IMPROVED MEDIAN EDGE DETECTION (IMED) FOR LOSSLESS IMAGE COMPRESSION

MUHAMMAD SHOIB AMIN✉,1, SUMMAIRA JABEEN[2], CHANGBO WANG[3], HASSAN ALI KHAN[1] AND ABDUL JABBAR[2]

[1]School of Software Engineering, East China Normal University, Shanghai, China, [2]Department of Computer Science, Zhejiang University, Hangzhou China, [3]School of Computer Science and Technology, East China Normal University, Shanghai, China
e-mail: 52184501030@stu.ecnu.edu.cn, 11821129@mail.zju.edu.cn, cbwang@cs.ecnu.edu.cn, hassankhanzae@gmail.com, jabbar@zju.edu.cn

(Received August 15, 2022; revised February 18, 2023; accepted March 8, 2023)

ABSTRACT

A wide variety of applications are used in lossless image compression models, especially in medical, space, and aerial imaging domains. Predictive coding improves the performance of lossless image compression, which highly relies on entropy error. Lower entropy error results in better image compression. The main focus of this research is to improve the prediction process by minimizing the entropy error. This paper proposes a novel idea for improved Median Edge Detection (iMED) predictor for lossless image compression. MED predictor is improved using k-means clustering and finding the local context of pixels using 20-Dimensional Difference (DDx20) for input images and updates the cluster weights using learning rates (μ_i) to minimize the prediction errors of pixels. The performance of the proposed predictor is evaluated on the standard grey-scale test images dataset and KODAK image dataset. Results are obtained and compared based on entropy error, bits per pixel (bpp), and computational running time in seconds(s) with the MED, GAP, FLIF, and LBP predictors. The performance of the proposed iMED predictor improves significantly in terms of the entropy error, bpp, and computational running time in seconds(s) after comparison with different state-of-the-art predictors.

Keywords: Difference Vector, K-means clustering, Learning rate, Lossless image compression, Median Edge Detector (MED), Predictive Coding.

INTRODUCTION

In many data storage and transmission applications, the importance of image compression has been increasing continuously (Li *et al.*, 2019; Marlapalli *et al.*, 2021). For example, in storage disks, many images need a large amount of storage space and more time for wireless transmission. Therefore, an efficient compression algorithm plays a vital role in saving storage space and transmission time (Siddeq and Rodrigues, 2017; Hameed *et al.*, 2020; Vura *et al.*, 2021). Image compression is a process in which compression is done by encoding image data with fewer bits, and redundant data is removed. Image compression is a cost-effective tool that preserves expensive resources, i.e., transmission bandwidth and data storage space. The basic categories of Image compression are lossy or lossless (LS) compression (Prasanna *et al.*, 2021; Qasim *et al.*, 2020). Various methods are used for encoding and decoding images for the digital image compression process. Lossless image compression technique is chosen where the nature of data is sensitive, like military or medical images. Lossless can achieve two to three times of total compression without losing image information (Wu *et al.*, 2012; Zhou and Kwan, 2018; Kwan and Luk, 2018; Kwan and Larkin, 2018; Rahman *et al.*, 2021). There are three basic types of lossless image compression methods, i.e., entropy coding, dictionary coding, and predictive coding. In entropy coding, the representation of symbols is done by taking the average number of bits. In dictionary coding, during the transmission of an image, the actual pixel values are represented by values in the database (dictionary). In predictive coding, the difference between the actual values and the predicted values of an image is transmitted (Shanmathi and Maniyath, 2017). A general view of the predictive lossless image compression scheme is presented in Fig. 1. Prediction of the current pixel value is calculated by using the predictor, and then its output is coded using an entropy coder, which removes the spatial redundancy. Information of an image is used by the context modeling block to improves the predictor's output.

In Lossless Predictive coding, the method of predictive coding depends upon context modeling, entropy coding, and a predictor (Tiwari and Kumar, 2005; Kamisli, 2016; Venugopal *et al.*, 2016). A large quantity of spatial redundancy is eliminated at the most important phase, i.e., predictor. Recently predictive coding has been used by many researchers in image compression due to the efficiency, symmetry, and simplicity of this coding scheme (Al-Mahmood and Al-Rubaye, 2014). Gradient Adjusted Predictor (GAP) (Tiwari and Kumar, 2008) is used in CALIC and JPEG_LS (Weinberger *et al.*, 2000), Median Edge Detector (MED) (Fouad, 2015), and Gradient Edge Detection (GED) (Shanmathi and Maniyath, 2017) are basic and well-known predictors used for lossless predictive coding. Gradient adjusted predictor (GAP) is built on the least mean square adaptation of linear predictive coefficients (Li and Orchard, 2001). The adaptation is based on linking neighbor pixels, and an optimal predictor will predict the intensity value of each pixel. MED consists of a static predictor, which is a switching predictor and is proficient at acclimatizing context of different types like smooth area, vertical edge, or horizontal edge (Weinberger *et al.*, 2000).

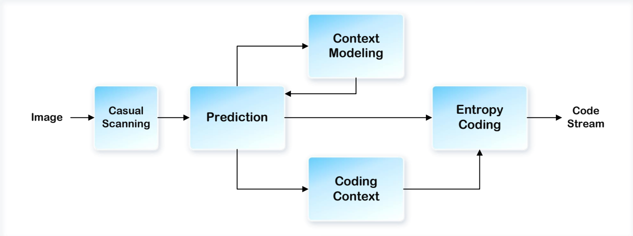


Fig. 1. *General predictive lossless image compression scheme.*

This paper proposes a new improved Median Edge Detection (iMED) predictor for lossless image compression. In this method, the MED predictor is improved by making clusters of an input image using k-means clustering and then finding the local context of pixels using DDx20. In lossless predictive coding, the predictor's performance highly relies on the difference between the actual value and entropy error of the pixels in an image. The performance of the predictor is increased with the reduction of the entropy error value. The value of entropy, bpp, and computational running time in second(s) of the prediction process is calculated to evaluate the iMED predictor performance. The proposed method results demonstrate significant improvement in entropy value, bpp, and computational running time(s) compared to state-of-the-art MED (Fouad, 2015), GAP (Tiwari and Kumar, 2008), FLIF (Sneyers and Wuille, 2016), and LBP (Novikov *et al.*, 2016) predictors.

In this paper, section 2 presents some related work about the predictive lossless compression techniques. And proposed iMED method is discussed in section 3. Experimental results and detailed discussion on the proposed predictor are discussed in section 4. Finally, section 5 concludes the paper.

RELATED WORK:

In lossless image compression, predictive coding is an important and well-known compression technique (Hussain *et al.*, 2018). This technique utilizes the spatial domain efficiently by involving two main steps, i.e., differentiation and prediction. In differentiation, the residual difference between the original and predicted image is calculated. In prediction, an approximation of the image to the original one is created (Al-Khafaji, 2012). The predicted image relies on the least difference values, either from the vertical or horizontal value or the left bottom value of the current pixel (Al-Khafaji and Al-Mahmood, 2016). In this section, some commonly used predictive techniques for lossless image compression and MED predictor are discussed.

