A Robust Intelligent Algorithm to Detect Counterfeiting using Transform Domain

Yue Zhang, Jingbing Li*

College of Information Science and Technology, Hainan University, Haikou, China

Abstract — As business competition grows, fake and low quality products are proliferating. Such products are a serious threat to consumers as well as the business community. In order to avoid such risks of low quality products, an anti-counterfeiting technology is emerging. These are based on digital technology and various algorithms are also introduced to retain the quality products in the market. Among these counterfeiting techniques, special marks are used which can be detected with the naked eyes, and some are readable by mobile devices. The main theme proposed in this paper is to propose an authentic security algorithm based on transform domain, which is visible to the naked eye for anti-counterfeiting purposes. Experimental results show the robustness of this algorithm for common attacks, geometrical attacks and local nonlinear attacks.

Keywords - Anti-counterfeiting; Transform domain; Feature vector; Automatic identification; Robustness.

I. INTRODUCTION

The area of anti-counterfeiting is becoming popular with time because of the presence of low quality products and to protect the legitimate right of the merchants as well as end users [1]. From last twenty years the anti-counterfeiting techniques have been developed and applied to protect the safety of the business community [2]. Traditional anti-counterfeiting techniques, such as laser printer based techniques are economical but the drawbacks of these approaches are not genuine and can be copied [3]. In the past years short message based anti-counterfeiting techniques has also been utilized, however the drawbacks emerged sometimes because of the network quality [4, 5]. Especially when the network becomes overloaded, these techniques couldn’t fulfill the anti-counterfeiting criteria. Recently two dimensional codes are also present and have been utilized widely because of their rapid identification [6, 7]. The problem with these techniques is every consumer can’t realize this automatic identification quickly and can also be copied. Another techniques called as RFID, has also been utilized for this purpose but this technique is not cost effective [8-10]. This article will focus on a robust, cost effective and an intelligent algorithm to main the authenticity of anti-counterfeiting. A vector based DCT coefficient signs comprising secure database features will be established. This algorithm will provide the features of easy identification, cost effective and avoid the risk of duplicity. Moreover it will provide the ways of fast detection based on software techniques.

II. THE FUNDAMENTAL THEORY

A. The discrete cosine transform (DCT)

Discrete cosine transform (DCT) is the orthogonal transformation method proposed by N.Ahmed et al in 1974. DCT transform uses the nature of the Fourier transform transforming the image into even function form, and then doing 2-D Fourier transform so that get results with cosine form. All DCT multiplications are on real numbers. When applied to an MxN size image or matrix, the 2D-Discrete Cosine Transform will compress all the energy information of the image and concentrates it in a few coefficients located in the upper left corner of the resulting real-valued MxN DCT/frequency matrix. 2-D discrete cosine transform is defined as:

\[ F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos \left( \frac{\pi (2x+1)u}{2M} \right) \cos \left( \frac{\pi (2y+1)v}{2N} \right) \]  

where M×N is anti-counterfeiting image size, \( f(x, y) \) corresponds to the value of the anti-counterfeiting image at the point \( (x, y) \) and \( F(u, v) \) is the DCT coefficient at the point \( (u, v) \) in frequency domain. The Formula shows that the sign of DCT coefficients are related to the phase of the component.

B. A method to obtain the feature vector of authentic work anti-counterfeiting image

Firstly, compute the original image using DCT. Then choose 9 low-frequency DCT coefficients (\( F(1,1), F(1,2), F(1,3), F(2,1), \ldots F(3,3) \)), which are shown in Table I. We can find the sign of DCT coefficients are almost unchanged even after attacks, shown as Table I. Therefore, we let “1” represents a positive or zero coefficient, and “0” represents a negative coefficient so that we get a sign sequence of low-frequency DCT coefficients as feature vector of authentic work anti-counterfeiting image, as shown in column “C12” in Table I. Finally, we compute the value of the normalized cross-correlation (NC) which used 32 bits sign sequence low-frequency DCT coefficients to obtain accurate
Finally, do the symbol operation on DCT coefficients to obtain the DCT coefficient matrix as:

\[ F(i,j) \] as the original authentic work anti-counterfeiting image. It is described as:

\[ F = \{ f(i,j) \mid f(i,j) \in R, 1 \leq i \leq N1, 1 \leq j \leq N2 \} \]

Where \( f(i, j) \) means the pixel gray value of the original authentic image.

