

# **SIL801 Special Topics in Multimedia System**

## **Project Report on Topic :-**

### **Multimedia Watermarking**

Anant Jain (2019MCS2557)

---

#### **Introduction :-**

Today, Multimedia data like text, image, audio , video etc are distributed or copied from one system to another. This data has great value for their owners.. They are owned by some authority or organisation. To provide copyright protection and copy prevention mechanism, Mutlimedia watermarking techniques are used.

#### **Multimedia WaterMarking :-**

The technique of multimedia watermarking can be applied to any type of data like text, image, audio and videos. There are various techniques for different type of these multimedia data.

Generally, Watermark is a data which is added in the host data (data on which watermark is applied) so that it cannot be removed. Mostly, It contain information about the status, origin and recipient of the host data. It is used for applications like copyright protection, data tracking and copy prevention mechanisms. It can help in identifying source and destination of the data.

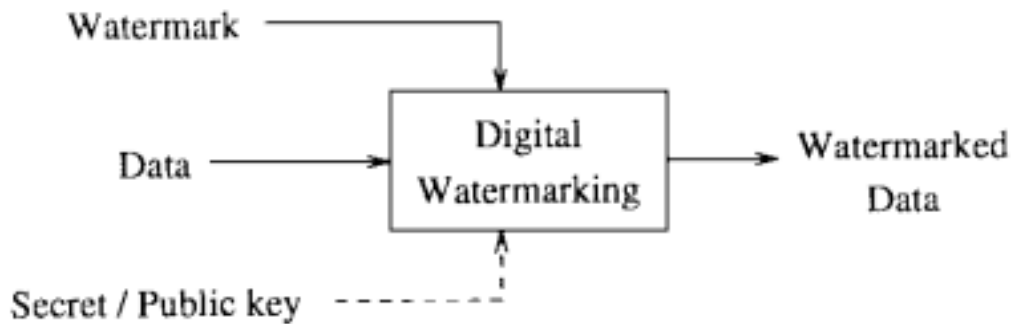
Watermarking are used according to the needs of the users. For image, Sometimes, Visible watermarks are required . And, Sometimes, Invisible watermarks are required. And Watermarking can be robust or fragile. Fragile Watermarking get effected by the modification in the data, while robust watermarking remain unaffected by these changes done by the attacker.

#### **Requirements in watermarking for all the media :-**

1. Watermark should provide all the information that it can.
2. A watermark should be secret and can be accessible by the authorised parties.
3. A watermark should remain in the data irrespective of various possible attacks. i.e. should be robust. But, In some application, this property is not required.
4. A watermark should, though being unremovable, be imperceptible.

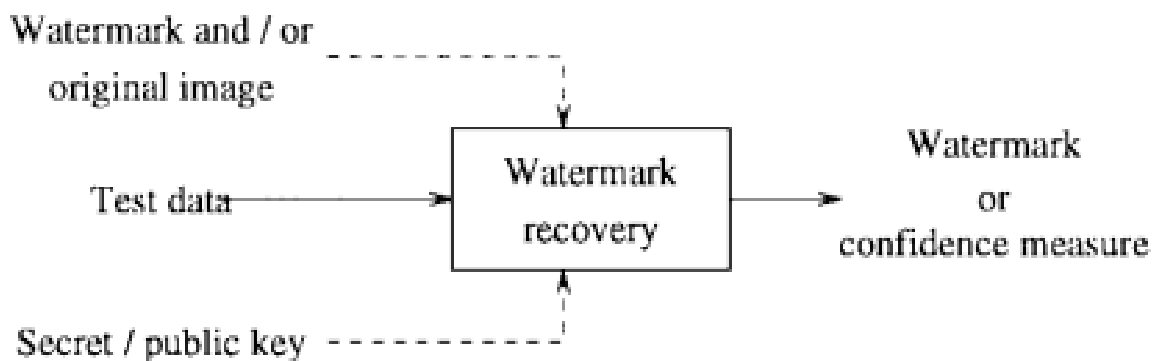
# Idea of Basic Watermarking Technique :-

## Watermarking Embedding mechanism :-



1. The input to watermarking embedding algorithm is the host data (either of any media), watermark data and some secret key/ public key.(optional in some cases).
2. The host data may depend upon the application , either compressed or uncompressed.
3. Watermark can be of any type like that of number , text data or image.
4. The secret key/ public key is used for giving security.
5. Watermarking algorithm are of various types for various types of multimedia data.

