Robust Digital Image Watermarking

Using Quantization and Back Propagation Neural Network

Nallagarla Ramamurthy
Research Scholar, JNTUA, Anantapur, A.P, India
ramamurthy1006@gmail.com

S. Varadarajan
Professor, Dept. of ECE, S.V. University college of Eng., Tirupati, India
varadasouri@gmail.com

Abstract

The copyright protection of digital content became a critical issue now-a-days. Digital image watermarking is one of the techniques used to protect digital content. A novel image watermarking approach based on quantization and back propagation neural network is proposed in this paper. The cover image is decomposed up to 3-levels using quantization and DWT. The bitmap is selected as a watermark. The back propagation neural network is implemented while embedding and extracting the watermark. The proposed watermarking algorithm is imperceptible and robust to some normal attacks such as JPEG compression, salt and pepper noise, median filtering, rotation, and cropping.

Keywords: Digital Image Watermarking, Back Propagation Neural Network, DWT, Bitmap, Imperceptible, Robust, JPEG compression, Cropping.

1. Introduction

Copyright protection of multimedia data has become critical issue due to the massive spreading of broadband networks, easy copying, and new developments in digital technology. As a solution to this problem, digital image watermarking
became very popular now-a-days. Digital image watermarking is a kind of technology, that embeds copyright information into multimedia content. An effective image watermarking scheme mainly includes watermark generation, watermark embedding, watermark detection, and watermark attack [4]. Digital image watermarking provides copyright protection to image by hiding appropriate information in original image to declare rightful ownership [1]. There are four essential factors those are commonly used to determine quality of watermarking scheme. They are robustness, imperceptibility, capacity, and blindness. Robustness is a measure of immunity of watermark against attempts to image modification and manipulation like compression, filtering, rotation, scaling, noise attacks, resizing, cropping etc. Imperceptibility is the quality, that the cover image should not be destroyed by the presence of watermark. Capacity includes techniques that make it possible to embed majority of information. Extraction of watermark from watermarked image without the need of original image is referred to as blind watermarking. The non-blind watermarking technique requires that the original image to exist for detection and extraction. The semi-blind watermarking scheme requires the secret key and watermark bit sequence for extraction. Another categorization of watermarks based on the embedded data is visible or invisible [8].

According to the domain of watermark insertion, the watermarking techniques fall into two categories: spatial domain methods and transform domain methods. Many techniques have been proposed in the spatial domain such as LSB (Least Significant Bit) insertion method, the patch work method and the texture block coding method [6]. These techniques process the location and luminance of the image pixel directly. The LSB method has a major disadvantage that the least significant bits may be easily destroyed by lossy compression. Transform domain method based on special transformations, and process the coefficients in frequency domain to hide the data. Transform domain methods include Fast Fourier Transform(FFT), Discrete Cosine Transform(DCT), Discrete Wavelet Transform(DWT), Curvelete Transform(CT), Counterlet Transform(CLT) etc. In these methods the watermark is hidden in the high and middle frequency coefficients of the cover image. The low frequency coefficients are suppressed by filtering as noise, hence watermark is not inserted in low frequency coefficients [6]. The transform domain method is more robust than the spatial domain method against compression, filtering, rotation, cropping and noise attack etc.

Yonghong Chen and jiancong Chen method[9] presents a blind image watermarking scheme that embeds watermark messages at different wavelet blocks based on the training of BPNN in wavelet domain. In this paper a new blind watermarking scheme to embed bitmap into the Blue(B) plane of the cover image is presented. This technique is based on training Back Propagation Neural Network (BPNN) in the Discrete wavelet Transform Domain. While embedding the watermark, a secrete key is generated to determine the beginning of the watermark location. BPNN is implemented to embed and extract the watermark. The experimental results show that the proposed watermark technique is invisible and robust to attacks such like compression, cropping,
rotation, median filtering, and salt\& pepper noise attack.

