A Novel Approach using Full Counterpropagation Neural Network for Watermarking

Prof. Ashish Bansal
Information Technology Department
Shri Vaishnav Institute of Technology and Science
Indore, India

Dr. Sarita Singh Bhadauria
Department of Electronics
Madhav Institute of Technology and Science
Gwalior, India

***Abstract*— Digital Watermarking offers techniques to hide watermarks into digital content to protect it from illegal copy or reproduction. Existing techniques based on spatial and frequency domain suffer from the problems of low Peak Signal to Noise Ratio (PSNR) of watermark and image quality degradation in varying degree. Earlier technique based on Full Counterpropagation Neural Network (FCNN) used the concept of embedding the watermark into synapses of neural net rather than the cover image to improve PSNR of watermark and to prevent image quality degradation. However, problems like "Proprietary neural net" and "sure win" still exist as explained in this work. This paper is an attempt to uncover and solve these problems. FCNN can be practically employed to obtain a successful watermarking scheme with better time complexity, higher capacity and higher PSNR with the suggested modifications.**

***Keywords*— Digital watermark, neural net, FCNN, Discrete Cosine Transform(DCT).**

I. INTRODUCTION

Digital watermarking should provide the qualities like imperceptibility, robustness, security of cover image. This paper is an attempt to uncover and solve problems related to the techniques used in [1] where FCNN was used to insert the watermark into synapses of FCNN rather than the cover image. A large number of techniques have been developed based on manipulating the bit plane of Least Significant Bit (LSB)[2], linear addition of watermark to cover image[2], using mid band coefficients of DCT transformed blocks to hide watermark[3], maximizing strength of watermark using Discrete Wavelet Transform(DWT) techniques[4], Using radial basis function(RBF)neural network to achieve maximum strength watermark[5], transforming color space of cover image and embedding watermark into saturation channel [6],Embedding watermark in the DC components of transformed blocks[7] etc. Principles of neurocomputing, and their usage in science and technology is well explained in [8] . Cox et al. [9] pointed that, in order for a watermark to be robust to attack, it must be placed in perceptually significant areas of the image. Schyndel et al. [10] generated a watermark using a m-sequence generator. Bas et al . [10] introduced a watermarking scheme using fractal codes. Bartolini et al. [11] utilized the properties of human visual system and generated watermark from DCT coefficients. Kundur and Hatzinakos [12] embedded the watermark in the wavelet domain where the strength of watermark was decided by the contrast sensitivity of the original image. Delaigle et el. [13] generated binary m-
sequences and then modulated on a random carrier. A method for casting digital watermarks on images and analyzing its effectiveness was given by I.Pitas[14] and immunity to subsampling was examined. Cox and Kilan [15] presented a secure algorithm for watermarking images using spread-spectrum techniques. Craver and Memon [16] proposed digital watermarks to resolve the copyright ownership.

However, these techniques suffer from the problems of unsatisfactory value of imperceptibility and robustness to various attacks as discussed in these papers. These techniques also have the problems related to security. Chun –Yu-Chang [1] proposed a wonderful technique of embedding the watermarks into synapses of FCNN rather than cover image. This helped to increase robustness and reduce imperceptibility problems to a great extent. However, this marvelous work suffers from a few problems discussed in the following sections which prevent its effective use in watermarking applications.

Section II discusses the previous technique[1] and its deficiencies. Section III suggests the remedies of these problems using block diagram of proposed scheme. Section IV provides the modified algorithm. Section V gives experimental results . Conclusion is given in Section VI followed by references.

II. EXISTING TECHNIQUES AND DEFICIENCIES

The first problem is "Proprietary FCNN". Anyone can train a FCNN with his own chosen set of weights to derive a watermark of his choice with any given image at the input layer of FCNN. Thus multiple claims may be made on a particular image by extracting different watermarks from the same cover image. This raises doubts on the ownership of the

digital content.

Second problem is "Sure Win". FCNN works on the principle of "competition learning". The cover image and the watermark are given at the input layer and the neuron of the input layer, which resembles most with this input pattern is declared winner and participates in producing the desired watermark at the output layer. This involves an iterative process of weight adjustments in both the layers. Thus, with each different image, one of the neurons must be the winner and may be trained to produce "some" watermark. It is quite possible that more than one input images resemble the weight pattern of the same neuron at the input layer. Thus, this neuron must be the winner in all the cases to produce the same watermark at the output layer for all the images. This raises problem of 'Authenticity', when one unauthentic image produces the correct watermark. The above problems require the need of an additional authentic information to be hidden in the image itself. This may be done by using techniques based on spatial or frequency domain. However, it is still profitable to keep this small information of encoding bits in the cover image instead of a much larger watermark image . The much higher capacity and much more robust watermark with little degradation of the input image is the real benefit of the FCNN, which is still preserved.

