International Journal of Computer Applications (0975 – 8887)
Volume 183 – No. 31, October 2021

Efficient Integer DCT based Image Watermarking Schemes in Smart Phones

Chetan K.R.
Associate Professor
Dept. of CSE, JNN College of Engg.,
Shimoga, India

ABSTRACT

Recent trends have dramatically changed the way one uses their mobile phones. What was once simply a cell phone has become a multi-purpose mobile computing device. Increased power and computing capabilities coupled with decreased size have made it possible to add functionality and combine previously separate operations into a single unit. Many mobile phones are equipped with high resolution digital cameras. Their users are capturing images and sharing them with others by sending them as email attachments, multimedia messages or via Bluetooth. Digital watermarking is adding an ownership information in multimedia contents to prove the authenticity.

This paper proposes an image watermarking scheme for ensuring authenticity of captured images. IMEI number is used as watermark. This work also aims at studying the robustness of the system under various kinds of attacks such additive noise, cropping, scaling attack. The proposed system has been tested on Android based Smart Phones. The results confirm the effectiveness of the scheme in providing efficient attack identification.

General Terms

Image Watermarking, Smart Phones, Robust watermarking, Copyright protection.

Keywords

Integer DCT, Watermark Generation, Bit shuffling, PSNR

1. INTRODUCTION

Images make up a major component of multimedia content. Examples of images are digital arts, illustrative diagrams, cultural heritage paintings in digitized form and digital photographs. When first multimedia messages appeared, they had very poor visual quality of images. Due to hardware limitations at that time, mobile devices had low megapixel cameras with simple optics. Typical image resolutions were 128x128 pixels with the maximum resolution of 640x480 pixels. In additional to camera limitations, displays were small and had low resolutions as well. This led to poor imaging experience on mobile device itself. When images taken by mobile phone were viewed on a PC screen, users were not always satisfied neither.

Modern mobile devices have overcome these difficulties with mobile imaging. They offer good experience of point and shoot use cases for typical users. Often cameras have a high number of megapixel, with good quality lens and xenon flash. It is clear that quality of mobile images is approaching very fast towards digital still cameras. Mobile displays are getting better as well, though they do not catch up with megapixel race comparing to desktop environment.

Advancement in mobile cameras has created threats to copyright protection and content integrity. For instance, images can be copied, modified, and distributed easily. Digital watermarking is a potentially good tool in enabling content protection. Watermarking in the contexts of image, audio or video data is well-known to be an effective technique to protect the intellectual property of electronic content. Digital watermarking provides the security knowing that no matter how or where images appear they carry the notice of ownership. Essentially, the technique embeds a secret message into a cover message within the content in order to prove the ownership of materials.

The rest of the paper is organized as follows. An exhaustive literature survey on watermarking on mobile phones is discussed in Section-2. Section-3 provides details of Design and Implementation of the proposed DCT based Image watermarking schemes on mobile phones. Results and analysis of the implemented system is discussed in Section-4. The paper concludes in Section-5 with directions for future work.

2. LITERATURE SURVEY

Mobile users are capturing images and sharing them with others by sending them as email attachments, MMS, via Bluetooth. The need for protecting copyright ownership of images has increased in the recent years. A survey of the work done on the protection of images through watermarking has been discussed in the following.

“Several watermarking schemes have been proposed, but digital watermarking schemes on mobile devices are scarce. In addition, watermark insertion is needed in devices with various features such as digital cameras, PDAs and mobile phones as multimedia services on those devices are heavily used [1-4]”.

In [5], an efficient watermarking algorithm that can be applied to mobile phones is proposed. In this a method for detecting the cause of destroying or changing hidden data in an MMS (Multimedia Messaging Service) message is proposed. MMS is a technology that allows mobile phones sending messages that includes multimedia objects esp. image. For copyright protection of images which are sent through MMS, one can use watermarking methods. But there is possibility of damaging or changing the hidden information while MMS is transferred.

For this purpose, in addition to the image, the copyright notes are also hidden in an audio file and sent along with the image. So, while extracting the hidden information from MMS message, if copyright notes which are extracted from both image and audio files are different, it can be concluded that the hidden data are damaged or changed. For hiding the data in image part of MMS message, a method called

"Steganography on mobile phone" is used [6,7]. For hiding data in audio part of MMS message method "Adaptive wavelet domain audio Steganography with high capacity and low error rate," is used.

In this paper [8], an improved method for hiding data into images for mobile phones is proposed. This method is used for secure data transfer from a computer to mobile phones. Information is hidden in an LSB of pixel colors. PNG image format had been used and message is hidden using a password.

The paper explored in [9] presents a blind image watermarking scheme for camera phone. This method is based on singular value decomposition (SVD) and wavelet decomposition. SVD is a numerical technique used to diagonalize matrices in numerical analysis. The singular values (SVs) of an image have very good stability. SVs represent intrinsic algebraic image properties. This method is mainly focused to JPEG compression as phone camera's output is JPEG compressed image because the output image captured by phone camera is compressed by JPEG.

