DIGITAL WATERMARKING IN IMAGE PROCESSING USING PYTHON

by

Name	Roll No.	Registration No:
SINCHITA BANERJEE	**11700314098**	**141170110280 of 2014-2018**
TRISHITA ROY	**11700314119**	**141170110301 of 2014-2018**
T. SRIYA AISHWARYA	**11700314116**	**141170110298 of 2014-2018**
PRITI SINHA	**11700314061**	**141170110243 of 2014-2018**

A comprehensive project report has been submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology
in
ELECTRONICS & COMMUNICATION ENGINEERING

Under the supervision of
Shri Monirul Purkait
(External Supervisor, VECC, Kolkata)
Dr. Soham Sarkar
Assistant /Associate / Professor

Department of Electronics & Communication Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015

CERTIFICATE OF APPROVAL

This is to certify that the project titled "**DIGITAL WATERMARKING IN IMAGE PROCESSING USING PYTHON** " carried out by

Name	Roll No.	Registration No:
SINCHITA BANERJEE	**11700314098**	**141170110280 of 2014-2018**
TRISHITA ROY	**11700314119**	**141170110301 of 2014-2018**
T. SRIYA AISHWARYA	**11700314116**	**141170110298 of 2014-2018**
PRITI SINHA	**11700314061**	**141170110243 of 2014-2018**

Optional in case of External Supervisor

.....
Shri Monirul Purkait
External Supervisor
Scientific Officer
VECC, Kolkata

.....
Dr. Soham Sarkar
Professor , Dept. of ECE
RCC Institute of Information Technology

for the partial fulfillment of the requirements for B.Tech degree in **Electronics and Communication Engineering** from **Maulana Abul Kalam Azad University of Technology, West Bengal** is absolutely based on his own work under the supervision of **Shri Monirul Purkait** . The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

.....
Dr. Abhishek Basu
Head of the Department (ECE)

RCC Institute of Information Technology

DECLARATION

"We Do hereby declare that this submission is our own work conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute and that, to the best of our knowledge and belief, it contains no material previously written by another neither person nor material (data, theoretical analysis, figures, and text) which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text."

.....

SINCHITA BANERJEE

Registration No:**141170110280** OF **2014-2018**

Roll No:**11700314098**

.....

TRISHITA ROY

Registration No:**141170110301 of 2014-2018**

Roll No: **11700314119**

.....

T SRIYA AISHWARYA

Registration No:**141170110298 of 2014-2018**

Roll No: **11700314116**

.....

PRITI SINHA

Registration No:**141170110243 of 2014-2018**

Roll No:**11700314061**

Date: 10.05.2018

Place: Kolkata

CERTIFICATE of ACCEPTANCE

This is to certify that the project titled "**DIGITAL WATERMARKING IN IMAGE PROCESSING USING PYTHON** " carried out by

Name	Roll No.	Registration No:
SINCHITA BANERJEE	**11700314098**	**141170110280 of 2014-2018**
TRISHITA ROY	**11700314119**	**141170110301 of 2014-2018**
T. SRIYA AISHWARYA	**11700314116**	**141170110298 of 2014-2018**
PRITI SINHA	**11700314061**	**141170110243 of 2014-2018**

is hereby recommended to be accepted for the partial fulfillment of the requirements for B.Tech degree in **Electronics and Communication Engineering** from **Maulana Abul Kalam Azad University of Technology, West Bengal.**

Name of the Examiners (Signature with Date)

1.……

2.……..…….

3.……...

4. ……..……..

CONTENTS

ABSTRACT

With the rapid growth and internet and networks techniques, multimedia data transforming and sharing is common to many people. Multimedia data is easily copied and modified, so necessarily for copyright protection is increasing. It is the imperceptible marking of multimedia data to “brand” ownership. Digital watermarking has been proposed as technique for copyright protection of multimedia data.

Digital watermarking invisibly embeds copyright information into multimedia data. Thus,digital watermarking has been used for copyright protection, finger protection, fingerprinting, copy protection, and broadcast monitoring. Common types of signals to watermark are images, music clips and digital video. The application of digital watermarking to still images is concentrated here. The major technical challenge is to design a highly robust digital watermarking technique, which discourages copyright infringement by making the process of watermarking removal tedious and costly.

LIST OF FIGURES

Chapter 1

Introduction

1.1. DIGITAL IMAGE PROCESSING

Digital Image Processing (DIP) deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focuses particularly on images. DIP focuses on developing a computer system that is able to perform processing on an image. The input of that system is a digital image and the system process that image using efficient algorithms, and gives an image as an output.

1.2. HOW DIP WORKS:

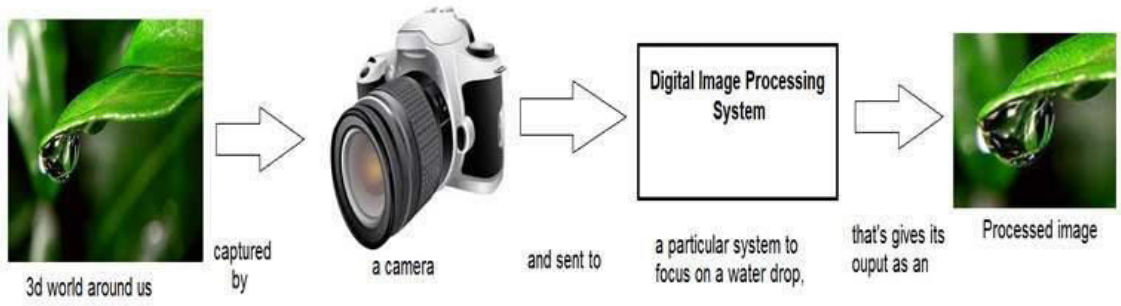

Figure 1.1

The digital image processing deals with developing a digital system that performs operations on an digital image.

