decompressed images using 6-bit binary representation

The results of various compressed & decompressed images using 6-bit binary representation are shown in Fig. 3 which details are shown in Table 1. Although compression using 4-bit binary representation has the most significant compression ratio but it also produces the biggest decompression errors anyway. Similarly, compression uses a 7-bit binary representation. Although having the smallest decompression errors but also yielding the lowest compression ratio. By using (7), the results show that image compression using 6-bit binary representation has the best performance.

Also, the image compression by using a particular binary bit representation less than it's original does not change the size of the original image, unless it only affects the visual quality. The fewer number of binary bits used will visually be darker, and vice versa. Because the manipulation does not involve the aspect of pixel neighbors, the compression results still have the same pattern as the original so it can be said to be classified as lossless compression. This technique also does not consider the security aspects so that the compression results are still possible to be accessed by all parties.

Table 1. Details of the compression and decompression of some original images using 4-bit, 5-bit, 6-bit, and 7-bit binary representations

No.	Size of original image	4-bit binary representation		5-bit binary representation		6-bit binary representation		7-bit binary representation	
		R	***PER***	***R***	***PER***	***R***	***PER***	***R***	***PER***
1.	697×565, $35.21KB$	74.30%	3.80%	67.24%	2.53%	53.05%	1.38%	22.38%	0.90%
2.	248×189, $8.34KB$	74.33%	5.84%	63.98%	3.64%	46.77%	2.51%	23.72%	1.71%
3.	225×150, $4.74KB$	64.75%	3.48%	57.04%	2.32%	44.75%	1.56%	20.49%	1.14%
4.	225×225, $9.46KB$	74.04%	5.18%	63.68%	3.39%	45.00%	2.15%	23.32%	1.14%
5.	247×208, $8.31KB$	73.87%	6.00%	64.07%	3.70%	45.90%	2.37%	22.24%	1.58%
average		72.26%	4.86%	63.20%	3.12%	47.09%	1.99%	22.43%	1.29%
performance (***P***)		**93.27%**		**95.07%**		**95.77%**		**94.23%**	

4. Conclusion

This study has applied the concept of image compression by using fewer binary bit representation options compared to the original. The results show that the use of 6-bit binary representation has the best performance. Image compression results show that the size is the same as the original image; the difference is visual. The compression image still has the same pattern as the original because it does not consider the neighbor aspect of its pixel. Techniques that have been used also do not consider the security aspects so that the compression image is still possible to be accessed by all parties.

To improvements in compression results with the compression ratio as large as possible and compression errors as small as possible and also consider the security of compression results will be the future work.

References

[1] 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, pp. 6–14, Mar. 2018, doi: 10.24014/ijaidm.v1i1.4599.

[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, no. 1, pp. 18–23, 2010, available at: Google Scholar.

[3] A. Kulkarni and A. Junnarkar, “Gray-Scale Image Compression Techniques: A Review,” *Int. J. Comput. Appl.*, vol. 131, no. 13, pp. 22–25, Dec. 2015, doi: 10.5120/ijca2015907519.

[4] A. Habib and D. Chowdhury, “An Efficient Compression Technique Using Arithmetic Coding,” *J. Sci. Res. Reports*, vol. 4, no. 1, pp. 60–67, Jan. 2015, doi: 10.9734/JSRR/2015/12846.

[5] S. Alcaraz-Corona and R. M. Rodriguez-Dagnino, “Bi-Level Image Compression Estimating the Markov Order of Dependencies,” *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 3, pp. 605–611, Jun. 2010, doi: 10.1109/JSTSP.2010.2048232.

[6] M. G. Reyes, X. Zhao, D. L. Neuhoff, and T. N. Pappas, “Lossy Compression of Bilevel Images Based on Markov Random Fields,” in *2007 IEEE International Conference on Image Processing*, 2007, pp. II-373-II–376, doi: 10.1109/ICIP.2007.4379170.

[7] 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, 2015, doi: 10.1007/s11042-014-2195-8.

[8] Shivaputra, S. H.S, and L. V, “An Efficient Lossless Medical Image Compression Technique for Telemedicine Applications,” *Comput. Appl. An Int. J.*, vol. 2, no. 1, pp. 63–69, Feb. 2015, doi: 10.5121/caij.2015.2106.

[9] V. G. Dubey and J. Singh, “Medical Image Compression and Decompression Using Huffman Encoding Technique,” *IJAIR*, pp. 335–338, 2012, available at: Google Scholar.

[10] S. Kumari, S. Khanna, and Taqdir, “Comprehensive Study of the Work Done in Image Processing and Compression Techniques for Redundancy,” *Int. Res. J. Eng. Technol.*, vol. 03, no. 01, pp. 855–860, 2016.

[11] P. Kavitha, “A Survey on Lossless and Lossy Data Compression Methods,” *Int. J. Comput. Sci. Eng. Technol.*, vol. 7, no. 03, pp. 110–114, 2016, available at: Google Scholar.

[12] S. S. Pandey, M. P. Singh, and V. Pandey, “Image Transformation and Compression using Fourier Transformation,” *Int. J. Curr. Eng. Technol.*, vol. 5, no. 2, pp. 1178–1182, 2015, available at: Google Scholar.

[13] X. Zhou, Y. Bai, and C. Wang, “Image Compression Based on Discrete Cosine Transform and Multistage Vector Quantization,” *Int. J. Multimed. Ubiquitous Eng.*, vol. 10, no. 6, pp. 347–356, Jun. 2015, doi: 10.14257/ijmue.2015.10.6.33, available at: Google Scholar.

[14] Z. Abidin and A. Alamsyah, “Wavelet based approach for facial expression recognition,” *Int. J. Adv. Intell. Informatics*, vol. 1, no. 1, pp. 7–14, Mar. 2015, doi: 10.26555/ijain.v1i1.7.

[15] S. Gomathi and T. Santhanam, “Performance Analysis of Haar Wavelet Transform and Huffman Coding Compression Techniques for Human Object,” *Int. J. Comput. Intell. Informatics*, vol. 6, no. 3, pp. 233–239, 2016, available at: https://www.periyaruniversity.ac.in/ijcii/issue/Vol6No3Dec2016/D6_PID0398.pdf.

[16] L. E. George and G. Al-Khafaji, “Image compression based on non-linear polynomial prediction

model," *Int. J. Comput. Sci. Mob. Comput.*, vol. 4, no. 8, pp. 91–97, 2015, available at: Google Scholar.

[17] M. Ammar and Y. Saleem, "Implementation of CCSDS Image Data Compression Standard on DSP Platform," *J. Sp. Technol.*, vol. V, no. 1, pp. 91–102, 2015, available at: Google Scholar.

[18] M. K. Islam, M. Moznuzzaman, M. F. Khatun, and R. Yesmin, "A Proposed Modification of Baseline JPEG Standard Image Compression Technique," *Int. J. Sci. Eng. Res.*, vol. 6, no. 8, pp. 180–186, 2015, available at: Google Scholar..

[19] C. Zhang, C. Wang, and B. Jiang, "Video Compression Algorithm Based on Directional All Phase Biorthogonal Transform and H.263," *Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 9, no. 3, pp. 189–198, Mar. 2016, doi: 10.14257/ijsip.2016.9.3.17.

[20] 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)*, Aug. 2018, pp. 49–55, doi: 10.1109/SAIN.2018.8673386.