

## Lossless Compression of Medical Images Using a Dual Level DPCM with Context Adaptive Switching Neural Network Predictor

**Emjee Puthooran\***

*Department of Electrical Engineering, Indian Institute of Technology Roorkee  
Roorkee, Uttarakhand, 247667, India*

**R S Anand**

*Department of Electrical Engineering, Indian Institute of Technology Roorkee  
Roorkee, Uttarakhand, 247667, India  
E-mail: anandfee@iitr.ernet.in*

**S Mukherjee**

*Department of Electrical Engineering, Indian Institute of Technology Roorkee  
Roorkee, Uttarakhand, 247667, India  
E-mail: shmeefee@iitr.ernet.in  
www.iitr.ernet.in*

Received 28 September 2012

Accepted 19 April 2013

### Abstract

A novel dual level differential pulse code modulation (DL-DPCM) is proposed for lossless compression of medical images. The DL-DPCM consists of a linear DPCM followed by a nonlinear DPCM namely, context adaptive switching neural network predictor (CAS-NNP). The CAS-NNP adaptively switches between three NN predictors based on the context texture of the predicted pixel in the image. Experiments on magnetic resonance (MR) images showed lower prediction error for the DL-DPCM compared to the GAP and the MED, which are used in benchmark algorithms CALIC and LOCO-I respectively. The overall improvement in data reduction after entropy coding the prediction error were 0.21 bpp (6.5%) compared to the CALIC and 0.40 bpp (11.7%) compared to the LOCO-I.

*Keywords:* dual level DPCM (DL-DPCM), neural networks, lossless image compression, medical image.

### 1. Introduction

Medical images provide anatomical and pathological details of the human body parts by noninvasive means. With the advancement of various imaging technology in the medical field, physicians tend to use medical images more extensively for disease diagnosis. Modalities such

as magnetic resonance (MR) and computed tomography (CT) occupy a considerable amount of data space, as they generate multiple slices for a single examination. Medical images may require to be stored for many years due to legal reasons. Modern medical practices such as telemedicine rely greatly on image transmission between healthcare organizations. Compressing the image before storage or transmission conserves storage

---

\* Corresponding author. E-mail : mgputhooran@gmail.com

space and bandwidth. Therefore, although numerous researches were carried out, medical image compression remains as an active field of study<sup>1</sup>.

Image compression techniques reduce the data size of an image by removing the irrelevant and redundant information. They are broadly classified into two — lossy and lossless. Lossy techniques<sup>2-5</sup> achieves higher compression ratios than lossless techniques, but causes permanent loss of information and may degrade image quality. Lossless compression techniques<sup>6,7</sup> preserve every detail in the original image.

Non-degraded images are essential for radiologists to interpret disease and to extract related diagnostic information. However, lossy compression alters the texture parameters of an image. Therefore, it may affect the diagnostic accuracy when interpreted by a radiologist. The accuracy of a computer aided diagnosis (CAD) system is also affected when lossy compressed image is used. In addition, legal reasons may require the medical images to be preserved without any loss of information. Therefore, lossless compression techniques are used most often to store medical images.

Most of the successful lossless coding schemes employ simple and efficient predictive coding techniques<sup>6</sup>. It involves pixel value prediction from causal neighboring pixels, modeling and entropy coding of the prediction error. Predictive coding technique is used by the two state-of-the-art lossless compression algorithms *viz.*, the context-based adaptive lossless image coding (CALIC)<sup>8</sup> and the low complexity context-based lossless image compression (LOCO-I)<sup>9</sup>. The LOCO-I is used at the core of the ISO standard for lossless image compression, JPEG-LS. The CALIC generally achieves a higher data reduction than the LOCO-I, with a higher computational complexity than LOCO-I. The JPEG-2000<sup>5</sup> includes a lossless mode based on integer wavelet transform. However, the lossless mode is slower and often achieves lower compression ratios than JPEG-LS.

There has been a continuous effort for improving the lossless image compression techniques by developing a variety of algorithms. Pan *et al.*<sup>10</sup> proposed a lossless embedded coding algorithm based on binary wavelet transform (BWT) called progressive partitioning binary wavelet-tree coder (PPBWC). Although the algorithm is fast, the compression performance is less than the benchmark lossless compression techniques. Zhang and Adjeroh<sup>11</sup> introduced prediction by partial approximate

matching (PPAM) technique for compression and context modeling. The algorithm needs to be trained offline before encoding and an average improvement of 0.03 bits per pixel (bpp) and 0.151 bpp over the CALIC and the JPEG-LS is achieved for a set of ten natural images. Zhao and He<sup>12</sup> proposed super-spatial structure prediction, in which the image area is partitioned into two regions: structure regions (SRs) and non-structure regions (NSRs). SRs are encoded with the super-spatial prediction, whereas NSRs are encoded with the CALIC. However, in this method only image with larger SRs achieves a higher compression ratio compared to CALIC.

Recently Taquet and Labit<sup>13</sup> proposed two variants of hierarchical oriented prediction (HOP) approaches using least square estimation (LSE) namely HOP-LSE and HOP-LSE<sup>+</sup>. The HOPE-LSE and the HOP-LSE<sup>+</sup> had an average improvement of 0.14 bpp and 0.21 respectively over CALIC, when tested on MR images. However, the computationally intensive least square optimization needs to be performed at both the encoder side as well as the decoder side.

