Developed JPEG Algorithm Applied in Image Compression

Abdullah A. Hussain¹, Ghadah K. AL-Khafaji² and Mohammed M. Siddeq³
Abdullah_amb@yahoo.com, hgkta2012@sc.uobaghdad.edu.iq, mamadmmx76@gmail.com
¹Dept. of Computer Science, College of Science, University of Baghdad, IRAQ
²Dept. of Computer Science, College of Science, University of Baghdad, IRAQ
³Computer Engineering Dept., Technical College/Kirkuk, Northern Technical University, IRAQ

Abstract: JPEG is most popular image compression and encoding, this technique is widely used in many applications (images, videos and 3D animations). Meanwhile, researchers are very interested to develop this massive technique to compress images at higher compression ratios with keeping image quality as much as possible. For this reason in this paper we introduce a developed JPEG based on fast DCT and removed most of zeros and keeps their positions in a transformed block. Additionally, arithmetic coding applied rather than Huffman coding. The results showed up, the proposed developed JPEG algorithm has better image quality than traditional JPEG techniques.

Keywords: JPEG, DCT, DPCM, Reduced matrix, and Split Zero from Non-Zero values.

1. Introduction

Today with immense facilities in communication and electronic devices that implies/such as computers, iPad, and smart phones; along with the ever growing size of multimedia data (i.e., text, audio, video, image) that/which becomes the essential part of our life, but the main obstacle in communication capacity limitation and media storage availability. To overcome this essential problem, the data compression process urgently required [1,2].

Image compression aims to minimizing/reducing the size of image information bytes without degrading the quality of the image to an unacceptable level [3]. Generally, Lossless and Lossy that based on the way upon exploiting image redundancy(s), where in the images compression techniques can be categorize into
two schemes, namely: former one/type of statistical redundancy removal with no information is lost, the compressed image is identical to the original image, while in the later one/type of psychovisual and/or statistical redundancy(s) degradation or loos allowed, the decompressed image will approximate to the original image. The utilization/use of one type over the other limited/determined by the application usage, the lossless exploits for criteria application of medical images, military, space and remote –sensing of low compression performances, on the other hand the lossy used in daily applications, TV, DVD, and video of higher compression achieved [4-7].

Currently a JPEG (Joint Photographic Expert Group) image compression standard technique is being extremely widely used with high performance of compression achieved and preserving quality without visual degradation, which implicitly means quite visually pleasing [8]. Simply/basically, the JPEG is of transform coding techniques that exploits the discrete cosine transform (DCT) which efficiently represent the spatial based image each segment (region) of size (8x8), followed by quantization process and zigzag ordering (from top left to bottom right) then finally/lastly encode the Dc and Ac coefficients [9,10].

The JPEG standard techniques, in spite of its proven effectiveness which paved the way for implementing efficient international standard image compression techniques such as JPEG2000, but still suffers from blocking artifacts, quantization effectiveness and large zigzag representation which overburden with insignificant overhead information [11].

Recently, many of compression techniques like JPEG and JPEG2000 are well recruited for using in deep learning models in order to effective image compression (representation) especially for Internet of Things (IoT) scenarios under the constraint of limited communication resources [12-13]. A large number of works have been introduced to develop / enhanced JPEG performance can be found in [14-19].

This paper is concerned with a developed JPEG to overcome the above mentioned problems to compress natural grayscale images that organized into the following sections, section 2 concerned with the most relevant works, sections 3, 4 and 5 discuss the proposed techniques, experimental results and conclusions respectively.