COMMON PREDICTION MODELS:

The proposed algorithm for lossless image compression (Azman *et al.*, 2019) can achieve the desired results by combining Integer Wavelet Transform (IWT) and Differential Pulse Code Modulation (DPCM). To analyze the performance of this hybrid algorithm, two parameters, i.e., compression ratio and entropy, are used. The experimental results show that the DPCM-IWT-Huffman sequence performs better than the IWT-DPCM-Huffman sequence in terms of low entropy and better compression ratio. Bits size is reduced by 48%, 36%, 34%, and 13% for the Cameraman, Lena, Pepper, and Baboon, respectively.

Among different lossless image compression techniques for the predictive coding approach, the medical images have low complexity and high coding efficiency. In the predictive coding approach, GED is based on prediction value. The author proposed an efficient prediction approach named Resolution Independent Gradient Edge Predictor (RIGED) (Sharma *et al.*, 2021). This algorithm supports 8-bit and 16-bit medical images data. Experimental results are compared with MED and GAP's well-known techniques of predictive coding. The analysis of experimental results shows that the RIGED method improvement percentage over MED and GAP methods are 30.39% and 0.92% respectively in terms of entropy for the medical images dataset. Therefore, the proposed method achieves better performance in terms of low entropy than MED and GAP, which is simple for implementation.

Prediction-based Transformation and Entropy Coding (PTEC) is another technique that is used for pixelated type images (Kabir and Mondal, 2018). The input image is divided hierarchically in the first stage of the PTEC method to predict the current pixel value using values of neighboring pixels. After that, in the latter phase, two different matrices are formed, where the first matrix contains the value of absolute error and the second matrix contains the prediction error's polarity. After these two stages, entropy coding is applied to these matrices. The results of PTEC are compared with the existing lossless state-of-the-art techniques, i.e., Edge-based Transformation and Entropy Coding (ETEC), Differential Pulse Code Modulation (DPCM), Joint Photographic Experts Group Lossless (JPEG-LS), and Set Partitioning in Hierarchical Trees (SPIHT). Experimental result shows that PTEC and ETEC provide much better compression as compared to other techniques. The computation time of PTEC is better than ETEC. PTEC performs better than ETEC if we consider both compression and computation time for pixelated and non-pixelated images.

LOCO-I (LOw COmplexity LOssless Compression for Images) is another algorithm used for near-lossless and lossless compression (Weinberger *et al.*, 2000). LOCO-I is based on a fixed context model in a simple form, in which high-order dependencies are also captured. The main strength is that this method can achieve similar or better compression ratios compared to other relevant techniques. For natural images, Li and Orchard (Li and Orchard, 2001) expose the superiority of lossless-based adaptation schemes and reduce the computational complexity. The main process of this approach involves, if the prediction error's magnitude is beyond the preselected threshold value, then prediction coefficients are updated instead of performing LS-based optimization on pixels.

In another image compression technique, Tiwari and Kumar (Tiwari and Kumar, 2008) tries to improve the capability in terms of accuracy with a slight increase in computational complexity of GAP. In this method, to find an optimal prediction for different slope bins Least Square (LS) technique is used. After the quantization process at the encoder and decoder end, various switched predictors are used. The strength of this approach is an encoder, in which the encoder is computationally much simpler. It can be used to compress Medical images.

(Avramović and Reljin, 2010) proposed Gradient Edge Detection (GED) predictor tends to use advantages of described MED (Haijiang *et al.*, 2005) and GAP predictors. GED is designed by combining these two techniques. In GED, values of five neighboring pixels are used to estimate the local gradient of pixels. It is a compromise between MED and GAP predictors. GED chooses among vertical edge, horizontal edge, or smooth area like MED predictor, but the mechanism for prediction is based on GAP predictor. The strength of GED comparing with GAP is that this gives about 1% higher bit rates but is considerably simpler.

MEDIAN EDGE PREDICTOR (MED):

The Median edge detector (MED) prediction method is used to generate the predicted values of the original pixels and calculate the prediction errors. MED is one of the most successful prediction schemes and serves as the core part of the LOCO-I algorithm (Weinberger *et al.*, 2000). It combined good compression efficiency with very low computational complexity. MED detects horizontal or vertical edge orientations based on the template consists of the $n, w, and\, nw$ neighbors of current pixel x, as shown in Fig. 2.

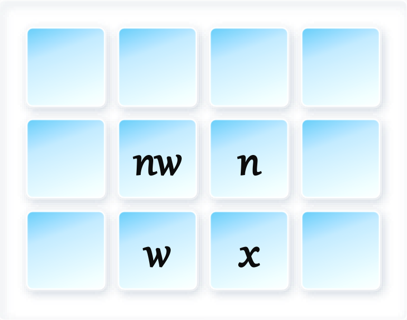


Fig. 2. *Neighboring pixels template of MED predictor.*

$$\begin{array}{lr} x' = min(n,w) & if nw \geq max(n,w) \\ x' = max(n,w) & if nw \leq max(n,w) \\ x' = n+w-nw & Otherwise \end{array} \quad (1)$$

MED predictor chooses a median value between n, w, and $n+w-nw$ as shown in Equation 1. The first two rows indicate an edge is assumed when the nw value is minimum or maximum compared to other neighboring pixels of x. Pixel n value is selected as a prediction in the case of vertical edge and w in case of horizontal edge. MED predictor predicts the current pixel x value according to the context of three neighboring pixels.

PROPOSED METHOD (IMED):

Methods for lossless predictive coding mainly depend on context model, entropy coding, and a predictor. Predictor is the most important part of the proposed method because it removes redundancies

