

Article

Edge-Based and Prediction-Based Transformations for Lossless Image Compression

Md. Ahasan Kabir * and M. Rubaiyat Hossain Mondal 

Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh; rubaiyat97@iict.buet.ac.bd

* Correspondence: kabir.ece07@gmail.com

Received: 7 February 2018; Accepted: 1 May 2018; Published: 4 May 2018



Abstract: Pixelated images are used to transmit data between computing devices that have cameras and screens. Significant compression of pixelated images has been achieved by an “edge-based transformation and entropy coding” (ETEC) algorithm recently proposed by the authors of this paper. The study of ETEC is extended in this paper with a comprehensive performance evaluation. Furthermore, a novel algorithm termed “prediction-based transformation and entropy coding” (PTEC) is proposed in this paper for pixelated images. In the first stage of the PTEC method, the image is divided hierarchically to predict the current pixel using neighboring pixels. In the second stage, the prediction errors are used to form two matrices, where one matrix contains the absolute error value and the other contains the polarity of the prediction error. Finally, entropy coding is applied to the generated matrices. This paper also compares the novel ETEC and PTEC schemes with the existing lossless compression techniques: “joint photographic experts group lossless” (JPEG-LS), “set partitioning in hierarchical trees” (SPIHT) and “differential pulse code modulation” (DPCM). Our results show that, for pixelated images, the new ETEC and PTEC algorithms provide better compression than other schemes. Results also show that PTEC has a lower compression ratio but better computation time than ETEC. Furthermore, when both compression ratio and computation time are taken into consideration, PTEC is more suitable than ETEC for compressing pixelated as well as non-pixelated images.

Keywords: Image compression; edge; SPIHT; computation time; pixelated image; JPEG-LS

1. Introduction

In today’s information age, the world is overwhelmed with a huge amount of data. With the increasing use of computers, laptops, smartphones, and other computing devices, the amount of multimedia data in the form of text, audio, video, image, etc. are growing at an enormous speed. Storage of large volumes of data has already become an important concern for social media, email providers, medical institutes, universities, banks, and many other offices. In digital media such as in digital cameras, digital cinemas, and films, high resolution images are needed. In addition to the storage, data are often required to be transmitted over the Internet at the highest possible speed. Due to the constraint in storage facility and limitation in transmission bandwidth, compression of data is vital [1–8].

The basic idea of compressing images lies in the fact that several image pixels are correlated, and this correlation can be exploited to remove the redundant information [9]. The removal of redundancy and irrelevancy leads to a reduction in image size. There are two major types of image compression—lossy and lossless [10–12]. In the case of lossless compression, the reconstruction process can recover the original image from the compressed images. On the other hand, images that go through the lossy compression process cannot be precisely recovered to its actual form. Examples of lossy

compression are some of the wavelet-based compressions such as embedded zerotrees of wavelet transforms (EZW), joint photographic experts group (JPEG) and the moving picture experts group (MPEG) compression.

A large number of research papers report image compression algorithms. For example, one study [13] is about discrete cosine transform (DCT)-based lossless image compression where the higher energy coefficients in each block are quantized. Next, an inverse DCT is performed only on the quantized coefficients. The resultant pixel values are in the 2-D spatial domain. The pixel values of two neighboring regions are then subtracted to obtain residual error sequence. The error sequence is encoded by an entropy coder such as Arithmetic or Huffman coding [13]. Image compression in the frequency domain using wavelets is reported in several studies [12,14–17]. In the method described in [14] lifting-based bi-orthogonal wavelet transform is used which produces coefficients that can be rounded without any loss of data. In the work of [18] wavelet transform limits the image energy within fewer coefficients which are encoded by “set partitioning in hierarchical trees” (SPIHT) algorithm.

In [19] JPEG lossless (JPEG-LS), a prediction-based lossless scheme, is proposed for continuous tone images. In [14] embedded zero tree coding (EZW) method is proposed based on the zero tree hypothesis. The study in [12] proposes a compression algorithm based on combination of discrete wavelet transform (DWT) and intensity-based adaptive quantization coding (AQC). In this AQC method, the image is divided into sub-blocks. Next, the quantizer step in each sub-block is computed by subtracting the maximum and the minimum values of the block and then dividing the result by the quantization level. In the case of intensity-based adaptive quantizer coding (IBAQC) reported in [12] the image sub-block is classified into low and high intensity blocks based on the intensity variation of each block. To encode high intensity block, it is required to have large quantization level depending on the desired peak signal to noise ratio (PSNR). On the other hand, if the pixel value in the low intensity block is less than the threshold, it is required to encode this value without quantization; otherwise, it is required to quantize the bit value with less quantization level. In case of the composite DWT-IBAQC method, IBAQC is applied to the DWT coefficients of the image. Since the whole energy of the image is carried by only a few wavelet (DWT) coefficients, the IBQAC is used to encode only the coarse (low pass) wavelet coefficients [12].

Some researchers describe prediction-based lossless compression [1,3,19–24]. Moreover, the combination of wavelet transform and the concept of prediction are presented in some studies [25,26]. In [25], the image is pre-processed by DPCM and then the wavelet transform is applied to the output of the DPCM. In [26], the image pixels are predicted by a hierarchical prediction scheme and then the wavelet transform is applied to the prediction error. Some work [5,9,11,27–30] applies various types of image transformation or pixel difference or simple entropy coding. An image transformation scheme known as “J bit encoding” (JBE) has been proposed in [11]. It can be noted that image transformation means rearranging the positions of the image components or pixels to make the image suitable for huge compression. In this [11] work, the original data are divided into two matrices where one matrix is for original nonzero data bytes, while the other matrix is for defining the positions of the zero/nonzero bytes.

A number of research papers use the high efficiency video coding (HEVC) standard for image compression [31–34]. The work in [31] describes a lossless scheme that carries out sample-based prediction in the spatial domain. The work in [33] provides an overview of the intra coding techniques in the HEVC. The authors of [32] present a collection of DPCM-based intra-prediction method which is effective to predict strong edges and discontinuities. The work in [34] proposes piecewise mapping functions on residual blocks computed after DPCM-based prediction for lossless coding. Besides, the compression using HEVC, JPEG2000 [35,36] and graph-based transforms [37] are also reported. Moreover, the work in [5] presents a combination of fixed-size codebook and row-column reduction coding for lossless compression of discrete-color images. Table 1 provides a comparative study of different image compression algorithms reported in the literature.

One special type of image is the pixelated images that are used to carry data between optical modulators and optical detectors. This is known as pixelated optical wireless communication system in the

literature. Figure 1 illustrates one example of a pixelated system [38]. In such systems, a sequence of image frames is transmitted by liquid crystal display (LCD) or light emitting diodes (LED) arrays. A smart-phone with camera or an array of photodiode with imaging lens can be used as optical receivers [6–8]. Such systems have the potential to have huge data rates as there are millions of pixels on the transmitter screens. The images created on the optical transmitter are required to be within the field of view (FOV) of the receiver imaging lens. Pixelated links can be used for secure data communication in banking and military applications. For instance, pixelated systems can be useful at gatherings such as shopping malls, retail store, trade shows, galleries, conferences, etc. where business cards, product videos, brochures, and photos can be exchanged without the help of the Internet (www) connection. The storage of pixelated images may be vital for offline processing. Since data are embedded within image pixels, the pixelated images must be processed by lossless compression methods. Any amount of loss in image entropy may lead to loss in the embedded data. A very important feature of pixelated images is that a single intensity value made of pixel blocks contains a single data, and this value of intensity changes abruptly at the transition of pixel blocks. This feature is not particularly exploited in the existing image compression techniques. Hence, none of the above-mentioned research reports are optimum for pixelated images as the special features of these images are yet to be exploited for compression. In fact, a new compression algorithm for pixelated images has been proposed by the authors of this paper in a very recent study [39]. This new algorithm is termed as edge-based transformation and entropy coding (ETEC) having high compression ratio at moderate computation time. In this previous study [39], the ETEC method is evaluated for only four pixelated images. This paper extends the study of ETEC method for fifty (50) different pixelated images. Moreover, a new algorithm termed as prediction-based transformation and entropy coding (PTEC) is proposed to overcome the limitations of computation time of ETEC. The main contributions of this paper can be summarized as follows:

- (1) Providing a framework for ETEC method as a combination of JBE and entropy coding, and then evaluating its effectiveness for compressing a wide range of pixelated images.
- (2) Developing a new algorithm termed as PTEC by combining the aspects of hierarchical prediction approach, JBE method, and entropy coding.
- (3) Comparing the proposed ETEC and PTEC schemes with the existing compression techniques for a number of pixelated and non-pixelated standard images.

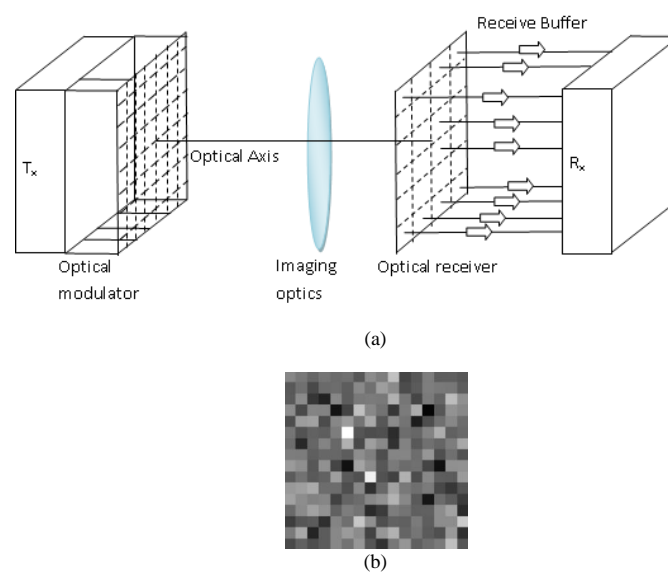


Figure 1. Illustration of (a) a pixelated optical wireless communication system [38] (b) a transmitted pixelated image.

The rest of the paper is summarized as follows. Section 2 describes JPEG-LS, SPIHT, Huffman coding, Arithmetic coding and other existing methods. Section 3 describes the new ETEC and PTEC methods. The results on different image compression methods are reported in Section 4. Finally, Section 5 presents the concluding remarks.