1.3. WHAT IS AN IMAGE ?

An image is nothing more than a two dimensional signal. It is defined by the mathematical function f(x,y) where x and y are the two coordinates horizontally and vertically. The value of f(x,y) at any point is gives the pixel value at that point of an image.

Figure 1.2

The above figure is an example of digital image which is nothing but a two dimensional array of numbers ranging between 0 and 255.

1.4. RELATIONSHIP BETWEEN A SIGNAL AND IMAGE

Signal

In physical world, any quantity measurable through time over space or any higher dimension can be taken as a signal. A signal is a mathematical function, and it conveys some information.

A signal can be one dimensional or two dimensional or higher dimensional signal. One dimensional signal is a signal that is measured over time. The common example is a voice signal.

The two dimensional signals are those that are measured over some other physical quantities. The example of two dimensional signal is a digital image.

Relationship

Since anything that conveys information or broadcast a message in physical world between two observers is a signal. That includes speech or (human voice) or an image as a signal. Since when we speak , our voice is converted to a sound wave/signal and transformed with respect to the time to person we are speaking to. Not only this , but the way a digital camera works, as while acquiring an image from a digital camera involves transfer of a signal from one part of the system to the other.

1.5. HOW A DIGITAL IMAGE IS FORMED

Since capturing an image from a camera is a physical process. The sunlight is used as a source of energy. A sensor array is used for the acquisition of the image. So when the sunlight falls upon the object, then the amount of light reflected by that object is sensed by the sensors, and a continuous voltage signal is generated by the amount of sensed data. In order to create a digital image, we need to convert this data into a digital form. This involves sampling and quantization. The result of sampling and quantization results in an two dimensional array or matrix of numbers which are nothing but a digital image.

1.6. WHY DIGITAL IMAGE PROCESSING REQUIRED

Digital information and data are transmitted more often over the internet now than ever before. The availability and efficiency of global computer networks for the communication of digital information and data have enhanced the popularity of digital media. Hence, information security is becoming more and more important for information intercommunication and transmission among people. In order to secure information against unauthorized illegal access, diverse methods such as symmetric and asymmetric encryption systems are used .

Traditionally, protection of digital data has been provided by a variety of encryption methods. However, encryption alone does not provide an adequate solution as it only provides for robust delivery of the content. Once the content is decrypted, it is no longer protected and the content may be illegally replicated or copied without any prevention. Thus, piracy in the presence of internet and computers is a major concern.To deal with piracy and counterfeiting of the multimedia data, digital watermarking technique has an edge over the other available techniques. Thus, last decades gaining attention on watermarking schemes.

CHAPTER 2

DIGITAL WATERMARKING

2.1. HISTORY OF WATERMARKING

The term "Digital Watermark" was coined by Andrew Tirkel and Charles Osborne in December 1992.

Two basic methods of information hiding are cryptography and steganography.The term steganography means “cover writing” and cryptography means “secret writing”. Cryptography is the study of methods of sending messages in distinct form so that only the intended recipients can remove the disguise and read the message. The message we want to send is called the plain text and disguised message is called ciphertext. The process of converting a plain text to a cipher text is called enciphering or encryption, and the reverse process is called deciphering or decryption. Encryption protects contents during the transmission of the data from the sender to receiver. However, after receipt and subsequent decryption, the data is no longer protected and is in the clear.Watermarking techniques are particular embodiments of steganography. The use of watermarks is almost as old as paper manufacturing.

Our ancients poured their half-stuff slurry of fiber and water on to mesh molds to collect the fiber, then dispersed the slurry within deckle frames to add shape and uniformity, and finally applied great pressure to expel the water and cohere the fiber. This process hasn’t changed too much in 2000 years. One by-product of this process is the watermark – the technique of impressing into the paper a form of image, or text derived from the negative in the mold, as the paper fibers are squeezed and dried.

The digitization of our world has expanded our concept of watermarking to include immaterial digital impressions for use in authenticating ownership claims and protecting proprietary interests. However, in principle digital watermarks are like their paper ancestors. They signify something about the token of a document or file in which they inherit. Whether the product of paper press or discrete cosine transformations, watermarks of varying degree of visibility are added to presentation media as a guarantee of authenticity, quality ownership and source.

2.2. DIGITAL WATERMARKING

The digital watermarking or watermarking explains the ways and mechanisms to hide the data and the data can be a number or text, in digital media, it may be a picture or video. The watermarking is a message that can be embedded into the digital data like video, pictures,

and text and the embedded data can be extracted later. The steganography is also another form of watermarking and in this, the messages are hidden in the content without making the people to note its presence. The Indian currency is a good example of watermarking and in the general watermarking procedure the genuine image undergoes the embedding procedure along with the watermark and the output generated will be a watermarked image.

2.3. GENERAL FRAMEWORK FOR WATERMARKING

Watermarking is the process that embeds data called a watermark or digital signature or tag or label into a multimedia object such that watermark can be detected or extracted later to make an assertion about the object. The object may be an image or audio or video.

In general, any watermarking scheme (algorithm) consists of three parts:

- The watermark
- The encoder (marking insertion algorithm)
- The decoder and comparator (verification or extraction or detection algorithm)

Each owner has a unique watermark or an owner can also put different watermarks in different objects the marking algorithm incorporates the watermark into the object. The verification algorithm authenticates the object determining both the owner and the integrity of the object.

ENCODING PROCESS:

The figure illustrates the encoding process.