### III. THE ALGORITHM

We choose an authentic work anti-counterfeiting image as the original authentic work anti-counterfeiting image. It is described as:

\[ F = \{ f(i,j) \mid f(i,j) \in R, 1 \leq i \leq N1, 1 \leq j \leq N2 \} \]

Where \( f(i, j) \) means the pixel gray value of the original authentic image.

#### A. Establish security database of authentic work anti-counterfeiting image feature vectors

**Step 1**: Acquire the feature vectors of the original authentic work anti-counterfeiting images using DCT

Firstly, the original image \( F(i,j) \) is computed using DCT to obtain the DCT coefficient matrix \( FD(i,j) \). Secondly, select the previous 4×8 coefficients to compose \( FD_{32}(i,j) \). Finally, do the symbol operation on DCT coefficients to obtain the feature vectors \( V(j) \). The procedure is described as:

\[
FD(i,j) = \text{DCT}_2(F(i,j))
\]

\[ V(j) = \text{Sign}(FD_{32}(i,j)) \]  \hspace{1cm} (2)

**Step 2**: Save the feature vectors of the original authentic work anti-counterfeiting images in database

Repeat operations as step 1, till all authentic work anti-counterfeiting images are handled. Then save these feature vectors in security database.

#### B. The automatic identification of authentic work anti-counterfeiting image

**Step 3**: Acquire the feature vector of the tested image

To acquire the feature vector \( V'(j) \) of tested image \( F'(i,j) \), do operations which is similar as step 1. The procedure is described as:

\[
FD'(i,j) = \text{DCT}_2(F'(i,j))
\]

\[
V'(j) = \text{Sign}(FD'_{32}(i,j))
\]  \hspace{1cm} (5)

**Step 4**: Calculate the Peak Signal to Noise Ratio (PSNR)

PSNR is used to assess the quality of the image. The bigger PSNR represents the better quality of the image. It is defined as:

\[
PSNR = 10 \log_{10} \left[ \frac{M \times N \times \max(I_{\alpha,j})^2}{\sum_{i,j}(I_{\alpha,j} - \Gamma_{\alpha,j})^2} \right]
\]  \hspace{1cm} (6)

**Step 5**: Calculate the normalized cross-correlation (NC) to determine whether the original image

Calculate the normalized cross-correlation (NC) using the formula as follow:

\[
NC = \frac{V(j) \times V'(j)}{V^2(j)}
\]  \hspace{1cm} (7)

