Vidya Pratishthan's

Kamalnayan Bajaj Institute of Engineering & Technology,

Baramati

DEPARTMENT OF COMPUTER ENGINEERING

Digital Steganography and Watermarking for Digital Images

PBL-II PROJECT REPORT

Submitted to the

Savitribai Phule Pune University

In partial fulfilment for the syllabus of PBL subject of

SECOND YEAR OF ENGINEERING

In

COMPUTER ENIGINERRING

Under Guidance of

Dr. G. J. Chhajed
Department of Computer Engineering

By

Nayana Ghalme	**2321049**
Revati Deshmukh	**2321050**
Gauri Jadhav	**2321051**
Divya Lokhande	**2321041**

VIDYA PRATISHTHAN'S KAMALNAYAN BAJAJ INSTITUTE OF ENGINEERING AND TECHNOLOGY

This is to certify that the project report entitles

Digital Steganography and Watermarking

for Digital Images

Submitted by

Nayana Sunil Ghalme	**2321049**
Revati Ravindra Deshmukh	**2321050**
Gauri Dhananjay Jadhav	**2321051**
Divya Rajendra Lokhande	**2321041**

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Dr. G. J. Chhajed** it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the project-based learning of **Computer Engineering Department**

Dr. G. J. Chhajed
Guide
Department of Computer Engineering

Dr. C.S.kulkarni
Head
Department of Computer Engineering

Dr. R. S. Bichkar
Principal,
Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering
and Technology, Baramati, Pune-413133

ACKNOWLEDGEMENT

Purpose of acknowledgements page is to show appreciation to those who contributed to conducting this project work / other tasks and duties related to the report writing. Therefore, when writing acknowledgements page, you should carefully consider everyone who helped during research process and show appreciation in the order of relevance. In this regard it is suitable to show appreciation in brief manner instead of using strong emotional phrases.

In this part of your work, it is normal to use personal pronouns like “I, my, me” while in the rest of the report this articulation is not recommended. Even when acknowledging family members and friends make sure of using the wording of a relatively formal register. The list of the persons you should acknowledged, includes guide (main and second), academic staff in your department, technical staff, reviewers, companies, family, and friends.

You should acknowledge all sources of funding. It is usually specific naming the person and the type of help you received. For example, an advisor who helped you conceptualize the project, someone who helped with the actual building or procedures used to complete the project, someone who helped with computer knowledge, someone who provided raw materials for the project, etc.

Name and Signature of Students:

1.	Nayana Sunil Ghalme	(2321049)
2.	Revati Ravindra Deshmukh	(2321050)
3.	Gauri Dhanjay Jadhav	(2321051)
4.	Divya Rajendra Lokhande	(2321041)

INDEX

1.Idea Of Project:

Steganography is the practice of concealing information within another message or physical object to avoid detection. Steganography can be used to hide virtually any type of digital content, including text, image, video, or audio content. That hidden data is then extracted at its destination.

In this project, we hide the text and image in image and we can hide the text without bothering the size. Also we can take the image of any size. The changes represent the hidden message but result if successful in no discernible change to the carrier. The information may be nothing to do with the carrier sound or image or it might be information about the carrier such as the author or a digital watermarking.

In computer, forensic field, Investigators should know about the Steganography that it can hide the data by using files without any harm. Steganography Projects is the most preferred method to hide data from the attackers. There are five different types in steganography Projects.

2.Motivation

Steganography is an additional step that can be used in conjunction with encryption in order to conceal or protect data. Steganography is a means of concealing secret information within (or even on top of) an otherwise mundane, non-secret document or other media to avoid detection.

Steganography received little attention in computing. Renewed interest because of industry desire to protect copyrighted digital work image and text.

Detect counterfeiter, unauthorised, presentation, embed key, embed author ID.

3.Sociteal Use

Watermarking is the most commonly used technique for owner identification, proof of ownership, authorship attribution, and verification and is generally used in the domain of digital copyright protection. Steganography is the practice of undetectably altering digital content to embed a secret message.

Steganography allows parties to a communication to hide different types of files over any communication medium in a way that it is hard for unauthorized persons to detect or read the concealed data.

4.Uers Of The Project

A project of steganography could be of interest to a variety of users, including:

Student:

Those studying computer science, cryptography, or information security may undertake steganography projects as part of their coursework or research. It offers a practical application of theoretical concepts and helps deepen understanding of encryption techniques.

Researchers:

Academics and professionals in the fields of cryptography, cybersecurity, and digital forensics might be interested in steganography projects to explore new methods of hiding information or to analyze the effectiveness of existing techniques.

Security Professionals:

Professionals working in information security may use steganography projects to test the security of systems or to develop countermeasures against potential threats involving hidden information.

Software Developers:

Developers interested in cybersecurity or privacy-enhancing technologies may work on steganography projects to create tools and applications for encrypting and hiding sensitive information in digital files.

Watermarking projects can attract a diverse range of users, including:

Privacy Advocate:

Individuals or groups concerned with privacy and data protection may explore watermarking projects as a means of embedding hidden messages or metadata into digital files to track their usage or prove ownership without revealing sensitive information.

Software Developers:

Developers interested in digital rights management (DRM), content protection, or cybersecurity may work on watermarking projects to create tools and software solutions for applying, verifying, and removing watermarks from digital media.

Photographers and Visual Artists:

Professionals and amateurs who create digital images may use watermarking to protect their work from unauthorized use or to establish ownership rights. Watermarks can be applied to photographs, illustrations, digital paintings, and other visual media.

Content Creators:

Individuals or organizations producing digital content such as videos, audio recordings, or documents may use watermarking to embed identifying information or copyright notices into their work. This helps deter piracy and provides a means of tracking unauthorized distribution.

5.Problem Statement

To hide the important data in the image without changing the original image to secure the data. To check the image receive by receiver is original or not, one symbolic or logo is hidden in original image.

6.Input and Output

In a steganography project, the input and output depend on the specific goals and implementation of the project. Here's a general overview:

Input:

Secret Message or Data: This is the confidential information that the user wants to hide within another file or medium. It could be text, an image, audio, video, or any other digital data.