Table 1. Technical scenarios of few existing lossless and near lossless image compression algorithms.

Ref. No.	Prediction Based	Wavelet Based	Pixel Difference Based	DCT	Entropy Coding	Image Encoder/Transformer	Image Type	Hierarchical Approach
[1]	Yes	No	No	No	Yes	PDT	Dithering	No
[3]	Yes	No	No	No	Yes	No	Continuous	Yes
[5]	No	No	No	No	Yes	Row-column reduction encoding	Map images	No
[9]	No	No	No	No	Yes	LZW	Continuous	No
[11]	No	No	No	No	Yes	J bit encoding	Continuous	No
[12]	No	Yes	No	No	No	AQC	Continuous	No
[13]	No	No	Yes	Yes	Yes	No	Continuous	No
[14]	No	Yes	No	No	No	Modified EZW	Continuous	No
[15]	No	Yes	No	No	No	No	All	No
[16]	No	Yes	No	No	No	No	Color	No
[17]	No	Yes	No	No	Yes	No	Continuous	No
[19]	Yes	No	No	No	Yes	No	Continuous	No
[20]	Yes	No	No	No	No	H.264/AVC	Hyper-spectral	No
[21]	Yes	No	No	No	Yes	Taylor series	Continuous	No
[22]	Yes	No	No	No	Yes	AQC	Continuous	Yes
[23]	Yes	No	Yes	No	Yes	S+P transform	Continuous	No
[24]	Yes	No	No	No	No	LS based	Natural	No
[25]	Yes	Yes	No	No	Yes	No	Medical image	No
[26]	Yes	Yes	No	No	No	Color transform	Color	Yes
[27]	Yes	No	No	Yes	Yes	Geometric, photometric transformation	JPEG image	No
[28]	No	No	No	No	Yes	Dynamic bit reduction	Continuous	No
[29]	No	No	Block diff	No	No	No	Fractal	No
[30]	No	No	No	No	No	AQC (3-level)	Continuous	No
[31]	Yes	No	No	Yes	Yes	residual coding	All	No
[32]	Yes	No	No	No	No	residual coding	All	No
[33]	Yes	No	No	No	No	residual coding	All	No
[34]	Yes	No	No	No	Yes	residual coding	All	No
[35]	Yes	Yes	No	No	Yes	residual coding	All	No
[36]	Yes	Yes	No	No	Yes	Embedded block coding	All	No
[37]	Yes	No	No	No	Yes	Graph based transforms	All	No

2. Existing Image Compression Techniques

The JPEG-LS compression algorithm is suited for continuous tone images. The compression algorithm consists of four main parts, which are fixed predictor, bias canceller or adaptive corrector, context modeler and entropy coder [19]. In JPEG-LS, the edge detection is performed by “median edge detection” (MED) process [19]. JPEG-LS uses context modeling to measure the quantized gradient of surrounding image pixels. This context modeling of the predication error gives good results for images with texture pattern. Next correction values are added to the prediction error, and the remaining or residual error is encoded by Golomb coding [40] scheme. SPIHT [18,41] is an advanced encoding technique based on progressive image coding. SPIHT uses a threshold and encodes the most significant bit of the transformed image, followed by the application of increasing refinement. This paper considers SPIHT algorithm with lifting-based wavelet transform for 5/3 Le Gall wavelet filter.

Differential pulse code modulation (DPCM) [42] predictor can predict the current pixel based on its neighboring pixels as mentioned in the JPEG-LS predictor. The subtraction of the current pixel intensity and the predictor output gives predictor error e . The quantizer quantizes the error value using suitable quantization level. In case of lossless compression, the quantized level is unity. Next, an entropy coding is performed to get the final bit streams. The predictor operator can be expressed by the following equation

$$\hat{x}_s(i, j) = a * I(i, j - 1) + b * I(i - 1, j - 1) + c * I(i - 1, j) + d * I(i - 1, j + 1) + \dots \quad (1)$$

where \hat{x}_s is predictor output, the terms a , b , c and d are constant, I is the intensity value and (x, y) represent the spatial indices of the pixels.

Arithmetic coding is an entropy coding used for lossless compression [43]. In this method, the infrequently occurring symbols/characters are encoded with greater number of bits than frequent occurring symbols/characters. An important feature of Arithmetic coding is that it encodes the full information into a single long number and represents current information as a range. Huffman coding [44] is basically a prefix coding method which assigns variable length codes to input characters/symbols. In this scheme, the least frequently occurring character is assigned with the smallest of the codes within a code table.

3. Proposed Algorithms

This section describes the recently proposed ETEC method and then proposes the PTEC method.

3.1. ETEC

The study of ETEC is extended in this paper with a detailed analysis of the ETEC algorithm. It has already been mentioned in Section I that each pixel block of a pixelated image carries a single intensity value or a single piece of data. The pixel blocks have abrupt transition and thus have many directional edges. The ETEC method can be described by three steps. In the first step, the special feature of pixelated images is used to calculate a residual error ϵ by using the following intensity gradient

$$\nabla I = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \tag{2}$$

where $\partial I/\partial x$ is the derivative with respect to the x direction, $\partial I/\partial y$ is the derivative with respect to the y direction, I is the intensity value and (x,y) represent the spatial indices of the pixels. The maximum change of gradient between two co-ordinates represents the presence of edge either in the vertical or the horizontal direction.

The edge pixels are responsible for the increase in the level of the residual error ϵ . It can be noted that for the presence of vertical edges, the value of ϵ can be reduced to obtain the vertical intensity gradient. Similarly, for the presence of horizontal edges, the value of ϵ can be reduced to obtain the horizontal intensity gradient. In order to detect a strong edge, a threshold T_h is applied to the residual error in between the previous neighbors. If the previous residual error is greater than the threshold T_h , then the present pixel $I(x,y)$ is considered to be on the edge. So, the direction of gradient is changed. This can be mathematically described as:

$$\text{if } \epsilon = \frac{\partial I_x}{\partial x} > T_h \tag{3}$$

$$\text{then } \epsilon = \frac{\partial I_x}{\partial x} \tag{4}$$

$$\text{and if } \epsilon = \frac{\partial I_y}{\partial y} > T_h \tag{5}$$

$$\text{then } \epsilon = \frac{\partial I_y}{\partial y} \tag{6}$$

As long as the previous residual error is less than the threshold, i.e., $\epsilon < T_h$, the scanning direction remains the same. After the whole scanning, the term ϵ contains lower entropy compared to the original image.

In the second step of ETEC method, two matrices **A** and **B** are generated to encode ϵ . The dimensions of the matrix **A** is $X \times Y$. The possible values of matrix **A** are 0 or 1 or 2 depending on the value of ϵ . The matrix **A** is assigned a value of 0 where $\epsilon(x,y)$ has a value of 0. Moreover, the matrix **A** is assigned values of 1 and 2 where $\epsilon(x,y)$ has a value greater than and less than 0, respectively. On the other hand, the matrix **B** is assigned with the absolute value of $\epsilon(x,y)$, except

for $\epsilon(x, y) = 0$. After assigning the values for the two matrices, run-length coding [45] is applied to **A**. This coding is applied to the values whose corresponding run is greater than other values. This method manipulates bits of data to reduce the size and optimize input of the other algorithm. Figure 2 shows the block diagram of step 2 of ETEC method.

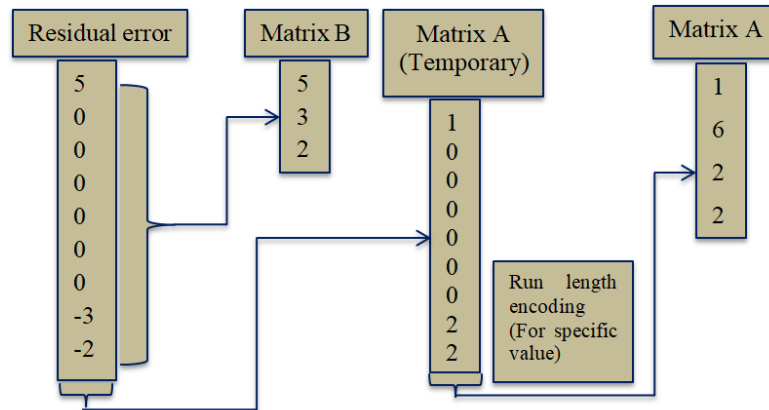


Figure 2. Modified J-bit encoding process.

In the third step of ETEC, Huffman or Arithmetic coding is applied to matrices **A** and **B**. Like other image compression methods, in ETEC, the general process of image decompression is just the opposite of compression. Figure 3 shows the flowchart of proposed ETEC algorithm.

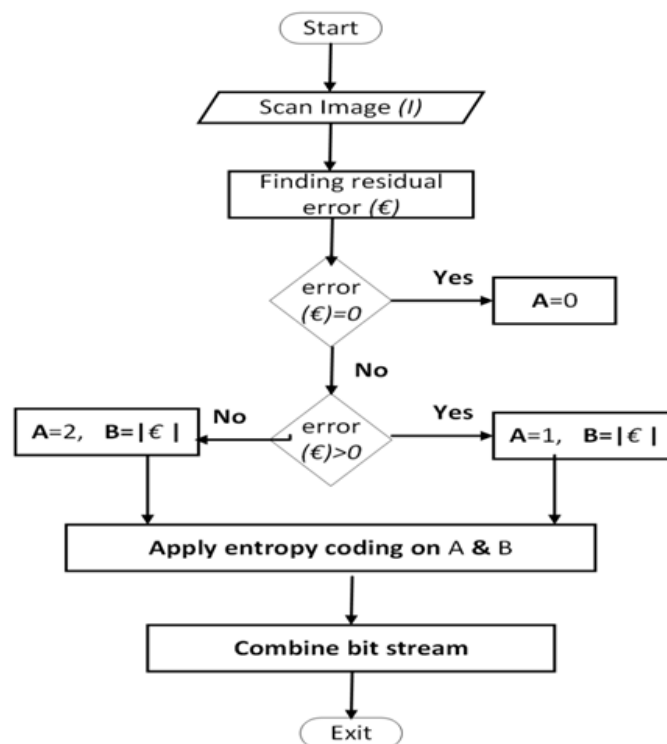


Figure 3. Flowchart of the “edge-based transformation and entropy coding” (ETEC) algorithm.

