African Journal of Advanced Pure and Applied Sciences (AJAPAS)
Online ISSN: 2957-644X
Volume 3, Issue 3, July-September 2024, Page No: 279-297
Website: *https://aaasjournals.com/index.php/ajapas/index*
ISI 2022-2023: 0.557 SJIFactor 2023: 5.689 معامل التأثير العربي 2023: (1.55)

Enhancement Image Compression using the Average Filtering and DCT

Ezzedin Muftah Enbiah [1], Khaled Almarimi [2*], Moammer H. Daw [3], Belgasem Alwerfali [4]
[1,3] Department of Computer Science & IT, College of Technical Science, Bani Walid, Libya
[2] Department of Computer Science, Faculty of Science, Bani Waleed University, Libya
[4] Higher Institute of Engineering Technology, Bani Walid, Libya

[*]Corresponding author: almaimekhaled@gmail.com

Received: August 03, 2024	Accepted: September 16, 2024	Published: September 24, 2024

Abstract:
Despite all the advances in storage and digital communication, the rapidly increasing volume of multimedia data continues to tax available resources. Compression of images becomes highly imperative to deal with the situation because of reduced requirements in terms of storage and transmission bandwidth while keeping the perceptible quality of the images. Among the lossy compression techniques for image compression, DCT-based block coding finds extensive application. However, it often causes highly perceived artifacts, mainly blocking effects, at higher compression rates. This research is conducted to minimize blocking artifacts generated by DCT-based image compression. The beginning gives an overview of data compression and the existing deblocking methods. Then it delves into DCT-based image compression and proposes a simple and effective solution to the blocking problem, depending on filtering the pixel value at the boundaries between the blocks. We test our proposed technique on different test images and analyze the results to know its effectiveness.

Keywords: Image compression, DCT, blocking effective, Lossy compression.

الملخص
على الرغم من كل التطورات في التخزين والاتصالات الرقمية، فإن الحجم المتزايد بسرعة من بيانات الوسائط المتعددة لا يزال يفرض ضغوطًا على الموارد المتاحة. يصبح ضغط الصور أمرًا ضروريًا للغاية للتعامل مع الموقف بسبب انخفاض المتطلبات من حيث التخزين وعرض النطاق الترددي للإرسال مع الحفاظ على الجودة الملموسة للصور. من بين تقنيات الضغط دو الفقد لضغط الصور يكون التشفير الكتلي القائم على DCT تطبيقًا واسع النطاق. ومع ذلك، فإنه غالبًا ما يتسبب في حدوث آثار شديدة الإدراك وخاصة التأثير الكتلي عند معدلات ضغط أعلى. يتم إجراء هذا البحث لغرض تقليل آثار التأثير الكتلي الناتجة عن ضغط الصور القائم على DCT. يعطي البداية نظرة عامة على ضغط البيانات وطرق إزالة التأثير الكتلي الحالية. ثم يتعمق في ضغط الصور القائم على DCT ويقترح حلاً بسيطًا وفعالاً لمشكلة التأثير الكتلي، اعتمادًا على تصفية قيمة البيكسل عند الحدود بين الكتل. نختبر تقنيتنا المقترحة على صور اختبار مختلفة ونحلل النتائج لمعرفة فعاليتها.

الكلمات المفتاحية: ضغط الصور، تحويل جيب التمام المتقطع، التأثير الكتلي، الضغط دو الفقد.

1. Introduction

Data compression refers to the transformation of a given file into a smaller file with fewer bits, to occupy less storage space and to transmit more files faster over the same bandwidth. Data decompression is the transformation of a given compressed file into an uncompressed file. During this process, the size of the file is enlarged, therefore occupying more space. Although data decompression is a required and necessary process, it is frequently omitted from the title. The most common type of data compression and decompression is the compression and decompression of text files.

Data compression is the method of uniformly reducing the size of data transforming data from one representation to another representation that requires fewer bits, thereby lowering storage and transmission expenses. Data decompression, the reverse operation of data compression, regards uniformly transforming data from a typically compressed representation to an uncompressed representation. This uncompressed representation has the same information content as the uncompressed representation. Recently, in light of the increasing demand for storing large amounts of data/pictures and transmitting them through networks, data compression has attracted considerable attention. There is a real need for telecommunications companies and television broadcasters to lower the required bandwidths (i.e., bring down expenses) and for the storage of images in personal computers and databases to use smaller storage mediums (thereby cutting expenses).

A typical implementation scenario of data compression and data decompression is shown in. The input to the data compression is an original data (uncompressed) representation, which consists of words of bits and comes from a source (e.g., camera, computer). The data compression results in a representation of the data that uses fewer bits and is typically accompanied with the designation "compressed" at the compressed data. The compressed data is accompanied with parameters that are needed for data decompression (e.g., prefix codes for Huffman coding), and is forwarded to the destination; the destination can be far away from the source (e.g., through radio transmission).

The compressed data is then fed to the data decompression algorithm. This algorithm "consumes" the compressed representation and produces (typically in one-to-one correspondence) the original data representation. This output is then forwarded to an application that "understands" the original data representation (e.g. display picture). It is worth mentioning that though both data compression and data decompression are numerically defined transformations of the data representation, they should be understood theoretically/algorithmically as two different domains. Indeed, a transformation may have different characteristics when viewed from the point of view of the data compression algorithm, compared to a transformation that is viewed from the point of view of the data decompression algorithm. For example, the representation of the data pictorially viewed, like in "Data Compression" and "Data Decompression" in, can be quite different. Hence, the data transformation is typically highly non-linear considering the data compression and data decompression algorithms separately. Figure 1 shows a platform based on the relationship between compression and decompression algorithms.