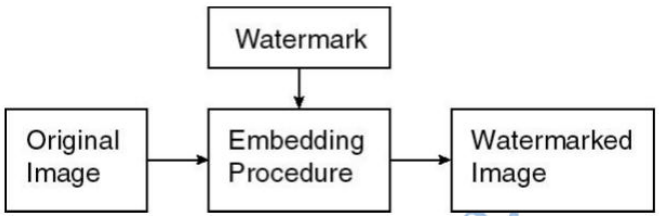

Figure 2.1

Let us denote an image by I, a signature by S = { s1, s2, … } the watermarked image by I'. E is an encoder function, it takes an image I and a signature S, and it generates a new image which is called watermarked image I', i.e. E (I, S) = I'.

DECODING PROCESS:

A decoder function D takes an image J (J can be a watermarked or unwatermarked image, and possibly corrupted) whose ownership is to be determined and recovers a signature S' from the image. In this process, an additional image I can also be included which is often the original and un-watermarked version of J. This is due to the fact that some encoding schemes may make use of the original images in the watermarking process to provide extra robustness against intentional and unintentional corruption of pixels. Mathematically,
D (J, I) = S'

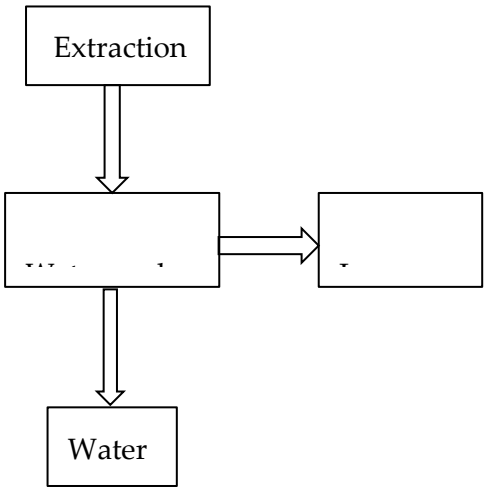

Figure 2.2

Depending on the way the watermark is inserted and depending on the nature of the watermarking algorithm, the method used can involve very distinct approaches. In some watermarking schemes, a watermark can be extracted in its exact form, a procedure we call watermark extraction. In other cases, we can detect only whether a specific given

watermarking signal is present in an image, a procedure we call watermark detection. It should be noted that watermark extraction can prove ownership whereas watermark detection can only verify ownership.

2.4. SYSTEM DESIGN

Visible watermark is a translucent overlaid into an image and is visible to the viewer. Visible watermarking is used to indicate ownership and for copyright protection. Whereas an invisible watermark is embedded into the data in such a way that the changes made to the pixel values are perceptually not noticed. Invisible watermark is used as evidence of ownership and to detect misappropriated images. Dual watermark is the combination of visible and invisible watermark. An invisible watermark is used as a backup for the visible watermark. According to Working Domain, the watermarking techniques can be divided into two types

a) Spatial Domain Watermarking Techniques

b) Frequency Domain Watermarking Techniques

In spatial domain techniques, the watermark embedding is done on image pixels while in frequency domain ater marking techniques the embedding is done after taking image transforms. Generally frequency domain methods are more robust than spatial domain techniques. According to the watermarking extraction process, techniques can be divided into three types

- Non-blind
- Semi-blind
- Blind

Non-blind watermarking schemes require original image and secret key for watermark detection whereas semi-blind schemes require secret key and watermark bit sequence for extraction. Blind schemes need only secret keys for extraction.

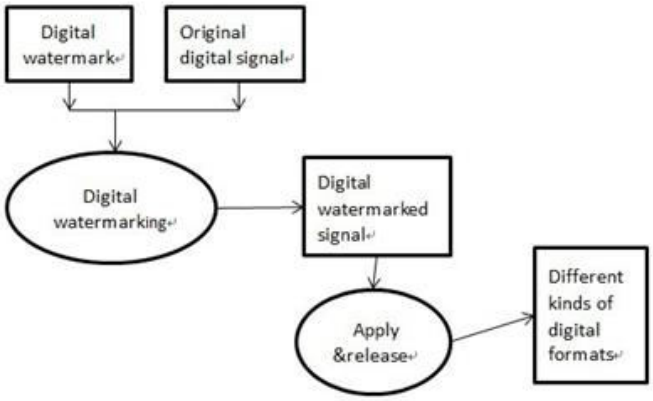

Figure 2.3

2.5. METHODOLOGY FOR IMPLEMENTATION

A watermarking system has a number of requirements. Obviously, different applications have different concerns therefore, there is no set of properties that all watermarking systems have to satisfy. This section highlights the common evaluation methods used for watermarking systems and indicates when they are important.

Invisibility:

The best way to evaluate invisibility is to conduct subject tests where both original and watermarked signals are presented to human subjects. However, due to the high volume of test images, subject tests are usually impractical. The most common evaluation method is to compute the peak signal-to-noise ratio (PSNR) between the host and watermarked signals. PSNR is defined as follows:

PSNR = 10log10 /2552 MSE

and

MSE = 1 n Xn i=1 (I_m(i) − I_w(i))2

where I_m and I_w are the original and watermarked images, respectively, n is the total number of pixels, and 255 refers to the highest possible image level in an 8-bit image. In general, the higher the PSNR, the better the signal quality.

Effectiveness :

Digital watermarking systems have a dependence on the input signal. Effectiveness refers to whether it is possible to detect a watermark immediately following the embedding process .

Although 100% effectiveness is ideal, it is often not possible to achieve such a high rate. For example, watermarking of a completely random signal is very difficult because of the lack of redundancies.

Efficiency :

Efficiency refers to the embedding capacity. For images, it is usually expressed in bits of information per pixel (bpp). A 512 x 512 image with 16 KB of embedded data has an embedding capacity of 0.5 bpp. The desired size of the watermark is application dependent.

Robustness :

Robustness is one of the most commonly tested properties in digital watermarking systems. In many applications, it is unavoidable that the watermarked signal would be distorted before it reaches the detector. Robustness refers to the ability for the detector to detect the watermark after signal distortion, such as format conversion, introduction of transmission channel noise and distortion due to channel gains.