2. Related works
A vast amount of work had been done to utilize and improve the performance of JPEG standard image compression techniques, here we concerned with the most related works to the above mentioned problems, including Aria (2002)[20] was proposed a novel method for post-processing of JPEG-encoded images, in order to reduce coding artifacts and enhance visual quality, in other words, shift the compressed images in vertical and horizontal directions by \((i, j)\) and re-applies JPEG to the shifted versions of the already-compressed image, and forms an average. By taking various shifts of JPEG, the original block boundaries will be exposed to the frequency shaping of the JPEG encoding process, thus the magnitude of the blockiness will be reduced. Also Chengyou et al. (2007) [21] introduced an improved JPEG compression algorithm based on ‘Haralick sloped-facet model’, where the traditional JPEG based on DCT process the whole image directly, it has not taken into account different areas after segmentation. Here, they are adopted four modes to compress images depends on the results of image segmentation based on Haralick sloped-facet model used. The experiment results show that the quality of reconstructed images is better than the traditional JPEG compression algorithm at the same bit rate. Sukhpal (2012) [22] presented deblocking algorithm in attempt to overcome ‘blocky’ artifacts in JPEG caused by the coarse quantization of DCT coefficients. The algorithm aims to filter the blocked boundaries by employing smoothening, detection of blocked edges and then filtering the difference between the pixels containing the blocked edge. The presented algorithm has been successful in reducing blocky artifacts in an image and therefore increases the subjective as well as objective quality of the reconstructed image. Mahmud et al. (2012) [23] suggested an efficient way to overcome overhead computation of complexity especially in quantization step. The suggested algorithm reduced complexity form \(O(n^2)\) to linear \(O(n)\) by keeping quantization step away from dividing the lower triangular matrix coefficients and simply by fill those coefficients with zero. The filling up of lower triangular can be carried out by a simple AND operation which is much lower costly than the division operation quantize only the upper triangular coefficients and reduce the computational cost thereby. Thus that the overall computational cost of the Baseline JPEG algorithm is reduced with less affecting the obtainable Compression Ratio (CR) and Peak Signal to Noise Ratio (PSNR). Lamia et al. (2017) [24] introduced an improved JPEG compression algorithm by modifying the luminance quantization table for color image. They observed that
compression ratio (CR), and objective fidelity measures of (mean-square-error (MSE)/ peak-signal- to-noise-ratio (PSNR)) values of proposed JPEG is outperform in contrast to conventional JPEG. This is because the quantization table is modified in a way so that only the lowest frequency DCT coefficients have significant amplitude values. Experimental results indicated that highly performance of modified quantization outperforms the standard quantization in terms of CR, MSE, and PSNR. Robby et al. (2017) [25] adopted a Zigzag Scan with Mapping method implemented in electronic circuit. It is able to accelerate the sorting process of DCT-quantized coefficients period because the input data can be immediately located in sequence position which has been determined without any value comparison and repetition process. Mapping process is run between input data sequence and zigzag position sequence. The mapping happens according to the same element position between those two vectors. This implementation result is directed to support a more efficient real-time JPEG compression process. In this case, efficient means faster process and using less circuit resources. Finally/lastly Remi (2018) [26] exploits a simple and efficient method for checking whether the used quantization tables are matched the standard ones or not. The work attempts to determine the exact quality factor of a JPEG image from its quantization tables. The work shows that quantization tables might exploited in steganography with side-information and to detect forgeries or identify image origin and mentioned that quantization tables are very useful in information hiding. Two types of quantization tables, the former one provides a standard way to obtain a quantization tables given a quality factor that is chosen between 1 and 100, the latter is non-standard quantization table where the JPEG norm accepts any quantization step between 1 and 255 as elements of the quantization table. Figure (1) shows an example of quantization matrices for quality factor 75.

\[
\begin{array}{cccccccc}
1 & 1 & 2 & 4 & 4 & 4 & 4 & 4 \\
1 & 1 & 3 & 4 & 4 & 4 & 4 & 4 \\
1 & 1 & 2 & 4 & 4 & 4 & 4 & 4 \\
2 & 3 & 4 & 4 & 4 & 4 & 4 & 4 \\
4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
\end{array}
\quad
\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 1 & 2 & 2 & 3 & 2 \\
1 & 1 & 1 & 2 & 3 & 3 & 3 & 2 \\
1 & 1 & 2 & 3 & 4 & 4 & 3 & 2 \\
1 & 2 & 3 & 4 & 5 & 4 & 3 & 2 \\
3 & 4 & 4 & 4 & 4 & 4 & 4 & 2 \\
\end{array}
\]

Fig. (1): Example of quantization matrices for quality factor 75 that adopted by [26].
3. Proposed Compression System

The proposed system aims to improve the traditional JPEG by overcoming the most/main related/inherited problems, the proposed system clearly illustrated in the following steps and also depicted in figure (2).