3.2. PTEC

The main purpose of the proposed PTEC algorithm is to optimize the compression ratio and computational time for pixelated images as well as for other continuous tone images. In the case of

a gray scale image, the signal variation is generally much smaller than that of a color image, but the intensity variation is still large near the edges of a gray scale image. For more accurate prediction of these signals and for accurate modeling of the prediction error, the hierarchical prediction scheme is used in PTEC. This method is described for the case where any image is divided into four subimages. At first, the gray scale image is decomposed into two subimages, i.e., a set of even numbered rows and a set of odd numbered rows, respectively. Figures 4 and 5 show the hierarchical decomposition of the input image X^0 . The input image is separated into two subimages: an even subimage X_e and an odd subimage X_o . Here the even subimage X_e is formed by gathering all even rows of the input image, and the odd subimage is formed of the collection of all odd rows of the input image. Each subimage is further divided into two subimages based on the even columns and the odd columns. Then X_{ee} is encoded and is used to predict the pixels in X_{eo} . In addition, X_{ee} is also used to estimate the statistics of prediction errors of X_{eo} . After encoding X_{ee} and X_{eo} , these are used to predict the pixels in X_{oe} . Furthermore, three subimages X_{ee} , X_{eo} , X_{oe} are used to predict a given subimage X_{oo} . With the increase in the number of subimages used to predict a given subimage, the probability of the prediction error may be decreased. To predict the pixels of the last subimage X_{oo} , maximum of eight (8) adjacent neighbors are used. This is evident from Figure 5. It can be noted that if the original image is divided into eight or more subimages instead of only four, the complexity and computation time will increase.

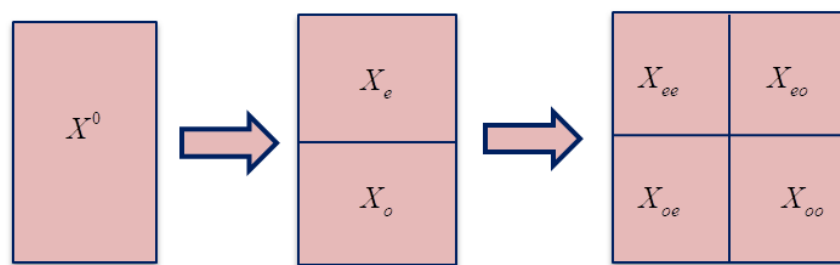


Figure 4. Illustration of hierarchical decomposition.

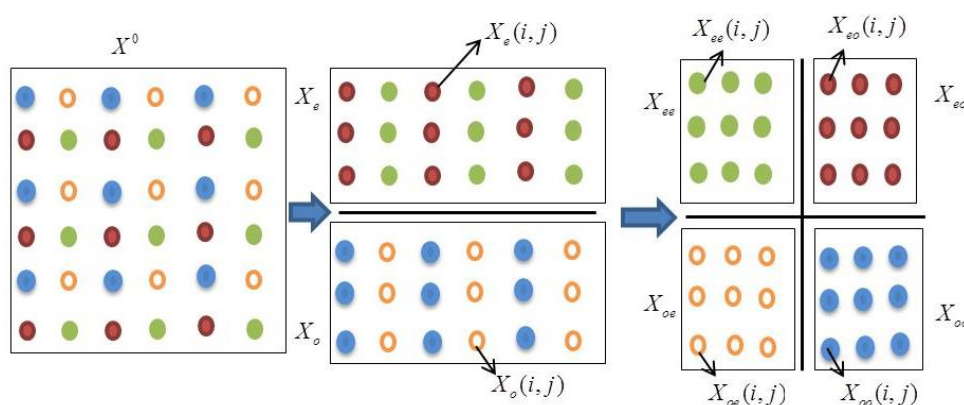


Figure 5. Input image and its decomposition image.

Suppose the image is scanned in a raster-scanning order; then the predictor is always based on its past casual neighbors (“context”). Figure 6 shows the order of the casual neighbors. The current pixels of the subimage X_{ee} are predicted based on the casual neighbors. A reasonable assumption made with this subimage source is the N th order Markovian property. This means in order to predict a pixel, N nearest casual neighbors are required. Then the prediction of current pixel $x(n)$ is predicted as follows:

$$\hat{X}(n) = \sum_{k=1}^N a(k)X(n - k) \tag{7}$$

where $a(k)$ is the prediction coefficient, and $X(n - k)$ is the neighbors of $X(n)$. For the prediction of X_{eo} pixels using X_{ee} , directional prediction is attached to avoid large prediction errors near the edge. For each pixel $X_{eo}(i, j)$ in X_{eo} , the horizontal predictor $\hat{X}_v(i, j)$ and vertical predictor $\hat{X}_h(i, j)$ are defined as shown in the following. Both $\hat{X}_v(i, j)$ and $\hat{X}_h(i, j)$ are determined by calculating the average of two different predictions. First, consider the case for $\hat{X}_{h1}(i, j)$. The prediction value, $\hat{X}_{h1}(i, j)$, is expressed as

$$\hat{X}_{h1}(i, j) = X_{eo}(i, j - 1) + \text{round} \left\{ \frac{X_{eo}(i - 1, j - 1) - X_{eo}(i - 1, j)}{2} \right\} \quad (8)$$

The second prediction value, $\hat{X}_{h2}(i, j)$, is expressed as

$$\hat{X}_{h2}(i, j) = \text{round} \left\{ \frac{X_{ee}(i, j) + X_{ee}(i, j + 1)}{2} \right\} \quad (9)$$

Now, the term $\hat{X}_h(i, j)$ is determined using the average of $\hat{X}_{h1}(i, j)$ and $\hat{X}_{h2}(i, j)$ as follows:

$$\hat{X}_h(i, j) = \text{round} \left\{ \frac{\hat{X}_{h1}(i, j) + \hat{X}_{h2}(i, j)}{2} \right\} \quad (10)$$

Similarly, the term $\hat{X}_v(i, j)$ can be expressed as follows:

$$\hat{X}_v(i, j) = X_{eo}(i - 1, j) + \text{round} \left\{ \frac{(X_{ee}(i - 1, j) - X_{ee}(i, j)) + (X_{ee}(i - 1, j + 1) - X_{ee}(i, j + 1))}{4} \right\} \quad (11)$$

Among these, one is selected as a predictor for $X_{eo}(i, j)$ from Equations (10) and (11). With these possible two predictors, the most common approach to encoding is mode selection; where better predictor for each pixel is selected and the mode selection is dependent on the vertical and horizontal edges. If $|X_{eo}(i, j) - \hat{X}_h(i, j)|$ is smaller than $|X_{eo}(i, j) - \hat{X}_v(i, j)|$, the horizontal edge is stronger than the vertical edge. Otherwise, the vertical edge is stronger than horizontal edge. For the prediction of X_{oe} using X_{ee} and X_{eo} , the vertical and horizontal edges as well as diagonal edges can be suitably predicted. For each pixel $X_{oe}(i, j)$ in X_{oe} , the horizontal predictor $\hat{X}_h(i, j)$, vertical predictor $\hat{X}_v(i, j)$, and diagonal predictor $\hat{X}_{dl}(i, j)$ (left), $\hat{X}_{dr}(i, j)$ (right) are defined in the following. Again, $\hat{X}_v(i, j)$, $\hat{X}_h(i, j)$, $\hat{X}_{dl}(i, j)$ and $\hat{X}_{dr}(i, j)$ are determined by taking the average of two different predictions. The term $\hat{X}_h(i, j)$ is determined as follows

$$\hat{X}_h(i, j) = X_{oe}(i, j - 1) + \text{round} \left\{ \frac{(X_{ee}(i, j - 1) - X_{ee}(i, j)) + (X_{ee}(i + 1, j - 1) - X_{ee}(i + 1, j))}{4} \right\} \quad (12)$$

Now, consider the case for $\hat{X}_v(i, j)$. The first prediction value, $\hat{X}_{v1}(i, j)$, is expressed as

$$\hat{X}_{v1}(i, j) = X_{ee}(i, j) + \text{round} \left\{ \frac{(X_{eo}(i, j - 1) - X_{eo}(i + 1, j - 1)) + (X_{eo}(i, j) - X_{eo}(i + 1, j))}{4} \right\} \quad (13)$$

The second prediction value, $\hat{X}_{v2}(i, j)$, is expressed as

$$\hat{X}_{v2}(i, j) = \text{round} \left\{ \frac{X_{ee}(i, j) + X_{ee}(i + 1, j)}{2} \right\} \quad (14)$$

The term $\hat{X}_v(i, j)$ is determined using the average of $\hat{X}_{v1}(i, j)$ and $\hat{X}_{v2}(i, j)$ as follows:

$$\hat{X}_v(i, j) = \text{round} \left\{ \frac{\hat{X}_{v3}(i, j) + \hat{X}_{v4}(i, j)}{2} \right\} \quad (15)$$

Now, consider the case for $\hat{X}_{dr}(i, j)$. The first prediction value, $\hat{X}_{dr1}(i, j)$, is expressed as

$$\hat{X}_{dr1}(i, j) = X_{eo}(i, j) + \text{round} \left\{ \frac{(X_{ee}(i, j) - X_{ee}(i + 1, j - 1)) + (X_{ee}(i, j + 1) - X_{ee}(i + 1, j))}{4} \right\} \quad (16)$$

The second prediction value, $\hat{X}_{dr2}(i, j)$, is expressed as

$$\hat{X}_{dr2}(i, j) = \text{round} \left\{ \frac{X_{eo}(i, j) + X_{eo}(i + 1, j - 1)}{2} \right\} \quad (17)$$

The term $\hat{X}_{dr}(i, j)$ is determined using the average of $\hat{X}_{dr1}(i, j)$ and $\hat{X}_{dr2}(i, j)$ as follows:

$$\hat{X}_{dr}(i, j) = \text{round} \left\{ \frac{\hat{X}_{dr1}(i, j) + \hat{X}_{dr2}(i, j)}{2} \right\} \quad (18)$$

Now, consider the case for $\hat{X}_{dl}(i, j)$. The first prediction value, $\hat{X}_{dl1}(i, j)$, is expressed as

$$\hat{X}_{dl1}(i, j) = X_{eo}(i, j - 1) + \text{round} \left\{ \frac{(X_{ee}(i, j - 1) - X_{ee}(i + 1, j)) + (X_{ee}(i, j) - X_{ee}(i + 1, j + 1))}{4} \right\} \quad (19)$$

The second prediction value, $\hat{X}_{dl2}(i, j)$, is expressed as

$$\hat{X}_{dl2}(i, j) = \text{round} \left\{ \frac{X_{eo}(i, j - 1) + X_{eo}(i + 1, j)}{2} \right\} \quad (20)$$

The term $\hat{X}_{dl}(i, j)$ is determined using the average of $\hat{X}_{dl1}(i, j)$ and $\hat{X}_{dl2}(i, j)$ as follows:

$$\hat{X}_{dl}(i, j) = \text{round} \left\{ \frac{\hat{X}_{dl1}(i, j) + \hat{X}_{dl2}(i, j)}{2} \right\} \quad (21)$$

Moreover, the selection of predictor is dependent on the presence of the directivity of the strong edges. By using Equations (12) and (21), it is possible to find an edge with a specified direction. Next, the residual error is encoded using modified J bit encoding. At the final stage, entropy coding is applied to the J bit encoded data.