People capture images on their multimedia phones but there is no security of ownership. This paper addresses the problem of providing copyright protection. It avoids illegal duplication of the images captured in mobile phones. It also resists various attacks like changing intensity, addition of noise and cropping attack. Following are the various objectives of our work:

- To generate algorithmically a watermarked image that is both imperceptible to the human eye and robust against attacks.
- To be compatible with all java supporting mobile phones.
- Instantly saving watermarked image into the gallery after its captured.
- To propose a robust algorithm which resists various attacks like addition of Noise, changing intensity, cropping attack.
- To apply watermark for already stored images on mobile phones.
- To embed multiple watermarks so as to make it robust against many attacks.

3. DESIGN AND IMPLEMENTATION

This paper discusses the design issues and overall system architecture in this section. A modular design of the system is also laid out.

3.1 Design Principles

The following are the issues involved in the design of image watermarking schema:

1. Int DCT

The integer Discrete Cosine Transform (IntDCT) is an integer approximation of the discrete cosine transform. Two significant advances of integer transforms are: no rounding of errors during computations and faster computations. Integer transforms are widely used for lossless transform coding. It avoids mismatch between forward and inverse transform. It also provides good energy compaction capability.

2. Dct coefficient Selection (DCS):

This DCS process is used to increase the invisibility qualities and also to find the coefficient with maximum magnitude excluding DC. As DCT packs energy in the low frequency regions, the 16 lower frequencies are screened to find the highest magnitude. This range of frequency is chosen because the high frequency components can be discarded in some image processing operation such as JPEG compression.

3. IMEI:

IMEI stands for International Mobile Equipment Identity and is a unique number given to every single mobile phone, typically found behind the battery. It is used as a watermark to be embedded into the host image.

4. Error Detection Coding:

Checksum error detection encoder is used in this technique where the numerical value representing the sum of the IMEI number is added. This is useful to check that the extracted number is correct or not.

5. Scrambling:

Scrambling process is used to reduce the spatial correlation between the host image and the embedded watermark. Linear congruence generation algorithm is used to do the scrambling.

6. Shuffling:

The scrambled binary IMEI number is shifted by a key factor before the embedding process to make attackers difficult to understand. This shift operation is carried out in a cyclic way.

3.2 System Architecture

Mobile phones have unique IMEI number that is embedded into the captured image using a 'key' in the embedding process. By this watermark images are generated. As these images are shared to others by sending via email attachments, MMS, Bluetooth, this may undergo some attacks. The watermark can be extracted using secret key which is only known to the intended users and not to the attackers. From this copyright ownership can be retained. The system architecture is depicted in Fig. 1.

3.3 Watermark Embedding

Watermark Embedding process is outlined in Fig. 2. The watermark i.e., IMEI number is converted to 1-D vector and the host image is divided into 'N' non-overlapping 8*8 sub blocks. The Color image is decomposed into three components, R, G and B. Inside each 8*8 blocks one int DCT coefficient is identified i.e., the optimum co-efficient. Check sum is added at the end of the IMEI number to make it 16 bit. This is converted into binary 64 digits watermark. The binary IMEI number is scrambled using a 'key' and then it is shuffled. The shuffled binary IMEI number is embedded into the optimum coefficient's DCT value. At last IDCT is applied to get the watermarked image.