Cover Medium: This is the file or medium in which the secret message will be hidden. It could be an image file (e.g., JPEG, PNG)

Original Content: This is the digital media (e.g., image, video, audio) to which the watermark will be applied. It could be a photograph, a video clip, a music track, or any other form of digital content.

Watermark Information: This includes the watermark itself, which could be a logo, text, image, or any other identifying information that the user wants to embed into the original content. Additionally, metadata such as copyright information or author details may also be included as part of the watermark.

The input might be an image file (original content) and a logo or text (watermark information).

Output:

Stego Medium: This is the result of embedding the secret message into the cover medium using steganographic techniques. The stego medium looks identical or very similar to the original cover medium but contains the hidden information.

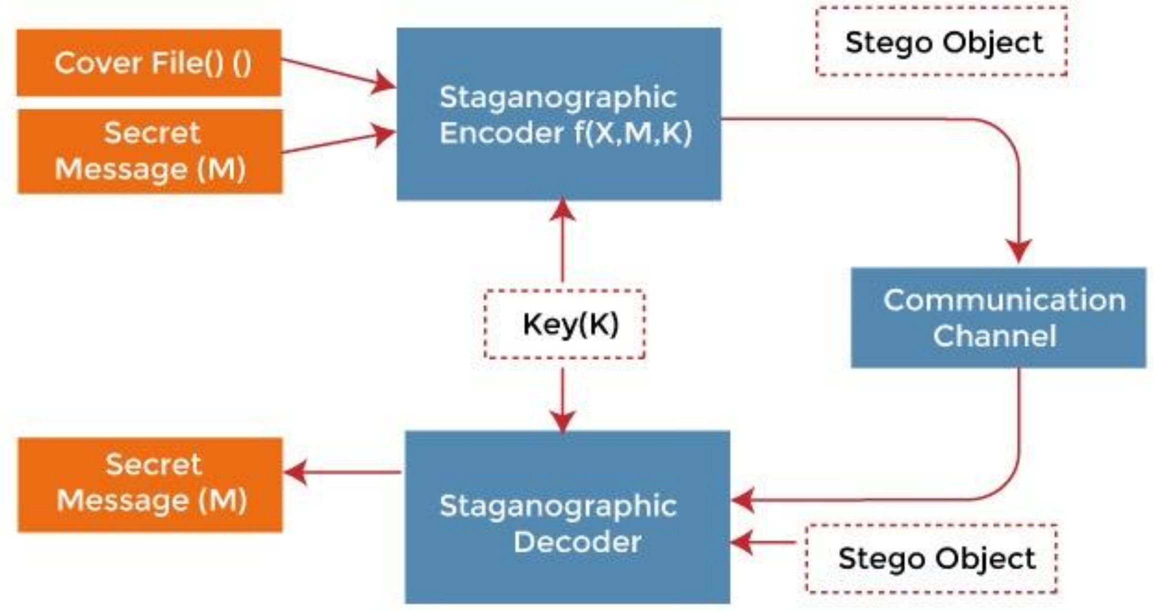
Decoded Message or Data: This is the secret message extracted from the stego medium. Decoding is the process of retrieving the hidden information using steganographic algorithms and techniques. The output could be text.

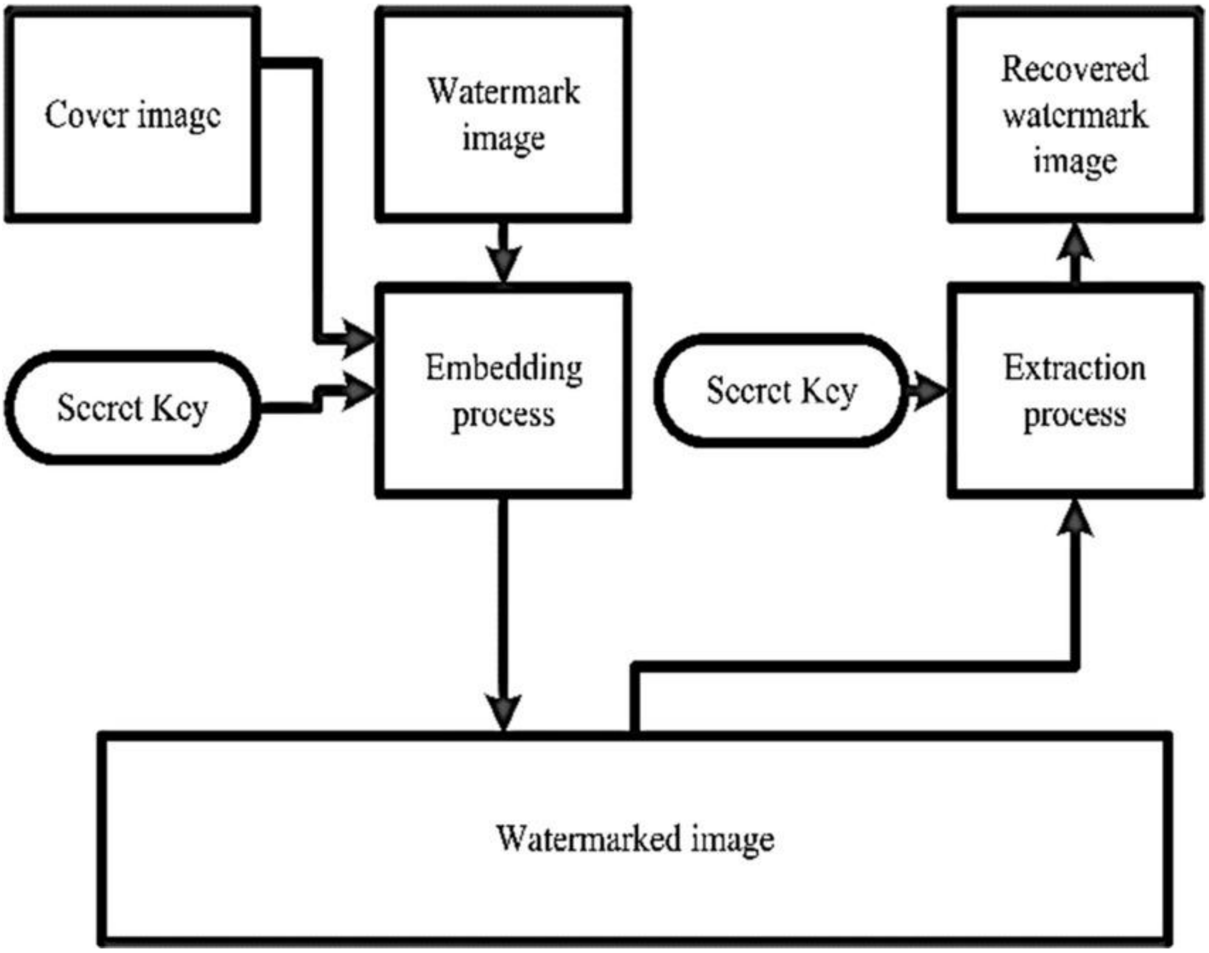
Watermarked Content: This is the result of embedding the watermark into the original content. The watermarked content preserves the original media while incorporating the watermark in a way that it's visible or hidden depending on the purpose of the watermark