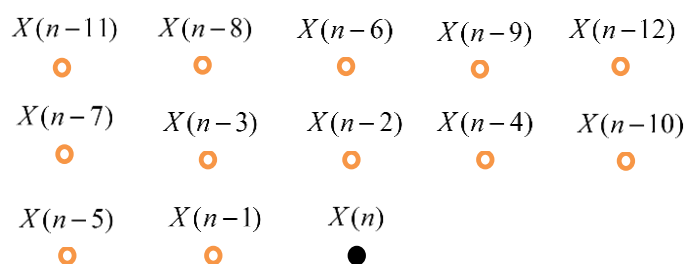


Figure 6. Ordering of the casual neighbors.

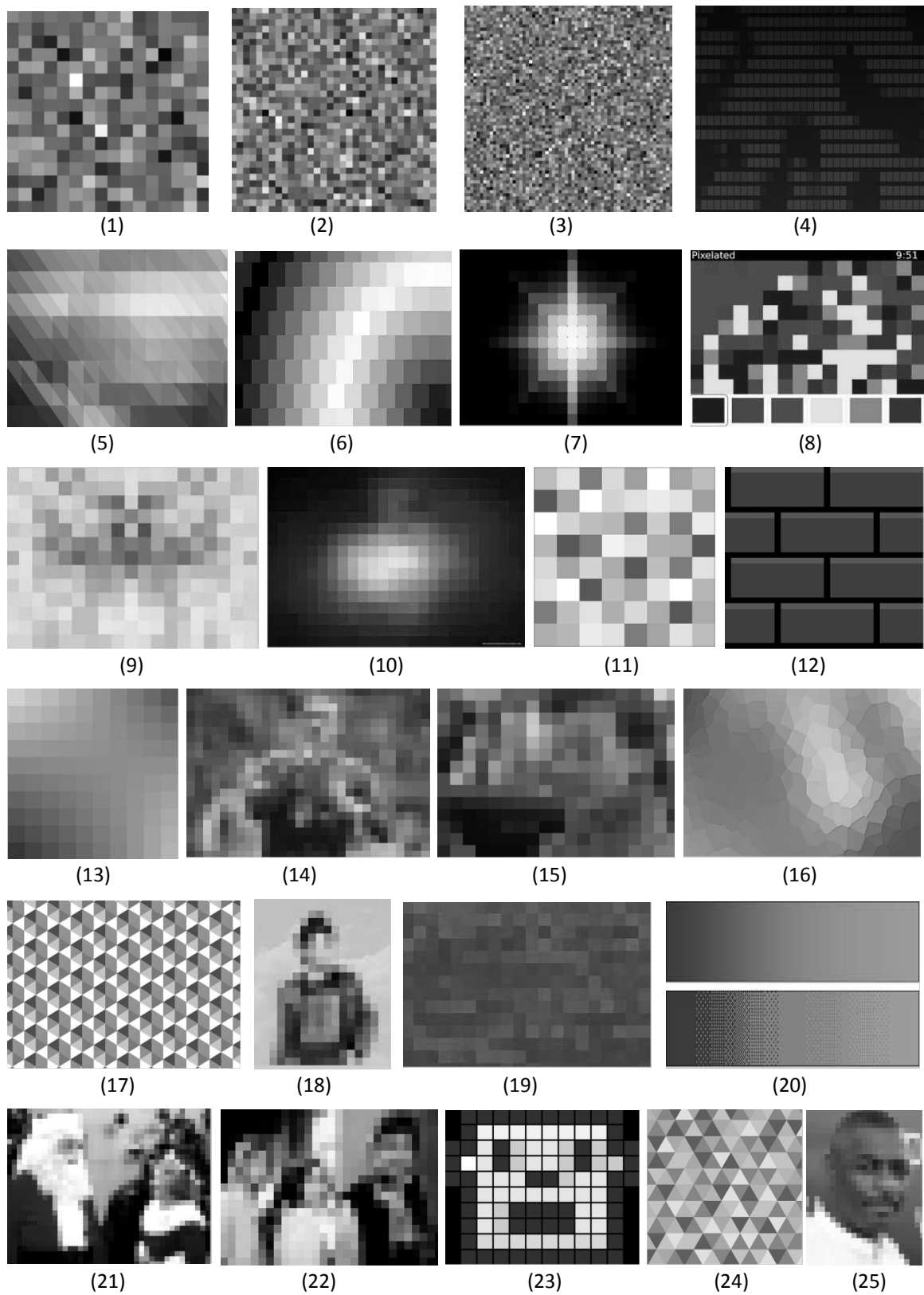
4. Results and Discussion

This section evaluates the performance of ETEC and PTEC schemes for various types of images. The evaluation is done with the help of MATLAB tool and computer having specifications of Intel core i3 (Intel, Shanghai, China), 3110 M 2.4 Hz processor, RAM 4 GB (Kingston, Shanghai, China), 1 GB VGA graphics card (Intel, Shanghai, China) and Windows 7 (32 bits) operating system (Microsoft, Shanghai, China). The intensity levels of the images are from 0 to 255 and the threshold term T_h is assumed to have a value of 20. It can be noted that this value of T_h has been selected as a near optimal

value. Since a high value of T_h may not recognize some edges in the images, whereas low values of T_h may unnecessarily consider any small transition as an edge

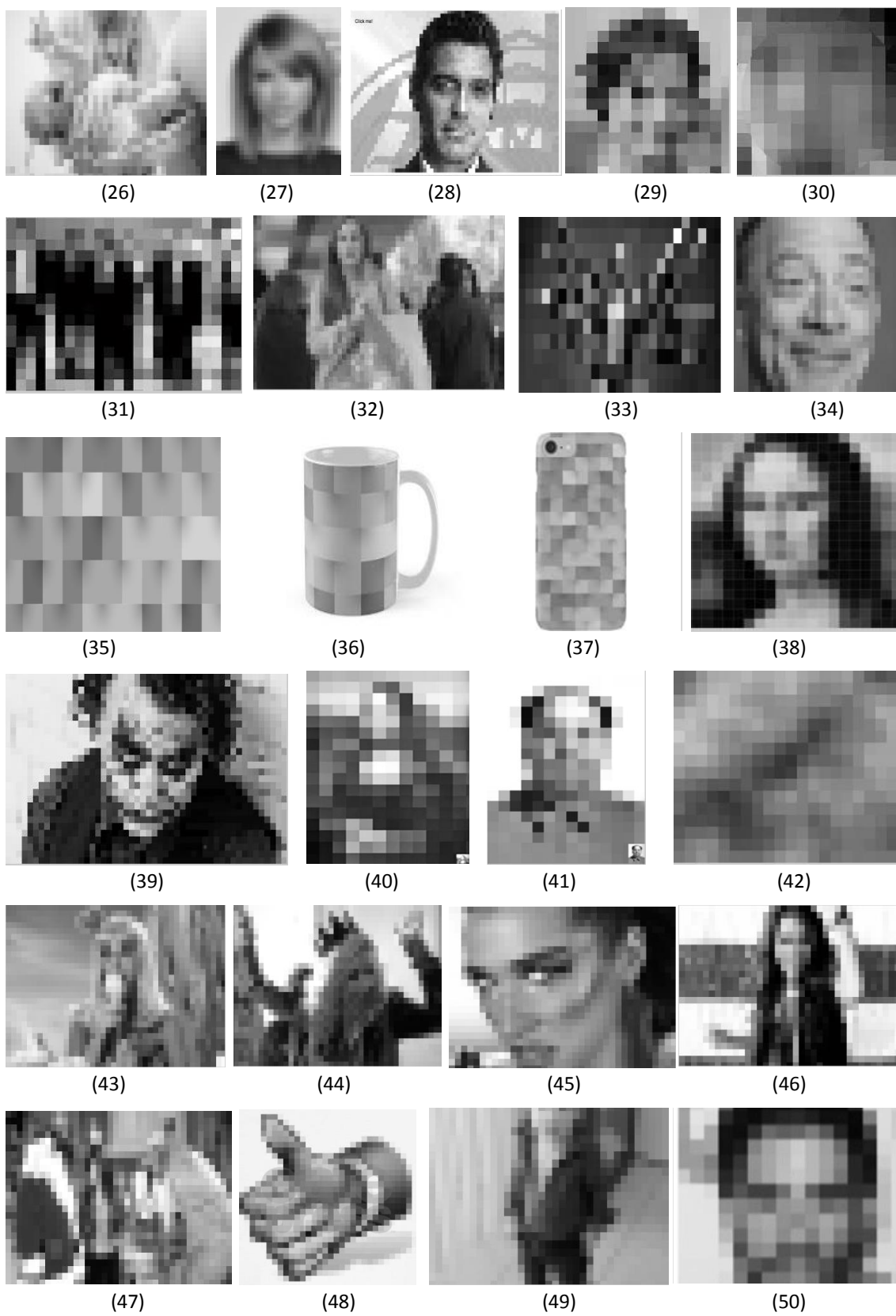
Figure 7 shows 50 different types of pixelated images used for evaluating the compression algorithms. Some of these images are created using MATLAB tool, and the remaining ones are available in [46–49]. Both Figure 7a,b have 25 images each. These images are made of different pixel blocks and each block is of different pixel sizes. Each pixel block has uniform intensity level. In some cases, a pixelated image may have very small pixel block or no block (each pixel block made of one pixel only). A number of metrics such as compression ratio, bits per pixel, saving percentage [28], and computation time are considered for comparing the algorithms. It can be noted that in this study the compression ratio is defined as the ratio of the size of the original image to the compressed image. Moreover, the saving percentage parameter is the difference between the original image and the compressed image as a percentage of the original image. Mathematically, the compression ratio is C_1/C_2 , and the saving percentage is $(1 - C_2/C_1)$ where C_1 and C_2 are the size of the original image and the compressed image, respectively. The bit per pixel parameter is obtained by dividing the image size (in bytes) by the number of pixels in the compressed image. The computation time is the total amount of time required to perform the image compression using MATLAB tool with the given computer specified earlier in this section.

