SF LAB II
Experiment 1

Submitted by

Aman Borkar

22CS06014

MTECH (CSE)

Under the supervision of

Dr. MANORANJAN SATPATHY

School of Electrical Science
INDIAN INSTITUTE OF TECHNOLOGY, BHUBANESWAR

2023

Introduction

Steganography is used for hiding a message in such a way that someone can not know the presence of the secret message.

Why do we use the Least Significant Bit in Steganography?

The Least Significant Bit (LSB) is a technique in which the last bit of each pixel is modified and replaced with the secret message's data bit. Suppose 255 representation in binary is

1	1	1	1	1	1	1	1
MSB							LSB

If we change the MSB bit from 1 to 0 then the value becomes 127 (99.99 % change) But if we change the LSB bit from 1 to 0 then the value becomes 254(very fewer changes in value) that's why we use LSB for hiding the secret message.

Working :

Step 1:To Hide the secret message in the image first we have to import all thepython libraries

Step 2: Upload the image that you want to use in

steganography.

Step 3: Use the prime number as a key

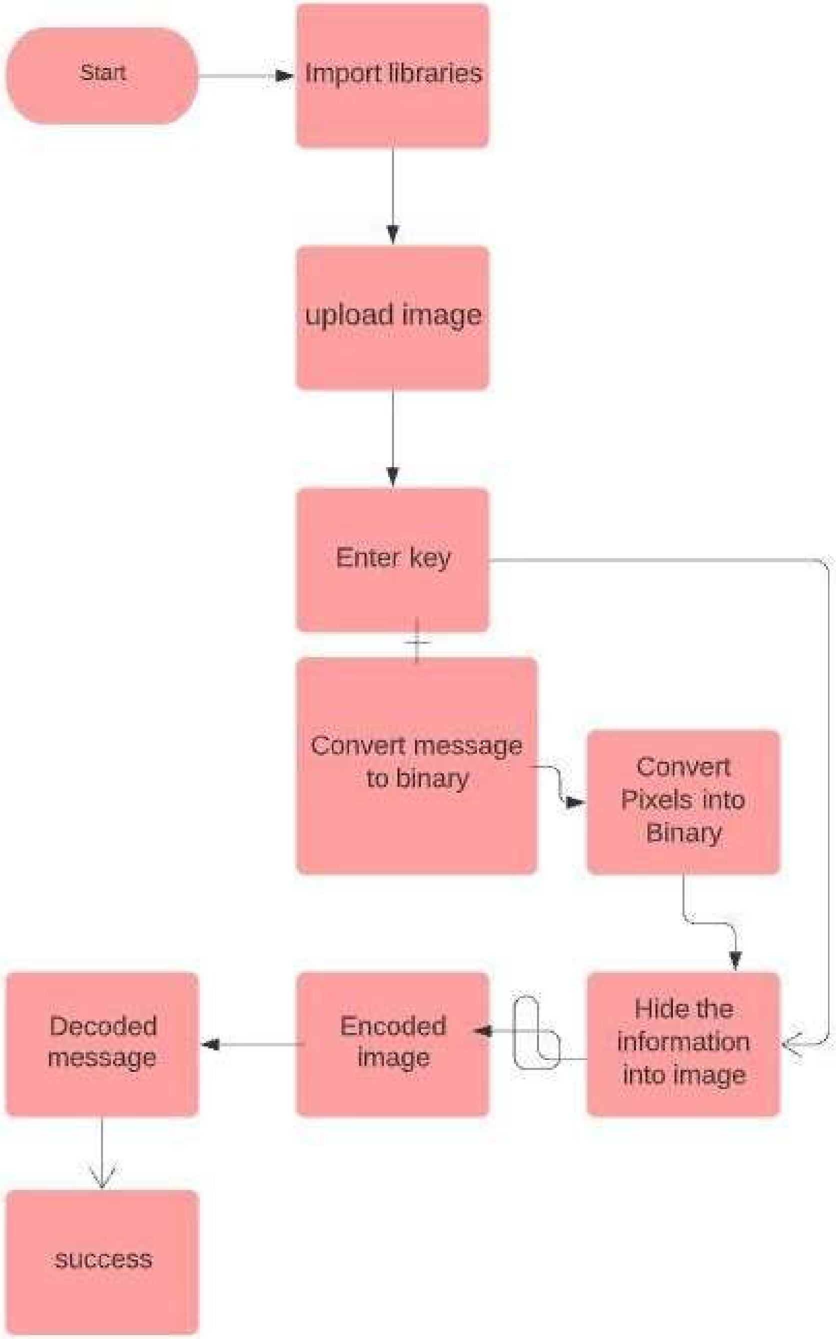
Step 4: convert the secret message and image pixels into binary