III. A NOVEL APPROACH USING FCNN FOR WATERMARKING

The above problems may be solved by using an encoded image rather than the original cover image at the input layer of FCNN. For embedding the cover image, first it is encoded using encoding bits and then the image is given to FCNN along with the desired watermark at the input layer. As shown in the Fig.1, cover image is converted into Discrete Cosine Transform (DCT) block by block and encoding bits are embedded in the mid band coefficients of the blocks . Inverse Discrete Cosine Transform (IDCT) of this embedded cover image is given to the input layer of FCNN with the desired watermark to obtain watermarked cover image and the watermark images at the output layer of FCNN. For the extraction of watermark from this watermarked image Fig. 2 can be referred. First the watermarked image is DCT converted blockwise. Then encoding bits are obtained from this image using the extraction algorithm and compared with the original encoding bits. If the match is found, then IDCT of this image is taken and given to the input layer of the trained FCNN to extract the watermark, otherwise error message is displayed. In case of multiple claims, the real owner may derive the encoded bits from the controversial image to justify his claim which is not possible for the others. This solves the first problem of "Proprietary neural net". Also, the suspected image is given to the trained FCNN for extraction only when the encoded bits are successfully derived from the suspected image. This encoded image is supplied as input to the FCNN. The algorithm to obtain output watermark works only when the image is authentic. Thus, the second problem of "sure win" is also resolved.

Embedding the entire digital watermark into the cover image is very restrictive. The quality of image degrades severely when a large information is embedded into its DCT coefficients. This results in a very poor PSNR value of the cover image. The basic idea behind watermarking is to embed secret information successfully into a digital image in such a way that the visual quality of the cover image is not much disturbed and it should not be possible for the user to distinguish between a normal image and a watermarked image. However, if the embedding strength of the watermark is kept low, it causes problems of low robustness. The embedded watermark is very easily destroyed by general image processing operations or malicious attacks. The watermark, should be robust enough for such image processing operations and attacks. Thus, there is a tradeoff between imperceptibility and robustness. With the present technique, even a large watermark can be successfully embedded as it is not embedded in the cover image but inside the synapses of a trained FCNN. Only a few encoding bits are required to be embedded in the transformed cover image, with much lesser distortion of the cover image.