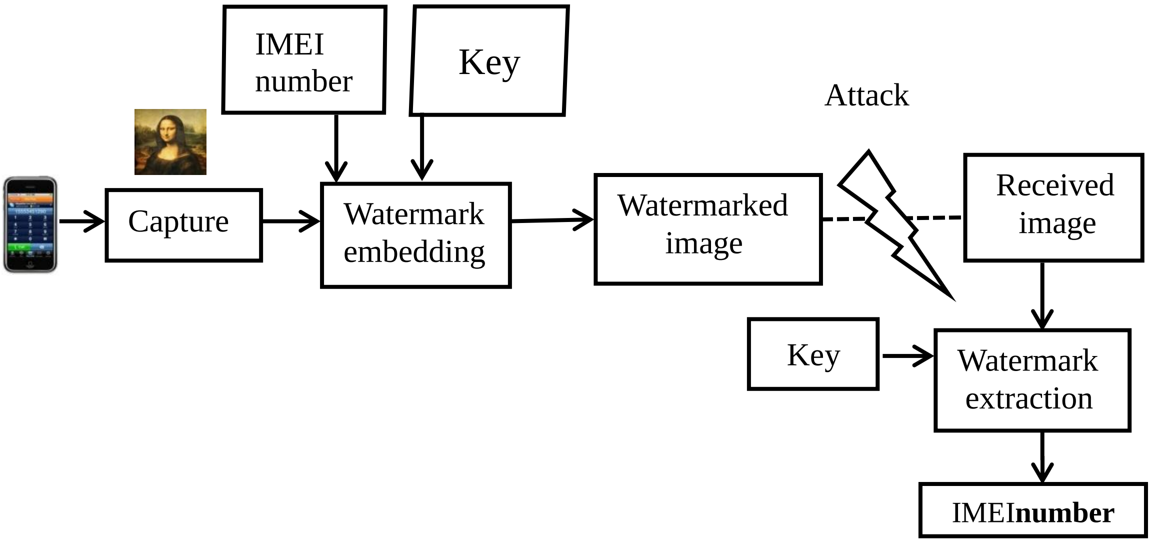


Fig. 1: System Architecture

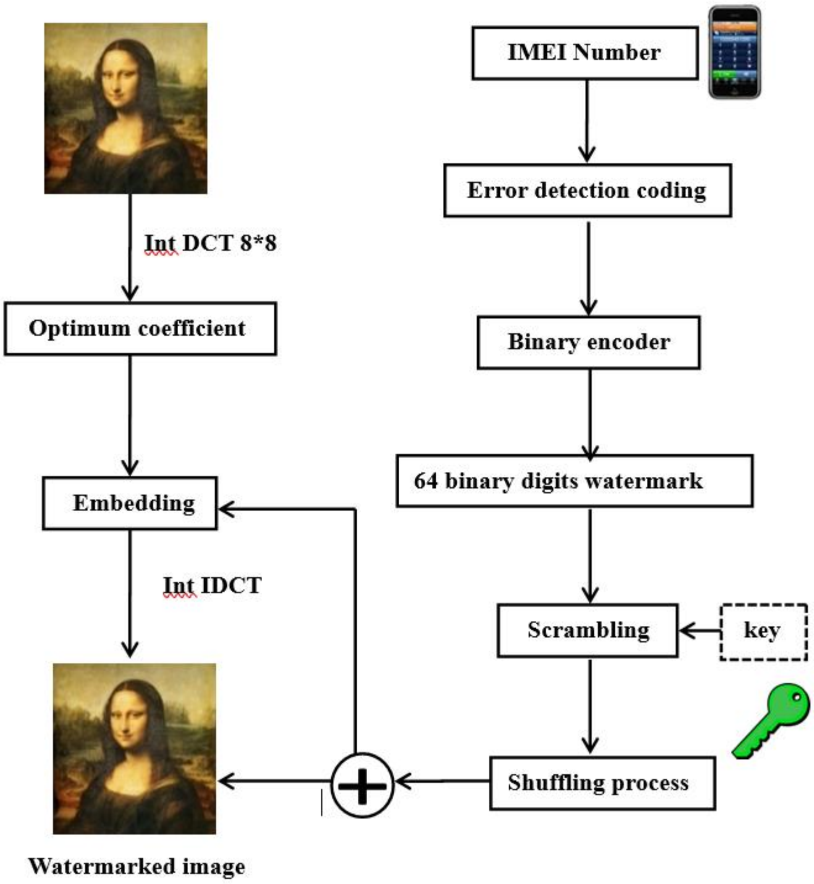


Fig. 2: Watermark Embedding

3.4 Watermark Extraction

The embedded watermarks information can be extracted by performing 8*8 DCT transform for the RGB-channel of the watermark host image and that carries the bits of the emedded watermarks using the requered secret key.The extraction formula defined in equation below is used to produce the reshuffled watermark. According to the key in the initial scrambling operation,the scrambled watermark is descrambled to retrieve the original watermark .The extracted formula is defined as follows:

$$\textit{If } Q\,(F_k\,(x, y)/\Delta) \textit{ is odd then } w\,(i, j) = 0$$
$$\textit{If } Q\,(F_k\,(x ,y)/\Delta) \textit{ is even then } w\,(i, j) =1$$

Where Q is rounded to the nearest interger.Δ has a value that is equal to the value used for the embedding process.Finally the original binary watermark is converted back to decimal to get the IMEI number. The watermark extraction mechanism is depicted in Fig. 3.

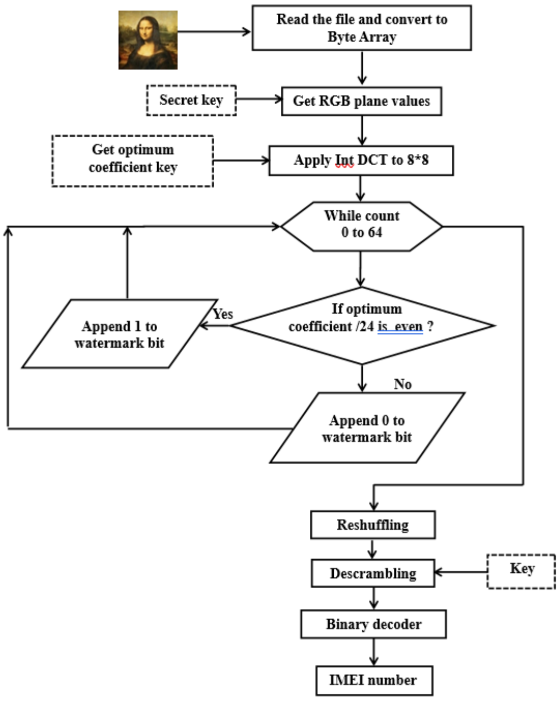

Read the file and convert to Byte Array
Secret key
Get RGB plane values
Get optimum coefficient key
Apply Int DCT to 8*8
While count 0 to 64
If optimum coefficient /24 is even ?
Yes
Append 1 to watermark bit
No
Append 0 to watermark bit
Reshuffling
Descrambling
Key
Binary decoder
IMEI number