2. Discrete wavelet Transform (DWT)

The discrete wavelet transform was invented by the Hungarian mathematician Alfred Haar. For an input represented by a list of $2^n$ numbers, the Haar Wavelet Transform simply pair up input values, storing the difference and passing the sum $[3]$. This process is repeated recursively, pairing up the sums, finally resulting in $2n-1$ differences and one final sum. DWT decomposes input image into four components namely LL, HL, LH, and HH. The lowest resolution level LL consists of the approximation part of the original image. Haar wavelet uses two types of filters. One is lowpass filter and the other is a high pass filter. The output of the low pass filter is obtained by averaging the input, while the output of the high pass filter is obtained from the differences of the inputs $[9]$. Low pass filter contained more information than high pass filter, because most of signal energy is concentrated in low pass filter. When an image is passed through a series of lowpass and high pass filters, DWT decomposes the image into sub bands of different resolutions $[1]$. Most of the energy of the image is concentrated in LL band. Hence modification of these low frequency sub bands would cost severe and unacceptable image degradation. So the watermark is not embedded in LL sub band. The good areas for watermark embedding are high frequency sub bands (vertical, horizontal and diagonal components). Human visual system is insensitive to these high frequency bands and effective watermark embedding is achieved without being perceived by human visual system. The basic idea of the DWT for two dimensional images described as follows. An image is first decomposed into four parts of high, middle, and low frequency sub components (LL1, HL1, LH1, and HH1) by critically sub sampling horizontal and vertical channels using sub component filters. The sub components HL1, LH1 and HH1 represent the finest scale wavelet coefficients. To obtain next level scaled wavelet components the sub component LL1 is further decomposed and critically sub sampled. This process repeated several times, which is determined by the application at hand $[5]$.

![Figure(1): Discrete Wavelet Transformation](image-url)
3. Back propagation Neural Network (BPNN)

A neural network represents a highly parallelized dynamic system with a directed graph topology that can receive the output information by means of reaction of its state on the input nodes. The ensembles of interconnected artificial neurons generally organized into layers of fields include neural networks. The behavior of such ensembles varies greatly with changes in architectures as well as neuron signal functions [2]. Artificial neural networks are massively parallel adaptive networks of simple non liner computing elements called neurons which are intended to abstract and model some of the functionality of the human nervous system in an attempt to partially capture some of its computational strengths. Neural networks are classified as feed forward and feedback networks. Back propagation network is of feed forward type. In BPNN the errors are back propagated to the input level.

The back propagation network with input, hidden and output layers is shown in figure (2). Bias is applied to both the hidden units and output units. The bias is always set to 1. The aim of this network is to train the net to achieve the balance between the ability to respond correctly to the input pattern that are used for training and the ability to provide good response to the input that are similar.

Figure (2): Back propagation neural network

The weight updating formula for BPNN is

\[
W_{jk}(t+1) = W_{jk}(t) + \alpha \delta_k y_j + \mu [W_{jk}(t) - W_{jk}(t-1)] \quad \ldots \quad (1)
\]

\[
V_{ij}(t+1) = V_{ij}(t) + \alpha \delta_j x_i + \mu [V_{ij}(t) - V_{ij}(t-1)] \quad \ldots \quad (2)
\]

where
Robust digital image watermarking

$W_{jk}$ are weights between hidden layer and output layer
$V_{ij}$ are weights between input layer and hidden layer
$\alpha$ is the learning rate parameter
$\mu$ is the momentum factor
$t$ represents time
$\delta_k$ is the error signal between output and hidden layers
$\delta_j$ is the error signal between hidden and input layers

Back Propagation Neural Network has good nonlinear approximation ability. It can establish the relationship between original wavelet coefficients and watermarked wavelet coefficients by adjusting the network weights and bias before and after embedding watermark. Owing to the use of neural network, we can extract watermark without the original signal and thus reduce the limit in practical applications. Using the neural network based on BPNN algorithm will meet the problem: how to determine the optimal structure of BPNN namely how to choose the network layers and the number of neurons. If we can’t select an appropriate network, it is difficult to significantly improve network performance, even if a large number of improvements have been made to the training algorithm.

The embedded technique of transform domain is proposed in this paper. And in order to improve the robustness, neural network is introduced, since it can fully approximate any complicated non-linear relationship. Thus the neural network model can well describe the relationship between selected wavelet packet coefficients and their neighborhood. For the multi-layer network, the number of input nodes and output nodes are determined by the problem itself. Choosing the size of network is mainly determining the size of nodes in hidden layer. The selection of nodes in hidden layer is very important for network training and studying. If hidden nodes are very few, the network will not have the necessary ability to learn and necessary information processing capabilities. Conversely, too many hidden nodes increase the complexity of network structure greatly, and is easier for neural network to fall into local minimum in the learning process. Meanwhile, it makes the network learn very slowly. The common method of selection is the trial and error method, generally based on experience to select the hidden layer of nodes, is very random. The empirical formula for determining the number of hidden nodes is

$n = (om + cm + d)^{1/2} \ldots (3)$

Where $n$ is the hidden nodes; ‘$m$’ is the number of input nodes; ‘$o$’ is the number of output nodes; ‘$c$’ & ‘$d$’ are the parameters to be determined. In general, the following posterior formula is used.