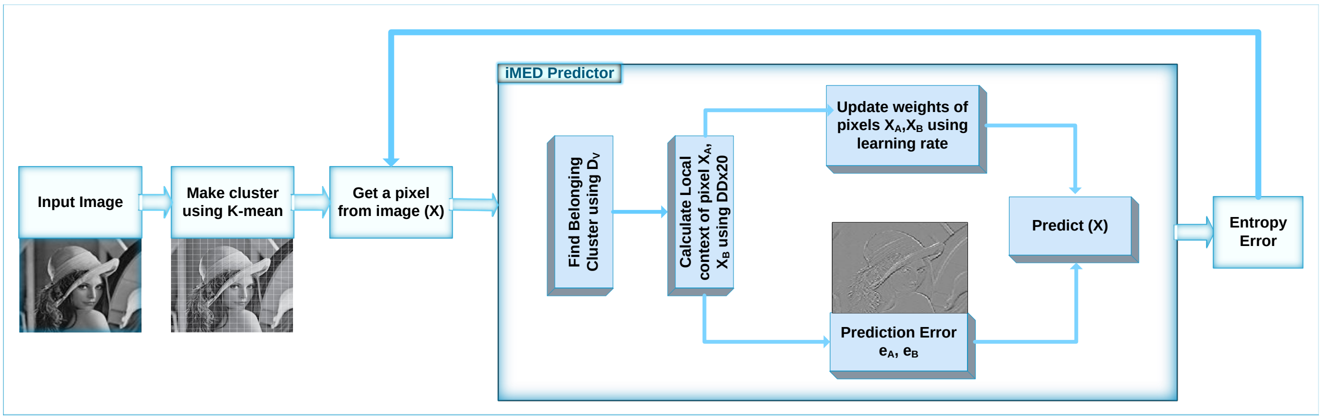

Input Image
Make cluster using K-mean
Get a pixel from image (X)
iMED Predictor
Find Belonging Cluster using D_V
Calculate Local context of pixel X_A, X_B using DDx20
Update weights of pixels X_A,X_B using learning rate
Predict (X)
Prediction Error e_A, e_B
Entropy Error

Fig. 3. *Flow Diagram of iMED. Prediction is made on each pixel after applying k-means clustering and using iMED Predictor.*

in the image by reducing the entropy error. The prediction mechanism of MED is improved by updating the original values of neighboring pixels X_A *and* X_B. Algorithm and detailed description of the proposed method is presented in section 3.1 and 3.2, respectively. The flow of the proposed predictor iMED is presented in Fig. 3. The core part of this method is the iMED predictor, which focuses on minimizing the entropy error using the local context of a pixel, as shown in Fig. 4. The local context of the current pixel (X) is calculated using a 20-Dimensional Difference (DDx20), as presented in Table 1.

ALGORITHM:

Step 1- Read an input image.

Step 2- Make clusters of the input image using the k-means algorithm.

Step 3- Chose a pixel (X) from the image and forward it to the iMED predictor to calculate entropy error.

3.1 Find the belonging cluster of a pixel using Difference Vector (D_v) as shown in Equation 4.

3.2 Find local context of a pixel X_A, X_B using 20-Dimensional Difference (DDx20) as shown in Equation 5 and 6, respectively.

3.3 Calculate prediction error e_A and e_B.

3.4 Update weights of pixels X_A *and* X_B using learning rate μ_i.

3.5 Predict value of current pixel (X)

Step 4- Calculate the entropy error.

Step 5- Take another pixel from the input image and repeat steps 3 and 4 for all pixels.

IMED PREDICTOR:

In this proposed iMED predictor, a novel context-sensitive prediction scheme is used, detailed description is explained as follows:

K-means clustering:

Read an input image, and calculate the total number of rows (height) and columns (width) of the image. Then total numbers of clusters for an input image is made using Equation 2.

$$N_C = \frac{\sqrt{width * height}}{10} \tag{2}$$

Where N_C is the number of clusters, N_C is formed optimally according to the height and width ratio of the input image, with ten as a denominator. This is an optimal value found during experiments. If this value decreases, then the predictor's performance also decreases, and if it increases, then the computational cost of the predictor increases. After that, we randomly select the number of training samples from an image using Equation 3. These random training samples calculate the distance between pixels and compute the cluster's centroid.

$$S_N = 500 * N_C \tag{3}$$

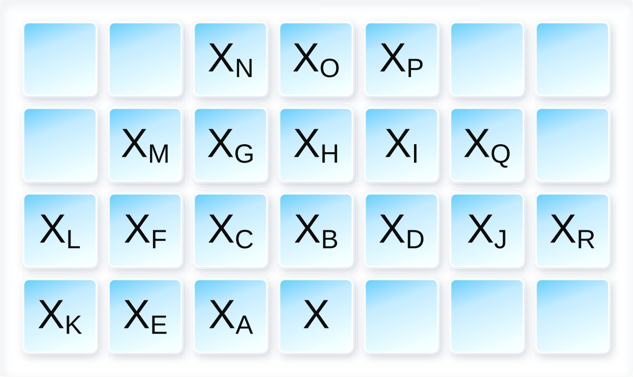
Where S_N is the number of training samples formed according to the total number of clusters N_C formed using Equation 2.

Difference Vector:

Difference Vector (D_V) is used to find the belonging cluster of a pixel (X). To randomly select the number of training samples S_N from cropped image, index numbers are randomly selected. Then this index number is used to get its D_V from the original image. D_V of each randomly selected pixels are obtained using Equation 4. Minimum distance from the centroid of each cluster and its index number against each pixel is obtained after calculating D_V.

$$D_v = [X_A - X_C, X_C - X_B, X_E - X_D, \\ X_A - X_E, X_B - X_H, S_F * X_A] \tag{4}$$

where $X_A, X_B, X_C, X_D, X_E, X_H$, are position of neighboring pixels and S_F is the Scaling Factor. We used six different combinations of neighboring pixels of X to calculate six D_V values. The distance of a pixel (X) is calculated from all N_C to find the belonging clusters with less computational cost.



		X_N	X_O	X_P		
	X_M	X_G	X_H	X_I	X_Q	
X_L	X_F	X_C	X_B	X_D	X_J	X_R
X_K	X_E	X_A	X			

Fig. 4. *Local Context of a pixel (X) using its neighboring pixel values.*

Calculate value of pixels X_A, X_B using local context of a pixel:

Image is ready for prediction after finding the belonging cluster of a pixel. The local context of the current pixel (X) is calculated using a 20-Dimensional Difference (DDx20), as presented in Table 1. DDX20 is used to update the neighboring pixels X_A *and* X_B values according to Equation 5 and 6, respectively. DDx20 is calculated for a pixel from its neighboring pixel values, as shown in Table 1 from $DDx20_1, DDx20_2, DDx20_3, \ldots, DDx20_{20}$. We take values from the top, right, and left neighboring pixels to make the prediction better.

Table 1. *20-Dimensional Difference (DDx20).*

$DDx20_1 = X_A - X_C$	$DDx20_{11} = X_H - X_O$
$DDx20_2 = X_C - X_B$	$DDx20_{12} = X_F - X_M$
$DDx20_3 = X_B - X_D$	$DDx20_{13} = X_K - X_L$
$DDx20_4 = X_A - X_E$	$DDx20_{14} = X_O - X_N$
$DDx20_5 = X_B - X_H$	$DDx20_{15} = X_J - X_Q$
$DDx20_6 = X_C - X_G$	$DDx20_{16} = X_J - X_R$
$DDx20_7 = X_E - X_F$	$DDx20_{17} = X_C - X_F$
$DDx20_8 = X_D - X_I$	$DDx20_{18} = X_I - X_P$
$DDx20_9 = X_D - X_J$	$DDx20_{19} = X_H - X_I$
$DDx20_{10} = X_E - X_K$	$DDx20_{20} = X_H - X_G$

$$X_A^{'} = X_A \sum_{i=1}^{20} w_i[X].DDx20_i \tag{5}$$

$$X_B^{'} = X_B \sum_{i=1}^{20} w_i[X].DDx20_i \tag{6}$$

We initialize the cluster weights w_i to zero and calculate the value of neighboring pixel $X_A^{'}$ according to Equation 5 using the local context of pixels. And calculate the value of neighboring pixel $X_B^{'}$ according to Equation 6. Where X_A *and* X_B are original values of neighboring pixels, these values are multiplied with weights of corresponding pixel and $DDx20_i$ value to get the new predicted value of $X_A^{'} and X_B^{'}$.