Fig. 3: Watermark Extraction

3.5.Integer DCT Coefficient Selection (IDCS)

The integer Discrete Cosine Transform (IntDCT) is an integer approximation of the discrete cosine transform. Two significant advances of integer transforms are: no rounding of errors during computations and faster computations. Integer transforms are widely used for lossless transform coding. The forward integer transform is used as follows:

$A = 13a + 13b + 13c + 13d,$
$B = 17a + 7b - 7c - 17d,$
$C = 13a - 13b - 13c + 13d,$
$D = 7a - 17b + 17c - 7d,$

where a, b, c, and d are 4 input pixels. The inverse transformation of transform coefficients A,B,C,D into 4 pixels a',b',c',d' is defined by:

$a' = 13A + 17B + 13C + 7D,$
$b' = 13A + 7B - 13C - 17D,$
$c' = 13A - 7B - 13C + 17D,$
$d' = 13A - 17B + 13C - 7D.$

The relation between a and a' is: a' = 676a. Besides, after 2-D transform, the integer transform coefficients are about 676 times the true DCT coefficients.

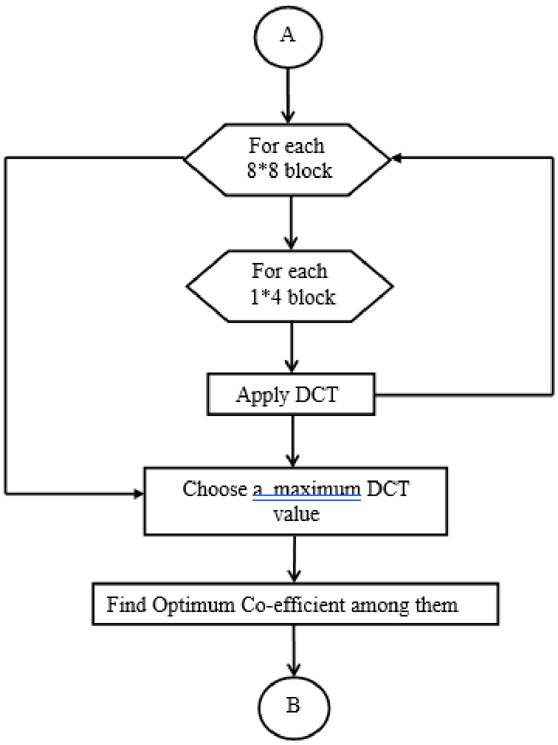

A
For each 8*8 block
For each 1*4 block
Apply DCT
Choose a maximum DCT value
Find Optimum Co-efficient among them
B

Fig. 4: DCT Coefficients selection

3.6 Generating watermark bits

IMEI is a 15 digit unique number given to every single mobile phone. One more digit is added at the end to make it 16 digits. The 16th digit is a check digit calculated using the Luhn algorithm.

The check digit is validated in three steps:

1. Starting from the right, double a digit every two digits (e.g., 7 → 14).
2. Sum the digits (e.g., 14 → 1 + 4).
3. Check if the sum is divisible by 10.

The 16 digit number is converted into 64 binary digits watermark. The binary IMEI number is randomly scrambled using a secret key. The scrambled process is carried out using Linear Congruential Generator. The linear congruential generator is a very simple example of a random number generator. All linear congruential generators use this formula:

$X_{n+1} = (aX_n + c) \pmod{m}$…(3)

Where : X_n is the sequence of pseudorandom values, and
m , $0 < m$— the "modulus"
a , $0 < a < m$ — the "multiplier"
c, $0 \leq c < m$ — the "increment"
X_0, $0 \leq X_0 < m$ — the "seed" or "start value" are integer constants that specify the generator.

After the scrambling process the binary IMEI number is shifted by a key factor before the embedding process. The shift operation is carried out in a cyclic way.