**Fig. (2): The Proposed Compression System.**

**Step 1:** Load input uncompressed image \( I \) of square /non-square size (i.e, \( N \times N / M \times N \)) either of gray based or color based, where in case of color based the conversion of averaging process utilized, such as:

\[
I = \frac{(I_{R} + I_{G} + I_{B})}{3} \quad (1)
\]

Where \((I_{R}, I_{G}, I_{B})\) corresponds to the color image bands of Red, Green, and Blue of high spectral redundancy.

**Step 2:** Apply normalization process which is elementary pre-processing step by select/choosing the maximum value of each input image \( I \) for level shift each pixel value in each block in order to convert it into singled integer by subtracting half of the maximum value from each pixel. So the new range of \( I \) is \((-\text{maximum value})\) to \((+\text{maximum value} - 1)\) which is surly more appropriate than traditional range which between -128 to + 127.

**Step 3:** Partitioning the normalized image \( I \) into fixed sized segments of non-overlapping base each of size (8×8), this step is essential due to local similarity or correlation.

**Step 4:** Perform the transform coding techniques of fast DCT base for each segmented blocks to efficiently transform from the spatial domain into frequency domain according the steps below:

**Constant DCT steps in MATLAB**

\[
i = 0; \\
\text{for } j = 0: \text{sizeofblock} - 1
\]
DCTtrans(i+1, j + 1) = sqrt(1 / sizeofblock) ... * cos ((2 * j + 1) * i * pi / (2 * sizeofblock));
end

for i =1: sizeofblock - 1
    for j = 0: sizeofblock - 1
        DCTtrans(i + 1, j + 1) = sqrt(2 / sizeofblock) ...
            * cos ((2 * j + 1) * i * pi / (2 * sizeofblock));
    end
end

Where \( \pi \approx 3.141592 \) and sizeofblock =8, it is represents DCT block 8x8. The few steps above are represents fast DCT used to transform each special domain block to frequency domain block, the output for the above program as shown in Figure (3- a).

Fig. (3-a): “DCTtrans” represents constant Discrete Cosine Transformation (DCT) values.

To illustrate the DCT process, assume we have a matrix size 8x8 (a special domain block), as shown in figure (3 - b). After that the formula in equation (2) applied to the special domain block as represented in figure (3 - b) to produce frequency domain block as represented in figure (3- c).

\[
\text{DCTmatrix} = \text{DCTtrans} \times \text{Special\_Domain} \times \text{DCTtrans}'
\]

(2)

Where “DCTtrans’” represents inverse matrix (inverse matrix DCT block),
Fig. (3-b): “spatial-domains” represents a block contain values from an image.

Fig. (3-c): “DCTmatrix” represents final frequency domain block by equation (2).

**Step 5:** Apply quantization process to eliminates dispensable coefficients by sacrificing the accuracy, and then the amount of image data is compressed, basically the quantization process impels division by transformed segmented block to the quantization matrix (QM) and rounding process, here we introduce a quantization matrices (see figure 4) along with adopting a Quantization Factor (QF) to control the quality of the JPEG image. Quantization factor is vary from range between 0.01 to 3.0 and it is used to increase / decrease image quality where the modified quantization matrix is multiply by specific value of this factor until make compressed image satisfy human’s subjective feelings better.

\[ Q = \text{Round} \left( \frac{\text{DCTmatrix}}{Q_m} \right) \] ................. (3)

\[ Q_m = QM \times QF \] ........................................... (4)

Fig. (4): Quantization matrices adopted

**Step 6:** Use the zigzag scan to order the segmented quantized block of size \( n \times n \) into \( 1 \times n^2 \), that implicitly ordering the values according to its importance, where easily
separate the Dc from Ac’s values, figure (5) shows the zigzag process with practical example

Fig. (5): The zigzag scan with mapping method.

**Step 7:** Applying reduce zigzag matrix size algorithm, that reduce the number of high frequency coefficients by resize the matrix, to obtain high compression gain the following sub-steps are applied:

A) DC coefficients occupied about more than 30% of the storage space in JPEG image [12]. Dc values, which is the first element of each segmented zigzag ordering, these values corresponds to the low frequency values (i.e., mean of each block) of highly correlation embedded. These values are coded using the simple DPCM techniques of differential base, namely, utilize the called Difference Between two Values (DBV), figure (6) shows example of DPCM encode/decode technique, such as;

\[ DC(i) = DC(i) - DC(i + 1) \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (5) \]

Where \( i = 1, 2, 3 \ldots n^2 - 1 \), \( n^2 \) is size of DC.