Security One of the major goals of a digital watermarking system is to protect digital content from illegal use and distribution. However, the protection is diminished if the attackers can estimate, remove, or insert a watermark.

2.6. TYPES OF DIGITAL WATERMARKS

Watermarks and watermarking techniques can be divided into various categories in various ways.

Watermarking techniques can be divided into four categories according to the type of document to be watermarked as follows:

a) Image Watermarking

b) Video Watermarking

c) Audio Watermarking

d) Text Watermarking

According to Human Perception, the watermarking techniques can be divided into three types

a) Visible Watermark

b) Invisible Watermark

c) Dual Watermark

Visible watermark is a translucent overlaid into an image and is visible to the viewer. Visible watermarking is used to indicate ownership and for copyright protection. Whereas an invisible watermark is embedded into the data in such a way that the changes made to the pixel values are perceptually not noticed. Invisible watermark is used as evidence of ownership and to detect misappropriated images. Dual watermark is the combination of visible and invisible watermark. An invisible watermark is used as a backup for the visible watermark.

Fig 1 Example of watermark on Indian currency

Figure 2.4 VISIBLE WATERMARKING

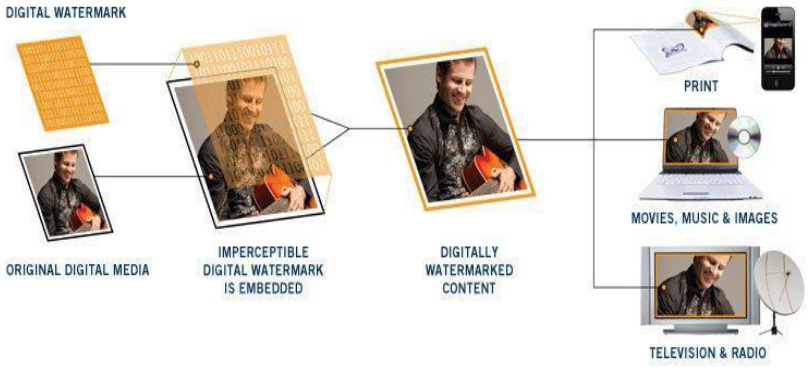

Figure 2.5 INVISIBLE WATERMARKING

2.7. APPLICATION OF DIGITAL WATERMARKS

VISIBLE WATERMARK:

Visible watermarks can be used in following cases

1. Visible watermarking for enhanced copyright protection. In such situations, where images are made available through Internet and the content owner is concerned that the images will be used commercially (e.g. imprinting coffee mugs) without payment of royalties. Here the content owner desires an ownership mark, that is visually apparent, but which doesn't prevent image being used for other purposes (e.g. scholarly research).
2. Visible watermarking used to indicate ownership originals. In this case, images are made available through the Internet and the content owner desires to indicate the ownership of the underlying materials (library manuscript), so an observer might be encouraged to patronize the institutions that owns the material.

INVISIBLE ROBUST WATERMARKS:

Invisible robust watermarks find application in following cases.

- Invisible Watermarking to detect misappropriated images. In this scenario, the seller of digital images is concerned, that his, fee-generating images may be purchased by an individual who will make them available for free, this would deprive the owner of licensing revenue.
- Invisible Watermarking as evidence of ownership. In this scenario, the seller the digital images suspects one of his images has been edited and published without payment of royalties. Here the detection of the seller's watermark in the image is intended to serve as evidence that the published image is property of seller.

INVISIBLE FRAGILE WATERMARKS:

Following are the applications of invisible fragile watermarks.

- Invisible Watermarking for a trustworthy camera. In this scenario, images are captured with a digital camera for later inclusion in news articles. Here, it is the desire of a news agency to verify that an image is true to the original capture and has not been edited to falsify a scene. In this case, an invisible watermark is embedded at capture time its presence at the time of publication is intended to indicate that the image has not been attended since it was captured.

- Invisible Watermarking to detect alternation of images stored in a digital library. In this case, images (e.g. human fingerprints) have been scanned and stored in a digital library the content owner desires the ability to detect any alteration of the images, without the need to compare the images to the scanned materials.

2.8. ATTACKS ON WATERMARKS

A watermarked image is likely to be subjected to certain manipulations, some intentional such as compression and transmission noise and some intentional such as cropping, filtering, etc. They are summarized below Lossy Compression:

Many compression schemes like JPEG and MPEG can potentially degrade the data's quality through irretrievable loss of data.

- Geometric Distortions: Geometric distortions are specific to images and videos and include such operations as rotation, translation, scaling and cropping.

Common Signal Processing Operations: They include the followings.

- D/A conversion
- A/D conversion
- Resampling
- Requantization
- Dithering distortion
- Recompression
- Linear filtering such as high pass and low pass filtering
- Non-linear filtering such as median filtering
- Color reduction
- Addition of a constant offset to the pixel values
- Addition of Gaussian and Non Gaussian noise
- Local exchange of pixels

- Other intentional attacks:
- Printing and Re scanning
- Watermarking of watermarked image (re watermarking)
- Collusion: A Number of authorized recipients of the image should not be able to come together (collude) and like the differently watermarked copies to generate an un-watermarked copy of the image (by averaging all the watermarked images).
- Forgery: A Number of authorized recipients of the image should not be able to collude to form a copy of watermarked image with the valid embedded watermark of a person not in the group with an intention of framing a 3rd party.
- IBM attack: It should not be possible to produce a fake original that also performs as well as the original and also results in the extraction of the watermark as claimed by the holder of the fake original.

2.9. DESIRED CHARACTERISTICS OF WATERMARKS

The desired characteristics of the watermarks are listed below.