Many researchers used computational intelligence for pixel value prediction and context modeling. Marusic and Deng<sup>14</sup> presented two adaptive prediction schemes based on adaptive neural network (AdNN) and local area training recursive-least mean square (LAT-RLMS) algorithm. The adaptation of the respective predictor coefficients is based on training of the predictors in a local causal area adjacent to the pixel to be predicted. The AdNN had an average improvement of 0.06 bpp over the CALIC and 0.24 bpp over the LOCO-I. The LAT-RLMS had an average improvement of 0.04 bpp over the CALIC and 0.22 bpp over the LOCO-I. However, on-the-fly training of predictors is required during encoding as well as decoding, which is computationally complex. Aiazzi *et al.*<sup>15</sup> proposed a fuzzy matching pursuits (FMPs) encoder, which consists of a space-varying linear-regression predictor obtained through fuzzy-logic techniques. It achieved an average improvement of 0.108 bpp and 0.29 bpp over the CALIC and the JPEG-LS respectively, when tested on a set of 24 natural as well as medical images.

Kau *et al.*<sup>16</sup> proposed the switching adaptive predictor (SWAP) with automatic fuzzy context modeling. The context of the coding pixel and the SWAP encoder switches between two predictors: the adaptive neural predictor (ANP) and the texture context

matching (TCM) predictor. The bit rate is improved by an average of 0.2 bpp compared with the CALIC for natural images. However, the online adaptive training used in ANP is computationally complex and both the encoder and the decoder need to perform the training.

Among the above techniques, the HOP-LSE<sup>+</sup> and the SWAP had some significant improvement in bpp compared to the CALIC. However, for both the HOP-LSE<sup>+</sup> and the SWAP, the predictor need to be optimized on-the-fly during encoding as well as decoding. This is a computationally intensive operation. In most of the medical imaging applications such as picture archiving and communication system (PACS) and telemedicine, an image is compressed only once but decompressed many times. Therefore, it is highly desirable to have a decoder with lesser computational complexity.

The focus of the present work is to develop a lossless compression technique with lower decoder complexity and higher coding efficiency for medical images such as MR. In this method a dual level differential pulse code modulator (DL-DPCM) has been proposed to obtain higher prediction accuracy. The DL-DPCM consists of a linear DPCM cascaded by a nonlinear DPCM. The nonlinear DPCM is realized by a context adaptive switching neural network predictor (CAS-NNP). After training, the neural network parameters are passed to the decoder as side information. This avoids the need of training the network while decompressing the image.

The rest of this paper is organized as follows: Section 2 provides the background theory and mathematical formulation. In section 3, details of the proposed method, dual level DPCM is provided. Section 4 introduces the data sets used for experiments. The software implementation details are given in section 5. In section 6 experimentation details are described. Section 7 includes the results and discussions and finally, the concluding remarks and future study are given in section 8.

## 2. Background Theory

### 2.1. Differential pulse code modulation

The differential pulse code modulation (DPCM) is a predictive decorrelation method. In DPCM, the current value of a signal is predicted from previously encoded values. The predicted value is subtracted from the original value to get the error signal. The error will be

small when the predicted signal nears to the original signal. An error signal containing mostly small values will have a peaked histogram, which can be efficiently coded by a variable length coder.

A two-dimensional DPCM (2D-DPCM) is most suitable for decorrelation of images, which are essentially two-dimensional signals. The 2D-DPCM predicts the current pixel value in a raster scan order from a causal template of neighboring pixels. Fig. 1 shows a causal template of 12 neighboring pixels,  $\{x_1, x_2, \dots, x_{12}\}$  and the pixel to be predicted,  $x_{i,j}$ . The significance of the causal template is that, it contains only the pixels that are already predicted in the raster scan order. During decompression, the same will be available at the decoder side as well.

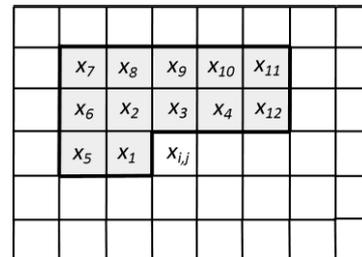


Fig. 1.  $x_{i,j}$  is the current pixel being predicted and  $C^{12} = \{x_1, x_2, \dots, x_{12}\}$  is a casual template (shaded region) with 12 neighboring pixels

Consider an image  $x$  having  $M \times N$  rows and columns, with pixel values  $x_{i,j}, i = 1, 2, \dots, M, j = 1, 2, \dots, N$ . The causal template  $C^n$ , with  $n$  number of neighboring pixels is defined as:

$$C^n = \{x_k : k = 1, 2, \dots, n\}. \tag{1}$$

The predicted pixel value  $\hat{x}_{i,j}$ , in a 2D linear DPCM (2D-LDPCM) scheme is given by:

$$\hat{x}_{i,j} = \sum_{k=1}^n \alpha_k x_k. \tag{2}$$

where,  $\alpha_k, k = 1, 2, \dots, n$  are predictor coefficients.

The prediction error is the difference between the original pixel value  $x_{i,j}$  and the predicted value  $\hat{x}_{i,j}$ :

$$\varepsilon_{i,j} = x_{i,j} - \hat{x}_{i,j}. \tag{3}$$

### 2.2. Artificial Neural Network

The structure of an artificial neural network with one input layer, one hidden layer and one output layer is shown in Fig. 2. The neurons of the input layer are connected to the hidden layer by the weight matrix  $W^1$  and that of hidden layer are connected to output layer by the weight matrix  $W^2$ . The number of layers and the number of neurons in each layer are chosen depending on the complexity of the problem by hit and trial method<sup>17</sup>. The output of the individual neuron is the weighted sum of inputs and bias value, mapped by an activation function (Eq. (4)), where  $w_i$  is the weight,  $x_i$  is the input to the neuron and  $b_i$  is the bias.