### TABLE I. CHANGE OF DCT LOW-FREQUENCY COEFFICIENTS WITH RESPECT TO DIFFERENT ATTACKS

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>C11</th>
<th>C12</th>
<th>C13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image processing</td>
<td>PSNR (dB)</td>
<td>F(1,1)</td>
<td>F(1,2)</td>
<td>F(1,3)</td>
<td>F(2,1)</td>
<td>F(2,2)</td>
<td>F(2,3)</td>
<td>F(3,1)</td>
<td>F(3,2)</td>
<td>F(3,3)</td>
<td>Sequence of coefficient signs</td>
<td>NC</td>
</tr>
<tr>
<td>Original image</td>
<td>86.48</td>
<td>62.10</td>
<td>4.43</td>
<td>0.43</td>
<td>3.60</td>
<td>0.04</td>
<td>0.12</td>
<td>-13.36</td>
<td>-3.01</td>
<td>-2.91</td>
<td>1111110000</td>
<td>1.00</td>
</tr>
<tr>
<td>Rotation (2°)</td>
<td>14.96</td>
<td>58.66</td>
<td>3.51</td>
<td>-1.96</td>
<td>2.21</td>
<td>10.82</td>
<td>0.19</td>
<td>-15.19</td>
<td>-3.40</td>
<td>-3.50</td>
<td>1101110000</td>
<td>0.94</td>
</tr>
<tr>
<td>Scaling (+2)</td>
<td>12.42</td>
<td>0.89</td>
<td>0.09</td>
<td>0.43</td>
<td>0.72</td>
<td>0.01</td>
<td>-2.67</td>
<td>-0.60</td>
<td>-0.58</td>
<td>1111110000</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>Scaling (+0.5)</td>
<td>31.07</td>
<td>2.22</td>
<td>0.21</td>
<td>1.07</td>
<td>1.81</td>
<td>0.02</td>
<td>-6.67</td>
<td>-1.51</td>
<td>-1.47</td>
<td>1111110000</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>Left translation (1%)</td>
<td>14.23</td>
<td>61.90</td>
<td>5.76</td>
<td>0.35</td>
<td>2.13</td>
<td>3.65</td>
<td>0.22</td>
<td>-13.34</td>
<td>-3.29</td>
<td>-3.06</td>
<td>1111110000</td>
<td>1.00</td>
</tr>
<tr>
<td>Down translation (5%)</td>
<td>11.91</td>
<td>59.21</td>
<td>4.99</td>
<td>0.57</td>
<td>-8.93</td>
<td>1.41</td>
<td>-0.50</td>
<td>-16.44</td>
<td>-3.42</td>
<td>-1.85</td>
<td>1110100000</td>
<td>0.70</td>
</tr>
<tr>
<td>Cropping (7% from Y)</td>
<td>61.04</td>
<td>5.80</td>
<td>0.84</td>
<td>-2.12</td>
<td>1.39</td>
<td>-0.89</td>
<td>-11.51</td>
<td>-1.89</td>
<td>-2.06</td>
<td>1101001000</td>
<td>0.59</td>
<td></td>
</tr>
<tr>
<td>Cropping (7% from X)</td>
<td>61.15</td>
<td>1.63</td>
<td>2.82</td>
<td>2.34</td>
<td>3.50</td>
<td>-0.52</td>
<td>-13.07</td>
<td>-2.98</td>
<td>-2.71</td>
<td>1111100000</td>
<td>0.66</td>
<td></td>
</tr>
<tr>
<td>Gaussian noise(3%)</td>
<td>12.88</td>
<td>70.94</td>
<td>3.42</td>
<td>0.56</td>
<td>1.89</td>
<td>2.93</td>
<td>0.04</td>
<td>-12.61</td>
<td>-2.97</td>
<td>-2.59</td>
<td>1111110000</td>
<td>0.82</td>
</tr>
<tr>
<td>Gaussian noise(5%)</td>
<td>10.75</td>
<td>74.62</td>
<td>4.05</td>
<td>1.30</td>
<td>1.32</td>
<td>2.97</td>
<td>1.01</td>
<td>-11.77</td>
<td>-2.29</td>
<td>-1.85</td>
<td>1111110000</td>
<td>0.75</td>
</tr>
<tr>
<td>JPEG compression (5%)</td>
<td>20.60</td>
<td>62.77</td>
<td>4.34</td>
<td>0.03</td>
<td>2.35</td>
<td>3.61</td>
<td>0.41</td>
<td>-12.89</td>
<td>-2.38</td>
<td>-3.58</td>
<td>1111110000</td>
<td>0.75</td>
</tr>
<tr>
<td>Median filter [3x3]</td>
<td>23.14</td>
<td>61.22</td>
<td>4.46</td>
<td>0.15</td>
<td>2.25</td>
<td>3.46</td>
<td>-0.02</td>
<td>-14.41</td>
<td>-3.29</td>
<td>-3.14</td>
<td>1111100000</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Unit of transform coefficient: 1.0e+002, NC used 32 bits sign sequence low-frequency DCT coefficients to obtain.
The larger the value of NC is, the more approximation between the tested authentic work anti-counterfeiting image $F'(i, j)$ and the original authentic work anti-counterfeiting image $F(i, j)$.

Step 6: Return the maximum value of NC to the user’s phone

All steps above are shown in the Fig. 1.