- Difficult to notice: The invisible watermarks should not be noticeable to the viewers nor should the watermark degrade the quality of the content. Ideally, it should be imperceptible . However, if a signal is truly imperceptible, then perceptual based lossy compression algorithm should, in principle, remove such signal. Of course, a just noticeable difference (JND) is usually observed by comparing two signals, e.g. compressed and uncompressed or watermarked and original.
- Robustness: In general, a watermark must be robust to transformations that include common signal distortions as well as D/A and A/D conversions and loss compression. Moreover, for images and video, it is important that the watermark survive geometric distortions such as translation, scaling and cropping etc. It has

been argued that robustness can only be attained if watermark is placed perceptually significant regions of an image.

But it has been already mentioned that watermark should be imperceptible, which is possible if watermark is plated in perceptually insignificant regions of an image. They are two conflicting requirements. It should be noted robustness actually comprises two separate issues:

- whether or not the watermark is still present in the data after distortion and whether the watermark detector can detect it.

It should also be noted that ability to embed robust watermarks in digital images does not necessarily imply the ability to establish ownership, unless certain requirements are imposed legally on the watermarking scheme.

- Tamper-resistance: As well as requiring the watermark to be robust to legitimate signal distortions, a watermark may also be subjected to signal processing that is solely intended to remove the watermark. It is important that a watermark be resistant to such tampering. There are a number of possible ways this may be achieved:
- Private Watermark: A private watermark where either the decoder requires knowledge of the un-watermarked content or the pseudo-random noise sequence that constitutes the watermark is only known to sender and receiver, are inherently more tamper resistant than public watermarks in which every body is free to decode the watermark
- Asymmetric encoder/decoder: If removal of a public watermark requires inverting the encoding, then it is highly desirable to make the encoder as complex as possible, especially if the watermark is only to be applied once. However if decoders must run in real time, then it is necessary for the decoding process to be simpler than encoding.
- Bit-rate: The bit rate of a watermark refers to them amount of information a watermark can encode in a signal. This is especially important for public watermarks. Low bit-rate watermarks are more robust .

- Modification and Multiple Watermarks: In some circumstances, it is desirable to alter the watermark after insertion. For example, in the case of digital video discs, a disc may be watermarked to allow only a single copy. Once this copy has been made, it is then necessary to alter the watermark on the original disc to prohibit further copies. Changing a watermark can be accomplished either (a) removing the 1st watermark and then adding a new one or (b) inserting a 2nd a watermark such that both are readable, but are overrides the other.
- Scalability: It is well known that computer speeds are approximately doubling every eighteen months, so that what looks computationally unreasonable today may very quickly become a reality. It is therefore, very desirable to design a watermark whole decoder is scalable with each generation of computers. Thus for example, the first generation of decoder might be computationally inexpensive but might not be as reliable as next generation decoders that can afford to expend more computation to deal with issues such as geometric distortions.
- Unambiguous: Retrieval of watermark should unambiguously identify the owner. The watermark should not need any interpretation as looking into the database of codes to interpret the watermark unless a standard body maintains it internationally.
- Universal: The same digital watermark should apply to all three media under consideration. This is potentially helpful in the watermarking of multimedia products. Also this feature is conducive to implementation of audio/image/video watermarking algorithm on common hardware.
- Minimum alternation of pixels: While watermarking high quality image and art works the amount of pixel modification should be minimum. Minimum Human intervention: Insert of watermark should require little human intervention or labor.

CHAPTER 3

A VISIBLE WATERMARKING TECHNIQUE FOR IMAGE DATA

3.1. INTRODUCTION

In visible watermarking of images, a secondary image, the watermark, is embedded on a primary image such that the watermark is intentionally perceptible to a human observer whereas in the case of invisible, the embedded image data that is not perceptible, but may be extracted by a computer program.

Some of the desired characteristics of visible watermarks are listed below A visible watermark should be obvious in both color and monochrome images.

- The watermark should spread in a large and important area of the image in order to prevent its deletion by clipping.
- The watermark should be visible yet must not significantly obscure the image details beneath it.
- The watermark must be difficult to remove, rather removing a watermark should be more costly and labor intensive than purchasing the image from the owner.
- The watermark should be applied automatically with little human intervention and labor.

3.2. PROPOSED WATERMARKING TECHNIQUE WITH OPENCV

The steps for watermark insertion using Transparent Overlay using opencv are described below.

- In order to construct a transparent overlay, two images are required:
    1. Original image.

2. An image(watermark) containing what we want to "overlay" on top of the first using some level of alpha transparency.

- This watermark is a PNG image with four channels: a Red channel, a Green channel, a Blue channel, and an Alpha channel used to control the transparency of each of the pixels in the image.
- Values in our alpha channel can range [0, 255], where a value of 255 is 100% opaque (i.e., not transparent at all) while a value of 0 is 100% transparent.
- Once we actually overlay the watermark on our image, the watermark will be semi-transparent, allowing us to (partially) see the background of the original image.

3.3. CREATING A WATERMARK USING OPENCV

- Open up a new file, name it watermark_dataset.py
- Import required Python packages. We'll be making use of the packages like imutils,argparse,cv2,os,numpy.
- Parsing our required command line arguments we require three command line arguments and can supply two additional (optional) ones.
- --watermark : Here we supply the path to the image we wish to use as the watermark. We presume that (1) this image is a PNG image with alpha transparency and (2) our watermark is smaller (in terms of both width and height) then all images in the dataset we are going to apply the watermark to.
- --input : This is the path to our input directory of images we are going to watermark.
- --output : We then need to supply an output directory to store our watermarked images.
- --alpha : The optional --alpha value controls the level of transparency of the watermark. A value of *1.0* indicates that the watermark should be 100% opaque (i.e., not transparent). A value of *0.0* indicates that the watermark should be *100%* transparent.
- --correct : Finally, this switch is used to control whether or not we should preserve a "bug" in how OpenCV handles alpha transparency.