$$y = f(\sum w_i x_i + b_i). \tag{4}$$

Activation function can be either linear or nonlinear. The two commonly used nonlinear activation functions are sigmoid (Eq. (5)) and hyperbolic tangent sigmoid function (Eq. (6)).

$$f(n) = \frac{1}{1 + e^{-n}}. \tag{5}$$

$$f(n) = \frac{2}{1 + e^{-2n}}. \tag{6}$$

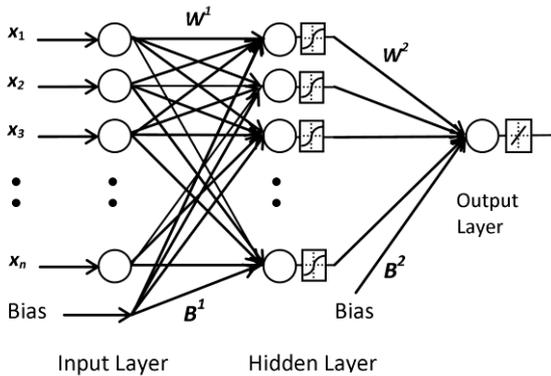


Fig. 2. The structure of an artificial neural network

The network weights and bias can be optimized by back-propagation algorithm. The standard back-propagation is a gradient descent method, in which the network weights are moved along the negative gradient of the performance function, calculated as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k. \tag{7}$$

where  $\mathbf{x}_k$  is a vector of current weights and biases,  $\mathbf{g}_k$  is the current gradient, and  $\alpha_k$  is the learning rate.

Improved learning performance are achieved by a variety of quasi-Newton algorithms and by using computational intelligence<sup>18</sup>.

### 2.3. Levenberg-Marquardt algorithm

The Levenberg-Marquardt (LM) algorithm<sup>19</sup> is a nonlinear least square algorithm used to approach a second-order training speed. When the performance function is computed as the mean square error, the Hessian matrix  $\mathbf{H}$  can be approximated from the Jacobian matrix as:

$$\mathbf{H} = \mathbf{J}^T \cdot \mathbf{J}. \tag{8}$$

where  $\mathbf{J}$  is the Jacobian matrix.

The Jacobian matrix contains first derivatives of the network errors with respect to the weights and biases. It can be computed through a standard backpropagation technique, which is less complex than computing the Hessian matrix.

The gradient  $\mathbf{g}$  is computed as:

$$\mathbf{g} = \mathbf{J}^T \cdot \mathbf{e}. \tag{9}$$

where  $\mathbf{e}$  is a vector of neural network errors.

The Levenberg-Marquardt algorithm updates the network parameters as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e}. \tag{10}$$

where  $\mathbf{x}_k$  is a vector of current weights and biases,  $\mu$  is a scalar parameter.

Eq. (10) reduces to Newton's method, if  $\mu$  is zero. If  $\mu$  is large, it becomes a gradient descent method. Newton's method is faster and more accurate near an error minimum. Therefore,  $\mu$  is decreased after a successful iteration and increased when an iteration would increase the performance function. This procedure ensures the reduction of performance function after each iteration.

### 3. The Proposed Method

The block diagram of the proposed method is shown in Fig. 3. In this scheme, the image is encoded sequentially, pixel by pixel, in a raster scan order. The inter-pixel redundancies present in the input image data

are removed in two stages by two DPCM which are cascaded *viz.*, 2D-LDPCM followed CAS-NNP. The 2D-LDPCM produces a prediction error image, which is the difference between the original image and the predicted image based on a subset of previously encoded pixels called the causal template. The CAS-NNP, which is adaptive and nonlinear, further removes the redundancies present in the error image.

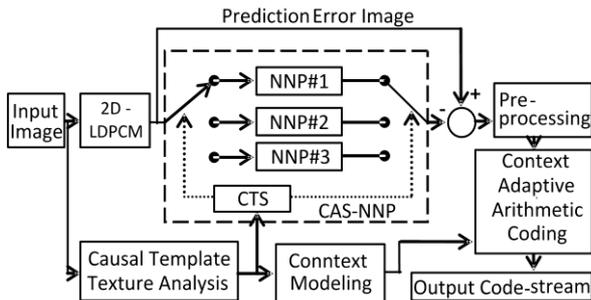


Fig. 3. Flow diagram of the proposed lossless image compression using the DL-DPCM

There are three NN predictors namely NNP#1, NNP#2 and NNP#3; each optimized to predict a specific area in the image *viz.*, plain-region, gradient-region and edge-region. The context texture switch (CTS) identifies the different regions of the image by texture analysis of the causal template. Finally, the prediction error image of CAS-NNP is entropy coded using context adaptive arithmetic coding after pre-processing to produce a binary output code-stream. The data size of the resulting binary code-stream will be less than the original image and the original image can be reconstructed from it without error.

### 3.1. Dual level DPCM

A linear predictor cannot accurately estimate a pixel from its causal template, since a nonlinear relationship exists between them. Different areas in an image have different mean pixel values. Since superposition theorem does not hold good for a nonlinear predictor, same variations in the causal template with different mean values will be presented to nonlinear predictor as different classes of inputs.

Improvements such as reduced complexity, higher accuracy and reduced learning time can be achieved if the mean value is subtracted from the causal template and only the variation is presented to the inputs of the nonlinear predictor. One method to realize this is to

divide the image into many blocks of sub-images and subtract the block mean value from each block. However, in this case the block mean value needs to be transmitted to the decoder, which will decrease the overall coding efficiency. Moreover, this will introduce sharp changes at the block edges, since the block mean values will be different for different blocks.