## Watermarking recovery mechanism :-



1. The input to the watermarking recovery mechanism are watermarked data, the secret or public key, and, depending on the method, the original data and the original watermark.
2. The output of this mechanism is either the recovered watermark or some kind of confidence measure indicating how given watermark is contained in the input data.

## **Design of the project :-**

1. This project will implement the watermarking algorithm for the image data.
2. In this, Images can be watermarked in the following ways :-
  - a. By visible methods :-** Watermarked data (text or image) will be embedded on the host image data.
  - b. By Invisible method :-** There will be some watermarking done on the host image which will be invisible, but can be detected through some mechanism. The invisible method is of two types:-
    - i. By fragile method :-** In this, Image will be watermarked by some fragile method. And, Watermarked image will be effected by some transformation. Due to this, Watermark in the transformed image cannot be detected.
    - ii. By Robust Method :-** In this, Image will be watermarked by some robust method. And, Watermarked image will be effected by some transformation. But, still, we are able to detect and recover the watermark in the image.
3. For all the methods, both embedding and recovery procedure will be implemented.
4. The project is implemented in the python language with the use of opencv and pillow library.

## **Implementation :-**

### **1. Visible Watermarking :-**

#### **Procedure for visible watermarking :-**

1. This method is implement by use of pillow library.
2. The input for this method is path of the main image, watermark message, path of output image and text size.
3. In the function, the watermark message is paste on given input main image, and the output image is formed to get desired result.



**Applying  
Visible watermarking**

----->

**algo, Watermark msg  
= 'hello lena'**



### **Visible watermarking method**

## **2. Fragile Invisible Watermarking method :-**

In this method, watermarking is done at the bit level. So, At any change in the watermarked image, watermark message is lost.

### **Procedure for fragile invisilbe watermarking method :-**

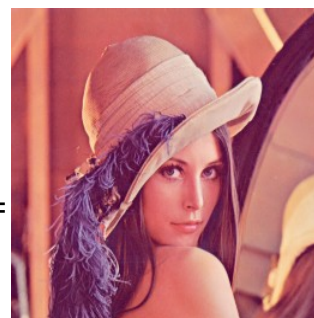
1. This input to this method are the path of input image, watermark message and path of the output image.
2. The watermark message is added with some end checker value. So, that message can be recovered at the time of recovery of watermark.
3. Now, Each character of the Message is covertred into binary form. Then, these are concatenated to form a list of 0 and 1.
4. After that , above list is inserted into the image. By this, Watermark message is put into the given image. And, Output image is formed ( containing watermark message, but it seems similiar to input image).