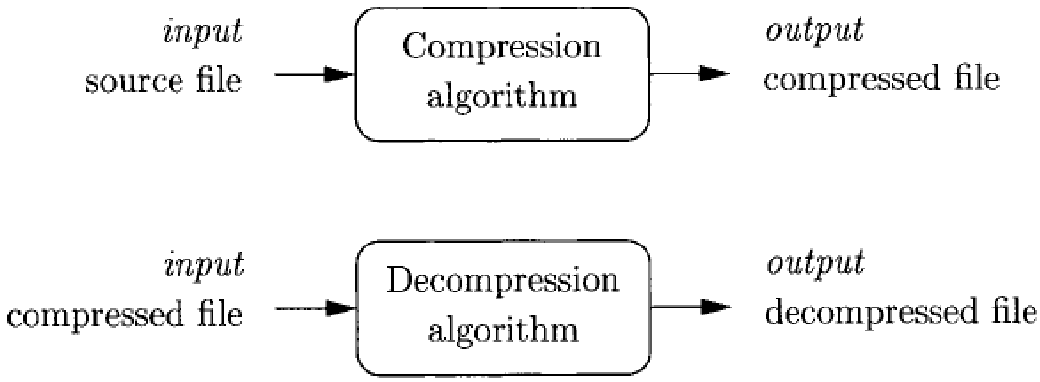

Figure 1: Compressor and de-compressor.

However, images can also be compressed and decompressed. A high-quality image compression and decompression process aims to reduce the number of bits in the image while maintaining the closest possible resemblance to the original image. There should be no distortions, lines, or edges left out or misplaced. A possible solution for this is JPEG compression. JPEG compression first transforms an image from the spatial domain to the frequency domain using DCT. After this transformation, the coefficients are quantized and entropy-coded with Huffman coding. JPEG decompression is the reverse procedure. It starts with the entropy-decoded image, the coefficients are unquantized and the image is transformed back to the spatial domain using the Inverse DCT.

JPEG compression is not a good solution for this task because of the poor image quality at high compression ratios. A better solution is the Joint Photographic Experts Group-compression (JPEG-compression). The objective of JPEG compression is to achieve the best image quality using few bits, but image quality deteriorates for high compression ratios.

Related Works

Image compression, a technique to reduce data volume for efficient transmission and storage, has been a focal point of research for several decades. Discrete Cosine Transform (DCT) is a cornerstone in many compression algorithms, especially in the realm of lossy compression. Early research focused on the comparison of DCT with other transforms, such as Discrete Wavelet Transform (DWT) [Katharotiya et al., 2011]. While DCT exhibited promising results, challenges like blocking artifacts arose during decompression. To address this, subsequent studies delved into post-processing techniques like DCT filtering and hybrid filtering [Kaur & Sethi, 2012]. The JPEG standard, heavily reliant on DCT, has been a subject of extensive analysis [Raid et al., 2014]. Researchers

have explored various aspects of JPEG compression, including DCT implementation optimization [Kaushik & Nain, 2014] and its application to different image types, such as grayscale images [Kulkarni & Junnarkar, 2015]. Efforts have also been made to refine DCT-based compression through quantization techniques and the mitigation of blocking artifacts [Pandey et al., 2015]. Recent years have witnessed a surge in research aimed at enhancing DCT-based compression performance. Adaptive approaches, such as Image Adaptive Transform (IAT), have been proposed to cater to high-resolution and multi-view images [Soh et al., 2017]. To further improve image quality, hybrid techniques combining DCT with other domains, like the pixel domain, have been investigated [Zhang et al., 2018]. Hardware implementation of JPEG compression using VHDL has been explored for efficiency gains [Shawahna et al., 2019]. Hybrid compression schemes, integrating DCT with DWT, have been studied to leverage the strengths of both transforms [Awadallah & Mahmoud, 2021]. Moreover, ongoing research focuses on optimizing DCT computations for resource-constrained systems while preserving image quality [Mefoued et al., 2023].
In summary, the literature showcases a rich history of DCT-based image compression research. While early works laid the foundation, subsequent studies have focused on refining DCT techniques, addressing challenges like blocking artifacts, and exploring hybrid approaches. The trend towards higher resolution and quality images has spurred the development of adaptive and computationally efficient DCT-based solutions.

Image compression

Image compression is a form of data compression used for digital images to decrease storage or transmission costs. This technique leverages visual perception and statistical properties of image data to yield better outcomes compared to general data compression methods. The objective is to encode the original image with fewer bits, aiming to minimize redundancy and enable efficient storage or transmission. Compression involves removing or consolidating specific parts of an image file. The following are some reasons for employing image compression:

• To improve your website, sites with uncompressed images can take longer to load, and can cause your visitors to be turned off because of this.

• To send and upload images, an uncompressed image can take some time to load, and some email servers have a file size limit.

• To reduce the storage impact on your hard drive.

There are two main categories of image compression processes: lossless image compression and lossy image compression. These categories differ in how they resize the image file. Lossless compression ensures that the image quality remains unchanged, while lossy compression removes some parts of the image to achieve a smaller file size. Here's a detailed explanation of each type:

1. Lossless compression

Lossless compression refers to the process of resizing images to a smaller version, this technique does not tamper with the image quality, and although it is an excellent way to resize image files, the result may not be very small, because lossless compression does not remove any part of the image.
For example, an image will be converted from (15) MB to (10) MB, however, its size will still be too large to be displayed on a web page, lossless image compression is especially useful when compressing text, because a small change in the original version can significantly change the meaning of the text or data.
Lossy compression is characterized by the fact that parts of the image remain intact, and the image quality does not lose any loss, and it is a reversible process, however, the disadvantages of this type are that the output image will appear very large, and decoding it is a challenge.

Lossless Compression Algorithms

Lossless compression algorithms, as the name suggests, are a type of data compression techniques that reduce file sizes without sacrificing any data. This implies that these algorithms can precisely reconstruct the original data from the compressed version. Various algorithms are developed based on specific types of input data or the anticipated redundancies in the uncompressed data. Here's a brief overview of some commonly employed compression algorithms.

bzip2: The bzip2 algorithm applies the Burrows-Wheeler algorithm along with RLE and Huffman encoding to compress data. Its usage is limited to compressing files without archiving them. Compressed files are commonly denoted with the .bz2 extension.

Huffman Encoding: This algorithm relies on a particular approach of identifying Each symbol results in a prefix symbol. Huffman encoding is a widespread method of generating prefix symbols compress files with extensions like.MPQ,. ACE, .JPEG,. Encoder are supported.

Lempel-Ziv Compression: The compression algorithm known as Lempel-Ziv, or LZ77 and LZ78, is widely recognized for its lossless data compression capabilities. These algorithms have paved the way for several variations, such as LZMA, LZW, LZSS, LZSS. Both are essentially programmers. In compression, LZ77 utilizes a sliding window, a concept that was subsequently discovered to be analogous to the explicit dictionary created by LZ78. Therefore, it becomes equivalent to decompressing the entire data. Files with. LZMA,. LZO,. LZ,.