The error signal obtained by subtracting the linear predictor estimate of an image from the original image will be automatically normalized. In this case there is no need to transmit any value to the receiver end and the sharp changes due to mean value subtraction will not be introduced as in the former case. Therefore, in the proposed method a linear DPCM (2D-LDPCM) is cascaded with a nonlinear DPCM (CAS-NNP) to achieve a higher overall coding efficiency.

#### 3.1.1. 2D-LDPCM

A causal template with four neighboring pixels is used in the 2D-LDPCM for pixel value prediction as shown in Fig. 4.

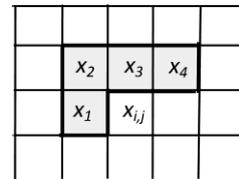


Fig. 4.  $C^d = \{x_1, x_2, x_3, x_4\}$  is the casual template (shaded region) used by 2D-LDPCM to predict the pixels  $x_{i,j}$

Firstly, an initial approximation of  $x_{i,j}$  is obtained by taking the average of two nearest neighboring pixel values  $x_1$  and  $x_3$ . Therefore, the value of coefficients corresponding to  $x_1$  and  $x_3$  (i.e.  $\alpha_1$  and  $\alpha_3$ ) is taken as  $1/2$ .

$$x_{i,j}(\text{initial}) = \alpha_1 \cdot x_1 + \alpha_3 \cdot x_3 = (1/2) \cdot x_1 + (1/2) \cdot x_3 \quad (11)$$

However, further improvement in prediction of  $x_{i,j}$  can be obtained by adding the difference between  $x_2$  and  $x_4$  to the initial prediction value. As the distance between  $x_2$  and  $x_4$  is double the distance between  $x_1$  and  $x_{i,j}$ , magnitude of the coefficients of  $x_2$  and  $x_4$  (i.e.  $\alpha_2$  and  $\alpha_4$ ) are taken as half of  $\alpha_1$  and  $\alpha_3$ .

$$\begin{aligned} x_{i,j}(\text{final}) &= x_{i,j}(\text{initial}) + \alpha_2 \cdot x_2 + \alpha_4 \cdot x_4 \\ &= (1/2) \cdot x_1 + (1/2) \cdot x_3 + (-1/4) \cdot x_2 + (1/4) \cdot x_4 \end{aligned} \quad (12)$$

Therefore, the set of coefficients used in the present work to predict  $x_{i,j}$  from the causal template is:

$$\alpha = \left\{ \frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4} \right\}. \tag{13}$$

### 3.1.2. Neural network predictor

Neural networks have the capability to model nonlinear functional relationships. They also have the ability of adapting to different classes of input data, by means of training. Since neural networks are parallel in architecture, they can also work efficiently with parallel computing hardware, where the computation time is reduced considerably.

A neural network (NN) architecture with I number of input neurons, H number of hidden layer neurons and O number of output layer neurons (I:H:O) is considered in the present work. The NN architecture (I:H:O)→(16:16:1) was decided empirically after repeated experimentation for different number of input layer neurons  $I \in \{4,6,8,\dots,16\}$  and different number of hidden layer neurons  $H \in \{2,4,8,\dots,16\}$ . It was observed that the NN architecture (I:H:O)→(16:16:1) yields the best performance in terms of prediction accuracy. The linear activation function gives a better accuracy, while nonlinear activation functions are used when the input – output relationship is nonlinear. Hence, hyperbolic tangent sigmoid activation function is used for the hidden layer and linear activation function is used for the output layer.

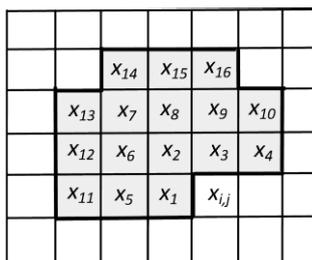


Fig. 5.  $C^{16} = \{x_1, x_2, \dots, x_{16}\}$  is the casual template (shaded region) used by NNP to predict the pixels  $x_{i,j}$

Levenberg-Marquardt algorithm is used for training as it is much faster than the gradient descent methods. Training data is prepared with 16 causal neighbouring values (Fig. 5) as input data set and the pixel to be predicted as the target. This is taken from the whole prediction error image produced by the 2D-LDPCM.

During the learning process, the NN predictor forms a relationship model between the neighbouring pixel values and the pixel value to be predicted. The learning process can be viewed as encoding the global image characteristics into the neural network. Thus, it is required to transmit the weight and bias values of the NN predictor to the decoder as side information. For the present case, there are 256 elements in the weight matrix  $W^1$ , each element representing a connection from input layer neuron to hidden layer neuron, and 16 elements in the weight matrix  $W^2$ , connecting hidden layer to the output layer. Similarly, there are 16 bias values for each neuron in the hidden layer and one bias value at the output layer. Thus, a total of 289 elements are transmitted as header information to the decoder. Using 16-bit floating point representation for each element, total of 4624 bits are sent as side information. This produces an overhead of 0.0176 bpp for a 512×512 gray scale image.

While predicting the pixels at the edges, some of the causal neighbouring pixel (or all of the neighbouring pixels in the case of the first pixel) are absent. This difficulty is overcome by padding the image with grey level value of 128. The use of other padding schemes, such as mirror padding will require transmitting the border pixel values to the decoder in advance, which will increase the overall bits per pixel (bpp) value.

### 3.1.3. Context adaptive switching

Since the image contains different textures at different areas, a single NN is inefficient to learn the whole pattern. By using three different NN predictors, each NN can be optimized to predict a particular class of texture. This also reduces learning time, since now the input pattern used by the individual NN has less variation compared to the original overall input data. The image is divided into three different regions: plain-region, gradient-region and edge-region. The different regions in the image are determined from the causal neighbouring pixels, so that the same steps can be carried out by the decoder. This avoids the overhead of sending the side information to detect different regions in the image.