Verification or Extraction Result: This depends on the type of watermark used: If it's a visible watermark, the output is simply the watermarked content, which can be visually inspected to see the embedded watermark.

The output would be a watermarked image file containing the embedded watermark. If it's an invisible watermark, the output might also include verification or extraction results confirming the presence of the watermark.

7.Flow Of Working(Input to Output):Modules of Project





Input Gathering:

Collect the secret message or data that needs to be hidden for steganography.
Obtain the watermark that will be added to the cover medium for watermarking.
Collect the cover medium e.g., an image, audio file, or video for both steganography and watermarking.

Preprocessing:

Preprocess the cover medium to ensure it's in a suitable format for both steganographic embedding and watermarking. Preprocess the secret message/data if needed e.g., convert it into binary format for steganography.

Steganographic Embedding:

Choose a steganographic algorithm e.g., LSB, DCT for embedding the secret message/data into the cover medium.
Implement the steganographic embedding algorithm to embed the secret message/data into the cover medium while minimizing perceptible changes.

Watermarking:

Choose a watermarking algorithm e.g., spatial domain, frequency domain for embedding the watermark into the cover medium.
Implement the watermarking algorithm to embed the watermark into the cover medium in a robust and imperceptible manner.

Output Generation:

Generate the steganographic output, which is the modified cover medium containing the embedded secret message/data.
Generate the watermarked output, which is the cover medium with the embedded watermark.

Extraction:

Develop an extraction algorithm for steganography to extract the hidden message/data from the steganographic output.
Develop an extraction algorithm for watermarking to extract the watermark from the watermarked output.
Implement the extraction algorithm to recover the watermark from the watermarked output.

Validation/Testing:

Ensuring the embedded message/data can be successfully extracted.
Assessing the imperceptibility of the modifications made to the cover medium.
Verifying the robustness of the watermark against various attacks.

Post-Processing:

If necessary, perform any post-processing on the extracted message/data or extracted watermark.

Output Presentation:

Present the extracted secret message/data to the authorized recipient or user for steganography. Present the extracted watermark for copyright protection or authentication purposes.

Maintenance/Updates:

Maintain the steganographic and watermarking systems by addressing any discovered vulnerabilities or weaknesses. Update the systems as necessary to adapt to changes in technology or security requirements.

8. Requirements

Steganography:

Imperceptibility:

The embedded information should be hidden in such a way that it is imperceptible to human senses. This means that the medium in which the information is hidden should appear unchanged to the casual observer.

Robustness:

The hidden information should be resilient to common attacks, such as compression, resizing, or noise addition. It should remain recoverable even if the carrier undergoes some modifications

Capacity:

The steganographic technique should provide sufficient capacity to hide the desired amount of information. The larger the capacity, the more information can be concealed, but this often comes at the cost of imperceptibility.

Security:

The embedded information should be secure from detection by unintended recipients or adversaries. This may involve encryption of the hidden data or using sophisticated embedding algorithms.

Watermarking:

Visibility:

Depending on the purpose, watermarks can be either visible or invisible. Visible watermarks are often used for asserting ownership, while invisible watermarks are used for authentication without disturbing the content's aesthetics.

Robustness:

Watermarks should withstand common transformations and attacks, such as resizing, cropping, compression, and even deliberate tampering. They should remain intact and detectable despite these manipulations.

Security:

Watermarks should be resistant to removal or alteration by unauthorized parties. Techniques such as embedding the watermark in perceptually significant areas or using cryptographic methods can enhance security.

Uniqueness:

Each watermark should be unique to the content it is associated with, allowing for accurate identification and tracking of the content.

Efficiency:

Watermark embedding and extraction processes should be efficient, especially for large-scale applications where a vast amount of content needs to be processed.

9. Resources required (H/W, S/W, language of coding)

Hardware:

Computing Devices:

You'll need a computer or computing device capable of running the software required for steganography and watermarking. This could be a desktop computer, laptop, or even a mobile device depending on your needs.

Storage:

Sufficient storage space is necessary for storing the digital media files like images, videos, audio that will be used for embedding or extracting hidden information.

Software:

Programming Languages:

Various programming languages can be used for implementing steganography and watermarking algorithms.

Python:

Widely used for its simplicity and availability of libraries like OpenCV and PIL

C/C++:

Provides high performance and low-level control, suitable for implementing complex algorithms efficiently.

Java:

Suitable for developing cross-platform applications and web-based solutions.

Libraries and Frameworks:

Depending on the chosen programming language, you may need to utilize specific libraries or frameworks for image/audio/video processing, cryptography, and data hiding techniques.

Image Editing Software:

Tools like Adobe Photoshop or GIMP can be useful for visualizing watermarks and evaluating the quality of steganographic embedding.