**Applying  
Fragile Invisible watermark**

----->

**Algorithm, Watermark msg =  
'hello lena'**



## **Fragile Invisible Watermarking method**

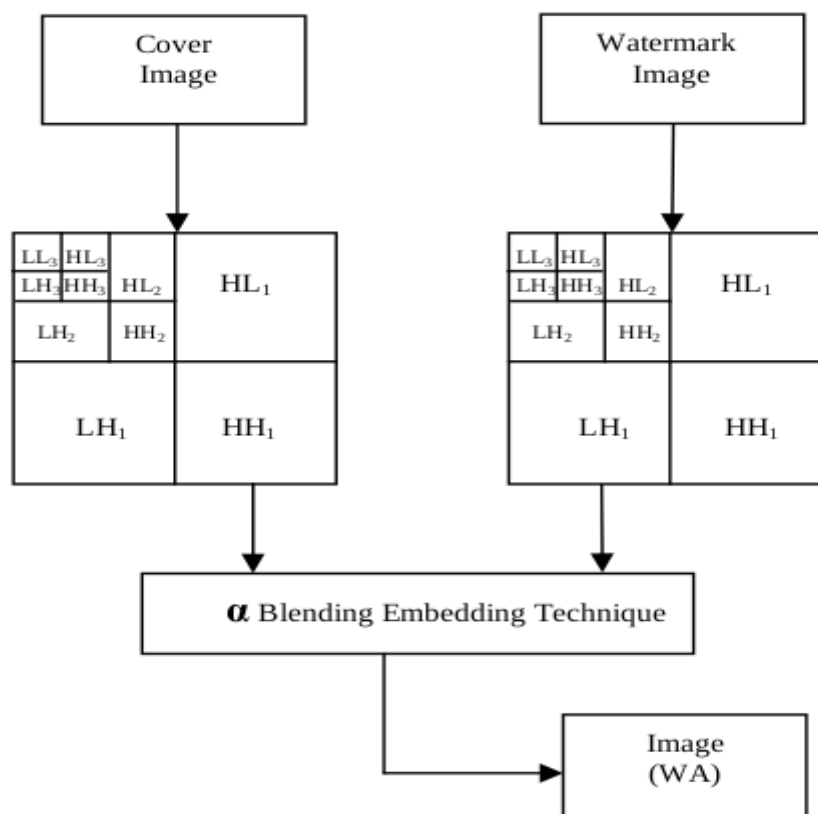
### Procedure for recovery of watermark in fragile invisilbe watermarking method :-

1. In this, the watermark message is obtained in the form of text.
2. During inserting watermark, the end checker value is added. Then, it will be used during recovery process.
3. The input to this procedure are path of watermarked image.
3. Here, We iterate from starting bit of the input image, adding one by one character to the content. If at the end of content, if we find the end checker value, then we get the required watermark message.

## **3. Robust Invisible Watermarking method :-**

In this, Watermarking is done by the 3-level discrete wavelet transform. It is a robust method.

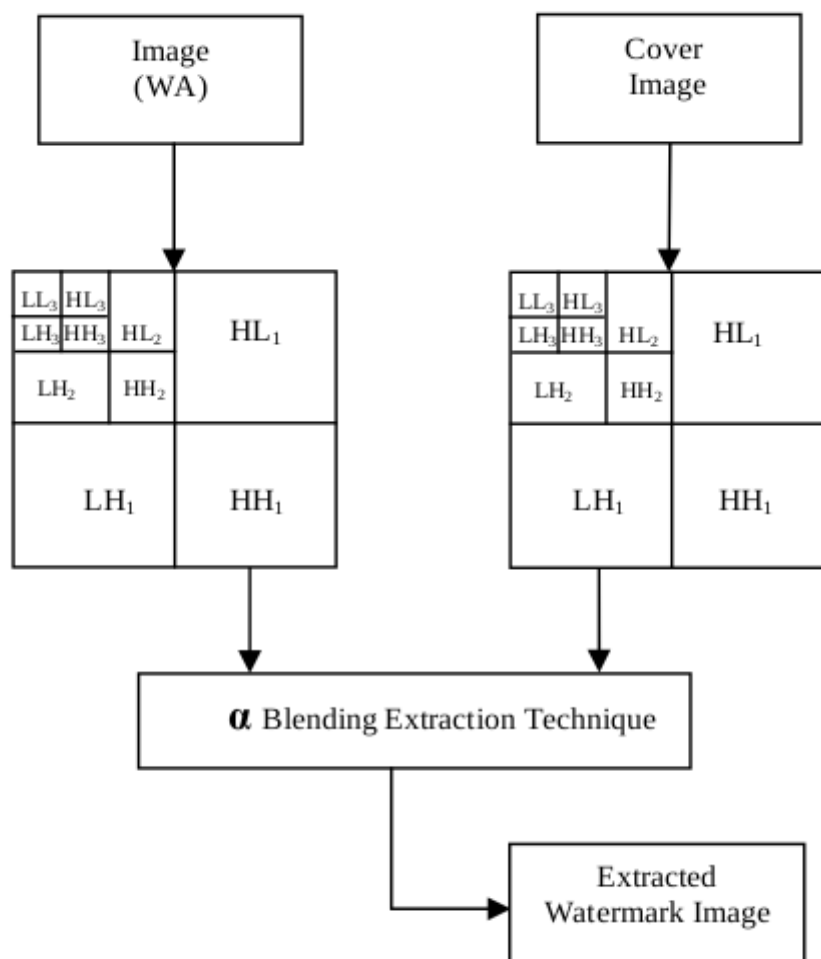
### Procedure for robust invisible watermarking method :-