A modified algorithm proposed by Kau et al.<sup>16</sup> is used for detecting the different regions. The variance of data in the plain-region is small, and that of an area with gradient-region and edge-region is large. The histogram will be evenly distributed in the case of gradient-region

but in the edge-region, histogram will have two peaks, representing the two sides of an edge. Ten causal neighbouring pixels  $x_1, x_2, \dots, x_{10}$  (similar to Fig. 5) are defined as the causal context ( $C^T$ ) for texture prediction.

In order to estimate the texture, mean  $\mu$  and variance  $\sigma^2$  of the context  $C^T$  are calculated. If the variance is less than a constant  $k_1$ , the texture is estimated as plain region. If the variance is greater than  $k_1$ , the texture is determined as gradient or edge region. The pixels in the context  $C^T$  are then divided into two sets: the pixels with grey level values less than  $\mu$  are formed into one group  $C^{TL}$  and the other are formed into group  $C^{TH}$ .

$$C^{TL} = \{\forall x_i \in C^T : x_i \leq \mu\}. \quad (14)$$

$$C^{TH} = \{\forall x_i \in C^T : x_i > \mu\}. \quad (15)$$

The variances of  $C^{TL}$  and  $C^{TH}$  are computed as  $\sigma_L^2$  and  $\sigma_H^2$  respectively. The relative variance  $\gamma$  is calculated as:

$$\gamma = \frac{\sigma^2}{0.01 + \sigma_L^2 + \sigma_H^2}. \quad (16)$$

A small value, 0.01 is added to the denominator of Eq. (16) so that it does not become zero when  $\sigma_L^2$  and  $\sigma_H^2$  are both zeros.

A region with an edge will have large variance ( $\sigma^2$ ), and small  $\sigma_L^2$  and  $\sigma_H^2$  values. Hence, if the variance  $\sigma^2$  is greater than or equal to  $k_1$  and relative variance  $\gamma$  is greater than or equal to a constant  $k_2$  then the context is estimated as an edge-region. If the variance  $\sigma^2$  is greater than or equal to  $k_1$  and the relative variance  $\gamma$  is less than  $k_2$ , then the context is estimated as gradient-region. The values of  $k_1$  and  $k_2$  in the algorithm were determined empirically as 25 and 2 respectively after experimentation for  $k_1 \in \{10, 25, 100\}$  and  $k_2 \in \{0.5, 1, 2, 3\}$ . The algorithm for the texture context estimation is listed below:

Calculate mean  $\mu$  and variance  $\sigma^2$  of  $C^T$

if ( $\sigma^2 < k_1$ )

assume: plain-region.

else,

Group all pixels in  $C^T$  less than  $\mu$  into  $C^{TL}$  and group the rest to  $C^{TH}$ .

Calculate the variances  $\sigma_L^2$  and  $\sigma_H^2$  of  $C^{TL}$  and  $C^{TH}$  respectively.

Calculate the relative variance,  $\gamma = \frac{\sigma^2}{0.01 + \sigma_L^2 + \sigma_H^2}$

if ( $\gamma < k_2$ )

assume: gradient-region.

else

assume: edge-region.

end

end

### 3.2. Lossless image coding

The prediction error from CAS-NNP is entropy coded after pre-processing, as explained below.

#### 3.2.1. Context Modelling and Entropy Coding

An adaptive arithmetic encoder<sup>20</sup> is used for entropy coding the prediction error. The advantages of adaptive arithmetic coding are that there is no need to transmit the symbol frequency table as side information to the decoder and it can encode efficiently when there is a variation in the symbol statistics during the encoding process. The adaptive arithmetic algorithm is inefficient while the frequency of the symbol table is being adapted to the source symbol statistics. Therefore, grouping the symbols into different sets is done which are having different frequency distribution. This improves the coding efficiency. However, for each symbol set, the symbol table is initialized to the same frequency at the beginning of the encoding process. Hence, increasing the number of symbol sets decreases the overall coding efficiency.

In order to improve the coding efficiency, error symbols are divided into different sets based on the contexts of the predicted pixel. The prediction errors are more at the top and left edge of the image and they form a separate set of symbols. A total of 13 different contexts were defined empirically based on the variance  $\sigma^2$  and the relative variance  $\gamma$  as defined in Eq. (16). Contexts with variance up to the value of 121 are classified into 11 classes on a logarithmic scale. Another class is formed with contexts having

$\sigma^2 \leq 400$  and  $\gamma \leq 64$ . The rest of the error symbols are grouped into another class. The following pre-processing is also done to improve the coding efficiency of the arithmetic encoder.

### 3.2.2. Error Remapping

When the image data has a dynamic range  $[0, L]$ , the prediction error takes values in the range  $[-L, +L]$ . This increases the total number of symbols in the error histogram, and reduces the coding efficiency of the entropy coder. However, for a predicted value  $\hat{x}_{i,j}$  of the pixel  $x_{i,j}$ , this can be mapped into the range  $[-L/2, L/2-1]$  since the error  $\varepsilon = x_{i,j} - \hat{x}_{i,j}$  can take values only in the range  $[-\hat{x}_{i,j}, L - \hat{x}_{i,j}]$ .

### 3.2.3. Histogram Tail Truncation

In the error histogram, large error occurs with very less frequency. Coding the values with very small frequency decreases the coding efficiency. The error symbols whose absolute values are larger than four times the standard deviation is replaced with an escape sequence. The corresponding error values are grouped into a separate symbol set and are entropy coded separately.