Calculate prediction error e_A and e_B:

Prediction error e_A for current pixel (X) with respect to the value of neighboring pixel A is calculated according to Equation 7.

$$e_A = X - X_A^{'} \tag{7}$$

Prediction error e_B for current pixel (X) with respect to the value of neighboring pixel B is calculated according to Equation 8.

$$e_B = X - X_B^{'} \tag{8}$$

Threshold values $(e_A^{'}, e_B^{'})$ are used to update the weight of pixels using the predicted value of neighboring pixels A and B. Threshold for errors are shown in Equations 9 and 10 according to neighboring pixels A and B, respectively.

$$e_A^{'} = sign(e_A * min(|e_A|, th_A)) \tag{9}$$

$$e_B^{'} = sign(e_B * min(|e_B|, th_B)) \tag{10}$$

where e_A and e_B are prediction errors with respect to neighboring pixels X_A *and* X_B of a corresponding cluster. th_A and th_B are threshold values for prediction error e_A and e_B.

Update weights of pixels X_A *and* X_B using learning rate:

Initialize the learning rate (μ_i) of the prediction process for DDx20 according to Equation 11. This learning rate is used to update the weights of pixels X_A *and* X_B.

$$\mu_i = \begin{cases} \frac{4}{10000} & fori \leftarrow 1,2,3 \\ \frac{2}{10000} & fori \leftarrow 4,5 \\ \frac{1}{10000} & fori \leftarrow 6,7,8,...,20 \end{cases} \tag{11}$$

$$w_{iA} = w_i + \mu_i * e_A^{'}(\frac{DDx20_i}{|DDx20_i| + 1}) \tag{12}$$

$$w_{iB} = w_i + \mu_i * e_B^{'}(\frac{DDx20_i}{|DDx20_i|+1}) \qquad (13)$$

Weights of neighboring pixels X_A *and* X_B are updated using Equations 12 and 13, respectively. We first initialize weight (w_i) to zero, then update the weight of corresponding pixels X_A *and* X_B according to the learning rate (μ_i), predicted threshold value of pixel A and B, and 20-Dimensional Difference (DDx20). This learning rate (μ_i) for DDX20 plays a vital role in improving the iMED predictor.

Predict the value of current pixel (X):

After getting the updated value of neighboring pixels X_A *and* X_B using DDx20, the current pixel (X) value is calculated according to Equation 14. And we choose the value of X_C without finding the local context of a pixel. Finally, the entropy value is calculated using these predicted values.

$$\begin{array}{rr} X' = min(X_A, X_B) & if X_C \geq max(X_A, X_B) \\ X' = max(X_A, X_B) & if X_C \leq max(X_A, X_B) \\ X' = X_A + X_B - X_C & Otherwise \end{array} \qquad (14)$$

EXPERIMENTS AND RESULTS:

In this section, detail about the experimental setup, dataset, and results are discussed. The performance of the proposed method is also compared with state-of-the-art predictors, i.e., MED (Fouad, 2015), GAP (Tiwari and Kumar, 2008), FLIF (Sneyers and Wuille, 2016), and LBP (Novikov *et al.*, 2016).

DATASET:

For comparisons, we performed experiments on a standard grey-scale images test dataset (Dataset 1, 2022) and True color KODAK images dataset (Dataset 2, 2022).

Dataset-1: Eight standard grey-scale test images (Lena, Cameraman, Livingroom, Mandrill, Peppers, Pirate, Woman_blonde, and Woman_darkhair) of size 256x256 are selected for experiments from standard grey-scale images test data to verify the performance of the proposed iMED predictor. These test data images are used in tiff (Tagged Image File) format with 8-bit depth, which is found frequently in the literature of image compression is shown in Fig. 5.

Dataset-2: Additionally, we also evaluate the performance of our proposed predictor by performing experiments on the True color KODAK images dataset. KODAK is a widely used dataset for testing the performance of compression models. This dataset contains 24 color images of type PNG as shown in Fig. 6, out of which eighteen images have 768x512 pixels (landscape), and six images have 512x768 pixels (portrait). The photographic quality of the KODAK dataset involves a variety of subjects in many locations under different lighting conditions. For experiments, the KODAK dataset is converted to grey-scale and resized to suitable resolution for prediction. All the landscape images are resized to 256x170 pixels, and all the portrait images are resized to 170x256 pixels.

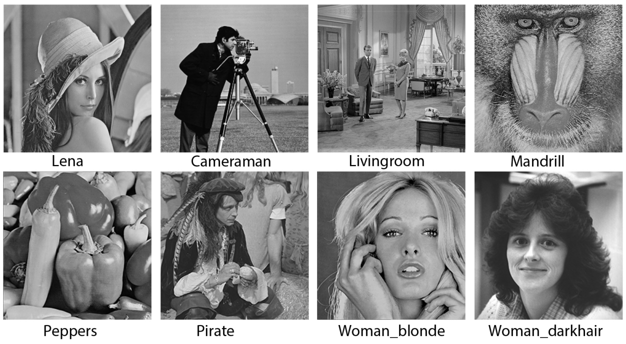


Fig. 5. *Standard grey-scale test images dataset.*

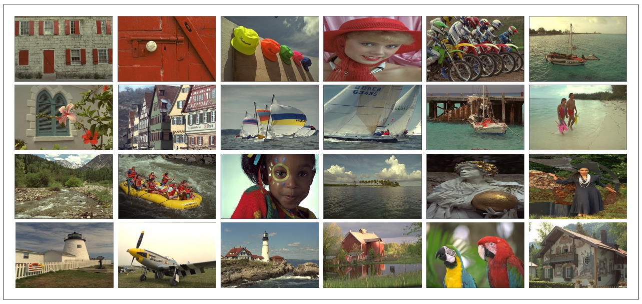


Fig. 6. *True color Kodak images dataset.*

EXPERIMENTAL DESIGN:

Experiments are carried out in the MATLAB R2021a environment. All the experiments are conducted and evaluated in terms of entropy and running time in seconds(s) on a machine with Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz laptop along with 32GB of RAM. System Type is 64-bit operating system Windows 11 Home edition, x64-based processor.