- Now that we have parsed our command line arguments, we can load our watermark image from disk
- The cv2.imread function loads our watermark image from disk using the cv2.IMREAD_UNCHANGED flag — this value is supplied so we can read the alpha transparency channel of the PNG image (along with the standard Red, Green, and Blue channels).
- The spatial dimensions (i.e., height and width) of the watermark image is taken.
- To ensure that each of the Red, Green, and Blue channels respected the alpha channel, the bitwise AND is taken.
- Start looping over each of the images in our --input directory. For each of these images, we load it from disk and its width and height is taken.
- It's important to understand that each image is represented as a NumPy array with shape (h, w, 3), where the *3* is the number of channels in our image — one for each of the Red, Green, and Blue channels, respectively.
- However, since we are working with alpha transparency, we need to add a 4th dimension to the image to store the alpha values. This alpha channel has the same spatial dimensions as our original image and all values in the alpha channel are set to *255*, indicating that the pixels are fully opaque and not transparent.
- Construct the overlay for our watermark. Again, the overlay has the exact same width and height of our input image.
- Finally, we construct our watermarked image by applying the cv2.addWeighted function.
- The output image are taken and written to the --output directory.

3.4. IMPLEMENTATION FOR WATERMARKING AN IMAGE:

```
# USAGE
#python watermark_dataset.py--watermark
pyimagesearch_watermark.png   --input input --output output
 # import the necessary packages
from imutils import paths
```

```
import numpy as np
import argparse
import cv2
import os
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-w", "--watermark", required=True,
  help="path to watermark image (assumed to be transparent PNG)")
ap.add_argument("-i", "--input", required=True,
  help="path to the input directory of images")
ap.add_argument("-o", "--output", required=True,
  help="path to the output directory")
ap.add_argument("-a", "--alpha", type=float, default=0.25,
  help="alpha transparency of the overlay (smaller is more
transparent)")
ap.add_argument("-c", "--correct", type=int, default=1,
  help="flag used to handle if bug is displayed or not")
args = vars(ap.parse_args())
alpha=args["alpha"]
beta= 1-args["alpha"]
print("alpha" + str(alpha))
print("beta" + str(beta))
# load the watermark image, making sure we retain the 4th channel
# which contains the alpha transparency
watermark = cv2.imread(args["watermark"], cv2.IMREAD_UNCHANGED)
```

```
(wH, wW) = watermark.shape[:2]
# split the watermark into its respective Blue, Green, Red, and
# Alpha channels; then take the bitwise AND between all channels
# and the Alpha channels to construct the actaul watermark
# NOTE: I'm not sure why we have to do this, but if we don't,
# pixels are marked as opaque when they shouldn't be
if args["correct"] > 0:
  (B, G, R, A) = cv2.split(watermark)
  B = cv2.bitwise_and(B, B, mask=A)
  G = cv2.bitwise_and(G, G, mask=A)
  R = cv2.bitwise_and(R, R, mask=A)
  watermark = cv2.merge([B, G, R, A])
# loop over the input images
for imagePath in paths.list_images(args["input"]):
  # load the input image, then add an extra dimension to the
  # image (i.e., the alpha transparency)
  image = cv2.imread(imagePath)
  (h, w) = image.shape[:2]
 image = np.dstack([image, np.ones((h, w), dtype="uint8") *
255])
  # construct an overlay that is the same size as the input
  # image, (using an extra dimension for the alpha transparency),
  # then add the watermark to the overlay in the bottom-right
  # corner
```

```
overlay = np.zeros((h, w, 4), dtype="uint8")
overlay[h - wH - 10:h - 10, w - wW - 10:w - 10] = watermark
# blend the two images together using transparent overlays
output = image.copy()
cv2.addWeighted(overlay, alpha, output, beta, 0, output)
# write the output image to disk
filename = imagePath[imagePath.rfind(os.path.sep) + 1:]
p = os.path.sep.join((args["output"], filename))
cv2.imwrite(p, output)
```