First, consider the compression ratio (denoted as CR) and bits/pixel parameters. In Table 2, compression ratio and bits/pixel metrics are compared for the proposed ETEC and PTEC techniques with the existing JPEG-LS, SPIHT, and DPCM methods. The comparison is done for the 50 pixelated images illustrated in Figure 7. The bits per pixel parameter of the first 15 images are plotted in Figure 8 for the proposed and existing compression algorithms. Now consider the saving percentage and computation time. Table 3 represents the percentage saving and computation time for the case of those 50 images. The computation time in seconds is also plotted for the first 15 images in Figure 9. It can be seen from Table 2 that for the pixelated images, the average bits per pixel for ETEC (0.299) and PTEC (0.592) are lower (better) than the existing JPEG-LS (0.836), SPIHT (2.105) and DPCM (2.17). Table 2 shows that the average compression ratio of ETEC (29.39) and PTEC (10.28) are better than SPIHT (3.178) and JPEG-LS (9.264) and DPCM (3.09). Table 2 also shows that the compression ratio of PTEC is not always better than JPEG-LS for all the 50 pixelated images. In particular, PTEC has better compression than JPEG-LS for pixelated images having large pixel blocks. For small pixel blocks, the compression performance of PTEC is worse than JPEG-LS. This is because of the hierarchical prediction of PTEC. In case of small pixel block images, the prediction error for the first subimage is very high due to the high randomness of pixel intensity. For large pixel block images, this problem is significantly reduced. Table 3 indicates that the computational time of ETEC (62.58 s) is worse than SPIHT (13.9 s), but better than JPEG-LS (526 s) and DPCM (17.48 s). Furthermore, PTEC method has a computation time of 18.406 s which is much better than ETEC (62.58 s) and comparable to SPIHT (13.9 s). So, for pixelated images and for the case where both compression and computation time are important, PTEC may be more suitable than ETEC, SPIHT, JPEG-LS and DPCM.



(a) Pixelated images: part 1.

Figure 7. Cont.



(b) Pixelated images: part 2.

Figure 7. Tested pixelated images [46–49].

Table 2. Comparison of bits per pixel and compression ratio for pixelated images.

Tested Image Name	JPEG-LS		Le Gall 5/3 + SPIHT (Subblock)		ETEC (Arithmetic)		PTEC		DPCM	
	Bits/Pixel	CR	Bits/Pixel	CR	Bits/Pixel	CR	Bits/Pixel	CR	Bits/Pixel	CR
1	0.573	13.95	2.105	3.8	0.162	49.52	0.393	20.35	2.29	3.49
2	0.715	11.19	2.822	2.84	0.176	45.42	0.812	9.85	2.70	2.96
3	1.236	6.47	1.084	7.38	0.493	16.23	2.128	3.76	3.98	2.01
4	1.202	6.65	3.021	2.65	1.145	6.98	1.702	4.70	3.25	2.46
5	0.814	9.83	2.465	3.25	0.764	10.48	0.905	8.84	1.89	4.22
6	0.834	9.59	2.085	3.84	0.457	17.52	0.783	10.22	2.61	3.06
7	0.964	8.3	1.495	5.35	0.734	10.9	1.08	7.41	2.10	3.81
8	2.098	3.81	3.172	2.52	1.527	5.24	2.241	3.57	4.14	1.93
9	1.744	4.58	2.846	2.81	1.271	6.29	1.956	4.09	3.70	2.16
10	1.618	4.94	2.555	3.13	1.36	5.88	1.831	4.37	3.52	2.27
11	0.282	28.4	2.462	3.25	0.046	175	0.166	48.21	1.39	5.72
12	0.297	26.98	5.01	1.59	0.027	297	0.183	43.65	2.11	3.79
13	0.879	9.09	2.017	3.97	0.385	20.77	0.843	9.49	0.96	8.31
14	1.055	7.58	3.724	2.15	0.241	33.24	0.502	15.94	2.56	3.12
15	0.836	9.56	2.105	3.8	0.299	26.73	0.592	13.52	2.17	3.68
16	1.612	4.96	2.695	2.97	1.703	4.70	1.751	4.57	2.465	3.25
17	2.279	3.51	4.662	1.72	2.396	3.34	2.312	3.46	3.714	2.15
18	2.265	3.53	3.295	2.43	2.209	3.62	2.281	3.51	4.020	1.99
19	1.723	4.64	2.536	3.15	1.509	5.30	1.699	4.71	3.173	2.52
20	2.838	2.82	3.669	2.18	2.422	3.30	2.689	2.98	4.348	1.84
21	0.713	11.23	2.172	3.68	0.713	11.23	1.117	7.16	3.030	2.64
22	0.094	85.47	1.623	4.93	0.094	85.47	0.305	26.24	1.774	4.51
23	0.778	10.28	2.462	3.25	0.254	31.51	0.783	10.22	3.237	2.47
24	1.588	5.04	3.827	2.09	1.865	4.29	1.599	5.00	3.030	2.64
25	1.539	5.20	2.963	2.70	1.594	5.02	1.597	5.01	1.770	4.52
26	1.398	5.72	2.669	3.00	0.948	8.44	1.377	5.81	2.180	3.67
27	1.590	5.03	3.499	2.29	1.608	4.98	1.594	5.02	1.932	4.14
28	0.683	11.72	2.492	3.21	0.087	92.28	0.914	8.75	1.946	4.11
29	1.750	4.57	2.724	2.94	1.264	6.33	1.729	4.63	3.721	2.15
30	1.678	4.77	2.617	3.06	1.376	5.81	1.633	4.90	3.540	2.26
31	1.468	5.45	2.006	3.99	1.931	4.14	1.882	4.25	4.494	1.78
32	2.900	2.76	3.724	2.15	3.067	2.61	3.030	2.64	4.571	1.75
33	0.422	18.95	1.868	4.28	0.025	326.28	0.086	92.55	1.460	5.48
34	1.257	6.36	2.346	3.41	0.766	10.45	1.192	6.71	2.606	3.07
35	0.876	9.14	2.075	3.86	0.803	9.96	0.872	9.17	1.843	4.34
36	1.198	6.68	2.271	3.52	1.118	7.16	1.236	6.47	2.410	3.32
37	1.586	5.04	2.759	2.90	1.630	4.91	1.590	5.03	3.053	2.62
38	1.714	4.67	2.756	2.90	3.101	2.58	2.920	2.74	4.908	1.63
39	1.965	4.07	2.985	2.68	1.889	4.23	1.961	4.08	4.020	1.99
40	1.701	4.70	3.105	2.58	1.727	4.63	1.766	4.53	3.478	2.30
41	0.990	8.08	2.466	3.24	0.847	9.44	0.986	8.11	2.540	3.15
42	1.510	5.30	2.630	3.04	1.180	6.78	1.594	5.02	3.376	2.37
43	1.078	7.42	2.525	3.17	0.742	10.79	1.077	7.43	2.508	3.19
44	1.135	7.05	2.522	3.17	0.832	9.62	1.105	7.24	2.581	3.10
45	0.988	8.09	2.298	3.48	0.608	13.16	0.974	8.21	2.410	3.32
46	0.944	8.48	2.321	3.45	0.601	13.31	0.905	8.84	2.332	3.43
47	1.560	5.13	3.189	2.51	1.242	6.44	1.610	4.97	3.463	2.31
48	1.428	5.60	2.889	2.77	1.141	7.01	1.518	5.27	3.125	2.56
49	1.792	4.46	3.197	2.50	1.924	4.16	1.900	4.21	3.419	2.34
50	1.261	6.34	2.362	3.39	0.903	8.86	1.252	6.39	2.888	2.77
Average	0.836	9.264	2.105	3.178	0.299	29.39	0.592	10.28	2.17	3.09

Table 3. Comparison of percentage saving and computation time for pixelated images.