Fig. (6): Example of DPCM encode/decode technique.

B) For Ac values, which usually overburden with insignificant values, here we resize the zigzag values by reducing (eliminating) uncorrelated values that limited by number of element to remove or neglected, as shown in the following example of figure (7). Note, the ‘RED’ line refers to number of column specified by user to delete some insignificant coefficients, the reason behind this process to increase compression ratio.
Step 8: Separating and counting, to improve the encoding process, each reduce values also filtered by separating of zero from non-zero values and represented efficiently, as illustrated in figure (8).

Step 9: The last step of JPEG is optimal coding of the resulting above data. Applying entropy encoder on the saving compressed information using the arithmetic coding technique by remove coding redundancies between the coefficients.

To reconstruct the compressed image all the above step reverse, starts by applying the decoding of arithmetic decoder, followed by inverse counting and merging (see figure 9), then applying the inverse of zigzag that implies reconstruct error-free the Dc’s values, followed by de-quantization process and inverse IDCT.

4. Experimental Results

The standard parameters were considered to measure the proposed compression method performances includes basically the peak signal to noise ratio
(PSNR) of objective base, and the compression ratio (CR) according to the equations below. The proposed method is implemented through MATLAB\textregistered R2012b program installed on computer processor (Intel Core I7), windows 7. It is applied on four tested images of varying details (characteristics), see figure (10) for an overview, with different quality parameters (QF=0.01, 0.05, 0.1, 0.5, 1, 2 and 3), and the number of columns (NC) implies a range of selected values (NC=8, 15 and 20).

\[ CR = \frac{\text{OriginalSize}(KB)}{\text{CompressedSize}(KB)} \] ................................. (6)

\[ \text{PSNR}(I,I_{\text{compressed}}) = 10 \log_{10} \frac{(255)^2}{MSE} \] ........................... (7)

\[ \text{MSE}(I,I_{\text{compressed}}) = \frac{1}{M \times N} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} [I_{\text{compressed}}(j,k) - I(j,k)]^2 \] ................. (8)

Where \( I \) corresponding to original image and \( I_{\text{compressed}} \) corresponding to compressed image.

The results shown in tables (1-4) and figure (11), demonstrates the performance of the proposed method for the tested images, which clearly showed how the increasing of QF increase CR and decreasing PSNR and vice versa. Also the compression ratio is decreased as number of columns increased and this affection is slightly decrease as the QF value is increase. The impact of changing QF value have more affection than changing NC value.
Fig. (11): Relationship between CR and PSNR using NC=8, 15 and 20
This because involving new NC means acquired more high frequency signals which have no a significant influence on compression ratio, while increasing of QF means acquired more quantization. Figure (12) shows the original tested images after conversion into gray based for color one, versus the compressed decode images using the proposed method with QF=3, NC=15 and block size 8x8.

Original images

![Original images](image1)

Decompressed images

![Decompressed images](image2)

Fig. (12): Example of the impact of QF=3 with NC=15 on four tested images.

The quality of reconstructed image and its compression ratio is varying according to these two factors (QF & NC) values and the impact of these factors is different based on image details. For more texture variation images like Lena and Iris, convenient CR gained with less significant on image quality (less PSNR variation). For less details variation images like Girl and Face, high CR gained with salient impact on image quality (more PSNR variation). Figure (13) an example of effects of (QF=0.01, QF=3) and (NC=8, NC=15, NC=20) for Lena tested image.

![Decompressed Lena image with different QF and NC](image3)

Fig. (13): Decompressed Lena image with different QF and NC
Table (5) compares the proposed system versus the standard JPEG techniques for high and low qualities respectively of tested images.