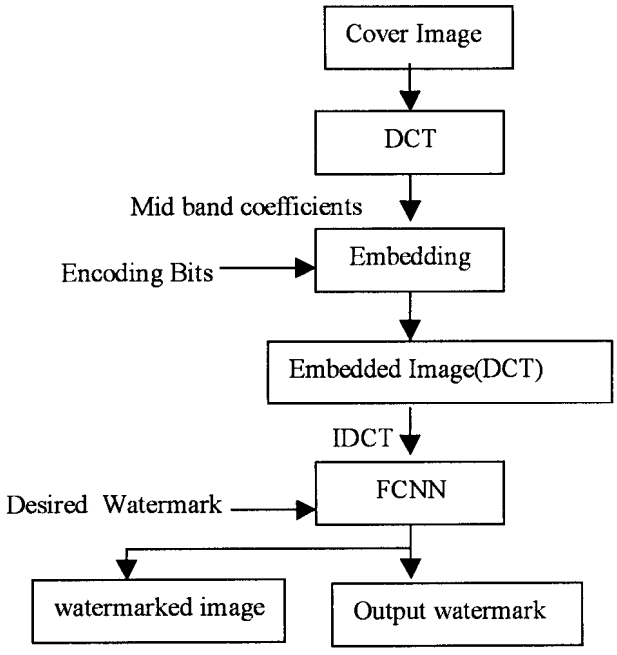

Fig. 1 Block Diagram : Watermarking the encoded image using FCNN

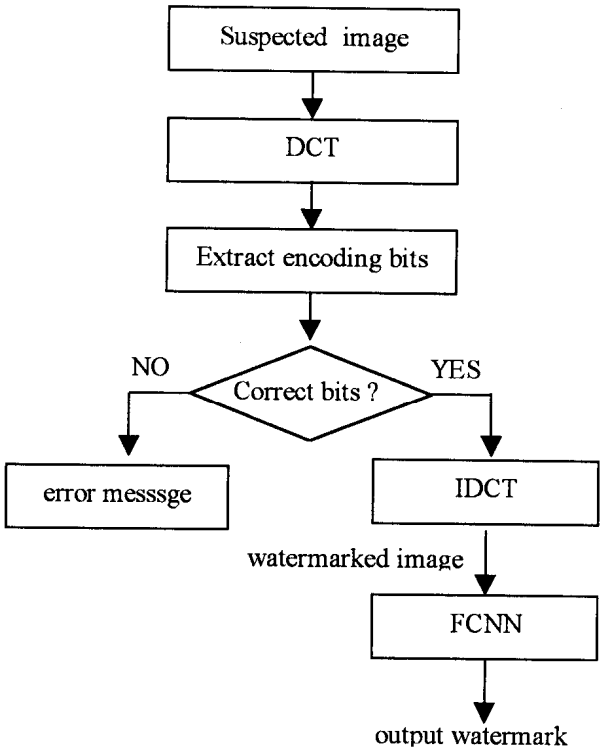

Fig. 2 Block Diagram : Extracting watermark from watermarked image

IV. ALGORITHM

A. EMBEDDING

The blocksize for image segmentation is set as

$$blocksize=8 \quad (1)$$

Following assumptions are made.

mc represents the number of rows in the cover image.

nc represents the number of columns in the cover image.

mm represents the number of rows in the encoding binary message matrix.

nm represents the number of columns in the encoding binary message marix.

The midband coefficients selection matrix is given as

$midband = [mid_{11}, mid_{12}, \ldots\ldots mid_{ij}, \ldots, mid_{88}]$ for $1<=i<=8$, $1<=j<=8$ and $mid_{ij} = 0$ or 1 (2)

($mid_{ij} = 1$ shows the presence of midband coefficient).

($mid_{ij}=0$ shows that it is not midband coefficient).

The cover image is given as

$cover\_image = [c_{11}, c_{12}, \ldots\ldots c_{ij}, \ldots\ldots c_{mc \times nc}]$ for $1<=i<=mc$, $1<=j<=nc$ (3)

The maximum length of message to be embedded is given as

$$max\_message = \frac{mc \times nc}{(blocksize)^2} \quad (4)$$

Each 8×8 block cotains one encoding message bit.

Let the binary message containing bits to be encoded in the image is given as

$$message = [m_{11}, m_{12}, \ldots m_{1_{nm}} \ldots, m_{ij}, \ldots m_{mm \times nm}] \text{ for } 1 \le i \le mm,\ 1 \le j \le nm,\ m_{ij} = 0 \text{ or } 1 \quad (5)$$

The message is reshaped as a column vector.

$$message = [m_{11}, m_{12}, \ldots\ldots, m_{mm \times nm}] \quad (6)$$

Now, a message vector is created with a length equal to max_message and initial bits same as desired message and rest of the bits initialized to '1'.

$$message\_vector = [mv_1, mv_2, \ldots\ldots mv_{mm \times nm}, \ldots\ldots, mv_{\max\_message}]$$

$$mv_i = message\ (i) \text{ for } 1 \le i \le (mm \times nm),$$

$$mv_i = 1,\ i > (mm \times nm) \quad (7)$$

Let, the cover image be termed as encoded image.

$$encoded\_image\ (i,j) = cover\_image\ (i,j) \text{ for } 1 \le i \le mc,\ 1 \le j \le nc \quad (8)$$

Now, generate a sequence containing as many random numbers as the sum of mid band coefficients in the 8×8 block termed as *pn_sequence_zero* to mark the presence of zero.

$$sum\_midband\_coefficients = \sum_{i=1}^{8}\sum_{j=1}^{8} midband(i, j) \quad (9)$$

$$pn\_sequence\_zero = \text{round}\left(2 \times \text{rand}\ (1, sum\_midband\_coefficients)\right) - 0.5 \quad (10)$$

Let us encode $R = p \times q$ blocks of the cover image with the message bits where,

$$p = mc \bmod 8 \quad (11)$$

$$q = nc \bmod 8 \quad (12)$$

Let the variables used in algorithm be initialized as

$$x=1, y=1, repeat\_times = 1 \quad (13)$$

Now, repeat the following steps *R* times to pick up *R* blocks of the cover image to encode *R* message bits. (One in each block) .