Evaluation on the benchmarks.

The main parameter to evaluate the performance or efficiency of a predictor is entropy. Entropy is the minimum number of bits required to represent complete information of an image. The efficiency of the predictor is considered high with a lower entropy value (Sharma *et al.*, 2021). Entropy H(X) of an image is calculated as shown in Equation 15,

$$H(X) = -\sum_{x\epsilon Y} p(X) logp(X) \qquad (15)$$

Where p(X) is the probability of symbol (X). Bits per pixels (bpp) is another evaluation parameter used to check the performance of the proposed predictor. Bpp is an exact metric that reflects the average number of bits required to encode the information of each pixel in an image. Additionally, the proposed predictor's performance is also evaluated on computational running time in terms of seconds(s). The computational complexity of a predictor is calculated based on two parameters, i.e., running time for algorithm execution and the number of operations required for its implementation. Running time for execution of a predictor increases with the complexity of the method. The efficiency of a predictor increases if its complexity and running time decreases. In these experiments, entropy, bpp, and computation time(s) are used to evaluate the proposed predictor's effectiveness compared to baseline predictors.

RESULTS AND DISCUSSION:

In experiments, dataset-1 and dataset-2, publicly available without any conflicts of interest, are used to calculate the value of entropy, bpp, and the computational running time in second(s). Three different experiments are performed on dataset-1 and dataset-2. In the first experiment, we calculate the entropy value of the proposed predictor (iMED) and other baseline predictors on both datasets. We calculate the bpp of the proposed predictor and baseline predictors in the second experiment. And similarly, in our third experiment, we calculate the computational running time in the second(s) of all the predictors for comparison.

Table 2. *Entropy for MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-1. Where Av. Ent = Average Entropy*

Image Name	MED	GAP	FLIF	LBP	iMED
Lena	0. 9954	0. 9131	0.8216	0.7561	0. 5203
Cameraman	0. 9972	0. 8807	0.7546	0.6054	0. 6197
Livingroom	0. 9982	0. 9647	0.7981	0.6549	0. 4195
Mandrill	0. 9996	0. 9941	0.8316	0.7615	0. 1922
Peppers	0. 9918	0. 9261	0.7615	0.5987	0. 4881
Pirate	0. 9984	0. 9636	0.8345	0.6885	0. 4072
Woman_blonde	0. 9976	0. 9509	0.8012	0.7916	0. 4222
Woman_darkhair	1.0000	0. 8534	0.6981	0.7659	0. 6698
Av. Ent	**0.9972**	**0.9308**	**0.7876**	**0.7028**	**0.4673**

Experiment 1:

Experiment 1 is performed on dataset-1 (Lena, Cameraman, Livingroom, Mandrill, Peppers, Pirate, Woman_blonde, and Woman_darkhair) and images of dataset-2 to calculate the entropy value. The entropy of the proposed predictor is compared with MED, GAP, FLIF, and LBP predictors. Detailed results from the perspective of entropy on dataset-1 are shown in Table 2. Average entropy summarizes results based on entropy for all the predictors, i.e., MED, GAP, FLIF, LBP, and iMED (ours). Their values are obtained as 0.9972, 0.9308, 0.7876, 0.7028, and 0.4673, respectively. It clearly shows that the proposed iMED predictor entropy significantly improves compared to MED, GAP, FLIF, and LBP predictors. Entropy comparison results on dataset-1 are shown in Fig. 7. The iMED predictor entropy value is low on all images of dataset-1 as compared to baseline predictors.

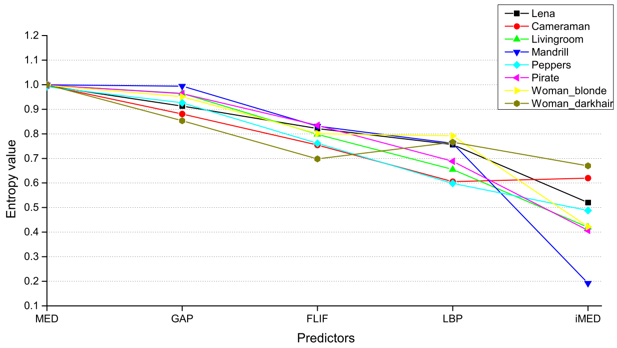


Fig. 7. *Entropy comparison of MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-1 (lower entropy is better)*

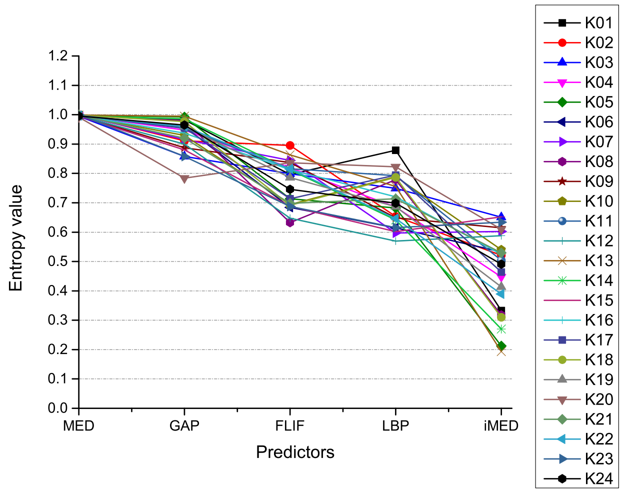


Fig. 8. *Entropy comparison of MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-2 (lower entropy is better)*

Detailed results from the perspective of entropy on dataset-2 are shown in Table 3. Average entropy values of the MED, GAP, FLIF, LBP, and iMED predictors are 0.9935, 0.9328, 0.7637, 0.7051, and 0.4727, respectively. It clearly shows that the proposed iMED predictor entropy significantly improves compared to MED, GAP, FLIF, and LBP predictors on dataset-2

Table 3. *Entropy for MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-2. Where Av. Ent = Average Entropy*