PPM: The PPM compression algorithm, also referred to as Prediction by Partial Matching, relies on prediction and context modeling. In order to anticipate the next symbol in a sequence, PPM models incorporate a series of preceding symbols from the uncompressed symbol stream. Moreover, the PPM algorithm is compatible with ZIP and 7Z files.

Run Length Encoding (RLE) : RLE is lossless RLE An algorithm for compression that relies on sequences comprising identical data values occurring in multiple adjacent data items. These sequences are known as runs. In RLE, each run is stored as a single data value and count. This method is particularly handy for data containing numerous runs, such as basic graphic images like lines, drawings, icons and animations. Examples of files that utilize RLE include .PSD, .PSB, and .TGA formats.

2. Lossy Compression

Lossy compression reduces the size of the image by removing some parts of the image, it removes the tags that are not very necessary, as if you choose this method, you can get a much smaller version of the image with minimal difference in quality, in addition, you can enjoy faster loading speed.

Lossy compression also works with quality parameters to measure the change in quality, in most cases, you should adjust these parameters, as if they are less than (90), the images may appear of low quality to the human eye, for example, you can convert an image of (15) MB to (2200) KB in addition to (400) KB.

Lossy compression has the advantage of being able to get very low image size and fast loading time, and is an ideal choice for websites. However, there are some disadvantages, which are that lossy compression leads to the loss of image components, and it is irreversible.

Lossy Compression Algorithms

Lossy compression algorithms represent a significant advancement in minimizing file storage requirements, albeit at the cost of sacrificing some details. These algorithms are developed based on the study of human data perception and are predominantly reliant on transform encoding. Here's a brief overview of some well-known lossy compression algorithms:

Wavelet Compression: is a widely used lossy compression algorithm for image compression. This algorithm employs a technique known as transform coding, in which a wavelet transform is first applied to generate coefficients equal to the number of pixels in the image. As the information is typically concentrated in a small number of coefficients, these coefficients can be more effectively compressed. Notable applications of wavelet compression include, DJVU, ECW for still images and JPEG 2000.

Cartesian Perceptual Compression (CPC): also referred to as CPC, is a method of high compression for black and white raster images from archival scans. The algorithm is frequently utilized in the online dissemination of geographical mapping, design blueprints and legal documents.

Fractional Compression: Fractal compression is an algorithm used for digital image compression based on fractals. It is particularly effective for natural images and textures, where certain parts of an image bear resemblance to other parts. Fractal algorithms transform these similar parts into fractal codes, which are then utilized to reconstruct the compressed image.

Discrete Cosine Transform (DCT)

DCT is a finite sequence of data points in terms of the sum of cosine functions fluctuating at different frequencies. It is used in most digital media, including digital images such as JPEG, HEIF, J2K, exif and DNG. DCT is an important technique utilized in signal processing, primarily in audio, image, and video compression. The DCT translates a signal into its elementary frequency components: that is, it converts spatial DCT-transformed objects into objects defined in a DCT frequency domain. The most commonly used DCT is a block-based 2D DCT similar to JPEG. In a DCT-transformed object, low frequencies are represented with relatively high precision, and accuracy in the representation of high frequencies can be sacrificed.

The DCT has certain advantages over a discrete sine transform (DST). It possesses the plus property that enhances the ability of the DCT to suppress noise. The signal-to-noise ratio (SNR) cost must be compensated with increased complexity in the encoding process. There is a need to explore potential DCT implementations better suited for non-power-of-two DCT sizes because current implementations typically work on 8, 16, and 32 point DCTs, which conform to the fast algorithm data structure based on the power-of-two notions. In signal compression processes, The signal's primary energy is focused on its low-frequency components low-frequency components, a general property of natural signals.

The DCT exploits the fact that most of the energy of the signal is concentrated in low-frequency components. If a signal is concentrated in a certain frequency band, the DCT blocks will provide a sparse (banded) representation of it, which when combined with quantization provides the basis for its compression.

The general steps of compression using DCT

1- The image is divided into k blocks of n x n (n = 8 typically) pixels, each of the pixels are denoted by P_{xy}. If the number of image rows (columns) is not divisible by 8, the bottom row (rightmost column) is duplicated as many times as needed.

2- The DCT in two dimensions is applied to each block. The result is a block (vector) W(i) of 64 transform coefficients $w_j^{(i)}$ (where j = 0, 1, . . . , 63). The k vectors W(i) become the rows of matrix W

$$\mathbf{W} = \begin{bmatrix} w_0^{(1)} & w_1^{(1)} & \dots & w_{63}^{(1)} \\ w_0^{(2)} & w_1^{(2)} & \dots & w_{63}^{(2)} \\ \vdots & \vdots & & \\ w_0^{(k)} & w_1^{(k)} & \dots & w_{63}^{(k)} \end{bmatrix}.$$

3- The 64 columns of W are denoted by $C^{(0)}$, $C^{(1)}$, . . . , $C^{(63)}$. The k elements of $C^{(j)}$ are ($w^{(1)}_j$, $w^{(2)}_j$, . . . , $w^{(k)}_j$). The first coefficient vector $C^{(0)}$ consists of the k DC coefficients (Direct Current).

4- Each vector $C^{(j)}$ is quantized separately to produce a vector $Q^{(j)}$ of quantized coefficients .The elements of $Q^{(j)}$ are then written on the compressed stream .

In practice, the DCT is used for lossy compression. For lossless compression (where the DCT coefficients are not quantized) the DCT is inefficient but can still be used, at least theoretically, because (i) most of the coefficients are small numbers and (ii) there often are runs of zero coefficients. However, the small coefficients are real numbers, not integers, so it is not clear how to write them in full precision on the compressed stream and still have compression. Other image compression methods are better suited for lossless image compression.

DCT aims to reduce the correlation within image data. Once the decorrelation is achieved, every transform coefficient can be encoded separately without compromising compression efficiency.