3.5. EXECUTION AND RESULT

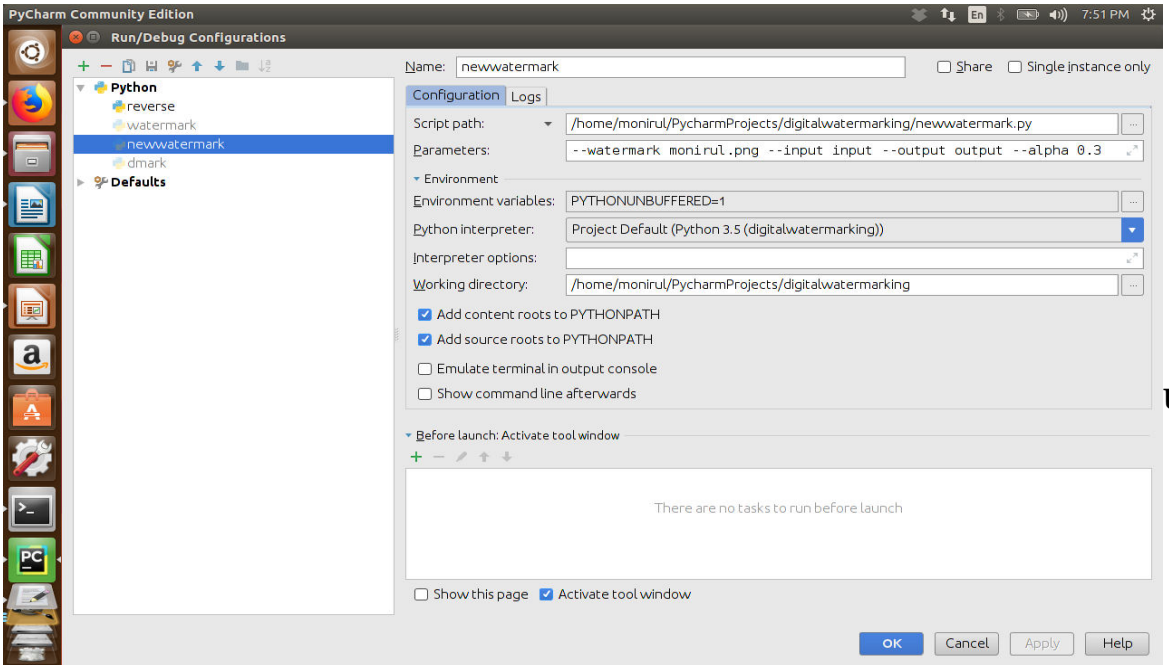

Figure 3.1

3.6. WATERMARKING RESULT

ORIGINAL IMAGE

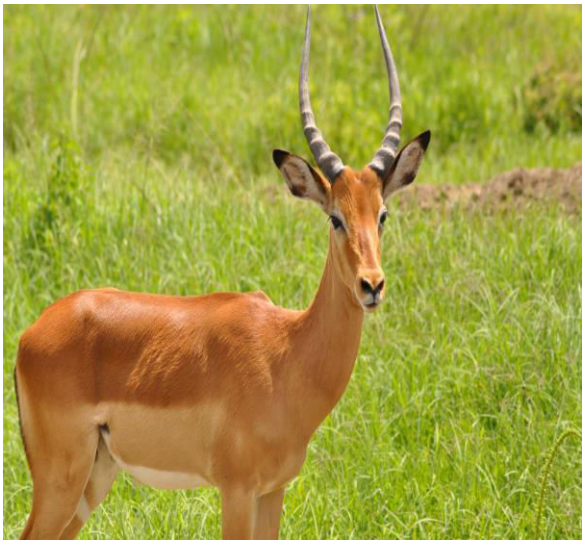

Figure 3.2(a)

WATERMARKED IMAGE

Figure 3.2(b)

The given images are watermarked

Figure 3.3(a)

Figure 3.3(b)

CHAPTER 4

A TECHNIQUE FOR REMOVAL OF A WATERMARK

4.1. PROPOSED TECHNIQUE FOR REMOVAL

- Import required Python packages. We'll be making use of the packages like imutils,numpy,argparse,cv2,os.
- Parsing our required command line arguments. We require three command line arguments and can supply two additional (optional) ones.
- --watermark : Here we supply the path to the image we wish to use as the watermark. We presume that (1) this image is a PNG image with alpha transparency and (2) our watermark is smaller (in terms of both width and height) then *all* images in the dataset we are going to apply the watermark to.
- --input : This is the path to our *input directory* of watermarked images we are going to de watermark.
- --output : We then need to supply an *output directory* to store our de watermarked images.
- --alpha : The optional --alpha value controls the level of transparency of the watermark. A value of *1.0* indicates that the watermark should be 100% opaque (i.e., not transparent).
- --correct : Finally, this is used to control whether or not we should preserve a "bug" in how OpenCV handles alpha transparency.
- The cv2.imread function loads our watermark image from disk using cv2.IMREAD_UNCHANGED.This value is supplied so we can read the alpha transparency channel of the PNG image (along with the standard Red, Green, and Blue channels).

- The spatial dimensions (i.e., height and width) of the watermark image is taken.
- To ensure that each of the Red, Green, and Blue channels respected the alpha channel, the bitwise AND is taken.
- Start looping over each of the images in our --input directory. For each of these images, we load it from disk and its width and height is taken.
- It's important to understand that each image is represented as a NumPy array with shape (h, w, 3), where the *3* is the number of channels in our image — one for each of the Red, Green, and Blue channels, respectively.
- However, since we are working with alpha transparency, we need to add a 4th dimension to the image to store the alpha values. This alpha channel has the same spatial dimensions as our original image and all values in the alpha channel are set to *255*, indicating that the pixels are fully opaque and not transparent.
- The overlay for our watermark is constructed. Again, the overlay has the exact same width and height of our input image.
- Finally, our image is constructed by applying the cv2.addWeighted function.
- then take our output image and write it to the --output directory.

4.2. IMPLEMENTATION FOR REMOVING WATERMARK:

```
# USAGE
#python watermark_dataset.py --watermark
pyimagesearch_watermark.png   --input input --output output
# import the necessary packages
from imutils import paths
import numpy as np
import argparse
import cv2
import os
# construct the argument parse and parse the arguments
```

```
ap = argparse.ArgumentParser()
ap.add_argument("-w", "--watermark", required=True,
   help="path to watermark image (assumed to be transparent PNG)")
ap.add_argument("-i", "--input", required=True,
   help="path to the input directory of images")
ap.add_argument("-o", "--output", required=True,
   help="path to the output directory")
ap.add_argument("-a", "--alpha", type=float, default=0.25,
   help="alpha transparency of the overlay (smaller is more transparent)")
ap.add_argument("-c", "--correct", type=int, default=1,
   help="flag used to handle if bug is displayed or not")
args = vars(ap.parse_args())
alpha= 1/(1-args["alpha"])
beta = (1 -1/(1- args["alpha"]))
print("alpha" + str(alpha))
print("beta" + str(beta))
# load the watermark image, making sure we retain the 4th channel
# which contains the alpha transparency
watermark = cv2.imread(args["watermark"], cv2.IMREAD_UNCHANGED)
(wH, wW) = watermark.shape[:2]
# split the watermark into its respective Blue, Green, Red, and
# Alpha channels; then take the bitwise AND between all channels
# and the Alpha channels to construct the actual watermark
```

```
# NOTE: I'm not sure why we have to do this, but if we don't,
# pixels are marked as opaque when they shouldn't be
if args["correct"] > 0:
	(B, G, R, A) = cv2.split(watermark)
	B = cv2.bitwise_and(B, B, mask=A)
	G = cv2.bitwise_and(G, G, mask=A)
	R = cv2.bitwise_and(R, R, mask=A)
	watermark = cv2.merge([B, G, R, A])
# loop over the input images
for imagePath in paths.list_images(args["input"]):
	# load the input image, then add an extra dimension to the
	# image (i.e., the alpha transparency)
	image = cv2.imread(imagePath)
	(h, w) = image.shape[:2]
	image = np.dstack([image, np.ones((h, w), dtype="uint8") *
255])
	# construct an overlay that is the same size as the input
	# image, (using an extra dimension for the alpha
transparency),
	# then add the watermark to the overlay in the bottom-right
	# corner
	overlay = np.zeros((h, w, 4), dtype="uint8")
	overlay[h - wH - 10:h - 10, w - wW - 10:w - 10] = watermark
	# blend the two images together using transparent overlays
	output = image.copy()
```

```
cv2.addWeighted(output, alpha, overlay, beta, 0, output)

# write the output image to disk

filename = imagePath[imagePath.rfind(os.path.sep) + 1:]

p = os.path.sep.join((args["output"], filename))

cv2.imwrite(p, output)
```