Step 1:

Let *cover_image(i*1:*i*2*)* be defined to select a block containing all elements of cover image (i1,i2), such that $x<=i1<=x+blocksize-1$, $y<=i2<=y+blocksize-1$.

Find the DCT transformation of cover image blockwise.

$$dct\_block = \text{DCT}\ (cover\_image\ (i_1 : i_2)) \quad (14)$$

Step 2:

The initial index of *dct_block* is set as

$$pos = 1 \quad (15)$$

if *message_vector(repeat_times)* = 0 then Embed the *pn_sequence_zero* into *dct_block* as per following equation.

$$dct\_block\ (jj,ii) = dct\_block\ (jj,ii) +$$

pn_sequence_zero(pos), $\forall jj, \forall ii : midband(jj, ii) = 1$,
for $1 \leq ii \leq blocksize$, $1 \leq jj \leq blocksize$, where
for each new pair *(jj,ii), pos = pos* + 1 (16)

Step 3: Now, encoded image block is obtained by taking the inverse DCT transform.
encoded_image (i_1: i_2) = IDCT(*dct_block*) for
$x \leq i_1 \leq x + blocksize - 1$, $y \leq i_2 \leq y + blocksize - 1$ (17)

Step 4: Now, x is incremented. If x crosses the total number of columns, it is reinitialized and next row is taken.
$x = x + blocksize$ (18)

$x = 1 \wedge y = y + blocksize \quad for \ (x+8) > nc$ (19)

Step 5: *repeat_times = repeat_times* + 1 (20)
Go to step 1 for *repeat_times*<= *R*

Now, this encoded image has to be supplied to the Full Counter Propagation Network at the input layer along with the desired watermark for training .
This encoded image can be represented as a column vector , and used as a cover image in FCNN.
$X = [x_1, x_2, x_3, ... x_{mc,nc}]$ (21)
where $mc \times nc$ is the total number of pixels in the encoded image.This image is supplied with the watermark
$Y = [y_1, y_2, y_3, ... y_m]$ to be embedded to the input layer of FCNN as per the procedure indicated in [1] . This FCNN after training, gives the watermarked image
$X=[x_1, x_2, x_3, ... x_{mw,nw}]$ and the desired watermark
$Y=[y_1, y_2, x_3, ... y_n]$ at the output layer.

B. EXTRACTING

The *midband* matrix and *blocksize* are taken same as in the embedding procedure . R= Total no. of image blocks as discussed in the embedding procedure.
Following assumptions are made.
mw represents the no. of rows in the watermarked image.
nw represents the no.of columns in the watermarked image.
mm represents the no. of rows in the encoding binary message matrix.
nm represents the no. of columns in the encoding binary message matrix.
Maximum size of message is given by

$$max\_message = \frac{mw \times nw}{blocksize^2} \quad (22)$$

Generate a *pn_sequence_zero* with same random state key as while embedding.
Taking x=1, y=1
Let encoded_*image(i1:i2)* be defined to select a block containing all elements (i1,i2) such that
x<=*i*1<=*x*+*blocksize*-1 , *y*<=*i*2<=*y*+*blocksize*-1.
Now, repeat the following steps *R* times to pick up *R* blocks of the *watermarked_image* to decode embedded *R* message bits. (One in each block) .
repeat_times = 1

Step 1:
The DCT coefficient of watermarked image is obtained blockwise as under.
dct_block =DCT(*encoded_image*(i_1:i_2)) (23)

The initial index of *dct_block* is set as
$pos = 1$ (24)

Embedded sequence is obtained as under.
sequence (pos) = dct_block (jj,ii), $\forall ii, \forall jj$:
for $1 \leq jj \leq blocksize$, $1 \leq ii \leq blocksize$, *midband (jj, ii)*=1
where, for all new pair (*jj, ii)* , pos = pos+1

Step 2:
Correlation of the obtained sequence is done with zero sequence.
correlat(repeat_times)=corr*(pn_sequence_zero,sequence)* (25)

Now, x is incremented. If x crosses the total number of columns, it is reinitialized and next row is taken.
$x = x + blocksize$ (26)

$x = 1 \wedge y = y + blocksize \ for \ x+8>nw$ (27)