1. The One-Dimensional DCT:- The most common DCT definition of a 1-D sequence of length N is

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \qquad (1)$$

for u= 0,1,2,…,N− 1. Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \qquad (2)$$

Where $f(x)$ is input data values (pixels), and for x= 0,1,2,…,N− 1 . In both equations (1) and (2) α(u) is defined as :-

$$\alpha(u) \begin{cases} \sqrt{\dfrac{1}{N}} & for \quad u = 0 \\ \sqrt{\dfrac{2}{N}} & for \quad u \neq 0 \end{cases} \qquad (3)$$

It is clear from (1) that for u=0 implies :-

$$C(u=0) = \sqrt{\frac{1}{N}} \sum_{x=0}^{N=1} f(x) \qquad (4)$$

The initial transform coefficient represents the average value of the sample sequence. In literature, it is commonly known as the Direct Current (DC) Coefficient. The remaining transform coefficients are referred to as alternating current (AC) coefficients.

To fix ideas, ignore the *f(x)* and *α(u)* component in (3.1). The plot of $\sum_{x=0}^{N-1} \cos\left[\frac{\pi(2x+1)u}{2N}\right]$ For *N* = 8 and varying values of u for u= 0,1,2,…,N− 1 are shown in figure2.

Based on our previous observations, the first waveform in the top-left corner (u = 0) maintains a constant (DC) value, while all other waveforms (u = 1,2,…, 7) produce waveforms with increasingly higher frequencies .

These waveforms belong to the cosine basis function. It is important to note that these fundamental functions are orthogonal, meaning that the multiplication and summation of any waveform in Figure 2 with another waveform result in a scalar value of zero. Likewise, when a waveform in Figure 2 is multiplied by itself and then summed over all sample points, the outcome is a constant scalar value. This independence characterizes orthogonal waveforms, meaning that none of the basis functions can be expressed as a combination of other basis functions.

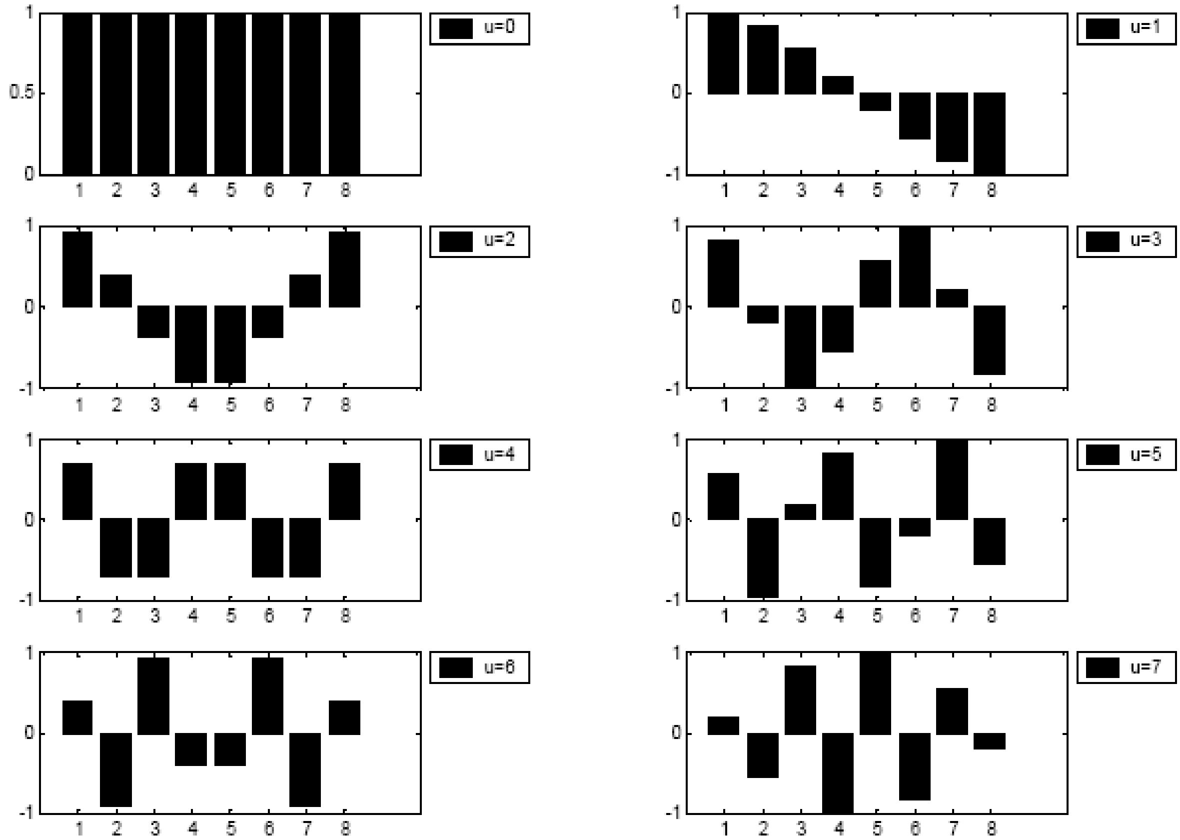

Figure 2: A basis function of cosine in one dimension (N=8)

When the input sequence contains more than N sample points, it can be divided into sub-sequences of length N. Each of these chunks can then undergo independent DCT application.

It's crucial to note that during each computation, the values of the basis function points remain constant, while only the values of f(x) change within each sub-sequence. This property is significant as it allows for pre-computation of the basic functions offline and their subsequent multiplication with the sub-sequences. This approach reduces the number of mathematical operations (such as multiplications and additions), resulting in enhanced computational efficiency.

2. The Two-Dimensional DCT:- This document aims to investigate the effectiveness of DCT in processing images, necessitating the expansion of concepts previously outlined to a two-dimensional framework. The 2-D DCT is a direct extension of the 1-D counterpart and is expressed as follows:

$$C(u,v)=\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)u}{2N}\right] \qquad (5)$$

for $u,v = 0,1,2,\ldots,N-1$ and $\alpha(u)$ and $\alpha(v)$ are defined in equation (3.3). The inverse transform is defined as

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)u}{2N}\right] \qquad (6)$$

For x and y ranging from 0 to N-1, the 2-D basis functions can be derived by the crosswise combination of the 1-D basis functions shown in Figure 2 with a similar set oriented vertically. The basis functions for N = 8 are shown in Figure 3. Clearly, these basis functions exhibit a progressive frequency increase in both the vertical and horizontal directions. The top left basis function of results from multiplication of the DC component in Figure 2 with its transpose. Hence, this function assumes a constant value and is referred to as the DC coefficient.