Table (1): The performance of the proposed system for Lena image

| QF | CR | PSNR (dB) | Image Size (KB) after Compression | QF | CR | PSNR (dB) | Image Size (KB) after Compression | QF | CR | PSNR (dB) | Image Size (KB) after Compression |
|----|----|-----------|-----------------------------------|----|----|-----------|-----------------------------------|----|----|-----------|-----------------------------------|
| 0.01 | 6.1352 | 36.5981 | 10.682 | 0.01 | 3.5231 | 37.4994 | 18.602 | 0.01 | 2.7582 | 38.1836 | 23.760 |
| 0.05 | 7.4346 | 36.5980 | 8.815 | 0.05 | 4.4689 | 37.4992 | 14.665 | 0.05 | 3.5750 | 38.1829 | 18.332 |
| 0.1 | 8.3913 | 36.5979 | 7.931 | 0.1 | 5.1815 | 37.4986 | 12.887 | 0.1 | 4.2021 | 38.1810 | 15.920 |
| 0.5 | 11.3956 | 36.5932 | 5.751 | 0.5 | 7.5790 | 37.4784 | 8.647 | 0.5 | 6.4561 | 38.1261 | 10.151 |
| 1 | 13.8554 | 36.5791 | 4.730 | 1 | 9.8167 | 37.4261 | 6.676 | 1 | 8.6243 | 38.0074 | 7.599 |
| 2 | 17.5183 | 36.5323 | 3.741 | 2 | 13.1731 | 37.2801 | 4.975 | 2 | 11.9048 | 37.7156 | 5.505 |
| 3 | 20.4928 | 36.4685 | 3.198 | 3 | 15.9766 | 37.1079 | 4.102 | 3 | 14.6515 | 37.4261 | 4.473 |

Table (2): The performance of the proposed system for Iris image

| QF | CR | PSNR (dB) | Image Size (KB) after Compression | QF | CR | PSNR (dB) | Image Size (KB) after Compression | QF | CR | PSNR (dB) | Image Size (KB) after Compression |
|----|----|-----------|-----------------------------------|----|----|-----------|-----------------------------------|----|----|-----------|-----------------------------------|
| 0.01 | 6.1596 | 36.8122 | 49.873 | 0.01 | 3.5382 | 38.5624 | 86.824 | 0.01 | 2.7820 | 39.4676 | 110.426 |
| 0.05 | 7.6285 | 36.8121 | 40.270 | 0.05 | 4.5519 | 38.5621 | 67.489 | 0.05 | 3.6501 | 39.4667 | 84.163 |
| 0.1 | 8.5433 | 36.8120 | 35.958 | 0.1 | 5.2045 | 38.5613 | 59.026 | 0.1 | 4.2211 | 39.4639 | 72.778 |
| 0.5 | 12.0527 | 36.8078 | 25.488 | 0.5 | 7.8596 | 38.5349 | 39.086 | 0.5 | 6.6915 | 39.3839 | 45.909 |
| 1 | 14.8170 | 36.7953 | 20.733 | 1 | 10.2506 | 38.4661 | 29.969 | 1 | 9.0748 | 39.2199 | 33.852 |
| 2 | 19.3695 | 36.7554 | 15.860 | 2 | 14.2137 | 38.2889 | 21.613 | 2 | 12.8967 | 38.8612 | 23.820 |
| 3 | 23.1273 | 36.7049 | 13.283 | 3 | 17.4209 | 38.0929 | 17.634 | 3 | 16.0225 | 38.5249 | 19.173 |
Table (3): The performance of the proposed system for Girl image

| QF  | CR  | PSNR (dB) | Image Size (KB) after Compression | QF  | CR  | PSNR (dB) | Image Size (KB) after Compression | QF  | CR  | PSNR (dB) | Image Size (KB) after Compression |
|-----|-----|-----------|-----------------------------------|-----|-----|-----------|-----------------------------------|-----|-----|-----------|-----------------------------------|
| 0.01 | 6.4881 | 38.9198 | 10.101                             | 0.01 | 3.7682 | 40.4438 | 17.392                             | 0.01 | 2.9847 | 41.2072 | 21.957                             |
| 0.05 | 8.0353 | 38.9196 | 8.156                              | 0.05 | 4.9010 | 40.4428 | 13.372                             | 0.05 | 3.9837 | 41.2043 | 16.451                             |
| 0.1  | 9.0307 | 38.9191 | 7.257                              | 0.1  | 5.6545 | 40.4399 | 11.590                             | 0.1  | 4.6711 | 41.1957 | 14.030                             |
| 0.5  | 13.0993 | 38.9038 | 5.003                              | 0.5  | 9.0770 | 40.3594 | 7.220                              | 0.5  | 8.0196 | 40.9994 | 8.172                              |
| 1    | 16.3513 | 38.8633 | 4.008                              | 1    | 12.0759 | 40.1849 | 5.427                              | 1    | 11.0052 | 40.6671 | 5.955                              |
| 2    | 21.0998 | 38.7312 | 3.106                              | 2    | 16.6801 | 39.7719 | 3.929                              | 2    | 15.6075 | 40.0283 | 4.199                              |
| 3    | 24.9281 | 38.5684 | 2.629                              | 3    | 20.3718 | 39.3674 | 3.217                              | 3    | 19.2980 | 39.5112 | 3.396                              |