$n = (o m +1.6799m + 0.9298)^{1/2} \ldots (4)$

In this paper, $m = 8$, $o = 1$, $n \approx 4.8$, taking $n = 5$, The neural network has three layers, there are 8 nodes in input layer, 5 nodes in hidden layer, 1 node in output layer.
4. Watermark Embedding

The cover image is resized with 512x512 pixels, the R, G and B planes are separated and blue (B) plane is selected to embed watermark. The bitmap is selected as watermark and is resized to 64x64 pixels. The DWT is applied to blue plane of cover image and watermark is embedded in high and middle frequency components. The quantization levels selected as $Q_1=16$ and $Q_2=6$. The back propagation neural network is used to embed and extract watermark. The training process is completed before embedding. After getting the coefficients from the watermark image, the relationship between the high frequency wavelet coefficients and the watermark can be established. The extra information is used to train the neural network to make it sure it must have the capability of memorizing the characteristics of relations between the watermarked image and the watermark. Sigmoid activation function is used in the hidden layer and linear activation function is used in the output layer.

**Figure (3): Watermark embedding.**

**Watermark Embedding Algorithm**

Step 1. Read the color image of size N x N.
Step 2. Resize the color image to 512x512 pixels and use it as a cover image.
Step 3. Select the Blue (B) plane to embed the watermark.
Step 4. Read the image of size 64x64 as the watermark.
Step 5. The frequency subcomponents \{HH1, HL1, LH1, \{HH2, HL2, LH2\}, \{HH3, LH3, LL3\}\} are obtained by computing the third level DWT of the Blue plane of RGB cover image.
Step 6. Select the beginning position of watermark using the secret key.
Step 7. Quantize the DWT coefficient $T_{(j+key)}$ by $Q$ as the input to the BPNN, then get the output of BPNN.
Step 8. Embed the watermark using the following equation

$$T'_{(j+key)} = BPNN(\text{round}(T_{(j+key)}/Q) + x_j) \ldots (5)$$

Where $x_j$ is the random watermark sequence.
Step 9. Perform IDWT on each coefficient to get watermarked image.

### 5. Watermark Extraction

The watermark extraction process is that anti-process of watermark embedding. The trained neural network is used in the extraction process, because neural networks have associative memory which can realize blind detection. The normalized correlation coefficient is used to detect the correlation between the original watermark and extracted watermark.

**Watermark Extraction Algorithm**

Step 1. Transform the watermarked image by the DWT.
Step 2. Quantize the DWT coefficient $T''(j)$ by $Q$ as the input of BPNN, then get the output of BPNN as $\text{round}(T''(j)/Q)$.
Step 3. Extract the watermark $x'$ using the equation

$$x' = T''(j) - \text{BPNN}($$ round$(T''(j)/Q)) \ldots \text{ (6)}$$

where $j=1 \text{ to } 8$.
Step 4. Measure the NC of the extracted watermark $x'$ and the original watermark $x$.

### 6. Experimental Results

The algorithm of watermark embedding and extraction are implemented using MATLAB. Pears image of size 256x256 is selected as the cover image. Gray scale bitmap image of size 64x64 Barbara is selected as the watermark. The PSNR of the watermarked image is calculated using the formula
\[ PSNR = 10 \log_{10} \frac{(R \ast R)}{MSE} \]  \hspace{1cm} (7)

Where \( R \) = maximum fluctuation in the input image = 511

\[ MSE = \sum_{j=1}^{r} \sum_{k=1}^{c} \frac{[W(j,k) - W'(j,k)]^2}{rc} \]  \hspace{1cm} (8)

Where \( r \) = number of rows
\( c \) = number of columns

\( W(j,k) \) and \( W'(j,k) \) represent blue plane of cover image and watermarked image.

\[ NC = \frac{\sum_{j} \sum_{k} W'(j,k) \ast W'(j,k)}{\sum_{j} \sum_{k} W(j,k) \ast W(j,k)} \]  \hspace{1cm} (9)