Tested Image No.	JPEG-LS		Le Gall 5/3 + SPIHT (Subblock)		ETEC (Arithmetic)		PTEC		DPCM	
	Saving %	Time (s)	Saving %	Time (s)	Saving %	Time (s)	Saving %	Time (s)	Saving %	Time (s)
1	92.83	295	73.68	20.8	97.98	134.5	95.09	22.93	71.33	31.5
2	91.06	309	64.79	16.7	97.8	133.65	89.85	28.35	66.23	35.05
3	84.54	495	86.45	38.5	93.84	126.74	73.40	49.21	50.24	48.23
4	84.96	751	62.26	18.72	85.69	92.12	78.72	30.42	59.39	29.26
5	89.83	389	69.23	17.52	90.46	78.63	88.69	20.82	76.32	19.64
6	89.57	501	73.96	14.94	94.29	74.13	90.22	19.31	67.32	25.25
7	87.95	405	81.31	7.25	90.83	43.01	86.50	17.92	73.76	17.44
8	73.75	235	60.32	3.55	80.91	3.76	71.99	9.9	48.18	9.34
9	78.2	246	64.41	3.61	84.12	3.66	75.55	7.02	53.81	8.66
10	79.76	247	68.05	3.40	83.00	3.57	77.12	8.23	56.00	7.95
11	96.48	114	69.23	2.34	99.43	16.18	97.93	5.53	82.51	9.17
12	96.29	92	37.3	7.44	99.66	16.03	97.71	5.57	73.63	12.75
13	89.01	840	74.81	24.84	95.19	164.5	89.46	15.01	87.97	7.26
14	86.82	862	53.49	34.5	96.99	221.58	93.73	44.17	67.95	38.56
15	91.47	407	68.85	21.22	98.92	93.44	92.60	3.14	72.84	45.29
16	79.84	788	66.31	18.71	78.71	98.08	78.12	20.82	69.19	15.67
17	71.51	779	41.72	25.15	70.05	102.80	71.10	35.04	53.57	28.52
18	71.69	180	58.81	2.49	72.39	2.77	71.49	4.8	49.75	4.23
19	78.47	196	68.29	2.76	81.14	2.70	78.77	5.09	60.33	4.35
20	64.53	277	54.14	3.79	69.73	4.68	66.39	5.61	82.17	3.60
21	85.23	149	72.85	2.17	86.03	1.26	86.03	3.89	62.12	2.78
22	94.85	625	79.71	25.72	98.83	255.55	96.19	39.48	77.83	33.47
23	90.27	360	69.22	11.10	96.83	31.93	90.21	12.72	59.54	9.52
24	80.16	370	52.16	11.24	76.68	23.60	80.02	5.57	62.12	3.75
25	82.52	833	66.63	18.71	88.15	95.19	82.79	25.75	72.75	22.43
26	80.76	1130	62.96	24.16	80.08	137.40	80.04	38.19	77.88	31.36
27	80.12	1152	56.27	27.60	79.90	142.31	80.08	39.71	75.85	35.09
28	91.46	407	68.85	21.22	98.92	93.44	88.57	27.05	75.67	23.68
29	78.13	234	65.96	3.21	84.20	3.40	78.39	5.47	53.49	3.39
30	81.64	291	74.92	4.89	75.86	4.64	76.47	7.52	43.82	23.5
31	63.75	326	53.45	3.86	61.66	8.09	62.12	6.85	42.86	31.52
32	79.02	173	67.29	2.43	82.80	2.41	79.59	4.09	55.75	6.15
33	94.72	118	76.65	6.98	99.69	16.48	98.92	11.32	81.75	5.43
34	84.28	1600	70.68	35.25	90.43	27.98	85.10	43.52	67.43	11.27
35	89.05	784	74.07	22.64	89.96	121.54	89.09	29.46	76.96	18.32
36	85.03	170	71.61	3.32	86.03	3.34	84.54	5.09	69.88	3.05
37	80.18	118	65.51	1.92	79.63	2.18	80.12	3.71	61.83	2.54
38	78.57	174	65.55	1.78	61.24	2.66	63.50	3.15	38.65	2.71
39	75.44	567	62.69	6.93	76.38	14.65	75.49	9.37	49.75	8.52
40	78.74	723	61.19	10.65	78.41	22.53	77.92	15.80	56.52	12.63
41	87.62	390	69.18	9.91	89.41	21.68	87.67	15.24	68.25	13.04
42	81.13	275	67.13	3.63	85.25	4.00	80.08	5.40	57.81	3.96
43	86.53	966	68.43	23.45	90.73	65.22	86.54	29.43	68.65	23.79
44	85.81	914	68.47	24.47	89.60	121.80	86.19	32.26	67.74	25.64
45	87.65	922	71.28	22.10	92.40	131.33	87.82	32.04	69.88	25.54
46	88.21	859	70.99	22.34	92.49	133.39	88.69	31.47	70.85	24.98
47	80.50	602	60.14	9.10	84.47	18.17	79.88	13.85	56.71	11.07
48	82.15	673	63.89	11.30	85.73	24.07	81.02	18.90	60.94	14.00
49	77.60	1254	60.03	20.83	75.95	119.82	76.25	27.27	57.26	21.77
50	84.24	717	70.47	13.68	88.71	62.66	84.35	17.86	63.90	14.45
Average	83.48	526	66.11	13.90	86.15	62.58	82.76	18.406	64.54	17.42

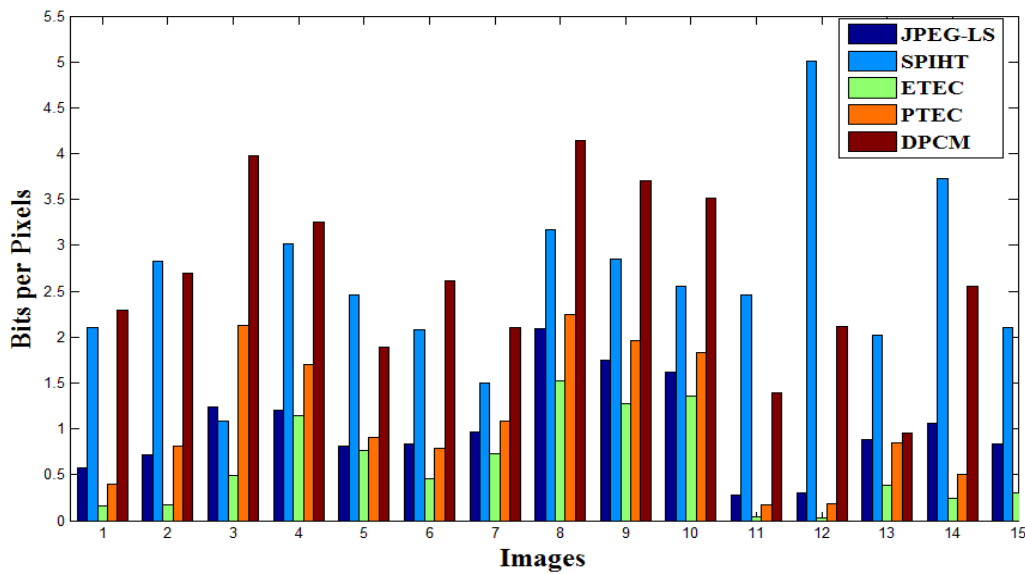


Figure 8. Comparison of bits per pixels for pixelated images.

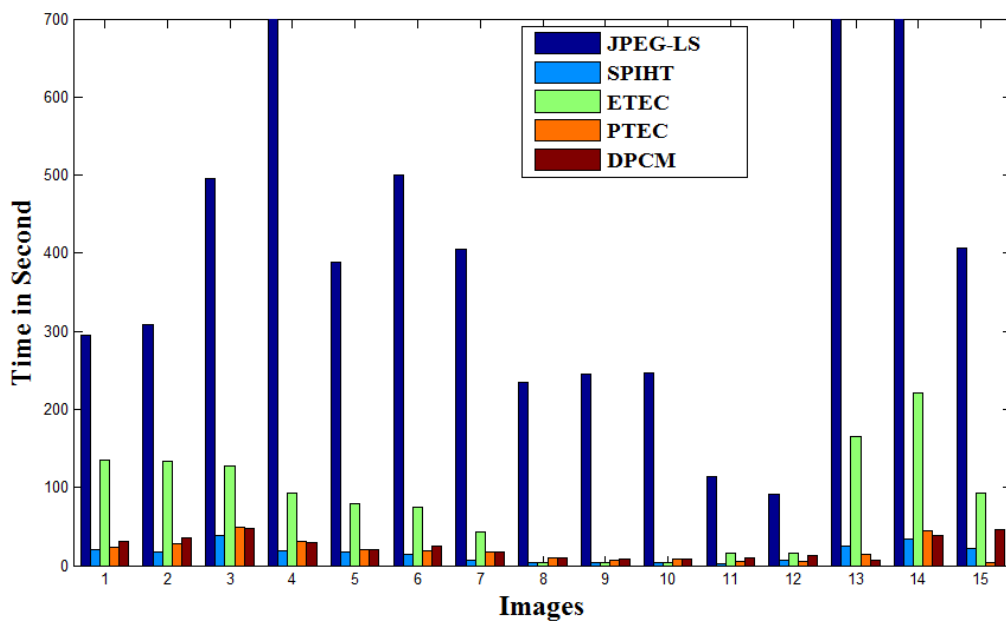


Figure 9. Comparison of computation time for pixelated images.

In the following, different compression algorithms are shown for standard images particularly, non-pixelated images. Figure 10 illustrates eight standard test images available in [50–54]. These images have a resolution of 512×512 pixels. All these eight images are used to test compression ratio of different algorithms. For example, the Lena image is of 2,097,152 bits, but this image results in 1,063,464, 1,399,968, 1,170,220, 1,145,985 and 1,263,345 bits by using compression schemes of JPEG-LS, SPIHT, ETEC, PTEC and DPCM, respectively. Therefore, for the use of JPEG-LS on Lena image, the compression ratio is 1.972 ($2,097,152 / 1,063,464$) and bits/pixel is 4.0567 ($1,063,464 / 512 \times 512$). Similarly, the compression ratio and bits/pixel values for different algorithms on different images can be easily obtained. These values are summarized in Table 4. Table 4 shows the comparison of compression ratio and bits per pixel representation of the ETEC and PTEC techniques with the existing JPEG-LS, SPIHT and DPCM for non-pixelated images. Figures 11 and 12 are the corresponding visual representation of Table 4 for the case of bits per pixel and compression ratio, respectively. When the

average compression ratio is considered, PTEC (2.06) is better than SPIHT (1.76), ETEC (1.93) and DPCM (1.72), but worse than JPEG-LS (2.16). Similarly, PTEC is better than SPIHT, ETEC and DPCM but worse than JPEG-LS in terms of average bits/pixel metric. Table 5 represents the percentage of saving area and computation time for the compression algorithms. It can be seen from Table 5 that the PTEC is better than SPIHT, ETEC and DPCM but worse than JPEG-LS in terms of percentage saving metric. Table 5 also shows that for the non-pixelated images, the average computation time of PTEC (74.50 s) is comparable to SPIHT (43.45 s) and DPCM (43.48 s), but better than ETEC (347.44 s) and JPEG-LS (2279.36 s). Note that PTEC has much better computation time than ETEC. This is because the use of hierarchical approach in PTEC. In the hierarchical approach, the computational data matrix is reduced to $\frac{1}{4}$ of the original data matrix. To handle a smaller matrix requires less time than handling a large one.

So, for non-pixelated images and for the case where both compression and computation time are important, PTEC, SPIHT and DPCM may be more suitable than ETEC and JPEG-LS.

Table 4. Comparison of bits per pixel and compression ratio for non-pixelated images.