10.Data required and source of collection

Image:

To hide the text data in image.

We have taken the binary, grey and colourful image from google to hide text data in image.

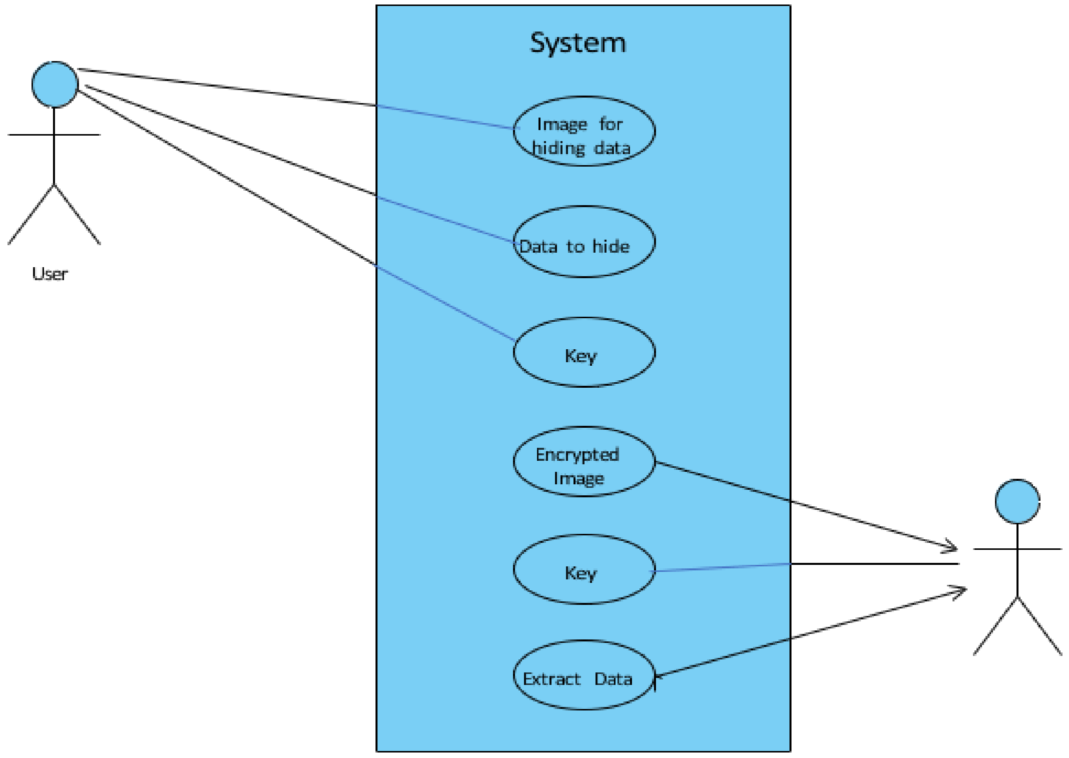
Image for watermark:

We have taken the input image from the google to hide one image in another image.

Python libraries:

Python libraries are required for image processing.

11. Use case diagram



12.Role and task of each group member

Name:	**Task Performed**

Nayana Ghalme	• Investigate various steganographic and watermarking techniques and algorithms. • Good communication with the group member to maintain the flow of project. • Work on actual algorithm. • Report writing.
Revati Deshmukh	• Document the project details, including the methodologies used, algorithms implemented, and experimental results. • Distributing the task among team members. • Report writing. • Ppt making.
Gauri Jadhav	• Review existing literature and research papers related to steganography and watermarking. • Work on research paper. • Search for the best research paper. • Collaborate to define project objectives and scope.
Divya Lokhande	• Suggestion to make the algorithm more simple. • Work on actual algorithm.

13. Alogorithm

Steganography:

```
import cv2
import string
import os
d={}
c={}

for i in range(255):
   d[chr(i)]=i
   c[i]=chr(i)


#print(c)

x=cv2.imread("image4.png")

i=x.shape[0]
j=x.shape[1]
```