## 4. Dataset Details

For validation, testing and comparative analysis of the proposed algorithm, medical image datasets are collected from three sources. The dataset details are given below:

(i) Massachusetts General Hospital (MGH) Dataset<sup>21</sup>: This dataset is collected from MGH Dataset Center for Morphometric Analysis internet brain segmentation Repository (IBSR). All scans were acquired at the NMR Center of Massachusetts General Hospital with 1.5 T General Electric Signa. This dataset consists of BMP images of size 256×256. This dataset consists of following images:

Table 1 MGH dataset details

MR sequence	Plane	Slices
T1-weighted images	Coronal	18
T2-weighted images	Coronal	18
Proton density	Coronal	18

(ii) MicroDicom Dataset<sup>22</sup>: This dataset consists of DICOM images of MR sequences-T1-weighted, T2-weighted, post contrast T1-weighted and Flair images of

normal subjects in coronal, axial and sagittal plane. All scans were acquired at Toronto Tri-Hospital MR Centre with 1.5 T General Electric Genesis Signa. The size of the images is 512×512. This dataset consists of following image sequences:

Table 2. MicroDicom dataset details

MR sequence	Plane	Slices
T1-weighted images	Sagittal	13
T1-weighted images	Coronal	13
T2-weighted images	Axial	18
T2-weighted images	Coronal	20
Post contrast T1-weighted images	Coronal	15
Post contrast T1-weighted images	Sagittal	15
Flair images	Coronal	12

(iii) OsiriX Dataset<sup>23</sup>: This dataset consists of DICOM images of MR sequence-T2-weighted images of brain tumor patients in axial and coronal plane. All scans were acquired with 1.5 T Philips Medical Systems, Achieva. The size of the images is 270×320. This dataset consists of following image sequences:

Table 3. OsiriX dataset details

MR sequence	Plane	Slices
T2-weighted images	Axial	22
T2-weighted images	Coronal	28

## 5. Software Implementation

The binary executable of CALIC is provided by the authors<sup>24</sup>, and that of LOCO-I is provided by HP Labs<sup>25</sup>. The algorithm is implemented in MATLAB 9.0 on a PC having Intel® CORE™2 Duo, 2.0 GHz using Neural Network Toolbox. The publicly available MATLAB implementation of the arithmetic coder developed by Karl Skretting<sup>26</sup> is used in this work for entropy coding.

## 6. Experimentation Details

The experiments are performed for lossless compression of medical image data consisting of MR images. The evaluation criteria and results and discussions are included in the following sub-sections.

### 6.1. Evaluation criteria

For quantitative analysis, three parameters are evaluated for the proposed method. These parameters are described below:

### 6.1.1. Bits per pixel

Bits per pixel (bpp) represents the average number of bits required to represent each pixel of the image. The total number of bits in the image data is divided by the total number of pixels in the image to find the bpp value, which is given by:

$$bpp = \frac{n_b}{M \times N}. \quad (17)$$

where,  $n_b$  is the number bits of the output code-stream,  $M \times N$  are the number of rows and columns in an image.

### 6.1.2. Mean square error

Mean square error (MSE) measures the average of the squares of the prediction errors.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (x_{i,j} - \hat{x}_{i,j})^2. \quad (18)$$

where,  $x_{i,j}$  is the pixel value of the  $i^{th}$  row and  $j^{th}$  column of the original image and  $\hat{x}_{i,j}$  is the  $i^{th}$  row and  $j^{th}$  column of the predicted image.

### 6.1.3. First order entropy

Entropy is the measure of average information content present in a data. The first order entropy can be calculated from a frequency distribution of symbol sets as:

$$H(z) = -\sum_{i=1}^L P(a_i) \log P(a_i). \quad (19)$$

where,  $P(a_i)$  is the probability of the symbol  $a_i$  in the data set containing  $L$  symbols.

## 7. Results and Discussions

In this section, compression results obtained with the DL-DPCM on MGH, MicroDicom and OsiriX datasets are discussed. These results are compared with that obtained with the CALIC and the LOCO-I.

Table 4 gives the average first order entropy values of prediction errors obtained for the original image, the MED, the GAP and the DL-DPCM. Similarly, Table 5 compares the average MSE values obtained. The GAP and the MED are the predictors used in the CALIC and the LOCO-I respectively<sup>8,9</sup>. It is observed from the results that an average improvement of 38.2% and 17.8% in entropy values are achieved against the MED and the GAP respectively. Similarly, improvement of 90% and 50.3% in case of MSE values. It is evident from the results that the proposed DL-DPCM is able to remove the statistical redundancies efficiently as compared to aforesaid predictors. The DL-DPCM gives lower entropies and RMS values for each dataset. The entropy values of the original image are also given in column 2 of Table 4 for comparison. It can be seen that there is significant improvement in average value from 5.87 to 3.25 bpp. Similarly, a low average MSE value (2.95) is obtained by the proposed method whereas the average RMS value is quite high for the MED and the GAP i.e. 28.81 and 5.70, respectively. Better results are achieved by the DL-DPCM as the CAS-NNP is used, in which each NN predictor is tuned for a particular area in the image (plain, gradient and edge). The 2D-LDPCM contributes to the performance of the CAS-NNP by presenting only the variation in the context template to the CAS-NNP rather than the original pixel values.

Table 6 presents the comparative bpp values of the compressed images obtained by the proposed method DL-DPCM and the existing methods. It can be noted that the proposed method achieved lower bpp values for all the datasets. The average bpp value of the CALIC is 3.27 and 3.46 for the LOCO-I whereas lower bpp value of 3.06 is obtained for the proposed algorithm. An improvement of 0.21 bpp compared to the CALIC and 0.40 bpp compared to the LOCO-I are achieved. The percentage improvement over the CALIC and the LOCO-I are 6.5% and 11.7% respectively. The improvement in bpp values is due to higher image data decorrelation capability of the proposed DL-DPCM.