4.3. EXECUTION AND RESULT

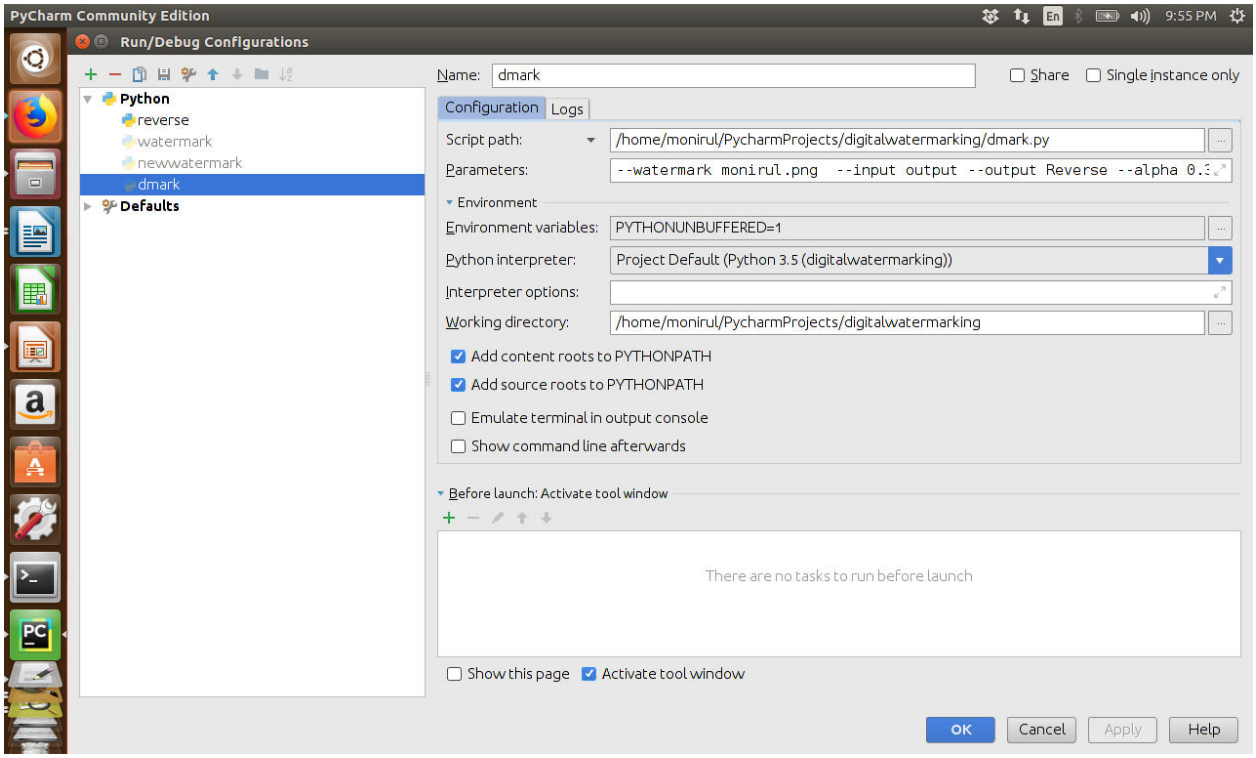

Figure 4.1

4.4. DEMARKING RESULT

Figure 4.2(a) WATERMARKED IMAGE

Figure 4.2(b) ORIGINAL IMAGE

CONCLUSIONS

To embed a hidden robust watermark to digital multimedia is the ultimate goal of watermarking system. Digital watermarking technology is an emerging field in computer science, cryptology, signal processing and communications. We have discussed the algorithms for watermarking and dewater marking of image as part of the project. The watermarking research is more exciting as it needs collective concepts from all the fields along with Human Psychovisual analysis, Multimedia and Computer Graphics. The watermark may be of visible or invisible type and each has got its own applications.

To maintain the security of the watermark, it should be embedded into randomly selected regions in some domain of the watermark signal. By doing this, it is difficult to remove the watermark.

REFERENCE

1. Methodologies in Digital Watermarking: Robust and Reversible Watermarking Techniques for Authentication, Security and Privacy Protection by Xin Cindy Guo
2. Digital Watermarking: A Tutorial, Dr. Vipula Singh
3. Selective Region Based Invisible Watermarking Using Asymmetric Key Encryption, Somenath Nag Choudhury CSE Department
4. Security and Robustness Enhancement of Digital Image Watermarking by Chittaranjan Pradhan
5. Watermarking of Digital Images by Saraju Prasad Mohanty