Model	K01	K02	K03	K04	K05	K06	K07	K08	K09	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23	K24	**Av. Ent**
MED	0.9995	0.9998	0.9950	0.9979	0.9986	0.9967	0.9983	0.9982	0.9975	0.9976	0.9971	0.9965	0.9988	0.9963	0.9954	0.9972	0.9984	0.9980	0.9983	0.9917	0.9982	0.9970	0.9999	0.9954	**0.9935**
GAP	0.9821	0.9107	0.8576	0.9480	0.9921	0.9590	0.9156	0.9795	0.8871	0.9294	0.9534	0.8999	0.9948	0.9859	0.8805	0.9371	0.9548	0.9764	0.9595	0.7838	0.9202	0.9598	0.8578	0.9646	**0.9328**
FLIF	0.7964	0.8952	0.7988	0.8193	0.7135	0.6842	0.8442	0.6326	0.8369	0.6942	0.8158	0.6473	0.8634	0.8123	0.6881	0.8125	0.7135	0.6914	0.7861	0.8361	0.6981	0.8156	0.6875	0.7459	**0.7637**
LBP	0.8786	0.6536	0.7488	0.6852	0.6824	0.6145	0.5971	0.7789	0.6485	0.7892	0.7925	0.5698	0.7557	0.6415	0.6018	0.7202	0.7946	0.7865	0.6871	0.8226	0.7136	0.6459	0.6152	0.6987	**0.7051**
iMED	0.3327	0.5209	0.6510	0.4478	0.2121	0.5326	0.6025	0.3180	0.6150	0.5402	0.5038	0.5879	0.1936	0.2698	0.6523	0.5240	0.4642	0.3103	0.4138	0.6093	0.5297	0.3890	0.6338	0.4908	**0.4727**

as well. Entropy comparison results on dataset-2 are shown in Fig. 8. The iMED predictor entropy value is low on all images of dataset-2 as compared to baseline predictors except K07, K12, K15, and K23, where the LBP entropy value is better than iMED.

Table 4. *Bpp for MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-1. Where Av. bpp = Average bpp*

Image name	**MED**	**GAP**	**FLIF**	**LBP**	**iMED**
Lena	5.5836	4.5997	5.3546	4.8954	4.0420
Cameraman	5.9305	4.3754	4.6255	4.6492	4.2378
Livingroom	5.6641	5.1831	4.9846	5.9863	4.4391
Mandrill	7.4788	6.8632	6.8715	4.9664	5.7646
Peppers	5.4063	4.5247	5.9592	4.1368	3.9929
Pirate	5.9298	5.3411	4.9584	5.6871	4.7565
Woman_blonde	5.6226	4.9374	5.1284	4.1545	4.3296
Woman_darkhair	5. 9442	6.8341	5.6828	4.9875	3.3588
Av. bpp	**5.9450**	**5.3323**	**5.4456**	**4.9321**	**4.3651**

Experiment 2:

Experiment 2 is performed to calculate the bpp value on dataset-1 (Lena, Cameraman, Livingroom, Mandrill, Peppers, Pirate, Woman_blonde, and Woman_darkhair) and images of dataset-2. First, we calculate the bpp value of MED, GAP, FLIF, LBP, and iMED predictors on dataset-1, which detailed results are shown in Table 4. Average bpp value summarizes results based on bpp for all the predictors, i.e., MED, GAP, FLIF, LBP, and iMED (ours). The bpp values for MED, GAP, FLIF, LBP, and iMED are 5.9450, 5.3323, 5.4456, 4.9321, and 4.3651, respectively. It clearly shows that the proposed iMED predictor takes less number of bits per pixel to store the final image compared to MED, GAP, FLIF, and LBP predictors. Bpp comparison results on dataset-1 are summarized in Fig. 9. The iMED predictor bpp value is low on all images of dataset-1 as compared to baseline predictors except on Mandrill and Woman_blonde image, where the LBP predictor bpp value is better than iMED. Detailed results from the perspective of bpp on dataset-2 are shown in Table 5. Average bpp values of the MED, GAP, FLIF, LBP, and iMED predictors are 6.1664, 5.1590, 5.2094, 4.0681, and 1.5535, respectively. It clearly shows that the proposed iMED predictor takes less number of bits per pixel to store the final image compared to MED, GAP, FLIF, and LBP predictors. Bpp comparison results on dataset-2 are summarized in Fig. 10. The iMED predictor bpp value is low on all images of dataset-2 as compared to baseline predictors.

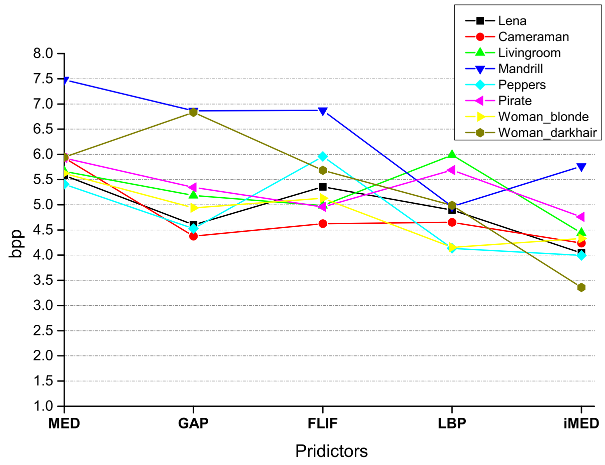


Fig. 9. *Bpp comparison of MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-1 (lower bpp is better)*

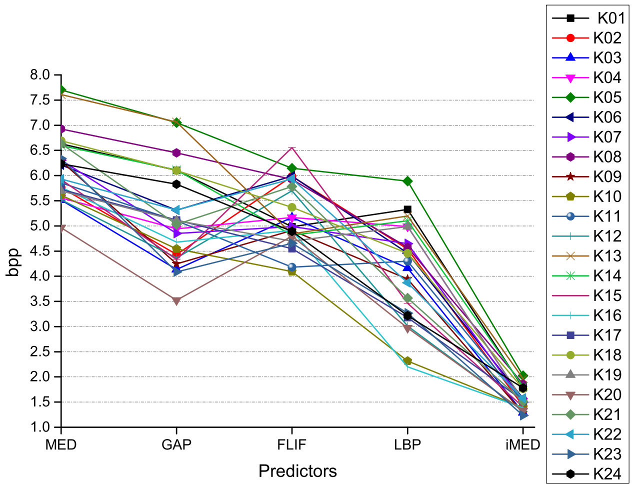


Fig. 10. *Bpp comparison of MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-2 (lower bpp is better)*

Table 5. *Bpp for MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-2. Where Av. bpp = Average bpp*

Model	K01	K02	K03	K04	K05	K06	K07	K08	K09	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23	K24	**A.bpp**
MED	6.6320	5.7621	5.5169	5.5555	7.7013	6.2108	6.3072	6.9248	6.2860	5.6116	5.8561	5.5186	7.6090	6.5972	5.9197	5.7686	5.7250	6.6921	5.6956	4.9631	6.6357	5.9320	6.3408	6.2338	**6.1664**
GAP	6.1031	4.4335	4.1344	4.9397	7.0509	5.3118	4.8517	6.4518	4.2441	4.5439	5.1020	4.3629	7.0618	6.1114	4.3318	4.6781	5.1164	6.1017	5.1051	3.5233	5.0290	5.3175	4.0836	5.8276	**5.1590**
FLIF	4.9823	5.9845	5.1684	5.1654	6.1459	5.9895	4.9875	5.9232	4.9125	4.0916	4.1791	5.6982	4.8413	4.8165	6.5565	4.9127	4.5468	5.3685	4.6975	4.7954	5.7826	5.9356	4.6546	4.8911	**5.2094**
LBP	5.3258	4.5385	4.1615	4.9877	5.8912	4.4613	4.6489	4.5732	3.9445	2.3164	4.2997	3.0124	5.1975	5.1026	3.4635	2.1986	3.1788	4.4568	4.9873	2.9721	3.5648	3.8713	3.2654	3.2164	**4.0681**
iMED	1.7793	1.4195	1.2900	1.5235	2.0248	1.5730	1.4985	1.8610	1.2806	1.3915	1.5012	1.3173	1.9665	1.8058	1.4052	1.3852	1.5325	1.7656	1.5366	1.3254	1.5138	1.5751	1.2314	1.7809	**1.5535**