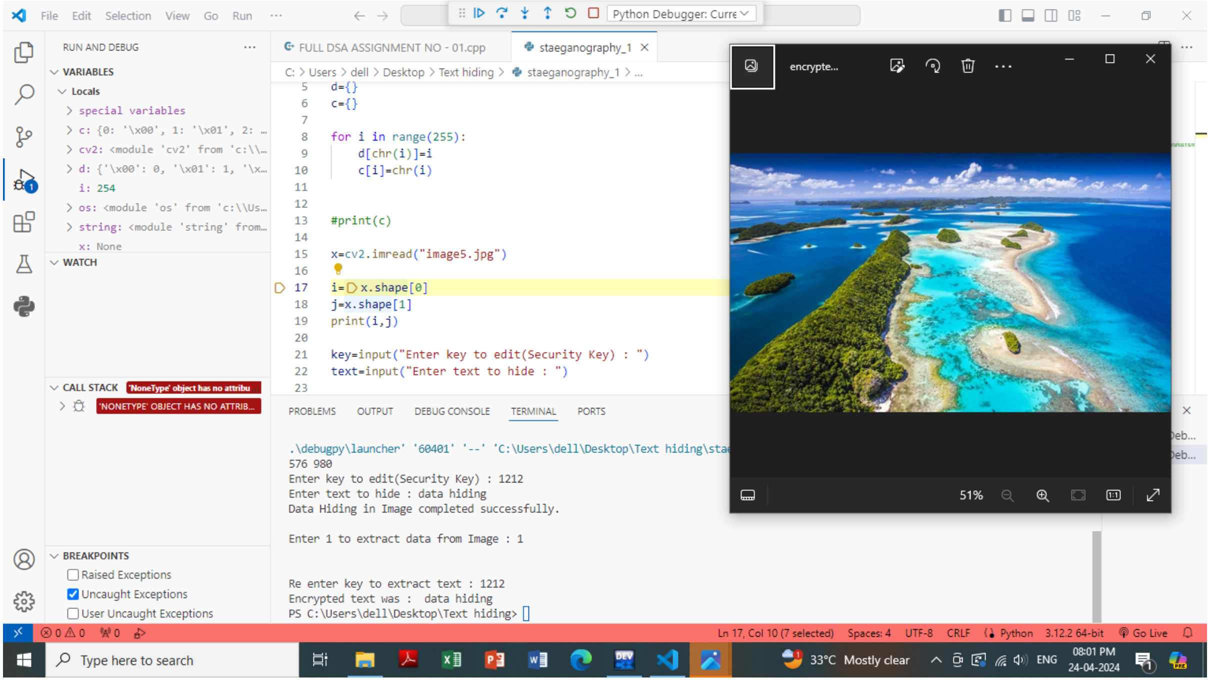
```
print(i,j)

key=input("Enter key to edit(Security Key) : ")
text=input("Enter text to hide : ")

kl=0
tln=len(text)
z=0 #decides plane
n=0 #number of row
m=0 #number of column

l=len(text)

for i in range(l):
   x[n,m,z]=d[text[i]]^d[key[kl]]
   n=n+1
   m=m+1
   m=(m+1)%3 #this is for every value of z , remainder will be between 0,1,2 .
i.e G,R,B plane will be set automatically.
            #whatever be the value of z , z=(z+1)%3 will always between 0,1,2 .
The same concept is used for random number in dice and card games.
   kl=(kl+1)%len(key)

cv2.imwrite("encrypted_img.jpg",x)
os.startfile("encrypted_img.jpg")
print("Data Hiding in Image completed successfully.")
#x=cv2.imread(“encrypted_img.jpg")


kl=0
tln=len(text)
z=0 #decides plane
n=0 #number of row
m=0 #number of column

ch = int(input("\nEnter 1 to extract data from Image : "))

if ch == 1:
   key1=input("\n\nRe enter key to extract text : ")
   decrypt=""

   if key == key1 :
      for i in range(l):
         decrypt+=c[x[n,m,z]^d[key[kl]]]
         n=n+1
         m=m+1
         m=(m+1)%3
```

```
            kl=(kl+1)%len(key)
        print("Encrypted text was : ",decrypt)
    else:
        print("Key doesn't matched.")
else:
    print("Thank you. EXITING.")
```

Watermarking:

```
import cv2
import numpy as np
import random

# Encryption function
def encrypt():
    img1 = cv2.imread('pic1.jpg')
    img2 = cv2.imread('pic2.jpg')

    # Check if images were successfully loaded
    if img1 is None:
        print("Error: Could not open or read 'pic1.jpg'")
        return
    if img2 is None:
        print("Error: Could not open or read 'pic2.jpg'")
        return

    # Resize img2 to match the dimensions of img1
    img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))

    for i in range(img1.shape[0]):
        for j in range(img1.shape[1]):
            for l in range(3):
                v1 = format(img1[i][j][l], '08b')
                v2 = format(img2[i][j][l], '08b')

                v3 = v1[:4] + v2[:4]

                img1[i][j][l] = int(v3, 2)

    cv2.imwrite('pic3in2.png', img1)

    # Display the encrypted image
    cv2.imshow('Encrypted Image', img1)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

# Decryption function
def decrypt():
    img = cv2.imread('pic3in2.png')
    width = img.shape[0]
    height = img.shape[1]

    img1 = np.zeros((width, height, 3), np.uint8)
    img2 = np.zeros((width, height, 3), np.uint8)

    for i in range(width):
        for j in range(height):
            for l in range(3):
                v1 = format(img[i][j][l], '08b')
                v2 = v1[:4] + chr(random.randint(0, 1) + 48) * 4
                v3 = v1[4:] + chr(random.randint(0, 1) + 48) * 4

                img1[i][j][l] = int(v2, 2)
                img2[i][j][l] = int(v3, 2)

    cv2.imwrite('pic2_re.png', img1)
    cv2.imwrite('pic3_re.png', img2)

    # Display the decrypted images
    cv2.imshow('Decrypted Image 1', img1)
    cv2.imshow('Decrypted Image 2', img2)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

# Driver's code
encrypt()
decrypt()
```