3.7 Scrambling and Shuffling process

Pseudocodes for scrambling and shuffling processes used in watermark embedding are given in the following:

```
Procedure Scramble ( array a ) return array
```

```
       choose key1 and key2 where gcd(key1,key2)=1
       a1=1
       c1=11
for each count k< array 'a' size do
{
     key2=key1
     key1=(a1*key1+c1)%64
  for each count j< array 'ran' size do
  {
   if(compare key1 equal to any jth element of 'ran' array)
    {
     key2=key2+1;
     key1=key2;
    }
  }
 if(key1 is not equal to any element of 'ran' array)
 {
  append key1 value to the end of 'ran' array
 }
}
for each count i< array 'a' size do
{
 Swap ran[i]th element of 'a' with ran[i+1]th element
}
}
```

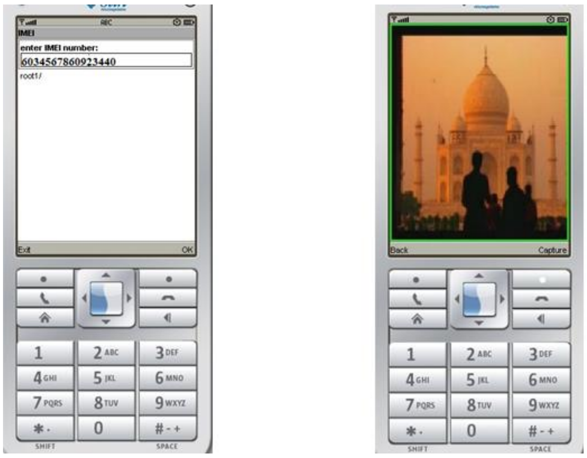
```
Procedure Shuffling ( array a, shifting amount b ) return array
       for each count i < b do
       {
          temp=a[0]
       for each count j < array 'a' size
        {
          a[i]=a[i+1];
          a[i]=temp;
        }
       }
```

4. RESULTS AND ANALYSIS

This section presents the results of the watermarking system for both instant clicking and the stored images. It also explores the effectiveness of DCT Co-efficient selection and analysis results. Different types of analysis like attack analysis, perception analysis; robustness analysis and computational complexity are evaluated and their results are presented.

4.1 Instant Embedding and Extraction Results

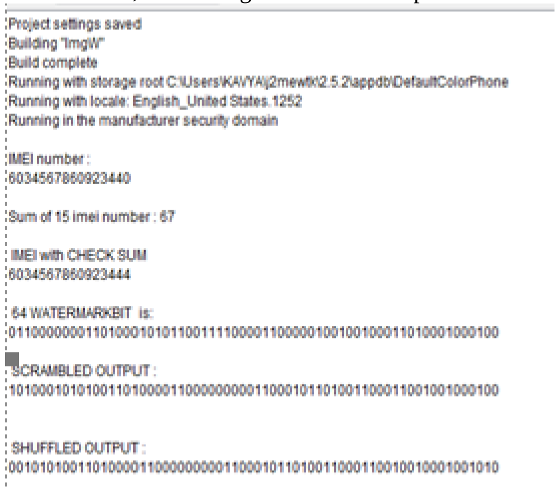
In this stage the capturing of an image and instantly embedding IMEI number is performed. Fig.5 shows a pre-screen of capture, in which takes IMEI number as input. It also shows the capturing event of the mobile phone. The IMEI number is usually unique to identify GSM, WCDMA, and iDEN mobile phones, as well as some satellite phones. It is usually found printed inside the battery compartment of the phone. It can also be displayed on the screen of the phone by entering *#06# into the keypad on most phones. The IMEI number is used by a GSM network to identify valid devices and therefore can be used for stopping a stolen phone from accessing that network. For example, if a mobile phone is stolen, the owner can call his or her network provider and instruct them to "blacklist" the phone using its IMEI number. This renders the phone useless on that network and sometimes other networks too, whether or not the phone's SIM is changed.



IMEI
enter IMEI number:
6034567860923440
root1/

Fig. 5: Capturing phase

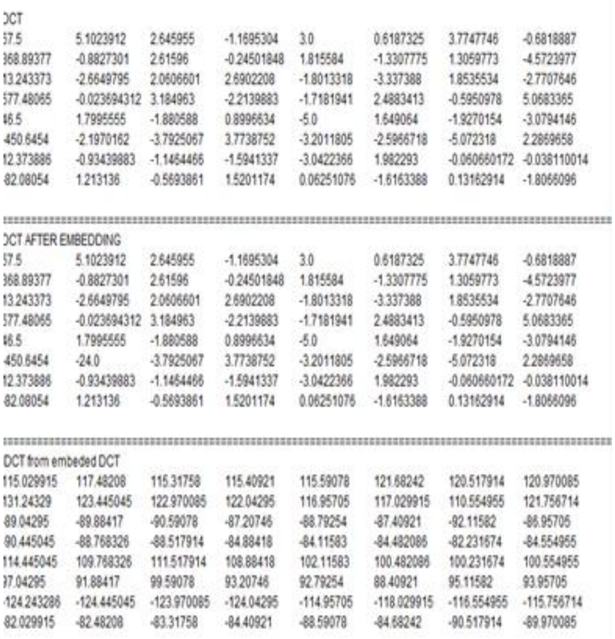
The watermark generation results are shown in Fig. 6. The different component displayed in the results are the input IMEI number, checksum added to that number, binary coding of that number, scrambling and shuffled outputs



```
Project settings saved
Building "ImgW"
Build complete
Running with storage root C:\Users\KAVYA\j2mewtk\2.5.2\appdb\DefaultColorPhone
Running with locale: English_United States.1252
Running in the manufacturer security domain

IMEI number :
6034567860923440

Sum of 15 imei number : 67

 IMEI with CHECK SUM
6034567860923444

 64 WATERMARKBIT  is:
0110000000110100010101100111100001100000100100100011010001000100

 SCRAMBLED OUTPUT :
1010001010100110100001100000000011000101101001100011001001000100


 SHUFFLED OUTPUT :
0010101001101000011000000000110001011010011000110010010001001010
```