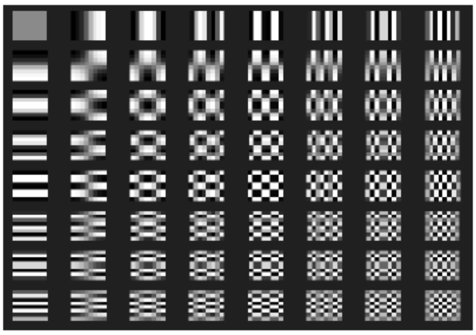

Figure 3: Two-dimensional DCT basis functions (N = 8)

Measures of Performance

The effectiveness of a compression algorithm can be evaluated using different criteria, depending on the specific application. While time efficiency is important, space efficiency becomes the primary concern when time is not a critical factor. The focus is on how well the data compression algorithm can conserve storage space. Assessing the overall performance of a compression algorithm is challenging due to its dependency on the presence of redundancy in the data it processes. Additionally, the behavior of the compression is influenced by whether an exact reconstruction of the original data is permitted.

1. Lossless compression :- For lossless compression algorithms, we measure the compression effect by the amount of shrinkage of the source file in comparison to the size of the compressed version. Following this idea, several approaches can be easily understood by the definitions below :-

1. **Compression ratio :-** This is simply the ratio of the output to the input file size of a compression algorithm, i.e. the compressed file size after the compression to the source file size before the compression as shown in equation (7).

$$Compression\ ratio = \frac{size\ after\ compression}{size\ before\ compression} \tag{7}$$

- **Compression factor: -** This represents the reverse of the *compression ratio*.
- **Saving percentage:** - This indicates the percentage of shrinkage as illustrated in equation (8).

$$saving\ perce\,ntae = \frac{size\ before\ compression - size\ after\ compression}{size\ before\ compression}\% \tag{8}$$

2. Lossy compression:-

Lossy compression requires evaluating the quality of the decompressed data and the compression impact. The term "fidelity" is commonly used to depict the similarity between the original and decompressed files, while the variance between the pre-compression source and post-decompression file is termed "distortion", which is often utilized in practical. applications.

Modeling and Coding:-

Data compression algorithm development for diverse data types involves two main phases: modeling and coding. The initial phase, modeling, focuses on extracting redundancy information from the data and representing it in the form of a model.

Methodology

The blocking effects reduction by using the averaging filter

This method falls under the abstract heading of image enhancement and restoration. Filters are applied to an image after it has been decompressed by DCT coefficient de-quantization and application of the inverse DCT via the smoothing linear filter, where these Pixel filters exploit the advantage of the neighbouring pixel correlation property of natural (photographic) images.

Proposed filter is performed for all points of the input image in away to smooth the input image by replacing the value of each pixel in an image with average value of the neighbourhood within 3X3 window. This reconstructed pixel would be affected by the neighbouring pixels of the corrupted image by using the proposed window shown in figure4 as follows:

$$P(x,y)_{new} = \frac{1}{12}\sum[P(x-1,y-1) + P(x-1,y) + P(x-1,y+1) + P(x,y-1) + 4P(x,y) + P(x,y+1) + P(x+1,y-1) + P(x+1,y) + P(x+1,y+1)] \qquad (9)$$

	x-1	x	x+1
y-1	1/12	1/12	1/12
y	1/12	4/12	1/12
y+1	1/12	1/12	1/12

Figure4: The proposed 3 x 3 filter

Since, the image composed from high details and low details information causes an effect on edges and low details. Thus, the image will be classified into two segments that are depending on edge segmentation. Therefore each segment is filtered independently on the other one and then are joined together to form the output filtered collected image as shown on the scheme in figure 5.

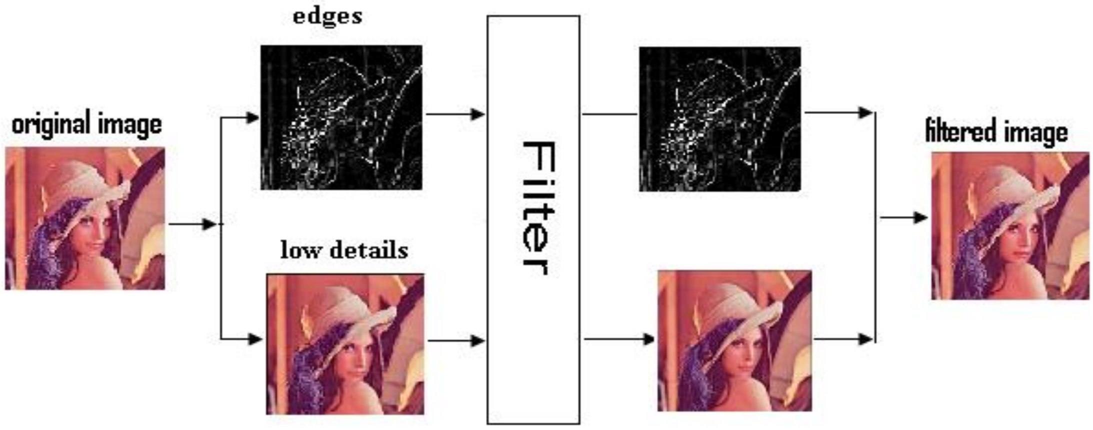

Figure 5: The scheme of the proposed technique using edges and low details.

When we apply this model on RGB color system, we first start by applying this filter on all color components individually (R, G ,and B). Then compose them to get the reconstructed result as RGB color system as shown in Figure 6.

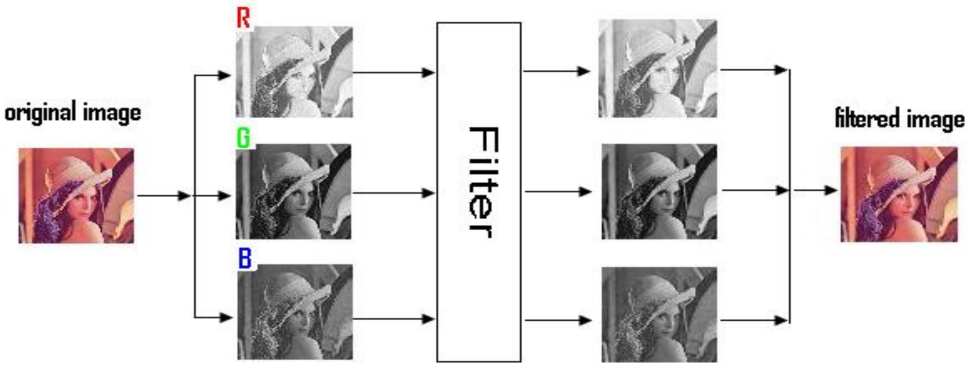

Figure 6: Stage of the proposed technique by using RGB image color.