Table 6. *Computational running time in second(s) for MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-2. Where Av. CT = Average Computational Time*

Model	K01	K02	K03	K04	K05	K06	K07	K08	K09	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23	K24	**Av. CT**
MED	23.90	14.92	15.43	13.22	12.30	28.67	15.94	12.41	15.17	15.12	14.46	15.35	14.34	14.21	18.23	16.96	15.69	22.35	15.38	40.34	17.46	15.31	17.80	15.93	**17.53**
GAP	20.79	14.05	15.41	13.13	12.41	23.14	14.33	12.51	14.69	14.07	14.37	15.12	12.72	12.79	15.30	14.47	15.16	12.90	13.72	21.65	15.07	13.37	14.77	13.07	**14.96**
FLIF	17.36	12.85	13.25	12.54	11.35	19.24	11.89	9.34	13.78	12.43	11.49	13.25	10.54	11.30	13.63	11.31	13.32	10.76	11.22	16.42	14.31	10.75	12.68	9.34	**12.68**
LBP	16.71	9.31	12.38	10.65	10.87	14.65	10.51	10.39	12.82	9.24	11.01	12.73	9.87	10.57	10.78	11.98	10.94	9.81	10.56	11.83	11.72	10.96	11.01	10.15	**11.31**
iMED	7.48	7.21	10.14	11.00	11.55	9.35	8.39	8.26	9.78	7.86	8.03	8.99	8.19	8.26	7.96	8.61	9.93	8.45	8.91	9.06	8.35	9.35	8.65	9.11	**8.87**

Experiment 3:

Experiment 3 is performed to calculate the computational running time in second(s) of proposed and baseline predictors on dataset-1 and dataset-2. First, we calculate the running time of MED, GAP, FLIF, LBP, and iMED predictors on dataset-1, which detailed results are shown in Table 7. Average Computational Time (C. Time) summarizes results based on running time in second(s) for all the predictors, i.e., MED, GAP, FLIF, LBP, and iMED (ours). The C. Time for MED, GAP, FLIF, LBP, and iMED are 25.23s, 20.27s, 18.35s, 15.59s, and 11.42s, respectively. It clearly shows that the proposed iMED predictor takes less computational running time in second(s) compared to MED, GAP, FLIF, and LBP predictors. C. Time comparison results on dataset-1 are summarized in Fig. 11. The iMED predictor C. Time is low on all images of dataset-1 as compared to baseline predictors. Detailed results from the perspective of C. Time on dataset-2 are shown in Table 6. Average C. Time of the MED, GAP, FLIF, LBP, and iMED predictors are 17.53s, 14.96s, 12.68s, 11.31s, and 8.87s, respectively. It clearly shows that the proposed iMED predictor takes less computational running time in second(s) compared to MED, GAP, FLIF, and LBP predictors. C. Time comparison results on dataset-2 are summarized in Fig. 12. The iMED predictor C. Time is less on all images of dataset-2 as compared to baseline predictors except on K04 and K05 images.

Results of experiment 1 and experiment 2 clearly show that entropy value and bpp values on both dataset-1 and dataset-2 are less in the case of iMED predictor compared to MED, GAP, FLIF, and LBP predictors. Additionally, according to experiment 3, the proposed predictor (iMED) improves in terms of computational running time in second(s) compared to baseline predictors MED, GAP, FLIF, and LBP. However, the limitation of the iMED predictor is that it takes too much computational time when running on high-resolution images due to k-means clustering and calculating the local context of a pixel using a 20-dimensional difference (DDx20).

Table 7. *Computational running time in second(s) for MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-1. Where Av. CT = Average Computational Time*

Image name	**MED**	**GAP**	**FLIF**	**LBP**	**iMED**
Lena	21.90	20.16	20.24	18.36	10.48
Cameraman	23.87	21.56	20.99	16.58	13.239
Livingroom	20.29	19.01	18.96	15.70	11.42
Mandrill	19.26	18.79	17.16	15.17	14.98
Peppers	21.92	21.92	19.91	17.69	10.73
Pirate	21.07	19.77	17.37	14.36	10.38
Woman_blonde	33.56	19.91	15.78	12.15	09.85
Woman_darkhair	39.93	22.41	16.42	14.71	10.31
Av. CT	**25.23**	**20.27**	**18.35**	**15.59**	**11.42**

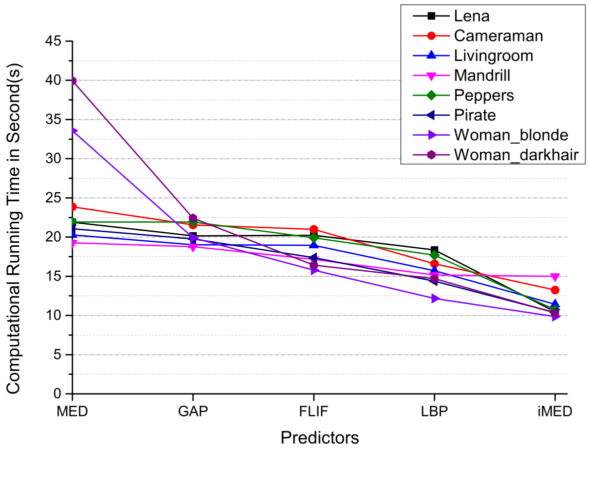


Fig. 11. *Computational running time in second(s) for MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-1 (lower is better)*

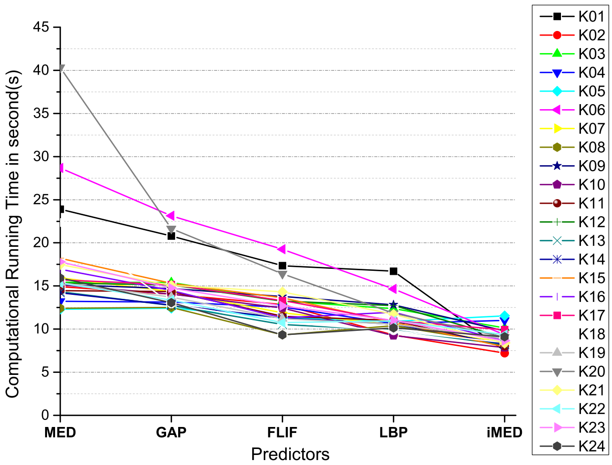


Fig. 12. *Computational running time in second(s) for MED, GAP, FLIF, LBP and iMED (ours) predictors on dataset-2 (lower is better)*