The performance evaluation of the method is done by measuring imperceptibility and robustness. The normalized correlation coefficient (NC) is used to measure the similarity between the cover image and the watermarked image. Peak Signal-to-Noise Ratio (PSNR) is used to measure the imperceptibility of the watermarked image. The robustness of the watermarked image is tested by attacks such as JPEG compression, cropping, median filtering, salt & pepper noise attack, and rotation. The cover image of size 512x512 and watermark of size 64x64 are shown in figure (5). The watermarked image and extracted watermark without any attack are shown in figure (5). The cropping attacked image and extracted watermark after cropping are shown in figure (6). The median filtering attacked image and extracted watermark after median filtering are shown in figure (7). The JPEG compression attacked image and extracted watermark after JPEG compression are shown in figure (7). The salt & pepper noise attacked image and extracted watermark after salt & pepper noise attack are shown in Figure (8). The rotation attacked image and extracted watermark after rotation attack are shown in figure (9). The PSNR of the watermarked image after all the attacks is higher than 49, from which we can say that the imperceptibility is excellent.

Figure (5): Cover Image, Watermark, Watermarked Image, and Extracted watermark without any attack.

Figure (6): Cropped Image and Extracted Watermark after Cropping
Figure (7): Median Filtering attacked Image, Extracted watermark, and JPEG Compressed Image and Extracted Watermark after Compression.

Figure (8): Salt & Pepper Noise attacked Image, Extracted Watermark after Salt and Pepper Noise attack.

Figure (9): Rotation attacked Image and Extracted Watermark after rotation attack.

Table 1: Comparison of Results

<table>
<thead>
<tr>
<th>Type of Attack</th>
<th>Intensity</th>
<th>MSE</th>
<th>PSNR</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watermarked image</td>
<td></td>
<td>0.9202</td>
<td>48.2396</td>
<td>0.9781</td>
</tr>
<tr>
<td>cropping</td>
<td>5%</td>
<td>0.4441</td>
<td>51.3593</td>
<td>0.9545</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>0.4648</td>
<td>50.4304</td>
<td>0.9439</td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>0.4609</td>
<td>50.4144</td>
<td>0.9177</td>
</tr>
<tr>
<td></td>
<td>35%</td>
<td>0.4802</td>
<td>49.5968</td>
<td>0.9081</td>
</tr>
<tr>
<td>JPEG Compression</td>
<td>Q=10</td>
<td>0.4285</td>
<td>50.5214</td>
<td>0.8249</td>
</tr>
</tbody>
</table>
Table 1: Performance of the Proposed Algorithm under different attacks

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median Filtering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>0.4468</td>
<td>49.6735</td>
<td>0.8045</td>
</tr>
<tr>
<td>5%</td>
<td>0.6455</td>
<td>50.1625</td>
<td>0.9859</td>
</tr>
<tr>
<td>10%</td>
<td>0.6243</td>
<td>50.2142</td>
<td>0.9927</td>
</tr>
<tr>
<td>15%</td>
<td>0.6548</td>
<td>50.4213</td>
<td>0.9859</td>
</tr>
<tr>
<td>20%</td>
<td>0.6215</td>
<td>50.1754</td>
<td>0.9842</td>
</tr>
<tr>
<td>Salt &amp; Pepper Noise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5%</td>
<td>0.6243</td>
<td>50.2142</td>
<td>0.9927</td>
</tr>
<tr>
<td>10%</td>
<td>0.6243</td>
<td>50.2142</td>
<td>0.9927</td>
</tr>
<tr>
<td>15%</td>
<td>0.6548</td>
<td>50.4213</td>
<td>0.9859</td>
</tr>
<tr>
<td>20%</td>
<td>0.6215</td>
<td>50.1754</td>
<td>0.9842</td>
</tr>
<tr>
<td>Rotation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4°</td>
<td>0.6324</td>
<td>50.2536</td>
<td>0.9845</td>
</tr>
<tr>
<td>8°</td>
<td>0.6243</td>
<td>50.2458</td>
<td>0.9854</td>
</tr>
<tr>
<td>12°</td>
<td>0.6208</td>
<td>50.2120</td>
<td>0.9745</td>
</tr>
<tr>
<td>16°</td>
<td>0.6142</td>
<td>50.0124</td>
<td>0.9658</td>
</tr>
</tbody>
</table>

7. Conclusion

In this paper the watermark is embedded into the blue plane of the color image using quantization. Algorithm is robust to Salt & Pepper noise attack, cropping and rotation but weak to JPEG compression, and median filtering attacks. The PSNR is better than the methods proposed in [6] and [9]. Robustness to JPEG compression and median filtering attacks can be improved by proper training of neural network and proper selection of quantization coefficients. Robustness can also be improved by applying Fuzzy Logic approach. This algorithm can also be applied to video images.

References

Robust digital image watermarking


Received: December, 2011