But when used LC representation (YC_bC_r) which based on luminance (Y) and chrominance (C), what we need is only to apply this on luminance (Y) Given that the human eye is more sensitive to small changes in brightness rather than in color, as illustrated in Figure 7.

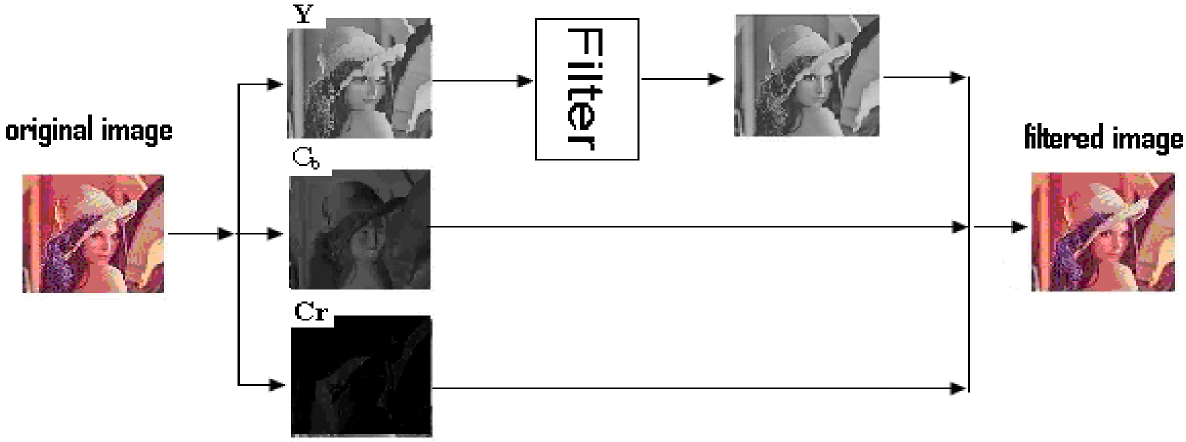

Figure 7: Stage of proposed technique for YC_bC_r image color.

Results

Computer simulations have been conducted to showcase the suggested algorithm that minimizes the blocking effects on JPEG decompressed images. The algorithm was tested on various images, including Lena, all of which were compressed using the standard JPEG code. We zoomed in part all our images shown in the following figures to show more our performance of the proposed technique. the original Lena image size is (768kb) as shown in figure 8. It is compressed to the size (13.5kb) as shown in figure 9 where are visible the block effects on this compressed image.In the first we compared between rate of pixel in the center of window when it equals ninth, third and half from total value for pixels where the results of Lena test image are shown in figures (10, 11,and 12) , also other results of the other images and the PSNR are measured according to these results are documented in table 1.


Figure 8: Original Lena image.

Figure 9: Compressed Lena image

Table 1: Result of filtered images at different rates of a center pixel in window.

Image	**MSE**				**PSNR**			
	Compressed	Filtered			compressed	Filtered		
		1/9	1/3	½		1/9	1/3	½
Lena.bmp	25.4666	25.4404	25.0765	25.0110	34.0711	34.0756	34.1381	34.1495
Bird.jpg	22.6667	22.3803	22.1684	22.1257	34.5769	34.6322	34.6735	34.6818
Baboon.png	33.1594	33.0916	32.1939	31.9208	32.9247	32.9336	33.0531	33.0901

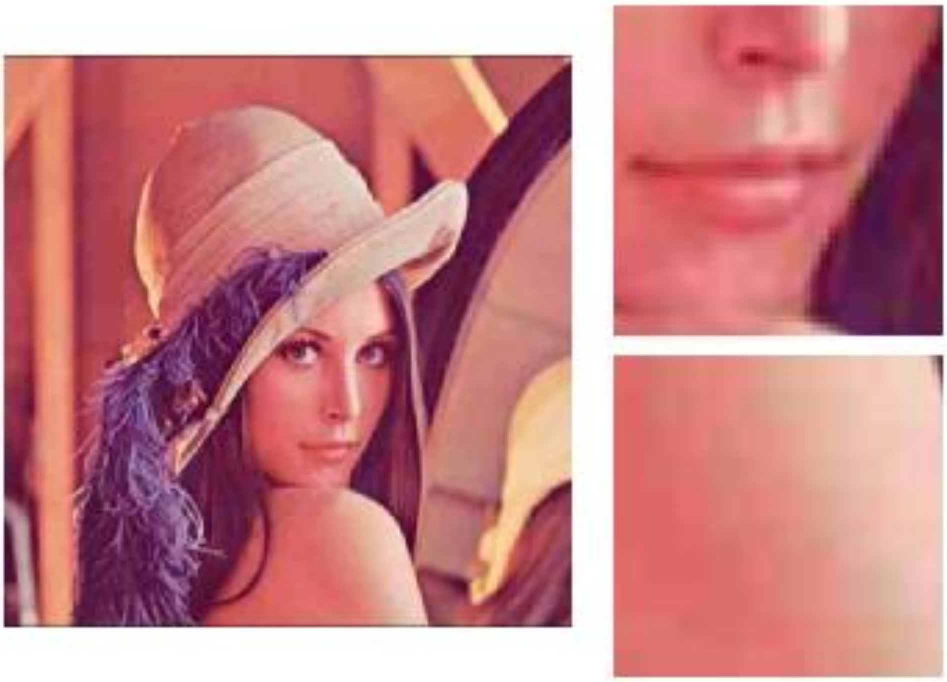

Figure 10: Filtered Lena images when the center pixel equals the ninth total value.

Figure 11: Filtered Lena images when the center pixel equals third total value.