CONCLUSION:

A new predictor is proposed in this paper, i.e., "improved Median Edge Detection (iMED) for lossless image compression." The iMED predictor is designed specifically to improve the performance of the MED predictor by minimizing the entropy value, bpp, and the computational running time in terms of seconds(s). In this method, the MED predictor is improved by making k-means clusters of an input image and finding the local context of each pixel using DDx20. The performance of the proposed predictor is evaluated on standard bench-marked grey-scale test images and the KODAK images dataset. Experimental results demonstrate the visible improvement of entropy value and bpp compared to MED, GAP, FLIF, and LBP predictors. Experimental results also show that the proposed predictor takes less computational running time in seconds(s) than state-of-the-art predictors. However, the limitation of the iMED predictor is that it takes too much running time in seconds(s) for high-resolution images (i.e., 512x512 or higher resolution images) due to the clustering process and finding the local context of each pixel using DDx20. This iMED predictor for higher resolution images needs to be improved in the future.

REFERENCES

REFERENCES

Al-Khafaji G. (2012). Intra and inter frame compression for video streaming. Computer Science.

Al-Khafaji G, Al-Mahmood H (2016). Lossless image compression using adaptive predictive coding of selected seed values. Int J Comput Appl, 141(4), 26-29.

Al-Mahmood H, Al-Rubaye Z (2014). Lossless image compression based on predictive coding and bit plane slicing. International Journal of Computer Applications, 93(1).

Avramović A, Reljin B (2010). Gradient edge detection predictor for image lossless compression. In Proceedings ELMAR-2010 (pp. 131-34). IEEE.

Azman NAN, Ali S, Rashid RA, Saparudin FA, Sarijari MA (2019). A hybrid predictive technique for lossless image compression. Bulletin of electrical engineering and informatics, 8(4), 1289-96.

Standard Gray-Scale test images. Image Databases. Retrieved August 4, 2022, from https://www.imageprocessingplace.com/root_files_V3/image_databases.htm

True Color Kodak Images. Kodak. Retrieved August 4, 2022, from http://r0k.us/graphics/kodak/

Fouad, MM (2015). A lossless image compression using integer wavelet transform with a simplified median-edge detector algorithm. Int J Eng Technol, 15(04), 68-73.

Hameed ME, Ibrahim MM, Abd Manap N, Mohammed AA. (2020). A lossless compression and encryption mechanism for remote monitoring of ECG data using Huffman coding and CBC-AES. Future generation computer systems, 111, 829-40.

Haijiang T, Sei-ichiro K, Kazuyuki T (2005). A Study of Bias Correction Methods for Enhancing Median Edge Detector Prediction. In 2005 IEEE 7th Workshop on Multimedia Signal Processing (pp. 1-4). IEEE.

Hussain AJ, Al-Fayadh A, Radi N (2018). Image compression techniques: A survey in lossless and lossy algorithms. Neurocomputing, 300, 44-69.

Kabir MA, Mondal MRH (2018). Edge-based and prediction-based transformations for lossless image compression. Journal of Imaging, 4(5), 64.

Kamisli F (2016). A low-complexity image compression approach with single spatial prediction mode and transform. Signal, Image and Video Processing, 10(8), 1409-16.

Kwan C, Luk Y (2018). Hybrid sensor network data compression with error resiliency. In 2018 Data Compression Conference (pp. 416). IEEE.

Kwan C, Larkin J (2018). Perceptually lossless compression for Mastcam images. In 2018

9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 559-65). IEEE.

Li F, Hong S, Wang L (2019). A novel near lossless image compression method. In 2019 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE.

Li X, Orchard MT (2001). Edge-directed prediction for lossless compression of natural images. IEEE Transactions on image processing, 10(6), 813-17.

Marlapalli K, Bandlamudi RS, Busi R, Pranav V, Madhavrao B. (2021). A review on image compression techniques. Communication Software and Networks, 271-79.

Novikov D, Egorov N, Gilmutdinov M. (2016). Local-adaptive blocks-based predictor for lossless image compression. In 2016 XV International Symposium Problems of Redundancy in Information and Control Systems (REDUNDANCY) (pp. 92-99). IEEE.

Prasanna YL, Tarakaram Y, Mounika Y, Subramani R. (2021). Comparison of Different Lossy Image Compression Techniques. In 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-7). IEEE.

Qasim AJ, Din R, Alyousuf FQA. (2020). Review on techniques and file formats of image compression. Bulletin of Electrical Engineering and Informatics, 9(2), 602-10.

Rahman MA, Hamada M, Shin J. (2021). The impact of state-of-the-art techniques for lossless still image compression. Electronics, 10(3), 360.

Shanmathi G, Maniyath SR (2017). Comparative study of predictors used in lossless image compression. Asian J Appl Sci Technol (AJAST), 1, 10-13.

Sharma U, Sood M, Puthooran E (2021). A novel resolution independent gradient edge predictor for lossless compression of medical image sequences. International Journal of Computers and Applications, 43(8), 764-74.

Siddeq MM, Rodrigues MA (2017). A novel high-frequency encoding algorithm for image compression. EURASIP Journal on Advances in Signal Processing, 2017(1), 1-17.

Sneyers J, Wuille P. (2016). FLIF: Free lossless image format based on MANIAC compression. In 2016 IEEE international conference on image processing (ICIP) (pp. 66-70). IEEE.

Tiwari AK , Kumar RVR (2005). A switched adaptive predictor for lossless compression of high resolution images. In IEEE International Conference on Communications, 2005. ICC 2005. 2005 (Vol. 2, pp. 1097-1101). IEEE.

Tiwari AK, Kumar RR (2008). Least squares based optimal switched predictors for lossless compression of images. In 2008 IEEE International Conference on Multimedia and Expo (pp. 1129-32). IEEE.

Venugopal D, Mohan S, Raja S (2016). An efficient block based lossless compression of medical images. Optik, 127(2), 754-58.

Vura S, Patil P, Patil, SB (2021). A study of different compression algorithms for multispectral images. Materials Today: Proceedings.

Weinberger MJ, Seroussi G, Sapiro G (2000). The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Transactions on Image processing, 9(8), 1309-24.

Wu J, Liang Q, Kwan C (2012). A novel and comprehensive compressive sensing-based system for data compression. In 2012 IEEE Globecom Workshops (pp. 1420-25). IEEE.

Zhou J, Kwan C (2018). A Hybrid approach for wind tunnel data compression. In 2018 Data Compression Conference (pp. 435). IEEE.