Step 3:
repeat_times = repeat_times+1 (28)
Go to step 1 for *repeat_times*<=*R*

Step 4:
Now, correlation of all sequences derived above is checked with *pn_sequence_zero* to mark the presence of '0' bit.
message_vector(kk) = 0 for *correlate(kk)*>0.55
and *message_vector (kk)* = 1 *(otherwise)* ,
$\forall$ *kk:* $1 \leq kk \leq mm \times nm$ (29)

Step 5:
extractflag =1, if, $\forall$ *i : message_vector (i) = message(i) and extractflag =0 (otherwise)* for $1 \leq i \leq mm \times nm$ (30)

Step 6:
if $extractflag = 0$
"Image is not authentic and not supplied to counterpropagation network for extracting the watermark."
Otherwise ,"Image is authentic and should be supplied to counterpropagation network for extracting the watermark".
Now, supply the encoded image X at the input layer of the trained counter propagation network and obtain the watermark at the output layer as per the procedure indicated in [1].

V. EXPERIMENTS CONDUCTED WITH AND THE RESULTS:

In order to show that the modified scheme of FCNN produces the correct results and eliminates the problems of proprietory neural net' and 'sure win', three experiments were conducted. In the first experiment, the cover images used are non encoded disc image of size(117×114) Fig. 3 and the DCT encoded disc image of size(117×114) Fig.4 respectively and the watermark image is Lena's image Fig.5. The binary two dimension message (4×8) used for encoding is taken as

message=[0 0 0 0 1 1 1 1;
0 0 0 0 1 1 1 1;
0 0 0 0 1 1 1 1;
0 0 0 0 1 1 1 1]; (31)

In the second experiment, the cover image is disc image(117×114) Fig.6 and the two different watermark images are taken as 'Lena' (117×114), Fig.7 and a peson's image(size) Fig.8 respectively. In the third experiment, the two different cover images taken are 'Disc image' (117×114) Fig.9 and the person's image(size) Fig.10. The watermark image chosen is 'Lena's' Image(117×114) Fig. 11 The watermark images chosen are much larger than that selected in [1].The figures shown display these images.

To calculate the Peak Signal to Noise ratio (PSNR),the following formula is used.

$$PSNR(DB) = 10\log_{10}\frac{X^2 peak}{\sigma_e^2} \quad (32)$$

Where,

$$\sigma_e^2 = \left(\frac{1}{MN}\right)\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{ij} - Z_{ij}\right)^2 \quad (33)$$

Where $M \times N$ is the size of cover image, X_{ij} is the gray level of (i,j) pixel of the cover image. Z_{ij} denotes the gray level of (i,j) pixel of the watermarked image.X^2peak shows the squared peak values of the cover image. The higher PSNR means more similar encoded image and the cover image. All experiments were conducted on genuine intel (R) CPU T-2050 @1.60GHZ, 504 MB OF RAM. The operating system used was Microsoft Windows XP Home edition, Version 2002, Service Pack 2. The random number generator state is taken as 100 both while embedding and extracting the encoded bits. While using the FCNN, number of neurons in the hidden layer is taken as 6. The initial learning rate for the input layer is taken as 0.95 and learning rate for the output layer is taken as 0.97. Learning rate of input layer is kept high initially and then reduced exponentially with each iteration for faster convergence as per following.

New learning rate=(previous learning rate)$\times$exp(-k/k0);

Where, k= iteration count. And k0 is fixed at 10. The threshold value is reduced from 1 to 0.00001 gradually in fractions of 0.1 as indicated in the table I.

Experiment 1:

In this experiment, the training of the FCNN is done with the help of non-encoded cover image Fig. 3 and the DCT-encoded cover image Fig. 4 respectively. Fig. 5 shows the output watermark.

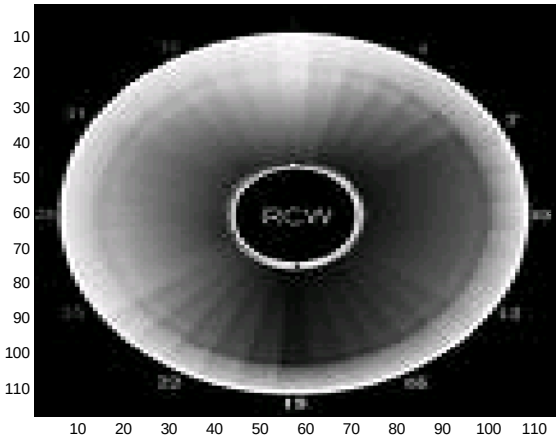

Fig.3 Non-Encoded cover image(Disc)