Table 4. First order entropy values for original image and prediction error images (i.e. difference of original and predicted image) obtained by the predictors MED, GAP and DL-DPCM

MR Image Dataset	Original Ent.	Ent. MED	Ent. GAP	Ent. DL-DPCM	% improvement over	
					MED	GAP
MGH-T1-weighted	5.98	5.33	4.03	3.50	34.2	13.1
MGH-T2-weighted	6.39	5.96	4.79	4.55	23.7	4.9
MGH- Proton density	6.37	5.72	4.41	4.12	28.1	6.7
MicroDicom T1-weighted (Sagittal)	5.58	4.83	3.32	2.51	48.1	24.5
MicroDicom T1-weighted (Coronal)	5.65	4.81	3.50	2.70	43.9	22.8
MicroDicom T2-weighted (Coronal)	6.17	5.48	4.19	3.35	38.9	20.0
MicroDicom Flair (Coronal)	6.61	5.73	4.11	3.02	47.2	26.4
MicroDicom T2-weighted (Axial)	6.28	5.51	4.06	3.16	42.6	22.2
MicroDicom post contrast T1-weighted (Coronal)	5.74	4.88	3.52	2.63	46.0	25.2
MicroDicom post contrast T1-weighted (Sagittal)	5.87	4.96	3.38	2.43	50.9	28.0
OsiriX –T2-weighted (Axial)	3.59	3.32	2.64	2.39	28.2	9.6
OsiriX –T2-weighted (Coronal)	6.22	6.30	5.19	4.64	26.3	10.6
AVERAGE	5.87	5.24	3.93	3.25	38.2	17.8

Note: Original Ent.: Original Entropy (i.e. First order entropy of the original image), Ent. MED: First order entropy of the prediction error image obtained by using MED predictor, Ent. GAP: First order entropy of the prediction error image obtained by using GAP predictor, Ent. DL-DPCM: First order entropy of the prediction error image obtained by using DL-DPCM predictor.

Table 5. MSE values for predicted image obtained by the predictors MED, GAP and DL-DPCM (i.e. MSE values between original and predicted images)

MR Image Dataset	MSE MED	MSE GAP	MSE DL-DPCM	% improvement over	
				MED	GAP
MGH-T1-weighted	30.96	6.56	3.15	89.8	52.0
MGH-T2-weighted	29.74	8.40	6.07	79.6	27.8
MGH- Proton density	39.37	7.02	4.57	88.4	34.9
MicroDicom T1-weighted (Sagittal)	14.28	2.83	1.41	90.1	50.2
MicroDicom T1-weighted (Coronal)	25.35	3.78	1.79	92.9	52.6
MicroDicom T2-weighted (Coronal)	34.31	6.07	2.92	91.5	51.9
MicroDicom Flair (Coronal)	28.76	4.47	1.94	93.2	56.5
MicroDicom T2-weighted (Axial)	34.46	4.84	2.17	93.7	55.1
MicroDicom post contrast T1-weighted (Coronal)	26.90	3.86	1.68	93.8	56.6
MicroDicom post contrast T1-weighted (Sagittal)	17.90	3.09	1.28	92.9	58.7
OsiriX –T2-weighted (Axial)	28.29	5.40	2.35	91.7	56.5
OsiriX –T2-weighted (Coronal)	35.46	12.11	6.03	83.0	50.2
AVERAGE	28.81	5.70	2.95	90.1	50.3

Note: MSE\_MED: MSE between original image and image predicted using MED predictor, MSE\_GAP: MSE between original image and image predicted using GAP predictor, MSE\_DL-DPCM: MSE between original image and image predicted using DL-DPCM predictor.

Table 6. Comparison of lossless compression performance of the proposed method with the CALIC and the LOCO-I.

MR Image Dataset	Bits per pixel (bpp) value			Bit gain over		% improvement over	
	CALIC	LOCO-I	Proposed	CALIC	LOCO-I	CALIC	LOCO-I
MGH-T1-weighted	3.41	3.61	3.32	0.08	0.28	2.4	7.8
MGH-T2-weighted	4.42	4.57	4.41	0.01	0.16	0.1	3.5
MGH- Proton density	3.96	4.13	3.94	0.02	0.19	0.5	4.5
MicroDicom T1-weighted (Sagittal)	2.70	2.98	2.48	0.22	0.50	8.0	16.8
MicroDicom T1-weighted (Coronal)	2.51	2.75	2.39	0.12	0.36	4.7	13.1
MicroDicom T2-weighted (Coronal)	3.20	3.47	2.94	0.27	0.53	8.3	15.3
MicroDicom Flair (Coronal)	3.51	3.85	3.01	0.51	0.84	14.4	21.9
MicroDicom T2-weighted (Axial)	3.44	3.72	3.10	0.34	0.63	9.9	16.8
MicroDicom post contrast T1-weighted (Cor.)	2.51	2.76	2.32	0.19	0.44	7.6	15.9
MicroDicom post contrast T1-weighted (Sag.)	2.68	2.96	2.40	0.28	0.57	10.6	19.1
OsiriX –T2-weighted (Axial)	1.64	1.79	1.53	0.11	0.25	6.5	14.3
OsiriX –T2-weighted (Coronal)	5.28	4.96	4.85	0.43	0.11	8.1	2.2
AVERAGE	3.27	3.46	3.06	0.21	0.40	6.5	11.7