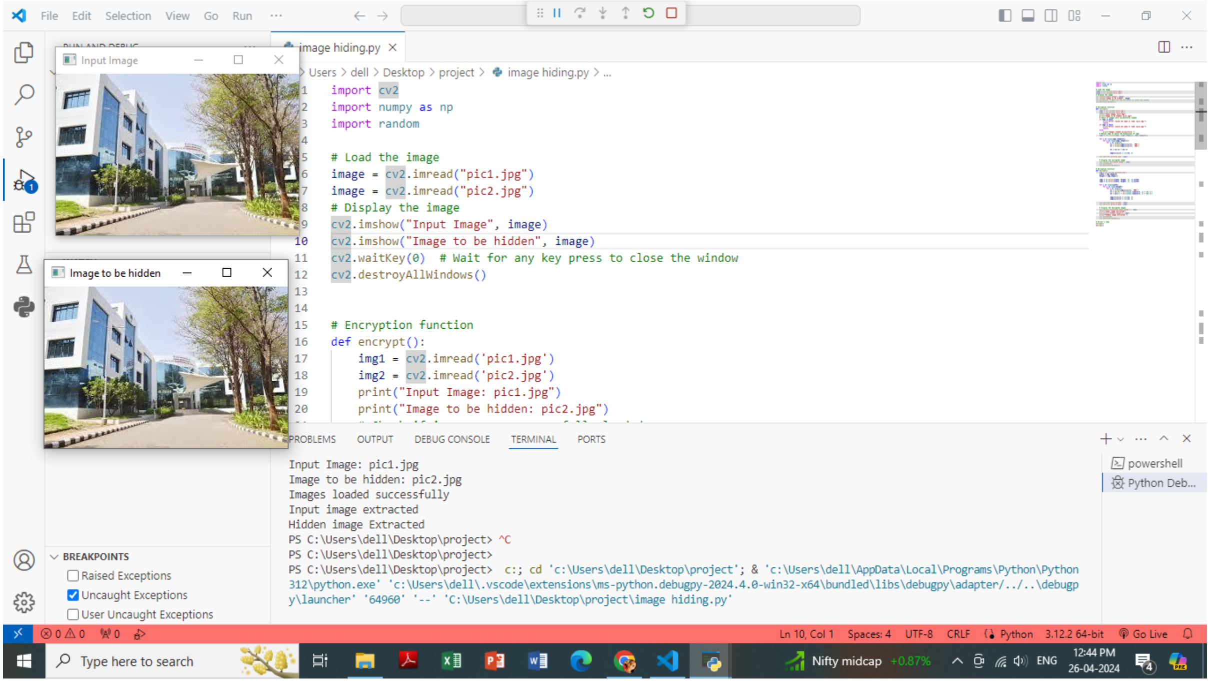
13. Screenshots of projects



RUN AND DEBUG
VARIABLES
Locals
special variables
c: {0: '\x00', 1: '\x01', 2: …
cv2: <module 'cv2' from 'c:\\…
d: {'\x00': 0, '\x01': 1, '\x…
i: 254
os: <module 'os' from 'c:\\Us…
string: <module 'string' from…
x: None
WATCH
CALL STACK
'NoneType' object has no attribu
'NONETYPE' OBJECT HAS NO ATTRIB…
BREAKPOINTS
□ Raised Exceptions
☑ Uncaught Exceptions
□ User Uncaught Exceptions
FULL DSA ASSIGNMENT NO - 01.cpp
staeganography_1
C: > Users > dell > Desktop > Text hiding > staeganography_1 > ...
encrypte...
51%
Ln 17, Col 10 (7 selected)
Spaces: 4
UTF-8
CRLF
Python
3.12.2 64-bit
Go Live
Type here to search
33°C Mostly clear
ENG
08:01 PM
24-04-2024

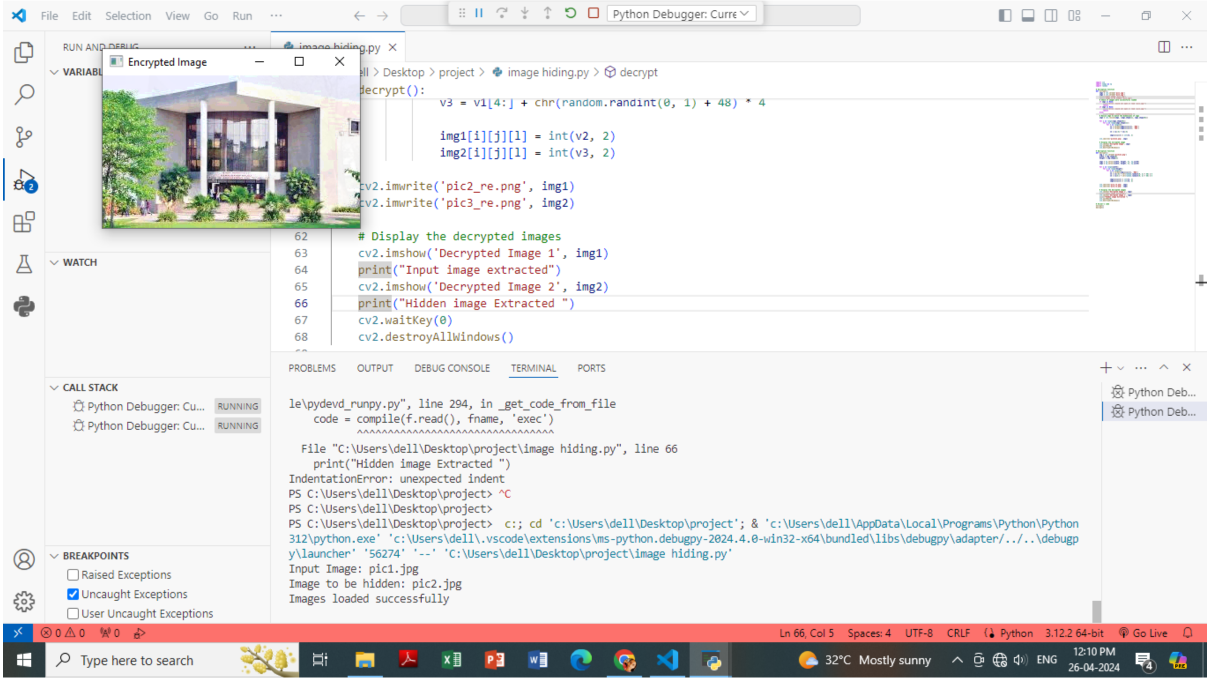
```python
d={}
c={}

for i in range(255):
    d[chr(i)]=i
    c[i]=chr(i)


#print(c)

x=cv2.imread("image5.jpg")

i=x.shape[0]
j=x.shape[1]
print(i,j)

key=input("Enter key to edit(Security Key) : ")
text=input("Enter text to hide : ")
```

```
.\debugpy\launcher' '60401' '--' 'C:\Users\dell\Desktop\Text hiding\stae
576 980
Enter key to edit(Security Key) : 1212
Enter text to hide : data hiding
Data Hiding in Image completed successfully.

Enter 1 to extract data from Image : 1


Re enter key to extract text : 1212
Encrypted text was :  data hiding
PS C:\Users\dell\Desktop\Text hiding>
```