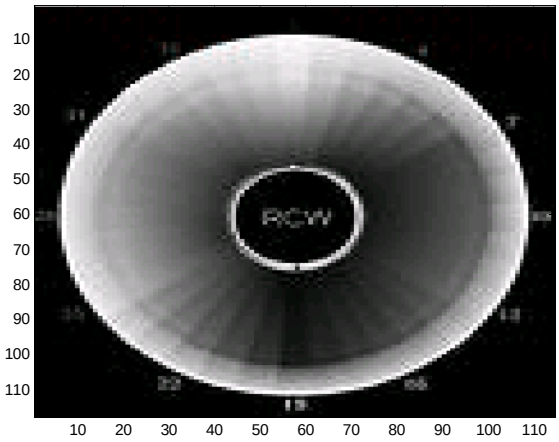

Fig.4 Encoded Cover Image(Disc)

Fig.5 Output watermark(Lena)

The table given below shows the threshold values, Number of epochs used in training, training time and PSNR values of the watermarked image and watermark obtained in both the cases. The encoding of the cover image is done with the help of message bits shown in equation (31).

TABLE I
(Non – Encoded image as cover image)

Threshold	PSNR of cover image	PSNR of watermark	Elapsed Training time	Number of epochs
1	129.2484	128.5260	0.3281	5
0.1	159.7064	158.9835	0.3750	6
0.01	190.1638	189.4412	0.4063	7
0.001	190.1641	189.4414	0.3906	7
0.0001	220.6211	219.8938	0.3594	8
0.00001	251.0791	250.3562	0.4063	9

TABLE II
(DCT-Encoded image as cover image)

Threshold	PSNR of cover image	PSNR of watermark	Elapsed Training time	Number of epochs
1	54.5174	94.8227	0.3285	5
0.1	54.5174	94.8246	0.3438	6
0.01	54.5174	94.8246	0.3594	7
0.001	54.5174	94.8248	0.3906	7
0.0001	54.5174	94.8246	0.3994	8
0.00001	54.5174	94.8246	0.3906	9

It is seen that by taking the DCT encoded cover image, the PSNR remains unaltered by the variation of threshold value. This is principally because of two reasons.A very fast convergence may result in directly jumping to the adjusted weights corresponding to the smallest threshold bypassing all other values of threshold.

As PSNR of the watermarked image has been taken with non encoded image, the error between the DCT encoded and the non encoded image is showing a constant difference resulting in the shown PSNR ratio. Thus, it is seen that a remarkable reduction in PSNR values is seen by introducing even a small matrix of message bits in the DCT midband. The consequence of hiding the entire watermark image of Lena of size 111*111 inside this cover image by encoding in the DCT coefficients may deteriorate the cover image very badly and imperceptibility feature shall be seriously affected . However, with the introduction of the proposed technique of FCNN in [1], this large image is also successfully embedded in the synapses of the neural network without much degradation in PSNR values and the suggestions in the current paper have removed the problems of 'proprietory neural net' and 'sure win' to make this scheme workable for watermarking applications though at the cost of a comparatively lower PSNR . The combination of using DCT encoding and neural network demonstrates that a much larger watermark image with little deterioration in quality and still preserving the real sense of watermarking may be obtained which may be very much useful. A much larger watermark may be embedded with this scheme as compared to the earlier techniques using techniques other than neural networks.

Experiment No.2

To demonstrate and eliminate the problem of 'Proprietory neural net', two different FCNN were trained using different set of weights to generate different watermark images of Lena(lena.jpg) (Fig.7) and a person (imag.jpg)(Fig. 8) using the same cover image of 'disc' (Fig. 6). This raises ownership issues.

However, the same message bit pattern is obtained from the cover image in both the cases, verifying the claim of the person who has successfully derived the message bit pattern from the cover image in both the cases. An unauthorized person can train a proprietary neural network to derive a watermark of his choice. However, it is not possible for that person to derive the same message bit pattern from both the images using the DCT coefficients.

Thus, multiple claims on the same digital image can be discarded easily. This provides solution to an important problem of 'ownership' and makes it possible to use the technique of FCNN for digital watermarking.

Fig. 6 shows the cover image and Fig. 7 and Fig. 8 show two different watermarks generated from the same cover image describing the problem. However, the correct owner derives the same message bit pattern from both the images successfully.