Figure 1. Authentic work anti-counterfeiting algorithm figure

IV. EXPERIEMENTS

In our experiments, we use 1000 groups of independent binary pseudo morph sequences. Every sequence consists of 32 bits. In the experiment, the 500th group is selected at random from the 1000 groups as the embedded feature vector. The authentic work anti-counterfeiting image’s size is $220 \times 76$, the image is shown in Fig. 2(a). The original authentic work anti-counterfeiting image is denoted as $F(i, j)$, where $1 \leq i \leq 220, 1 \leq j \leq 76$. The corresponding coefficient matrix is $FD(i, j)$. We select the 32 bits sequence of coefficient signs as the feature vector $V(j)$ selected from $FD(i, j)$, where $1 \leq i \leq 4, 1 \leq j \leq 8$. Security database is comprised of 32 bits feature vectors so that database can save more space than original images. By this way, we improve the rate of automatic identification.

In this simulation, PSNR is used for objectively evaluating the quality of the tested image, and the NC is used to objectively evaluate the results of similarity detection.

Fig 2(b) is the NC values between the 1000 available pseudo morph sequences and the extracted feature vector, which is achieved by using DCT and symbolic operation. It can be seen from Fig. 2(a) that the original authentic work anti-counterfeiting image. The similarity can be detected clearly, NC=1.0.

A. Common attacks

1) Common attacks

Table II is the experimental data of anti Gauss noise interference of anti-counterfeiting image. It can be seen that when the Gauss noise intensity of up to 13%, the PSNR of the anti-counterfeiting image down to 7.32dB. Then extract the anti-counterfeiting image, the correlation coefficient NC=0.76. The anti-counterfeiting image can still be accurate to extract. This shows that the algorithm has strong anti Gauss noise ability. Fig. 3(a) shows the authentic work anti-counterfeiting image under Gaussian attacks (2%). PSNR=14.35dB. The similarity can be detected with NC=0.94, as shown in Fig. 3(b).

<table>
<thead>
<tr>
<th>Noise Intensity (%)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR(dB)</td>
<td>17.12</td>
<td>14.35</td>
<td>12.88</td>
<td>10.75</td>
<td>8.18</td>
<td>7.32</td>
</tr>
<tr>
<td>NC</td>
<td>0.88</td>
<td>0.94</td>
<td>0.82</td>
<td>0.75</td>
<td>0.65</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Figure 3. Gaussian noise (noise density=2%): (a) Adding Gaussian noise; (b) Similarity detector

2) JPEG attacks.

Table III is anti JPEG compression experimental data for anti-counterfeiting image. When the compression quality is only 2%, we still can extract the anti-counterfeiting image, NC=0.75. The results show that the algorithm is robust to JPEG attacks. The authentic work anti-counterfeiting image with JPEG attacks (3%) is shown in Fig. 4(a). PSNR=17.89dB. The similarity can still be detected, NC=0.75, as shown in Fig. 4(b).

<table>
<thead>
<tr>
<th>JPEG Compression (%)</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR(dB)</td>
<td>17.12</td>
<td>20.60</td>
<td>24.01</td>
<td>27.24</td>
<td>29.11</td>
<td>30.33</td>
</tr>
<tr>
<td>NC</td>
<td>0.75</td>
<td>0.75</td>
<td>0.93</td>
<td>1.0</td>
<td>1.0</td>
<td>0.82</td>
</tr>
</tbody>
</table>
B. Geometrical attacks

1) Rotation attacks.

Table IV is anti rotation attacks experimental data for anti-counterfeiting image. We can see when the anti-counterfeiting image clockwise rotation $10^\circ$, NC=0.69, the anti-counterfeiting image can still be accurate to extract. Fig.5(a) shows the authentic work anti-counterfeiting image rotated clockwise by $2^\circ$, PSNR =14.96dB. Fig.5(b) shows that the similarity can be detected with NC=0.94. Therefore we can conclude that our scheme is robust against rotation attacks.