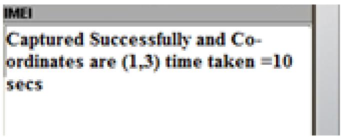
Fig. 6: Generation of watermark bits

Int DCT is applied to each block of 64X64. The pixels in the block and the corresponding Int-DCT coefficients are displayed in Fig. 7. The time taken for embedding is computed and this allows to check whether embedding can be performed in the real time. It has been observed that the time taken depends on the type of the cell, but still even with a low-end mobile, only a fraction of seconds is taken for both embedding and extraction operations. The value of the time taken is displayed in Fig. 8. In Fig. 8, the co-ordinates where watermark is embedded is also shown. This is required as the same has to be produced for valid extraction and further claiming of copyright.



```
DCT
57.5         5.1023912    2.645955     -1.1695304   3.0          0.6187325    3.7747746    -0.6818887
368.89377    -0.8827301   2.61596      -0.24501848  1.815584     -1.3307775   1.3059773    -4.5723977
13.243373    -2.6649795   2.0606601    2.6902208    -1.8013318   -3.337388    1.8535534    -2.7707646
577.48065    -0.023694312 3.184963     -2.2139883   -1.7181941   2.4883413    -0.5950978   5.0683365
46.5         1.7995555    -1.880588    0.8996634    -5.0         1.649064     -1.9270154   -3.0794146
450.6454     -2.1970162   -3.7925067   3.7738752    -3.2011805   -2.5966718   -5.072318    2.2869658
12.373886    -0.93439883  -1.1464466   -1.5941337   -3.0422366   1.982293     -0.060660172 -0.038110014
82.08054     1.213136     -0.5693861   1.5201174    0.06251076   -1.6163388   0.13162914   -1.8066096

====================================================================================================
DCT AFTER EMBEDDING
57.5         5.1023912    2.645955     -1.1695304   3.0          0.6187325    3.7747746    -0.6818887
368.89377    -0.8827301   2.61596      -0.24501848  1.815584     -1.3307775   1.3059773    -4.5723977
13.243373    -2.6649795   2.0606601    2.6902208    -1.8013318   -3.337388    1.8535534    -2.7707646
577.48065    -0.023694312 3.184963     -2.2139883   -1.7181941   2.4883413    -0.5950978   5.0683365
46.5         1.7995555    -1.880588    0.8996634    -5.0         1.649064     -1.9270154   -3.0794146
450.6454     -24.0        -3.7925067   3.7738752    -3.2011805   -2.5966718   -5.072318    2.2869658
12.373886    -0.93439883  -1.1464466   -1.5941337   -3.0422366   1.982293     -0.060660172 -0.038110014
82.08054     1.213136     -0.5693861   1.5201174    0.06251076   -1.6163388   0.13162914   -1.8066096

====================================================================================================
DCT from embeded DCT
115.029915   117.48208    115.31758    115.40921    115.59078    121.68242    120.517914   120.970085
131.24329    123.445045   122.970085   122.04295    116.95705    117.029915   110.554955   121.756714
89.04295     -89.88417    -90.59078    -87.20746    -88.79254    -87.40921    -92.11582    -86.95705
90.445045    -88.768326   -88.517914   -84.88418    -84.11583    -84.482086   -82.231674   -84.554955
114.445045   109.768326   111.517914   108.88418    102.11583    100.482086   100.231674   100.554955
97.04295     91.88417     99.59078     93.20746     92.79254     88.40921     95.11582     93.95705
124.243286   -124.445045  -123.970085  -124.04295   -114.95705   -118.029915  -116.554955  -115.756714
82.029915    -82.48208    -83.31758    -84.40921    -88.59078    -84.68242    -90.517914   -89.970085
```

Fig. 7: IntDCT Results



IMEI

Captured Successfully and Co-ordinates are (1,3) time taken =10 secs

Fig. 8: User informative display after capture

4.2 Effectiveness of DCT Co-efficient Selection

The IntDCT block consists of 8×8 coefficients. The 16 lower frequencies are screened to find the coefficient with the highest magnitude and register its location. This process is repeated for all DCT blocks. The location which is repeated more is selected. This location will vary from one image to another according to the spatial frequency contents of the image. One binary bit of the watermark will be embedded in this location. Screening the Int DCT blocks again after embedding resulted in totally different locations from the previously registered optimum locations in the original un-watermarked images. The security analysis of the watermarking scheme is analyzed in the following section.

Table 1 shows the optimum coefficient locations for original images and after embedding respectively. This verifies the method is secure and attacker would not be able to use the Int DCS process again to detect the originally selected locations.