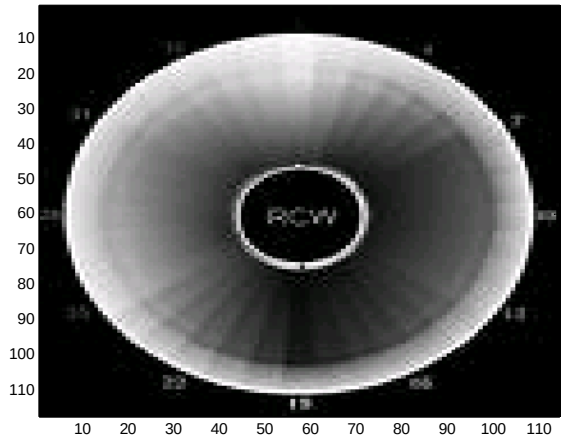

Fig. 6 Cover Image(disc)

Fig. 7 First watermark(Lena)

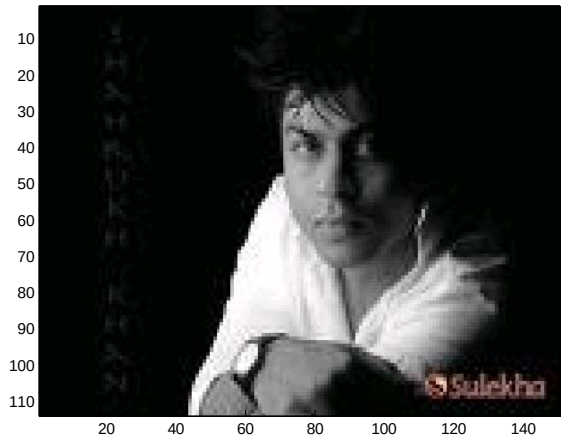

Fig. 8 Second Watermark(Person)

(Message bits decoded from cover image)

(Case I)

0	1	0	1	0	1	1	1
0	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1

(Case II)

0	1	0	1	0	1	1	1
0	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1

The message bits derived above are almost same as message bits shown in equation (31).

Experiment No.3

To demonstrate and eliminate the problem of ‘Sure win’, same watermark image of ‘Lena’,(Fig. 11) was obtained using different cover images of ‘ DCT encoded disc (first)(Fig.9) and a person’s image (Fig.10). This raises authenticity issues where two different cover images may extract the same watermark. This is due to the same winning neuron during competition learning in both the cases.

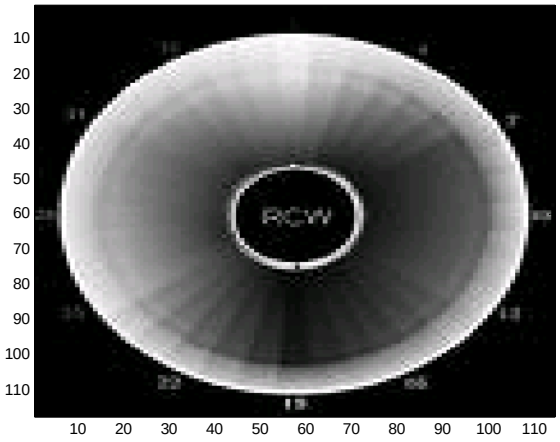

Fig. 9 First Cover Image(Disc)

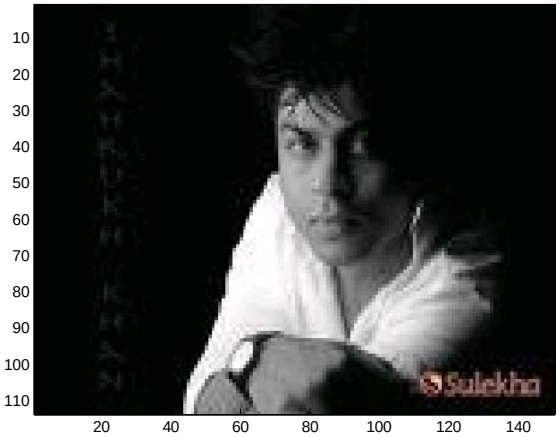

Fig. 10 Second Cover Image(Person)

Fig. 11 Output Watermark(Lena)

However, when the proposed scheme was applied to extract the message bits from both the cover images, it successfully derived almost correct message bits as per equation No. (31) from the 'disc' image (Fig. 9)and derived [111..], a pattern of all 1..s from the person's image (Fig.10), indicating that the second image is not authentic. Thus the second problem of 'sure win' is also eliminated.