<table>
<thead>
<tr>
<th>Rotational Degree</th>
<th>0°</th>
<th>1°</th>
<th>2°</th>
<th>3°</th>
<th>4°</th>
<th>5°</th>
<th>6°</th>
<th>7°</th>
<th>10°</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR(dB)</td>
<td>86.48</td>
<td>19.67</td>
<td>14.96</td>
<td>12.97</td>
<td>12.05</td>
<td>11.60</td>
<td>11.31</td>
<td>11.06</td>
<td>10.51</td>
</tr>
<tr>
<td>NC</td>
<td>1.0</td>
<td>1.0</td>
<td>0.94</td>
<td>0.75</td>
<td>0.67</td>
<td>0.55</td>
<td>0.68</td>
<td>0.76</td>
<td>0.69</td>
</tr>
</tbody>
</table>

2) Scaling attacks.

Table V is anti scaling attacks experimental data for anti-counterfeiting image. We can see when the anti-counterfeiting image scaling factors low to 0.2, NC=0.94, we still can extract the anti-counterfeiting image. Fig.6(a) shows the authentic work anti-counterfeiting image shrunk with a scale factor of 2, PSNR =14.96dB. Fig.5(b) shows that the similarity can still be detected, with NC=0.88. Therefore we can conclude that our scheme is robust against scaling attacks.

<table>
<thead>
<tr>
<th>Scaling Factors</th>
<th>0.2</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>1.1</th>
<th>1.5</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>0.94</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

3) Translation attacks.

Table VI is anti left translation attacks experimental data for anti-counterfeiting image. We can see when the anti-counterfeiting image left translation 7%, NC=0.57, the anti-counterfeiting image can still be accurate to extract. Fig.7(a) shows the authentic work anti-counterfeiting image left translation 3%, PSNR =10.59dB. Fig.5(b) shows that the similarity can be detected, with NC=0.82. Therefore we can conclude that our scheme is robust against translation attacks.

<table>
<thead>
<tr>
<th>Left Translation (%)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR(dB)</td>
<td>14.23</td>
<td>11.82</td>
<td>10.59</td>
<td>9.72</td>
<td>9.57</td>
<td>9.46</td>
<td>9.36</td>
</tr>
<tr>
<td>NC</td>
<td>1</td>
<td>0.94</td>
<td>0.92</td>
<td>0.69</td>
<td>0.69</td>
<td>0.57</td>
<td>0.57</td>
</tr>
</tbody>
</table>

C. Local nonlinear attacks

1) Pinch distortion attacks.

Table VII is the anti pinch distortion attacks test data for anti-counterfeiting image. The distortion parameter is the distortion factor, the larger distortion factor, the higher frequency of the distortion. When the distortion factor is 70, the anti-counterfeiting image PSNR is low to 11.31dB, but NC=0.70, we still can extract the anti-counterfeiting image. Fig.8(a) shows the authentic work anti-counterfeiting image under the pinch distortion (40%) attacks, PSNR= 12.08dB. Fig.8(b) shows that the similarity can still be detected, with NC=0.88. The results show that the scheme is robust against pinch distortion attacks.


<table>
<thead>
<tr>
<th>Distortion Factor (%)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR(dB)</td>
<td>18.03</td>
<td>14.14</td>
<td>12.72</td>
<td>12.08</td>
<td>11.75</td>
<td>11.50</td>
<td>11.31</td>
</tr>
<tr>
<td>NC</td>
<td>0.94</td>
<td>0.94</td>
<td>0.88</td>
<td>0.88</td>
<td>0.82</td>
<td>0.82</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Figure 8. Pinch distortion attacks (40%): (a) under pinch distortion attack; (b) Similarity detector

V. CONCLUSION

A robust intelligent algorithm for authentic work anti-counterfeiting Based on DCT is proposed in this paper. It is discernible to the anti-counterfeiting mark using the naked eye. The experimental results show this algorithm can realize automatic identification and have strong robustness by combining DCT, the feature vector, and data base. In addition, the algorithm can improve the rate of identification and save a lot of storage space.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (No. 61263033), International Science and Technology Cooperation Project of Hainan (NO. KJHZ201504) and the Institutions of Higher Learning Scientific Research Special Project of Hainan (NO. Hnkyzx2014-2).

REFERENCES