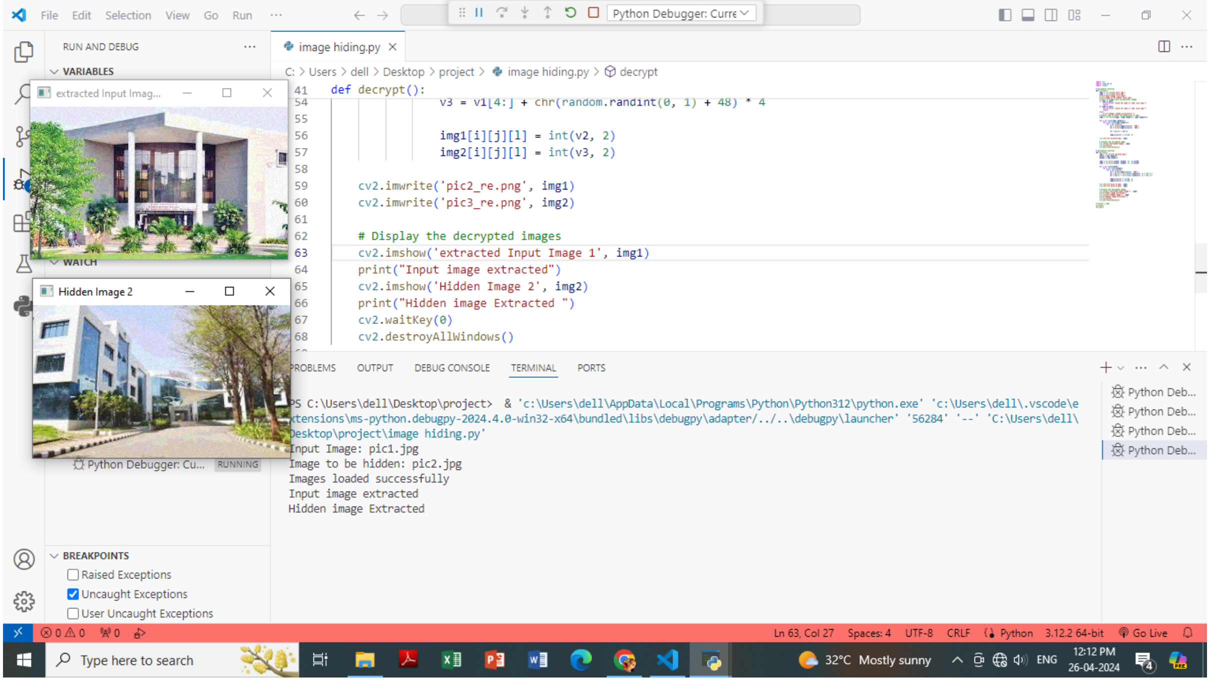
```python
import cv2
import numpy as np
import random

# Load the image
image = cv2.imread("pic1.jpg")
image = cv2.imread("pic2.jpg")
# Display the image
cv2.imshow("Input Image", image)
cv2.imshow("Image to be hidden", image)
cv2.waitKey(0)  # Wait for any key press to close the window
cv2.destroyAllWindows()


# Encryption function
def encrypt():
    img1 = cv2.imread('pic1.jpg')
    img2 = cv2.imread('pic2.jpg')
    print("Input Image: pic1.jpg")
    print("Image to be hidden: pic2.jpg")
```

```
Input Image: pic1.jpg
Image to be hidden: pic2.jpg
Images loaded successfully
Input image extracted
Hidden image Extracted
PS C:\Users\dell\Desktop\project> ^C
PS C:\Users\dell\Desktop\project>
PS C:\Users\dell\Desktop\project>  c:; cd 'c:\Users\dell\Desktop\project'; & 'c:\Users\dell\AppData\Local\Programs\Python\Python
312\python.exe' 'c:\Users\dell\.vscode\extensions\ms-python.debugpy-2024.4.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugp
y\launcher' '64960' '--' 'C:\Users\dell\Desktop\project\image hiding.py'
```



```
decrypt():
            v3 = v1[4:] + chr(random.randint(0, 1) + 48) * 4

            img1[i][j][l] = int(v2, 2)
            img2[i][j][l] = int(v3, 2)

cv2.imwrite('pic2_re.png', img1)
cv2.imwrite('pic3_re.png', img2)

# Display the decrypted images
cv2.imshow('Decrypted Image 1', img1)
print("Input image extracted")
cv2.imshow('Decrypted Image 2', img2)
print("Hidden image Extracted ")
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
le\pydevd_runpy.py", line 294, in _get_code_from_file
    code = compile(f.read(), fname, 'exec')
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\dell\Desktop\project\image hiding.py", line 66
    print("Hidden image Extracted ")
IndentationError: unexpected indent
PS C:\Users\dell\Desktop\project> ^C
PS C:\Users\dell\Desktop\project>
PS C:\Users\dell\Desktop\project>  c:; cd 'c:\Users\dell\Desktop\project'; & 'c:\Users\dell\AppData\Local\Programs\Python\Python
312\python.exe' 'c:\Users\dell\.vscode\extensions\ms-python.debugpy-2024.4.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugp
y\launcher' '56274' '--' 'C:\Users\dell\Desktop\project\image hiding.py'
Input Image: pic1.jpg
Image to be hidden: pic2.jpg
Images loaded successfully
```



```
def decrypt():
            v3 = v1[4:] + chr(random.randint(0, 1) + 48) * 4

            img1[i][j][l] = int(v2, 2)
            img2[i][j][l] = int(v3, 2)

    cv2.imwrite('pic2_re.png', img1)
    cv2.imwrite('pic3_re.png', img2)

    # Display the decrypted images
    cv2.imshow('extracted Input Image 1', img1)
    print("Input image extracted")
    cv2.imshow('Hidden Image 2', img2)
    print("Hidden image Extracted ")
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