Table (4): The performance of the proposed system for Face image

| QF  | CR  | PSNR (dB) | Image Size (KB) after Compression | QF  | CR  | PSNR (dB) | Image Size (KB) after Compression | QF  | CR  | PSNR (dB) | Image Size (KB) after Compression |
|-----|-----|-----------|-----------------------------------|-----|-----|-----------|-----------------------------------|-----|-----|-----------|-----------------------------------|
| 0.01 | 8.6550 | 39.1123 | 7.572                              | 0.01 | 4.9641 | 40.5646 | 13.202                             | 0.01 | 3.9199 | 41.2809 | 16.719                             |
| 0.05 | 10.6928 | 39.1122 | 6.129                              | 0.05 | 6.4638 | 40.5641 | 10.139                             | 0.05 | 5.2513 | 41.2794 | 12.480                             |
| 0.1  | 12.1004 | 39.1119 | 5.416                              | 0.1  | 7.5113 | 40.5625 | 8.725                              | 0.1  | 6.2019 | 41.2749 | 10.567                             |
| 0.5  | 17.6885 | 39.1043 | 3.705                              | 0.5  | 11.8983 | 40.5231 | 5.508                              | 0.5  | 10.2947 | 41.1748 | 6.366                              |
| 1    | 21.2159 | 39.0845 | 3.089                              | 1    | 15.0312 | 40.4302 | 4.360                              | 1    | 13.3966 | 40.9724 | 4.892                              |
| 2    | 26.2669 | 39.0146 | 2.495                              | 2    | 19.9501 | 40.1663 | 3.285                              | 2    | 18.3574 | 40.5013 | 3.570                              |
| 3    | 30.3127 | 38.9198 | 2.162                              | 3    | 23.9620 | 39.8894 | 2.735                              | 3    | 22.4438 | 40.0821 | 2.920                              |
Table (5): Comparison with JPEG technique

| Image name          | CR  | PSNR | Image Size (KB) after Compression |
|---------------------|-----|------|----------------------------------|
| Lena (High quality) | 6.202 | 38.9 | 10.48                           |
| Lena (Low quality)  | 8.783 | 37.6 | 7.49                            |
| Iris (High quality) | 8.649 | 40.3 | 34.8                            |
| Iris (Low quality)  | 15.925 | 38.4 | 18.9                            |
| Girl (High quality) | 18.18 | 40   | 10.56                           |
| Girl (Low quality)  | 26.08 | 38.3 | 7.36                            |
| Face (High quality) | 20.4  | 40.81| 9.41                            |
| Face (Low quality)  | 28.82 | 38.8 | 6.66                            |

5. Conclusions

1- The tested images are of varying details of complex nature such as Lena and iris, of less complexity such as Girl and of moderate nature of large background facial image, since JPEG is applicable to all the mentioned types which is a challenge to be appropriate to different image structure.

2- The QF and NC corresponds to the compression control parameters, which affects the proposed system. There is a trade-off that must be made between image details and selecting of these parameters.

3- The proposed system achieved better quality to the standard JPEG with superiority in CR obtained.

References:

[1] Candra, R., Sarifuddin, M. and Sunny, A. (2016). Optimum zigzag scan mapping method on FPGA device, 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE). IEEE.

[2] Yeganegi, F., Hassanzade, V. and Ahadi, M. (2018). Comparative performance evaluation of SVD-based image compression. In Electrical Engineering (ICEE), Iranian Conference on (pp. 464-469). IEEE.