## **Robust Invisible Watermarking method**

1. The input to this method are path of main input image, watermark message, path of output image, scaling factor of main image and scaling factor of watermark.
2. In this , a new image is formed containing watermark message.
3. Now, 3-level Discrete Wavelet Transform is applied on the main input image and newly formed watermark image.
4. Now, The low level component of main input image and newly formed image are multiplied by their respective scaling factor input. After that, there are added. (i.e. Alpha blending method)
5. After that, 3-level inverse discrete wavelet transform is applied on above output to get final watermarked image.

Procedure for recovery of watermark in robust invisible watermarking method :-



**Robust Invisible Watermarking recovery method**

1. The input to this method are the path of input watermarked image ,path of original image, path of output watermark image, scaling factor of origin image and scaling factor of watermark.
2. In this, Watermark message is obtained in the form of image.
3. First, 3-level Discrete Wavelet Transform is applied on the main input watermarked image and original image.
4. Now, The 3-level frequency components of the watermark image is obtained by use of the components of given watermarked and original image.
5. After that, 3-level inverse discrete wavelet transform is applied on above output to get final watermark image.
6. For the correct output, choosing proper scaling factors are very important.

## **Explanation of the Code :-**

In this project, There are two python file are implemented :-

### **1. wm\_schemes.py :-**

In this , function for visible watermarking, fragile invisible watermarking and robust invisible watermarking are implemented.

**Running command :-** `python wm_schemes.py`

After running this command, the program ask for input for further working from the user. User have to enter desired input for further functioning.

### **2. wm\_recovery.py :-**

In this , function for recovery for fragile invisible watermarking and robust invisible watermarking are implemented.

**Running command :-** `python wm_schemes.py`

After running this command, the program ask for input for further working from the user. User have to enter desired input for further functioning.

## **References :-**

1. <http://www.cse.iitd.ac.in/~pkalra/sil801/Projects/ProcIEEEHartungKutter.pdf>
2. [https://en.wikipedia.org/wiki/Digital\\_watermarking](https://en.wikipedia.org/wiki/Digital_watermarking)
3. [https://www.tutorialspoint.com/python\\_pillow/python\\_pillow\\_creating\\_a\\_watermark.htm](https://www.tutorialspoint.com/python_pillow/python_pillow_creating_a_watermark.htm)

4. <https://www.pyimagesearch.com/2016/04/25/watermarking-images-with-opencv-and-python/>
5. Image Watermarking Using 3-Level Discrete Wavelet Transform (DWT)
6. DIGITAL IMAGE OF WATERMARKING: A SURVEY Reza Aghababaeyan a\* ,  
Seyed Fariborz Seyed Asrari b\*
7. Content Based Watermarking of Images Mohan S Kankanhalli School of  
Computing National University of Singapore, Singapore
8. <https://pillow.readthedocs.io/en/4.2.x/reference/Image.html>
9. <https://pywavelets.readthedocs.io/en/latest/ref/2d-dwt-and-idwt.html>
10. <https://www.geeksforgeeks.org/opencv-python-tutorial/>