```
PS C:\Users\dell\Desktop\project>  & 'c:\Users\dell\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\dell\.vscode\e
xtensions\ms-python.debugpy-2024.4.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '56284' '--' 'C:\Users\dell\
Desktop\project\image hiding.py'
Input Image: pic1.jpg
Image to be hidden: pic2.jpg
Images loaded successfully
Input image extracted
Hidden image Extracted
```

Input :-

Data to be hidden:- VPKBIET

Output:

Image with hidden data

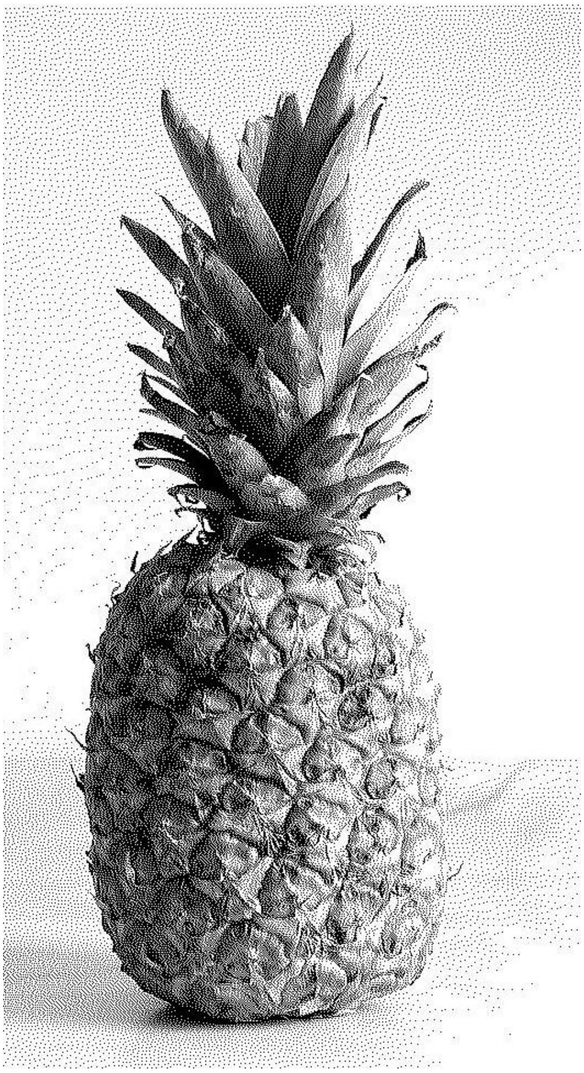
Input:-

Data to be hidden:-VPKBIET

Output:-

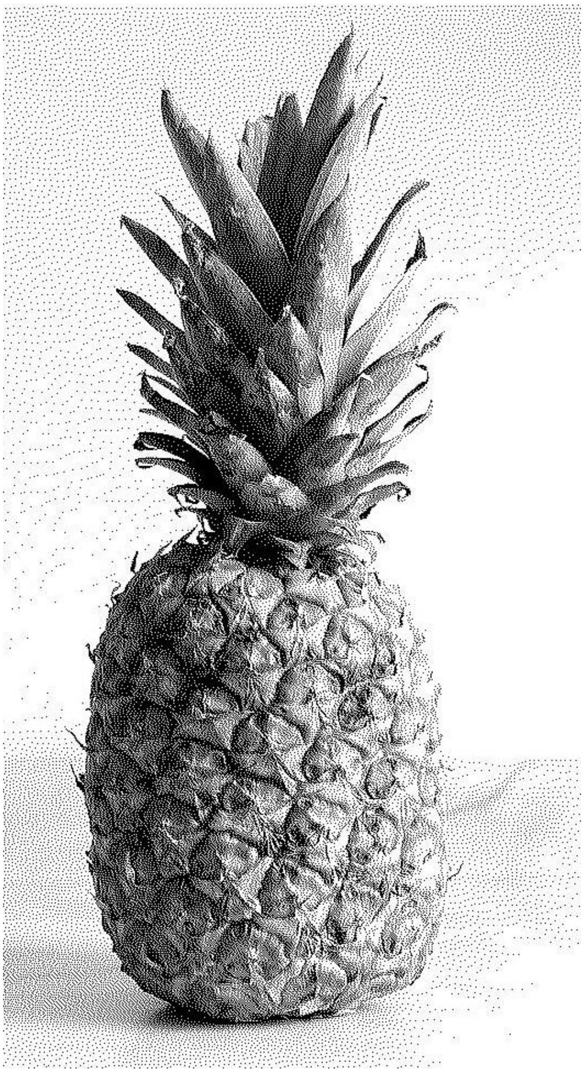
Image with hidden data

Input:



Data to be hidden:-VPKBIET

Output:



- **Watermarking:**

Input

Image to be hidden:

Output:-

Image after hidden another image

14. Gap Analysis :

Requirements Defined / Requirements fulfilled	Providing image to hide data	Providing image to hide another image	Providing key extract the data
Providing image to hide data	**YES**	**-----**	**YES**
Providing image to hide another image	**-----**	**YES**	**-----**
Providing key extract the data	**YES**	**-----**	**YES**

15. Learning through project work process and skills developed

Technical Skills:

We have gain proficiency in programming languages, algorithms, and techniques relevant to steganography and watermarking.

Problem solving:

We tried to overcome the various problem that require creative thinking.

Research Skills:

We learned learn how to find and evaluate academic papers, documentation, and other resources to deepen your understanding.

Collaboration and Teamwork:

Collaborate with peers or mentors on projects, sharing ideas, troubleshooting problems, and providing feedback. This fosters teamwork skills, communication, and the ability to work effectively in a collaborative environment.

Documentation and Presentation:

Document our project work, including methodologies, results, and conclusions. Present our findings through reports, presentations, or demonstrations. This strengthens our ability to communicate complex technical concepts effectively and enhances our writing and presentation skills.

16. Rate your learning and usefulness between 1 to 5 (1: Poor, 5: Excellent)

★ ★ ★ ★ ☆

4: Very Good

17.Conclusion

Steganography and watermarking are powerful techniques with diverse applications in digital media security and information protection. Our experiments have demonstrated the effectiveness of various steganographic and watermarking techniques in concealing information and protecting digital assets. Through project work focused on steganography and watermarking, we not only gain a deep understanding of these techniques but also develop essential skills in research, experimentation, problem-solving, and communication.

18.Reference

M. Kumar, and A. Rani, "A Short Survey Steganography", International Journal of Computer
Science and Technology, 4(1), 181-185, 2013.

A. Kumar, and K. Pooja, "Steganography- A Data Hiding Technique." International Journal of Computer Applications, 9(7), 19-23, 2010.

Internet: F. Zioner , LSB-Steganography, https://www.kitploit.com/2018/02/lsb-steganography python-program-to.html, 2018.