Tested Image No.	JPEG-LS		Le Gall 5/3 + SPIHT (Subblock)		ETEC (Arithmetic)		PTEC		DPCM	
	Bits/Pixel	CR	Bits/Pixel	CR	Bits/Pixel	CR	Bits/Pixel	CR	Bits/Pixel	CR
lena	4.0567	1.972	5.3404	1.498	4.464	1.792	4.364	1.83	4.81	1.66
peppers	4.5050	1.775	5.1543	1.552	4.998	1.601	4.704	1.70	4.84	1.65
ankle	2.93	2.73	3.704	2.16	3.265	2.45	2.694	2.97	3.940	2.03
brain	2.52	3.17	3.54	2.26	3.019	2.65	2.74	2.92	3.791	2.11
mri_top	3.60	2.22	4.372	1.83	3.922	2.04	3.774	2.12	4.678	1.71
boat	4.8618	1.645	5.4581	1.465	5.15	1.553	5.067	1.58	5.35	1.49
barbara	4.8280	1.657	5.3452	1.496	5.559	1.439	5.402	1.48	5.74	1.39
house	3.8535	2.076	4.4817	1.785	4.176	1.915	4.227	1.89	4.64	1.72
Average	3.89	2.16	4.67	1.76	4.32	1.93	4.12	2.06	4.72	1.72

Table 5. Comparison of percentage saving and computation time for non-pixelated images.

Tested Image No.	JPEG-LS		Le Gall 5/3 + SPIHT (Subblock)		ETEC (Arithmetic)		PTEC		DPCM	
	Saving %	Time (s)	Saving %	Time (s)	Saving %	Time (s)	Saving %	Time (s)	Saving %	Time (s)
lena	49.29	2934	33.24	59.7	44.2	330.26	45.451	97.16	39.92	53.70
peppers	43.69	1705	35.57	76.27	37.527	390.89	41.197	104.6	39.52	54.87
ankle	63.37	1548.7	53.70	22.13	59.18	398.25	66.33	37.63	50.74	27.29
brain	68.45	1761.67	55.75	36.62	62.26	500.39	65.75	61.43	52.61	29.94
mri_top	54.95	1941.48	45.36	34.25	50.98	393.27	52.83	62.12	41.52	33.76
boat	39.23	3352	31.77	58.19	35.625	397	36.665	106.3	33.03	60.86
barbara	39.65	4037	33.18	50.70	30.517	355.47	32.473	109.1	28.22	72.41
house	51.83	955	43.98	9.7	47.794	14.13	47.163	17.6	41.96	15.01
Average	51.30	2279.36	41.57	43.45	46.01	347.44	48.48	74.50	40.85	43.48

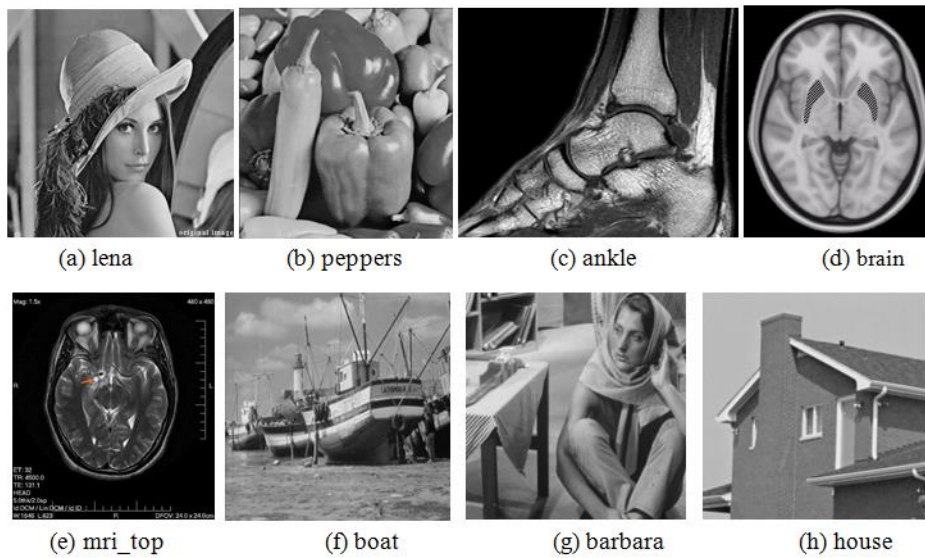


Figure 10. Standard test images: (a) lena [51] (b) peppers [51] (c) ankle [52] (d) brain [53] (e) Mri_top [54] (f) boat [51] (g) barbara [50] (h) house [51].

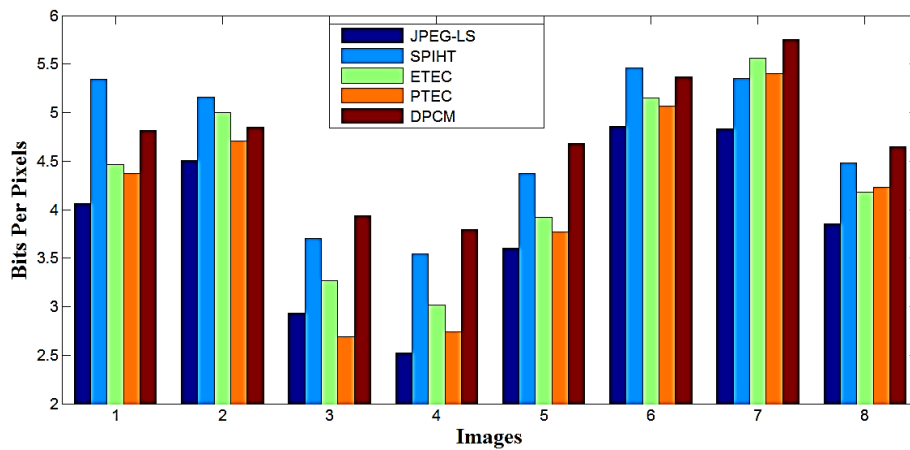


Figure 11. Comparison of bits per pixels for non-pixelated images.

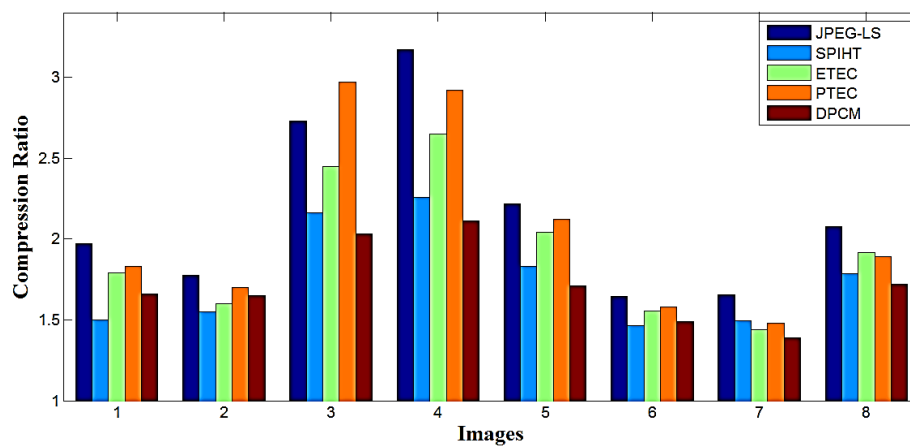


Figure 12. Comparison of compression ratio for non-pixelated images.

5. Conclusions

This work describes two algorithms for compression of images, particularly pixelated images. One algorithm is termed as ETEC, which has recently been conceptualized by the authors of this paper. The other one is a prediction-based new algorithm termed as PTEC. The ETEC and PTEC techniques are compared with the existing JPEG-LS, SPIHT, and DPCM methods in terms of compression ratio and computation time. For the case of pixelated images, the compression ratio for PTEC is around 10.28, which is worse than ETEC (29.39) but better than JPEG-LS (9.264), SPIHT (3.178), and DPCM (3.09). In particular, for images having large pixel-blocks, the PTEC method provides a much greater compression ratio than JPEG-LS. In terms of average computational time, the PTEC (18.406 s) is comparable with SPIHT (13.90 s) and DPCM (17.42 s) for pixelated images, and better than JPEG-LS (526 s) and ETEC (62.58 s). The compression ratio of PTEC (2.06) for non-pixelated images is comparable with JPEG-LS (2.16), but better than SPIHT (1.76), ETEC (1.93), and DPCM (1.72). Therefore, for the cases where compression ratios, as well as computational time, are required and for the case of pixelated images, PTEC is a better choice than ETEC, JPEG-LS, SPIHT, and DPCM. Moreover, for the case of non-pixelated images, PTEC, along with DPCM and SPIHT, are better choices than ETEC and JPEG-LS when both compression ratio and computational time are important. Therefore, PTEC is an attractive candidate for lossless compression of standard images including pixelated and non-pixelated images. The proposed PTEC method may be modified in future by applying an error correction algorithm to the prediction error caused by hierarchical prediction. The resultant values will be encoded by JBE and entropy coder as usual.

Author Contributions: M.A.K. performed the study under the guidance of M.R.H.M. Both M.A.K. and M.R.H.M. wrote the paper.

Funding: This research received no external funding.