(Extracted bits from the Disc cover image of Fig. 9)

0	1	0	1	0	1	1	1
0	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1

(This is almost same as the message bits as per equation(31)).(Incorrect Extracted bits from the person's cover image Fig. 10)

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

VI. CONCLUSIONS:

In this paper, attempts have been made to remove the deficiencies in the scheme of digital watermarking using FCNN . Using encoded image instead of actual cover image has solved the problems like "proprietary neural network" and "sure win" and also helped to sustain the authenticity while preserving the other advantages of robustness, imperceptibility and high capacity of watermark. With this modification, FCNN can be practically employed to obtain a successful watermarking scheme with better time complexity, higher capacity and higher PSNR. However, trained weight matrix of the FCNN is required to extract watermark from the given image.

REFERENCES

[1] Chun-Yu-Chang,"The Application of a Full Counterpropagation Neural Network to Image Watermarking", 2005, IEEE

[2] R.G.Van Schyndel,A.Z.Tirkel and CF.Osborene, "A Digital Watermark" in Proc. IEEE International Conf. Image processing,1994,vol.2 pp 86-92.

[3] Ahmidi N. Safabaksh R. "A Novel DCT Based Approach for Secure Color Image Watermarking " in Proc. ITCC 2004 International Conference Information Technology:Coding and computing,2004,vol 2,pp 709-713.

[4] K.J.Davis and K.Najarian " Maximizing Strength of Digital Watermarks Using Neural Networks", in Proc. International Joint Conf. Neural Network ,2001,vol 4, pp. 2893-2898.

[5] Zhang Zhi Ming,Li Rong-Yan,Wang Lei,"Adaptive Watermark Scheme with RBF Neural Networks, in Proc. 2003 International Conf. Neural Networks and Signal Processing,2003,vol 2. pp.1517-1520.

[6] Ren –Junn Hwand,Chuan-Ho Kao and Rong-Chi Chang, "Watermark in Color Image" in Proc. First International symposium on cyber worlds,2002 ,pp 225-229.

[7] Fengsen Deng and Bingxi Wang,"A Novel Technique for Robust Image Watermarking in the DCT Domain" in Proc. Of the 2003 International Conf. Neural Networks and Signal Processing,2003,vol.2,pp.1525-1528.

[8] Fredric M.Ham and Ivica Kostanic, "Principles of Neurocomputing for Science & Engineering", Mc.GrawHill, Singapore,2001, pp,136-140.

[9] J.Cox,J.Kilian , "A Secure Robust Watermark for Multimedia" in Proc. First International Workshop, vol 1174 of Lecture notes in computer science ,pp. 185-206.

[10] R.Schyndel, A.Tirkel, and C.Osborne, "A Digital Watermark" in Proc.IEEE Int. Conf. on Image Processing,Nov. 1994 ,Vol II,pp.86-90.

[11] F.Bartolini,M.Barni,V.Cappellini ad A.Piva, "Mask Building for Perceptually Hiding Frequency Embedded Watermarks", in Proc. Int.Conference on Image Processing ,Oct. 1998,vol. I,pp. 450-454.

[12] D.Kundur and D. Hatzinakos, " A Robust Digital Image Watermarking Method using Wavelet – Based Fusion",in Proc,IEEE Int. Conf. on Image Processing , Oct. 1997, vol. I, pp. 544-547.

[13] J.Delaigle,C.De Vleeschouwer, and B. Macq, "Psychovisual Approach to Digital Picture Watermarking", Journal of Electronic Imaging,vol.7,No.3,pp.628-640,July 1998.

[14] I.Pitas , "A Method for Signature Casting on Digital Images",in Proc,IEEE Int. Conf. on Image Processing ,Sept 1996,vol.III,pp.215-218.

[15] I.Cox,J Kilan, " Secure Spread Spectrum Watermarking for Images,Audio and Video" , in Proc. IEEE International Conference on Image Processing ,1996,vol 3,pp. 243-246.

[16] S.Craver ,N. Memon , "Resolving Rightful Ownership with Invisible Watermarking Techniques:Limitations,Attacks and Implications",IEEE Trans.,Vol 16,No. 4,pp. 573-586,1998.