[3] Sayood, K. (2017). Introduction to data compression. Morgan Kaufmann.

[4] Ghadah, Al-K. (2012). Intra and Inter Frame Compression for Video Streaming, Ph.D. thesis, Exeter University, UK

[5] Gupta, M. and Garg, K. (2012). Analysis of image compression algorithm using DCT. International Journal of Engineering Research and Applications (IJERA), 2(1), 515-521.
[6] Ghanbari, M. (2003). Standard codecs: Image compression to advanced video coding (No. 49).

[7] Tuba, E., Tuba, M., Simian, D. and Jovanovic, R. (2017). JPEG quantization table optimization by guided fireworks algorithm. In International Workshop on Combinatorial Image Analysis (pp. 294-307). Springer, Cham.

[8] Wallace, K. (1992). The JPEG still picture compression standard. IEEE transactions on consumer electronics, 38(1), xviii-xxxiv.

[9] Raid, M., Khedr, M., El-Dosuky, A. and Ahmed, W. (2014). Jpeg image compression using discrete cosine transform-A survey. arxiv preprint arxiv:1405.6147.

[10] Deepthi, K. and Ramprakash, R. (2013). Design and Implementation of JPEG Image Compression and Decompression. International Journal of Innovations in Engineering and Technology (IJIIET), 2(1), 90-98.

[11] Sun, M. et al. (2020). Reduction of JPEG compression artifacts based on DCT coefficients prediction. Neurocomputing, 384, 335-345.

[12] Qiu, H. et al. (2020). Deep Residual Learning based Enhanced JPEG Compression in the Internet of Things. IEEE Transactions on Industrial Informatics.

[13] Krishnaraj, N. et al. (2019). Deep learning model for real-time image compression in Internet of Underwater Things (IoUT). Journal of Real-Time Image Processing, 1-15.

[14] Zhai, G. et al. (2008). Efficient image deblocking based on postfiltering in shifted windows. IEEE Transactions on Circuits and Systems for Video Technology, 18(1), 122-126.

[15] Hu, Y., Meng, F. and Wang, Y. (2012). Improved JPEG compression algorithm based on saliency maps. 5th International Congress on Image and Signal Processing, Chongqing, 2012, pp. 262-266.

[16] Siddeq, M. and Ghadah Al-Khafaji. (2013). Applied Minimized Matrix Size Algorithm on the Transformed Images by DCT and DWT used for Image Compression. International Journal of Computer Applications 70.15 (2013).
[17] Starosolski, R. (2014). New simple and efficient color space transformations for lossless image compression. Journal of Visual Communication and Image Representation, 25(5), 1056-1063.

[18] Messaoudi, A., Benchabane, F. and Srairi, K. (2019). DCT-based color image compression algorithm using adaptive block scanning. Signal, Image and Video Processing, 13(7), 1441-1449.

[19] AlKandari, M., AlRasheedi, H. and Ahmad, I. (2019). Enhanced JPEG Algorithm for Colored Images Based on FPGA Implementation. In 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO) (pp. 1-5). IEEE.

[20] Nosratinia, Aria. (2002). Enhancement of JPEG-compressed images by reapplication of JPEG. The journal of VLSI signal processing 27 (2002): 1291-1298.

[21] Wang, C. et al. (2007). An improved JPEG compression algorithm based on sloped-facet model of image segmentation. International Conference on Wireless Communications, Networking and Mobile Computing. IEEE.

[22] Singh, S. (2012). An algorithm for improving the quality of compacted JPEG image by minimizes the blocking artifacts. arxiv preprint arxiv:1208.1983.

[23] Hasan, M., Kamruddin N. and Tanzeem N. (2012). Computational complexity reduction of jpeg images. International Journal of Scientific and Technology Research 1.4 (2012): 72-75.

[24] Alam, L. et al. (2017). An improved JPEG image compression algorithm by modifying luminance quantization table. International Journal of Computer Science and Network Security (IJCSNS) 17.1: 200.

[25] Candra, R., et al. (2017). The Implementation of an Efficient Zigzag Scan. Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9.2 (2017): 95-98.

[26] Cogranne, R. (2018). Determining JPEG image standard quality factor from the quantization tables. arxiv preprint arxiv: 1802.00992.