Figure 12: Filtered Lena images when the center pixel equals half total value.

when we applied the filter shown in figure 4 on the corrupted image, we obtained an enhanced and restored image that appear in figure 13.

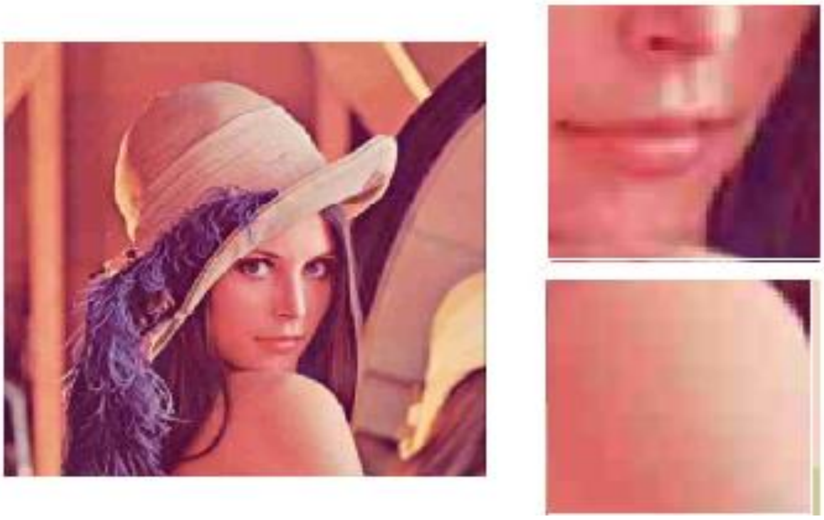

Figure 13: 3x3 window filtered Lena image.

But when we are tried to apply the other filters (5x5, and 7x7) on the same corrupted Lena image we obtained the results shown in figures 14 , 15.

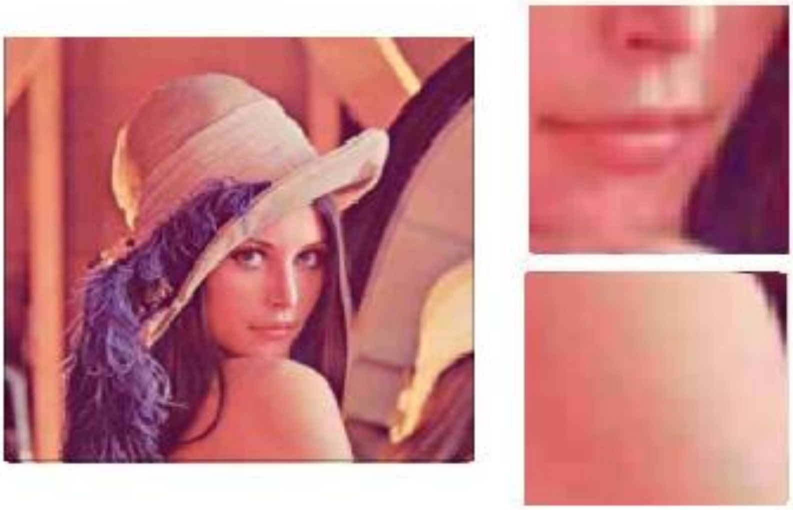

Figure 14: 5x5 window filtered Lena image.

Figure 15: 7x7 window filtered Lena image.

Thus, different results are obtained by using those previous filters (5x5, and 7x7), also other results of the other images and then the PSNR is measured to derive the artifact measure as shown in Table 2.

Table 2: Results of filtered image at (5x5, and 7x7) filters.

Image	MSE				PSNR			
	compressed	Filtered			compressed	Filtered		
		3x3	5x5	7x7		3x3	5x5	7x7
Lena.bmp	25.4666	25.0765	27.2681	30.0131	34.0711	34.1381	33.7742	33.3577
Bird.jpg	22.6667	22.1684	23.5015	25.6659	34.5769	34.6735	34.4198	34.0372
Baboon.jpg	47.7041	50.8621	56.9669	60.2735	31.3452	31.0669	30.5746	30.3295

When we work under RGB color system, the compressed image divided to three fundamental color component as illustrated in figure 16.

Figure 16: RGB color component of Lena image.

The proposed filter applied on RGB test image components is shown in figures 17(a),18(a), and 19(a) separately whereas the deblocking result is shown in figures 17(b),18(b) and 19(b).
All filtered image components are composed to result in filtered image as shown in Figure 20. Thus, we could notice the improvement after applying the proposed technique.

(a) (b)

Figure 17: Result after applying a filter on R test image component.

(a) (b)

Figure 18: Result after applying filter on G test image component.

(a) (b)

Figure 19: Result after applying filter on B test image component.

Figure 20: Filtered Lena image under RGB system.

The detailed result for only Lena image, as the results for the other images are similar to those of the Lena image, it indicates consistency across the test images. Where the figures indexed by (a) for those figures represent the original images, and the figures indexed by (b) for those figures represent the compressed images, and the figures indexed by (c) for those figures represent the filtered images, whereas PSNR derive the artifact measure which is shown in table 3.

Table 3: Results of filtered image for RGB system color.

Image	Size		MSE		PSNR	
	Original image	Compressed image	Compressed image	Filtered image	Compressed image	Filtered image
Lena	768k	13.5k	25.4666	25.0959	34.0711	34.1348
Baboon	55.6k	27.2k	47.7041	50.8781	31.3452	31.0655
Desert_Libya	19.3k	6.65k	22.4428	24.0446	34.6200	34.3206

When it is used the filter under LC representation (YCbCr), where the Lena compressed image is shown in Figure 9 the only luminance (Y) component is used as shown in figure 21 and therefore is filtered and the result is shown in figure 22 whereas the final result of filtered image is shown in figure 23.

Figure 21 : Lena luminance (Y) component.

Figure 22: Result of filtered Lena luminance (Y) component.

Figure 23: Filtered Lena image under (YCbCr) system.

When we applied this technique by using (YCbCr) on the other images the results where the figures indexed by (a) represent the luminance (Y) components, and the figures indexed by (b) represent the Filtered luminance (Y) components, and the figures indexed by (c) for those figures represent the final results, whereas PSNR derive the artifact measure which is shown in Table 4.

Table 4: Results of filtered image for YCbCr system color.