Table 1 optimum locations

Sample Images	Optimum Location (Before)	Optimum Location (After)
Image1	(1,3)	(1,7)
Image2	(1,7)	(0,7)
Image3	(2,7)	(0,3)
Image4	(7,3)	(1,3)
Image5	(7,7)	(1,7)

4.3 Attack Analysis

The attacker can try to guess the inputs given while extraction. The attacks fall into four cases: (i) Entering wrong co-ordinates and wrong keys (ii) Entering correct co-ordinates and wrong keys (iii) Entering wrong co-ordinates and correct keys (iv) Entering correct co-ordinates and correct keys. It is observed that the system defends all such attack scenarios and thus claims to be robust. The accuracy of each cases is shown in plot of Fig. 9. It can be observed from the plot that accuracy for all the cases is well above 90%.

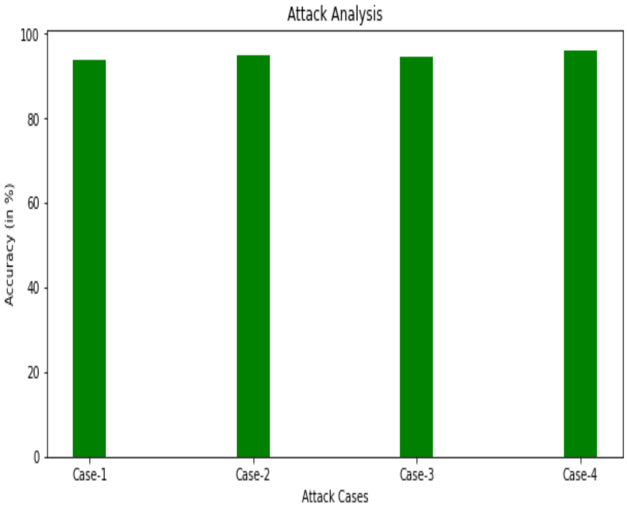


Fig. 9: Accuracy during attacks

4.4 Fidelity Analysis

Image fidelity is a characteristic of an image that measures the perceived image degradation (typically, compared to an ideal or perfect image). Imaging systems may introduce some amounts of distortion or artifacts in the signal, so the quality assessment is an important problem.

The degree of watermarking is defined as the number of watermarks redundantly embedded in the image to increase robustness. The impact of degree of watermarking on PSNR is studied. PSNR values for various degrees have been tabulated in Table 2 and corresponding plot is shown in Fig. 10. It can be observed that the plot does not change much irrespective of the change in the number of watermarks being embedded. The range of PSNR has found to be in 46-48dB which is easily acceptable.

Table 2 degree of watermark v/s PSNR

Degree of Watermark	PSNR (in DB)
3	48.57
6	46.78
8	45.98
10	45.11

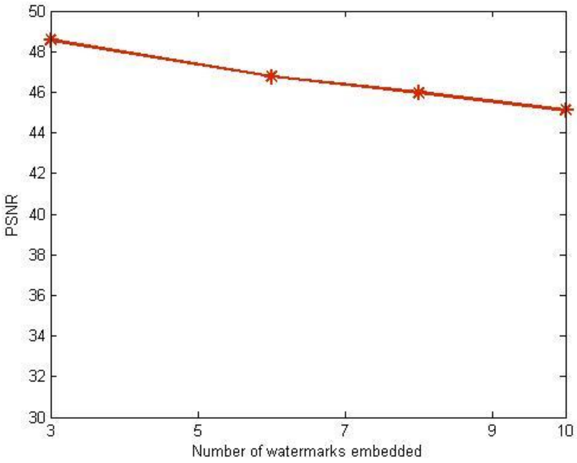


Fig. 10: Degree of watermarking vs PSNR

4.5 Robustness Analysis

The intensity of the image is checked for embedding 10 watermarks. The intensity has been gradually increased from 5% to 25% and the number of watermarks correctly recovered has been computed. Fig. 11 shows the plot of intensity variations v/s number of watermarks correctly recovered. It can be observed that the rate of recovery does not depend on the rate at which intensity is changed. It is almost above 90% independent of the percentage of intensity change performed.

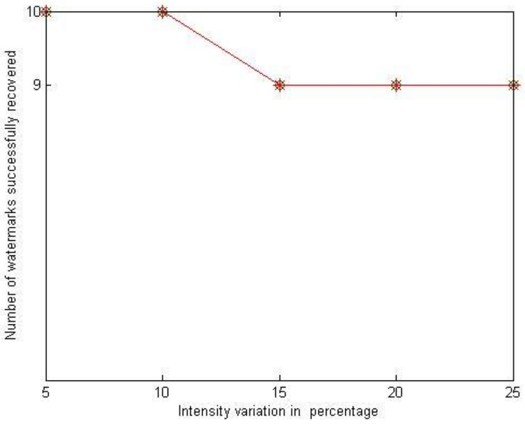


Fig. 11: Impact on robustness for change in intensity

The effect of cropping on the watermarked image has been studied. The watermarking degree is fixed to be 10. Fig. 12 shows the graph plotted for cropping percentage v/s number of watermark recovered. It can be inferred from the graph that number of watermarks are recoverable despite a large amount of crop. The cropping can be defended by embedding a greater number of watermarks and this makes the scheme robust