## 8. Conclusions

A lossless compression scheme, with a dual level DPCM has been implemented. The dual level DPCM is realized by cascading the 2D-DPCM (linear DPCM) and the CAS-NNP (nonlinear DPCM). It is observed that the cascaded architecture of linear and nonlinear DPCM achieved higher decorrelation for MR images with varying intensity and texture patterns. The CAS-NNP switches between three different NN predictors based on the causal context information. The comparative analysis with the MED and the GAP shows that lower first order entropy and MSE values are obtained with DL-DPCM. It is also observed that the lossless coding using the DL-DPCM achieved lower bpp values compared to the state-of-the-art methods the CALIC and the LOCO-I. The improvement achieved using CAS-NNP, which a nonlinear predictor shows that the functional relationship between the pixels is considerably nonlinear in nature. The proposed lossless compression scheme with lower decoder complexity is useful for efficient storage and telecommunication of voluminous medical image data such as MR. Future studies involve optimization of the NNP architecture with techniques such as optimal brain damage.

## References

1. I. N. Bankman, (ed.), *Handbook of Medical Image Processing and Analysis*. (Academic Press, San Diego, 2000).
2. G. K. Wallace, The JPEG still picture compression standard. *IEEE Trans on Consumer Electronics*, 38 (1) (1992) 18–34, doi: 10.1109/30.125072.
3. J. M. Shapiro, Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Trans on Signal Processing*, 41 (12) (1993) 3445–3462, doi: 10.1109/78.258085.
4. A. Said and W. A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans on Circuits and Systems for Video Technology*, 6 (3) (1996) 243–250, doi: 10.1109/76.499834.
5. A. Skodras, C. Christopoulos and T. Ebrahimi, The JPEG 2000 still image compression standard. *IEEE signal processing magazine*, 18 (5) (2001) 36–58, doi: 10.1109/79.952804.
6. B. Carpentieri, M. J. Weinberger, and G. Seroussi, Lossless compression of continuous-tone images. *Proceedings of the IEEE*, 88 (11) (2000) 1797–1809, doi: 10.1109/83.568923.
7. A. Przelaskowski, Experimental Comparison of Lossless Image Coders for Medical Applications, in *Computer*

- Vision and Graphics* (Springer Netherlands, 2006), pp. 216–221, doi: 10.1007/1-4020-4179-9\_31
8. X. Wu and N. Memon, Context-Based, Adaptive, Lossless Image Coding. *IEEE Trans. on Communications*, 45 (4) (1997) 437–444, doi: 10.1109/26.585919.
  9. M. J. Weinberger, G. Seroussi, and G. Sapiro, The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Trans on Image Processing*, 9 (8) (2000) 1309–1324, doi: 10.1109/83.855427.
  10. H. Pan, W. Siu, and N. F. Law, Lossless image compression using binary wavelet transform. *IET Image Processing*, 1 (4) (2007) 353–362, doi: 10.1049/iet-*ipr*:20060195.
  11. Y. Zhang and D. A. Adjeroh, Prediction by Partial Approximate Matching for Lossless Image Compression, *IEEE Trans. on Image Processing*, 17 (6) (2008) 924–935, doi: 10.1109/TIP.2008.920772.
  12. X. Zhao and Z. He, Lossless Image Compression Using Super-Spatial Structure Prediction. *IEEE Signal Processing Letters*, 17 (4) (2010) 383–386, doi: 10.1109/LSP.2010.2040925.
  13. J. Taquet and C. Labit, Hierarchical oriented predictions for resolution scalable lossless and near-lossless compression of CT and MRI biomedical images. *IEEE Trans on Image Processing*, 21 (5) (2012) 2641–52, doi: 10.1109/TIP.2012.2186147.
  14. S. Marusic and G. Deng, Adaptive prediction for lossless image compression. *Signal Processing: Image Communication*, 17 (5) (2002) 363–372, doi: 10.1016/S0923-5965(02)00006-1.
  15. B. Aiazzi, L. Alparone, and S. Baronti, Fuzzy logic-based matching pursuits for lossless predictive coding of still images. *IEEE Trans on Fuzzy Systems*, 10 (4) (2002) 473–483, doi: 10.1109/TFUZZ.2002.800691.
  16. L. J. Kau, Y. P. Lin, and C. T. Lin, Lossless image coding using adaptive, switching algorithm with automatic fuzzy context modelling. *IEE Proc.-Vis. Image Signal Process.*, 153 (5) (2006) 684–694, doi: 10.1049/*ip-vis*:20045256.
  17. R. O Duda, P. E. Hart and D. G Stork, *Pattern Classification* (Wiley, New York, 2001).
  18. P. Chen, Y. Lu and Y. Chen, Extremal Optimization Combined with LM Gradient Search for MLP Network Learning. *Int. J. Comput. Intell. Syst.*, 3 (5) (2010) 622–631, doi: 10.1080/18756891.2010.9727728.
  19. M. T Hagan and M. Menhaj, Training Feedforward Networks with the Marquardt Algorithm. *IEEE Trans on Neural Networks*, 5 (6) (1994) 989–993, doi: 10.1109/72.329697.
  20. A. Said, Arithmetic Coding, in *Lossless Compression Handbook* (Elsevier Science, USA, 2003), pp. 101-152.
  21. <http://www.cma.mgh.harvard.edu.ibrs>.
  22. <http://www.microdicom.com/downloads.html>.
  23. <http://pubimage.hcuge.ch:8080/>.
  24. <http://www.ece.mcmaster.ca/~xwu/calicexe/>.
  25. <http://www.hpl.hp.com/loco/jlsrefV100.zip>.
  26. <http://www.ux.uis.no/~karlsk/proj99/index.html>.