Image	Size		MSE		PSNR	
	Orginal image	Compressed image	Compressed image	Filtered image	Compressed image	Filtered image
Lena	768k	13.5k	25.4666	25.6000	34.0711	34.0484
Baboon	55.6k	27.2k	47.7041	51.2961	31.3452	31.0300
Desert_Libya	19.3k	6.65k	22.4428	24.4025	34.6200	34.2565

When the proposed technique is applied as the scheme is depicted in figure 5, the image will classify the high details as shown in figure 24(a) and low details as shown in figure 25 (a). After that all segments of image are filtered as shown in figures 24(b) and 25 (b) and then they composed to obtain the result of the filtered image as shown in figure 26. The PSNR measured for the compressed and improved Lena image, and other images were filtered by the proposed technique, where then the result are shown in table 5 and to watch and observe the other mentioned images.

Table 5: Results of filtered image by the proposed technique.

Image	MSE		PSNR	
	Compressed	filtered	Compressed	filtered
Lena.bmp	25.4666	24.9642	34.0711	34.1576
Bird.jpg	22.6667	22.0153	34.5769	34.7036
Baboon.jpg	47.7041	49.2972	31.3452	31.2026

(a) (b)

Figure 24: Edges of Lena image.

(a) (b)

Figure 25: Low details of Lena image.

Figure 26: Result of proposed schema.

Discussions

From the results presented in Table 1 we noticed that the different PSNR's values for the different images are increases of the filtered images respect the values of the original compressed images, but from the observation of the filtered images that are shown in figures (10, 11 and 12), we can see the block effects is reduced for high rate in figures (10 and 11) and less rate in figure 12 with some distortion in small features which is increased in figure 10.

By observing the results in Table 2 we will observe the values of PSNR are improved when the number of the center pixel's neighbours lesser in value. Also, we can notice that in figures (13,14 and 15), the edges and high details appear clearer in the 3x3 window.

From the table 3 and table 4 we can notice the difference between using various color systems where there are small difference between using RGB color system and YCbCr color system where as the filtering luminance (Y) components are used only in YCbCr color system.

With the relation of the outcomes that are presented existing in table 5, we could observe the performance of the scheme which are shown in figure 5 where the values of PSNR are increased up to the previous results in tables (1, 2, 3 and 4). Also, this performance has appeared in figure 26 where the block effects is reduced clearly with the retained to the most of edges perfectly, and the small features like the pupils of the eyes are also still intact. These results are very near to those results obtained from the other methods, where the difference in PSNR between the compressed and the filtered images ranges between ±1dB.

Conclusion

In this study, a novel approach has been suggested to minimize block effects in block-transform compressed images. The method involves employing an average filter to address pixel value discrepancies at block boundaries, with a focus on categorizing the image into low-detail areas and edges. It is used to smooth neighbouring pixel so that to preserve the subtle image details. This model is selected to be simple and easy for realization. this technique could be used for LC representation (YCbCr) system color because it could be used only on luminance (Y) without using the other two components.

The algorithm's performance was evaluated based on the PSNR of the filtered image. The results indicated an increase in the measured PSNR for most test images. However, it's important to note that the PSNR may not entirely capture the improvement in image quality.

References

[1] A.M.Raid, W.M.Khedr, M. A. El-dosuky and Wesam Ahmed, " Jpeg Image Compression Using Discrete Cosine Transform - A Survey" , International Journal of Computer Science & Engineering Survey (IJCSES) Vol.5, No.2, April 2014.

[2] Abdelkader Mefoued, Nasreddine Kouadria, Saliha Harize and Noureddine Doghman, “Improving image encoding quality with a low-complexity DCT approximation using 14 additions”, Journal of Real-Time Image Processing (2023) 20:58.
[3] Abhishek Kaushik*, Er. Deepti Nain, “Image Compression Algorithms Using Dct”, Int. Journal of Engineering Research and Applications www.ijera.com ISSN : 2248-9622, Vol. 4, Issue 4(Version 1), April 2014, pp.357-364.
[4] Ahmad Shawahna, Md. Enamul Haque, and Alaaeldin Amin, “JPEG Image Compression using the Discrete Cosine Transform: An Overview”, Applications, and Hardware Implementation arXiv:1912.10789v1 [cs.MM] 1 Nov 2019.
[5] Aman George and Ashish Sahu, “Analysis of Image Compression Algorithm using DCT”, IJSTE - International Journal of Science Technology & Engineering | Volume 3 | Issue 12 | June 2017 ISSN (online): 2349-784X.
[6] Anilkumar Katharotiya1* Swati Patel1 Mahesh Goyani, “ Comparative Analysis between DCT & DWT Techniques of Image Compression”, Journal of Information Engineering and Applications ISSN 2224-5758 (print) ISSN 2224-896X (online) Vol 1, No.2, 2011.
[7] Asawari Kulkarni and Aparna Junnarkar, “ Gray-Scale Image Compression Techniques: A Review”, International Journal of Computer Applications (0975 – 8887) Volume 131 – No.13, December2015.
[8] Jae Woong Soh, Hyun-Seung Lee and Nam Ik Cho, “An Image Compression Algorithm Based on the Karhunen Lo`eve Transform”, Proceedings of APSIPA Annual Summit and Conference 2017 12 - 15 December 2017, Malaysia.
[9] Nancy Awadallah Awad and AmenaMahmoud, “Improving Reconstructed Image Quality via Hybrid Compression Techniques”, Computers,Materials & Continua(CMC, 2021, vol.66, no.3).
[10] Parmjeet Kaur and Poonam Sethi, “ Blocking Artifacts Reduction in Block based Discrete Cosine Transform Compressed Images”, International Journal of Computer Applications (0975 – 8887) Volume 42– No.6, March 2012.
[11] S.S.Pandey, Manu Pratap Singh & Vikas Pandey, “ Block wise image compression & reduced blocks artifacts using discrete Cosine Transform”, International Journal of Scientific and Research Publications, Volume 5, Issue 3, March 2015 1 ISSN 2250-3153.
[12] Xiaoshuai Zhang, Wenhan Yang, Yueyu Hu, Jiaying Liu y, “ Dmcnn: Dual-Domain Multi-Scale Convolutional Neural Network For Compression Artifacts Removal”, arXiv:1806.03275v2 [cs.CV] 11 Jun 2018.

.