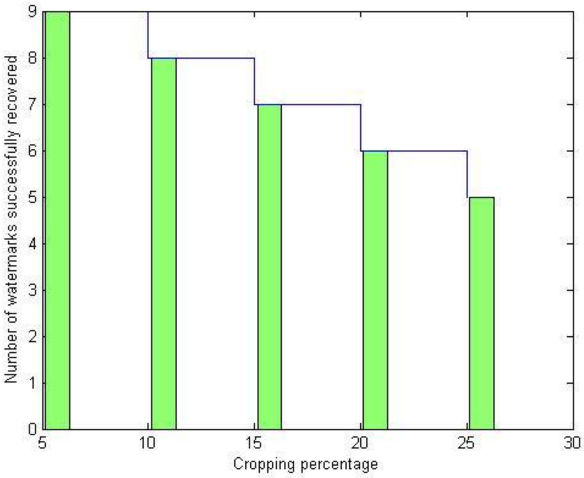


Fig. 12: Impact on robustness for change in cropping

4.6 Computation Complexity

Evaluating the computational complexity of the implemented watermarking system is of utmost importance as the embedding and extraction modules have to be run on mobile devices. Fig. 13 shows the plot of degree of watermark v/s time taken. It can be seen that time taken increases, as the number of watermarks increases. But the total time taken is not beyond 12 seconds and the time taken is acceptable. The time complexity can be reduced with high-end mobile phones having faster processors.

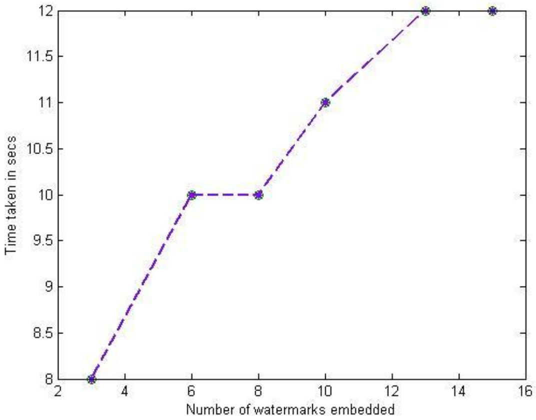


Fig. 13: Degree of watermarking vs time taken

5. CONCLUSION

A secure blind watermarking algorithm of color images using mobile IMEI number has been successfully achieved. An integer DCT coefficient selection (IDCS) process has been applied to increase the invisibility qualities; this process manages to find the coefficient with the maximum magnitude. This approach is robust against several attacks like additive noise, change in intensity, cropping etc. Instantly saving watermarked image into the gallery is also achieved. This approach even embeds watermark to the still images from gallery. User need to provide a key to extract the hidden data. A single bit of wrong prediction by attacker result in failure. From this intellectual property of the owner is retained.

The performance of the watermark system can be enhanced further by introducing a faster alternative method. The Watermarking system can be tested for other kind of attacks.

6. REFERENCES

[1] S.Katzenbeisser and F.Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking. London: Artech House,2000.

[2] A.Al-Gindy a, H.Al-Ahmad b, R.Qahwaji a and A.Tawfik c ,"A New Watermarking Scheme For Colour Images Captured By Mobile Phone Cameras", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.7, July 2009

[3] B.Verrma, S.Jain, D.P.Agarwal and A.Phadikar, "A new colour image watermarking scheme," Infocomp, Journal of Computer Science, vol.5,pp. 37-42,2006.

[4] Jie Liang, Trac Tran and Pankaj Topiwala, "A 16-bit architecture for H.26L, treating DCT transforms and quantization", in Test Model Long Term Number 5 (TML-5).Sterrett Pl., #322 Columbia, MD 21044 USA

[5] K.Yogalakshmi and 2R.Kanchana "Blind Watermarking Scheme for Digital Images", in Journal of Technology And Engineering System(IJTES),Jan –March 2011- Vol.2.No.3.

[6] Jie Liang et.al.,"A 16-bit architecture for H.26L, treating DCT transforms and quantization", in ITU - Telecommunications Standardization Sector, Thirteenth Meeting: Austin, Texas, USA, 2-4 April, 2011

[7] W. Lu, H. Lu, and F. L. Chung, "Robust digital image watermarking based on sub-sampling" in applied mathematics and computation, 2006, pp. 886-893.

[8] How to retrieve the Device Unique ID from android device http://developer.samsung.com/android/technical-docs/How-to-retrieve-the-DeviceUnique-ID-from-android-device

[9] Getting IMEI Number and other Details. http://www.learn-androideasily.com/2013/05/getting-imei-number-and-other-detail-html"