WAVELET-BASED IMAGE COMPRESSION USING SUPPORT VECTOR MACHINE LEARNING AND ENCODING TECHNIQUES

Rakib Ahmed

Gippsland School of Computing and Information Technology
Monash University, Gippsland Campus
Australia.
Rakib.Ahmed@infotech.monash.edu.au

ABSTRACT
This paper presents a method of compressing still images combining the powerful features of support vector machine (SVM) for machine learning with discrete wavelet transform (DWT) in image transformation. DWT, based on the 'haar' wavelet, has been used to transform the image and the coefficients acquired from DWT are then trained with SVM using Gaussian kernel. SVM has the property that it selects a minimal number of coefficients to model the training data for a predefined level of accuracy. The coefficients are then quantized and encoded using the Huffman coding algorithm. The performance of the proposed method is aspiring and comparable with the existing image compression standards.

KEY WORDS
Image Compression, Discrete Wavelet Transform, Support Vector Machine, Regression.

1. Introduction

Image compression is one of the major technologies that enables the revolution of multimedia. Image compression techniques find several applications in the areas like, Internet, digital photography, medical, wireless and document imaging, image archives and databases, security and investigation, printing, scanning, and facsimile.

Machine learning algorithms have been used often in image compression. A method using the back-propagation algorithm in a feed-forward network is described in [1]. The compression ratio of the image recovered using this algorithm was generally around 8:1 with an image quality much lower than JPEG, one of the most well-known image compression standards. The compression scheme presented by Amerijckx et al. [2] based on vector quantization (VQ) of the discrete cosine transform (DCT) coefficients by the Kohonen map, differential coding by first order predictor and entropic coding of the differences gave better performance than JPEG for compression ratios greater than 30:1.

Robinson and Kecman in [3] and [4] have used image compression algorithms based on SVM learning of the DCT coefficients. The method has produced better image quality than JPEG in higher compression ratios.

Compression based on DCT has some drawbacks as described in the following section. The latest standard of still image compression JPEG2000 uses the state-of-the-art discrete wavelet transform (DWT) technology with the view of overcoming these limitations. In this paper an image compression algorithm based on wavelet technology is proposed that uses the support vector machine learning algorithm to achieve the goal. The result of compression is quite satisfactory and aspiring.

2. Discrete Wavelet Transform

Block-based DCT techniques are usually suffered from blocking artifacts at higher compression ratios (low bit rates). On the other hand, compressions based on Wavelet techniques provide substantial improvement in picture quality at lower bit rates [5].

If $f(t)$ is any square integrable function satisfying

$$\int_{-\infty}^{+\infty} \left|f(t)\right|^2 dt < \infty \quad \textbf{(1)}$$

the continuous time wavelet transform of $f(t)$ with respect to a wavelet is defined as

$$W(a,\tau) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{|a|}} \Psi^* \left(\frac{t-\tau}{a}\right) dt \quad \textbf{(2)}$$

where the real variables a and τ are dilation and translation parameters, respectively, and $*$ denotes complex conjugation [6].

The wavelet may be defined as

$$\Psi_{a\tau}(t) = |a|^{-1/2}\,\Psi\left(\frac{t-\tau}{a}\right) \quad (3)$$

The function, referred to as the mother wavelet, satisfies two conditions – it integrates to zero and is square integrable, or has finite energy.

In the wavelet transform, the window size in the time domain varies with frequency, i.e., longer time window for lower frequency and shorter time window for higher frequency. For image data, time-frequency plane concept becomes a space-frequency plane. The wavelet transform allows the spatial resolution and frequency bandwidth to vary in the space-frequency plane thereby results in achieving better bit allocation for active and smooth areas. For image compression using DCT one major difficulty is to choose the block size. The choice of the block size is a trade off between handling active areas and smooth areas of the image.

It is preferred to represent *f(t)* as a discrete superposition sum rather than an integral for digital image compression. Equation (3) now becomes

$$\Psi_{k,l}(t) = 2^{-k/2}\,\Psi\left(2^{-k}t - l\right) \quad (4)$$

where $a = 2^k$ and $\tau = 2^k l$ for discrete space with k and l both integers.