Acknowledgments: This work is a part of Master's thesis of the author M.A.K. under the supervision of the author M.R.H.M. submitted to the Institute of Information and Communication Technology (IICT) of Bangladesh University of Engineering and Technology (BUET). Therefore, the authors would like to thank IICT, BUET for providing research facilities.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Koc, B.; Arnavut, Z.; Kocak, H. Lossless compression of dithered images. *IEEE Photonics J.* **2013**, *5*, 6800508. [[CrossRef](#)]
2. Jain, A.K. Image data compression: A review. *Proc. IEEE* **1981**, *69*, 349–389. [[CrossRef](#)]
3. Kim, S.; Cho, N.I. Hierarchical prediction and context adaptive coding for lossless color image compression. *IEEE Trans. Image Process.* **2014**, *23*, 445–449. [[CrossRef](#)] [[PubMed](#)]
4. Kabir, M.A.; Khan, M.A.M.; Islam, M.T.; Hossain, M.L.; Mitul, A.F. Image compression using lifting based wavelet transform coupled with SPIHT algorithm. In Proceedings of the 2nd International Conference on Informatics, Electronics & Vision, Dhaka, Bangladesh, 17–18 May 2013.
5. Alzahir, S.; Borici, A. An innovative lossless compression method for discrete-color images. *IEEE Trans. Image Process.* **2015**, *24*, 44–56. [[CrossRef](#)] [[PubMed](#)]
6. Mondal, M.R.H.; Armstrong, J. Analysis of the effect of vignetting on MIMO optical wireless systems using spatial OFDM. *J. Lightwave Technol.* **2014**, *32*, 922–929. [[CrossRef](#)]
7. Mondal, M.R.H.; Panta, K. Performance analysis of spatial OFDM for pixelated optical wireless systems. *Trans. Emerg. Telecommun. Technol.* **2017**, *28*, e2948. [[CrossRef](#)]
8. Perli, S.D.; Ahmed, N.; Katabi, D. PixNet: Interference-free wireless links using LCD-camera pairs. In Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MOBICOM), Chicago, IL, USA, 20–24 September 2010.
9. Shantagiri, P.V.; Saravanan, K.N. Pixel size reduction loss-less image compression algorithm. *Int. J. Comput. Sci. Inf. Technol.* **2013**, *5*, 87. [[CrossRef](#)]

10. Ambadekar, S.; Gandhi, K.; Nagaria, J.; Shah, R. Advanced data compression using J-bit Algorithm. *Int. J. Sci. Res.* **2015**, *4*, 1366–1368.
11. Suarjaya, A.D. A new algorithm for data compression optimization. *Int. J. Adv. Comput. Sci. Appl.* **2012**, *3*, 14–17.
12. Al-Azawi, S.; Boussakta, S.; Yakovlev, A. Image compression algorithms using intensity based adaptive quantization coding. *Am. J. Eng. Appl. Sci.* **2011**, *4*, 504–512.
13. Mandyam, G.; Ahmed, N.; Magotra, N. Lossless Image Compression Using the Discrete Cosine Transform. *J. Vis. Commun. Image Represent.* **1997**, *8*, 21–26. [[CrossRef](#)]
14. Munteanu, A.; Cornelis, J.; Cristea, P. Wavelet-Based Lossless Compression of Coronary Angiographic Images. *IEEE Trans. Med. Imaging* **1999**, *18*, 272–281. [[CrossRef](#)] [[PubMed](#)]
15. Taujuddin, N.S.A.M.; Ibrahim, R.; Sari, S. Progressive pixel to pixel evaluation to obtain hard and smooth region for image compression. In Proceedings of the 6th International Conference on Intelligent Systems, Modeling and Simulation, Kuala Lumpur, Malaysia, 9–12 February 2015.
16. Oh, H.; Bilgin, A.; Marcellin, M.W. Visually Lossless Encoding for JPEG2000. *IEEE Trans. Image Process.* **2013**, *22*, 189–201. [[PubMed](#)]
17. Yea, S.; Pearlman, W.A. A Wavelet-Based Two-Stage Near-Lossless Coder. *IEEE Trans. Image Process.* **2006**, *15*, 3488–3500. [[CrossRef](#)] [[PubMed](#)]
18. Usevitch, B.E. A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000. *IEEE Signal Process. Mag.* **2001**, *18*, 22–35. [[CrossRef](#)]
19. Weinberger, M.J.; Seroussi, G.; Sapiro, G. The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS. *IEEE Trans. Image Process.* **2000**, *9*, 1309–1324. [[CrossRef](#)] [[PubMed](#)]
20. Santos, L.; Lopez, S.; Callico, G.M.; Lopez, J.F.; Sarmiento, R. Performance Evaluation of the H.264/AVC Video Coding Standard for Lossy Hyperspectral Image Compression. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 451–461. [[CrossRef](#)]
21. Al-Khafaji, G.; Rajab, M.A. Lossless and Lossy Polynomial Image Compression. *OSR J. Comput. Eng.* **2016**, *18*, 56–62. [[CrossRef](#)]
22. Wu, X. Lossless Compression of Continuous-Tone Images via Context Selection, Quantization, and Modeling. *IEEE Trans. Image Process.* **1997**, *6*, 656–664. [[PubMed](#)]
23. Said, A.; Pearlman, W.A. An Image Multiresolution Representation for Lossless and Lossy Compression. *IEEE Trans. Image Process.* **1996**, *5*, 1303–1310. [[CrossRef](#)] [[PubMed](#)]
24. Li, X.; Orchard, M.T. Edge-Directed Prediction for Lossless Compression of Natural Images. *IEEE Trans. Image Process.* **2001**, *10*, 813–817.
25. Abo-Zahhad, M.; Gharieb, R.R.; Ahmed, S.M.; Abd-Ellah, M.K. Huffman Image Compression Incorporating DPCM and DWT. *J. Signal Inf. Process.* **2015**, *6*, 123–135. [[CrossRef](#)]
26. Lohitha, P.; Ramashri, T. Color Image Compression Using Hierarchical Prediction of Pixels. *Int. J. Adv. Comput. Electron. Technol.* **2015**, *2*, 99–102.
27. Wu, H.; Sun, X.; Yang, J.; Zeng, W.; Wu, F. Lossless Compression of JPEG Coded Photo Collections. *IEEE Trans. Image Process.* **2016**, *25*, 2684–2696. [[CrossRef](#)] [[PubMed](#)]
28. Kaur, M.; Garg, E.U. Lossless Text Data Compression Algorithm Using Modified Huffman Algorithm. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2015**, *5*, 1273–1276.
29. Rao, D.; Kamath, G.; Arpitha, K.J. Difference based Non-linear Fractal Image Compression. *Int. J. Comput. Appl.* **2011**, *30*, 41–44.
30. Oshri, E.; Shelly, N.; Mitchell, H.B. Interpolative three-level block truncation coding algorithm. *Electron. Lett.* **1993**, *29*, 1267–1268. [[CrossRef](#)]
31. Tan, Y.H.; Yeo, C.; Li, Z. Residual DPCM for lossless coding in HEVC. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 2021–2025.
32. Sanchez, V.; Aulí-Llinàs, F.; Serra-Sagristà, J. DPCM-Based Edge Prediction for Lossless Screen Content Coding in HEVC. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2016**, *6*, 497–507. [[CrossRef](#)]
33. Lainema, J.; Bossen, F.; Han, W.J.; Min, J.; Ugur, K. Intra Coding of the HEVC Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1792–1801. [[CrossRef](#)]
34. Sanchez, V.; Aulí-Llinàs, F.; Serra-Sagristà, J. Piecewise Mapping in HEVC Lossless Intra-Prediction Coding. *IEEE Trans. Image Process.* **2016**, *25*, 4004–4017. [[CrossRef](#)] [[PubMed](#)]

35. Hernández-Cabronero, M.; Marcellin, M.W.; Blanes, I.; Serra-Sagristà, J. Lossless Compression of Color Filter Array Mosaic Images with Visualization via JPEG 2000. *IEEE Trans. Multimedia* **2018**, *20*, 257–270. [[CrossRef](#)]
36. Taubman, D.S.; Marcellin, M.W. JPEG2000: Standard for interactive imaging. *Proc. IEEE* **2002**, *90*, 1336–1357. [[CrossRef](#)]
37. Egilmez, H.E.; Said, A.; Chao, Y.H.; Ortega, A. Graph-based transforms for inter predicted video coding. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3992–3996.
38. Hranilovic, S.; Kschischang, F.R. A pixelated MIMO wireless optical communication system. *IEEE J. Sel. Top. Quantum Electron.* **2006**, *12*, 859–874. [[CrossRef](#)]
39. Kabir, M.A.; Mondal, M.R.H. Edge-based Transformation and Entropy Coding for Lossless Image Compression. In Proceedings of the International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox’s Bazar, Bangladesh, 16–18 February 2017; pp. 717–722.
40. Huffman, D. A method for the construction of minimum redundancy codes. *Proc. IRE* **1952**, *40*, 1098–1101. [[CrossRef](#)]
41. Miaou, S.-G.; Chen, S.-T.; Chao, S.-N. Wavelet-based Lossy-to-lossless Medical Image Compression using Dynamic VQ and SPIHT Coding. *Biomed. Eng. Appl. Basis Commun.* **2003**, *15*, 235–242. [[CrossRef](#)]
42. Tomar, R.R.S.; Jain, K. Lossless Image Compression Using Differential Pulse Code Modulation and Its Application. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2016**, *9*, 197–202. [[CrossRef](#)]
43. Chen, Y.-Y.; Tai, S.-C. Embedded Medical Image Compression using DCT based Subband Decomposition and Modified SPIHT Data Organization. In Proceedings of the IEEE Symposium on Bioinformatics and Bioengineering, Taichung, Taiwan, 21–21 May 2004; pp. 167–175.
44. Sharma, M. Compression using Huffman Coding. *Int. J. Comput. Sci. Netw. Secur.* **2010**, *10*, 133–141.
45. Salomon, D. *A Concise Introduction to Data Compression*; Springer: London, UK, 2008.
46. Wallpaperswide. Available online: <http://wallpaperswide.com/pixelate-wallpapers.html> (accessed on 23 April 2018).
47. Freepik. Available online: https://www.freepik.com/free-photo/pixelated-image_946034.htm (accessed on 23 April 2018).
48. Famed Pixelated Paintings. Available online: <https://www.trendhunter.com/trends/digitized-classic-paintings> (accessed on 23 April 2018).
49. Pixabay. Available online: <https://pixabay.com/en/pattern-super-mario-pixel-art-block-1929506/> (accessed on 23 April 2018).
50. Image Processing Place. Available online: http://www.imageprocessingplace.com/root_files_V3/image_databases.htm (accessed on 23 April 2018).
51. Computational Imaging and Visual Image Processing. Available online: <https://www.io.csic.es/PagsPers/JPortilla/image-processing/bls-gsm/63-test-images> (accessed on 23 April 2018).
52. Wikimedia Commons: Sprgelenkli. Available online: <https://commons.wikimedia.org/wiki/File:Sprgelenkli131107.jpg#filelinks> (accessed on 23 April 2018).
53. Wikimedia Commons: Putamen. Available online: <https://commons.wikimedia.org/wiki/File:Putamen.jpg> (accessed on 23 April 2018).
54. Wikimedia Commons: MRI Glioma 28 Yr Old Male. Available online: https://commons.wikimedia.org/wiki/File:MRI_glioma_28_yr_old_male.JPG (accessed on 23 April 2018).