The corresponding wavelet transform can be rewritten as

$$W(k,l) = \int_{-\infty}^{+\infty} f(t)\,\psi^*_{kl}(t)\,dt \quad (5)$$

and the inverse transform as

$$f(t) = \sum_{k=-\infty}^{+\infty}\sum_{l=-\infty}^{+\infty} d(k,l)\,2^{-k/2}\Psi\left(2^{-k}t - l\right) \quad (6)$$

The values of the wavelet transform at those a and τ are represented by

$$d(k,l) = W(k,l)/C \quad (7)$$

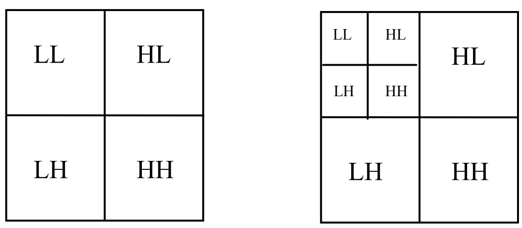
The *d(k,l)* coefficients are referred to as the discrete wavelet transform of the function *f(t)*. If the descretization is also applied to the time domain letting $t = mT$, where m is an integer and T is the sampling interval chosen according to Nyquist sampling theorem, then the discrete time wavelet transform is defined as

$$w_d(k,l) = \sum_{m=-\infty}^{+\infty} f(m)\,\psi^*_{kl}(m) \quad (8)$$

and the inverse discrete time wavelet transform as

$$f(m) = \sum_{m=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} d(k,l)\,2^{-k/2}\Psi\left(2^{-k}m - l\right) \quad (9)$$

The decomposition of an image using discrete wavelet transform comprises of a chosen low pass and a high pass filter, known as Analysis filter pair. The low pass and high pass filters are applied to each row of data to separate the low frequency and the high frequency components. These data can be sub-sampled by two. The filtering is then done for each column of the intermediate data finally results in a two dimensional array of coefficients containing four bands of data, known as low-low (LL), high-low (HL), low-high (LH) and high-high (HH). Each coefficient represents a spatial area corresponding to one-quarter of the original image size. The low frequencies represent a bandwidth corresponding to $0 < |\omega| < \pi/2$, while the high frequencies represent the band $\pi/2 < |\omega| < \pi$. It can be possible to decompose the LL band in the same way up to any level, resulting in pyramid-structured decomposition as shown Fig 1. The LL band at the top of the pyramid containing approximate coefficients holds the most significant information and the other bands containing details coefficients have lesser significance. Thus the degree of significance is decreasing from the top of the pyramid to the bands at the bottom.



a) 1st-level decomposition   b) 2nd-level decomposition

Fig. 1. Two-dimensional wavelet transform.

2. Support Vector Machine Learning

Support vector (SV) machines developed by Vapnik [8] can be used not only for classification problems but also for regression analysis, i.e., function estimation [9]. The SV machine implements the idea of mapping the input vectors x into a high-dimensional feature space Z through some chosen nonlinear mapping.

SV approximation to regression takes place if Regression is estimated in the set of linear functions $f(x,w) = (w^T.x) + b$, the problem of regression estimation is defined as that of risk minimization with respect to an ε-insensitive

($\varepsilon \geq 0$) loss function

$$L(y - f(x,w)) = |y - f(x,w)|_\varepsilon \quad \textbf{(10)}$$

where

$$|y - f(x,w)|_\varepsilon = \begin{cases} 0, & \text{if } |y - f(x,w)| \leq \varepsilon \\ |y - f(x,w)| - \varepsilon, & \text{otherwise.} \end{cases} \quad \textbf{(11)}$$

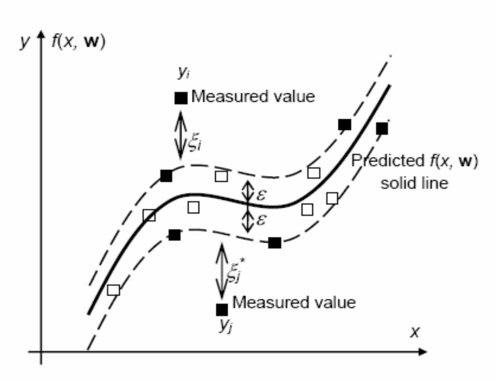


Fig. 2. Support Vector Machine Parameters.

The ε-insensitivity loss function (Equation 10) defines an ε - tube (Fig. 2). If the predicted value is within the tube, the loss is zero. In solving regression problems, SVM performs linear regression in *n*-dimensional feature space using ε -insensitivity loss function. At the same time, it tries to reduce model capacity by minimizing $\|w\|^2$, in order to ensure better generalization. These can be achieved by minimizing risk R

$$R_{w,\xi,\xi^*} = \left[\frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{l}\xi + \sum_{i=1}^{l}\xi^*\right)\right] \quad \textbf{(12)}$$

where *C* is a constant

under constraints

$$\begin{aligned} & y_i - w^T x_i - b \leq \varepsilon + \xi, && i = 1, \ldots, l \\ & w^T x_i + b - y_i \leq \varepsilon + \xi^*, && i = 1, \ldots, l \\ & \xi \geq 0, && i = 1, \ldots, l \\ & \xi^* \geq 0, && i = 1, \ldots, l \end{aligned} \quad \textbf{(13)}$$

where ξ and ξ* are slack variables and positive valued, shown in Fig. 2 for measurements above and below an ε - tube, respectively. Lagrange multipliers α_i and α_i*, corresponding to ξ and ξ*, will be nonzero values for training points above and below an ε - tube. For data points inside the tube, both multipliers equal zero. The constant *C*, which influences a trade-off between an approximation error and the weights vector norm $\|w\|$, is a design parameter chosen by the user. Insensitivity zone ε is the another most relevant learning parameters that can be utilized in constructing SV machines for regression. Increase in ε decreases the number of SVs at the cost of accuracy of approximation.

The constrained optimization problem can now be solved by forming a primal variables Lagrangian $L_p(\mathbf{w}, \xi, \xi^*)$

$$\begin{aligned} & L_P(w, b, \xi, \xi^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*) \\ & = \frac{1}{2} w^T w + C\left(\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\xi_i^*\right) \\ & - \sum_{i=1}^{l}\alpha_i^*\left[y_i - w^T x_i - b + \varepsilon + \xi_i^*\right] \\ & - \sum_{i=1}^{l}\alpha_i\left[w^T x_i + b - y_i + \varepsilon + \xi_i\right] \\ & - \sum_{i=1}^{l}\left(\beta_i^*\xi_i^* + \beta\xi_i\right) \end{aligned} \quad \textbf{(14)}$$

This problem can be solved in a dual space [3] and the solution may be given by

$$f(x,y) = \sum_{i=1}^{N_{SV}}\left(\alpha_i^* - \alpha_i\right)G(x_i, x) \quad \textbf{(15)}$$

subject to constraints

$$0 \leq \alpha_i^* \leq C, \qquad i=1, \ldots, l$$
$$0 \leq \alpha_i \leq C, \qquad i=1, \ldots, l$$

where N_{SV} is the number of support vectors and $G(x_i,x)$ is the kernel function.

4. Proposed Method

This section explains the proposed algorithm for compressing the coefficients found by applying discrete wavelet transform on an image data. The experimental

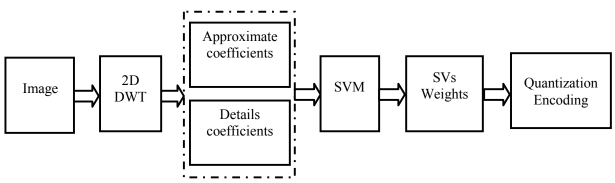


Fig. 3. Schematic diagram of the proposed image compression algorithm.

image is tiled into blocks, tile size of image-block being chosen by the user. The complete image may also be treated as a block. The two-dimensional discrete wavelet transform is applied on each tile treating them as one single image. The ‘haar’ wavelet has been used in this paper. The other wavelets may also be found suitable. The resulting approximate coefficients and details coefficients are then stored for each sub-image.

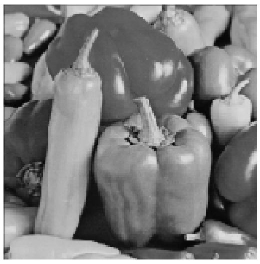
Support vector machine algorithm for regression analysis is then applied to each matrix of coefficients. The SV machine produces a minimum number of SVs required to generalize the training data within a predefined error (shown in Fig 2 as ε-insensitivity tube). It is found from the experiment that SVM performs better for not too large sets of coefficients as its training data. The coefficients are then quantized in predefined levels and encoded using Huffman coding principle. The proposed compression algorithm is shown schematically in Fig. 3.

5. Simulation Results

The gray-scale peppers image of size 512 X 512 (shown in Fig. 4a) has been taken to test the compression capability of the proposed method. The image to be compressed is first tiled into some blocks, for example it may be divided into blocks of 128 X 128 sub-images, or the whole image may be treated as one block. Dividing the image into blocks improves time complexity for compressing the image. The two-dimensional discrete wavelet transform iss applied on the sub-images treating each of them as a complete individual image. It gives the approximate coefficients, CA and details coefficients, DH, DV and DD.

For the experiment, the whole image of size 512 X 512 was treated as a block and after applying DWT the coefficient matrices generated were each of size 256 X 256. The support vector regression learning algorithm was applied on each set of coefficients. It was found that while applying SVM learning, time complexity increased for data sets more than a certain limit (such as for 256 X 256 coefficient matrix of the image in Fig. 4a) and the performance also deteriorates. To overcome this, the coefficient matrices were divided into blocks and then the SVM learning was applied individually on each block. This process has not only improved time complexity but also provided better generalization.

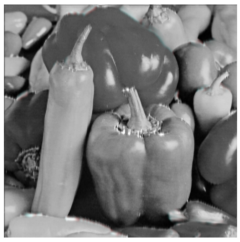
After the SVM regression algorithm was applied to the coefficient matrices for a predefined accuracy, two sets of coefficients for each matrix, namely, support vectors (SVs) and corresponding weights were achieved according to equation (15). These were then quantized and encoded.



a) Original (512 X 512)

Compression Ratio 36:1
PSNR 28.34

Compression Ratio 35:1
PSNR 28.129



Compression Ratio 42:1
PSNR 27.375

Compression Ratio 40:1
PSNR 27.212

b) Proposed Method c) JPEG

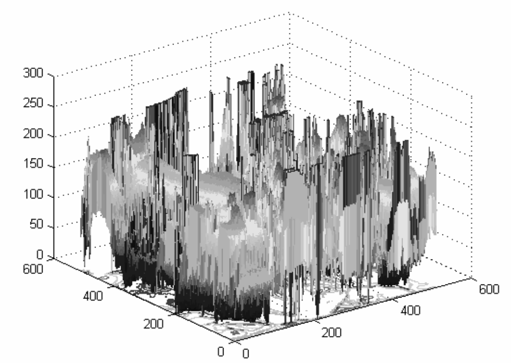
Fig. 4. Results of Experiments.

In the reverse process, the image was reconstructed following the decoding and dequantization process and by using the weights achieved thereby. Thus greater compression ratio is possible to achieve.

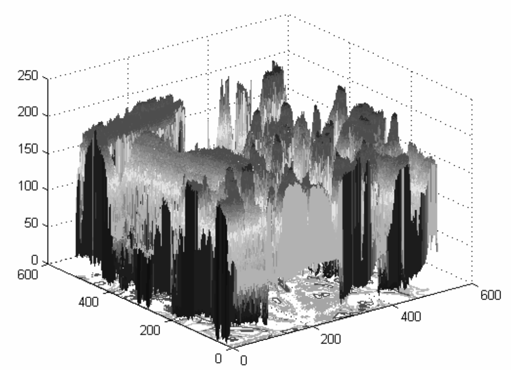
Fig. 4 shows the original image and the result of compression indicating compression ratio and peak signal to noise ratio (PSNR). Fig. 5 shows the three dimensional mesh plot of the original and the recovered image with 42:1 ratio of compression.

5. Conclusion

It may be concluded from the results of the experiment that the proposed method of compressing still images has shown aspiring performances. The proposed method has been performed better than JPEG. However, the comparison of compression ratio and picture quality with the latest standard, such as JPEG2000, is yet to be done. The method is very much in the early stage and there are several areas to improve the method in order to achieve better picture quality and higher ratio of compression, such as application of appropriate encoding techniques and wavelets, selection of suitable SV kernel.



(a)



(b)

Fig. 5. 3D mesh plot of the image (a) Original Image (b) Recovered Image with 42:1 compression.

References:

[1] M H Hassoun, Fundamentals of Artificial Neural Networks, Cambridge, MA: MIT Press, 1995.

[2] C. Amerijckx, M. Verleysen, P. Thissen, and J. Legat, Image Compression by self-organized Kohonen map, IEEE Trans. Neural Networks, vol. 9, pp. 503–507, 1998.

[3] J. Robinson and V. Kecman, Combining Support Vector Machine Learning with the Discrete Cosine Transform in Image Compression, IEEE Transactions on Neural Networks, Vol 14, No 4, July 2003.

[4] Jonathan Robinson, The Application of Support Vector Machines to Compression of Digital Images, PhD dissertation, School of Engineering, University of Auckland, New Zealand, February 2004.

[5] D S Taubman, M W Marcellin, JPEG2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, 2002.

[6] Y Q Shi and H Sun, Image and Video Compression for Multimedia Engineering –Fundamentals, Algorithms and Standards, CRC Press LLC, 2000.

[7] D. Chai and A. Bouzerdoum, JPEG2000 Image Compression: An Overview, Australian and New Zealand Intelligent Information Systems Conference (ANZIIS'2001), Perth, Australia, pp. 237-241 Nov. 2001.

[8] V N Vapnik, The Nature of Statistical Learning Theory, Springer, 2000.

[9] V Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, The MIT Press, 2001.