Step 5: Hide the secret message into the image by altering the LSB

Step 6: retrieve the hidden message from the stegno image.

Step 7: compare the images(before encoding, after encoding).

Flow chart :


Start
Import libraries
upload image
Enter key
Convert message to binary
Convert Pixels into Binary
Hide the information into image
Encoded image
Decoded message
success

Algorithm

Keyset

Initialize the keyset with a start value of row and column and key-value key set(start_x,start_y, key) where a key is a prime number.

Encoding

Start with data_index =0, take an input _image and message and convert it into binary_message.

Then check if the image is RGB then Access each pixel on the gap of the key, Extract each channel from the pixel and convert each channel into an 8-bit Binary string then modify the LSB of each channel with binary_message

If an image is a single-channel image (grayscale image) then access each pixel on the gap of the key, Extract each pixel and convert it into a binary string then modify the LSB of each pixel with binary_message.

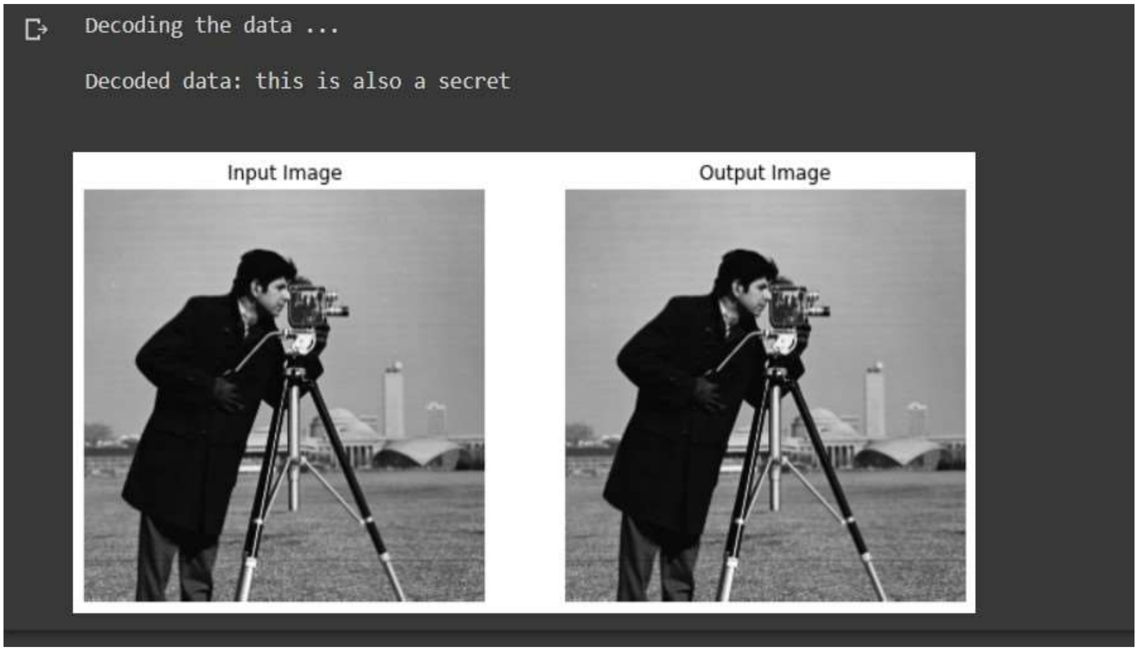
Decoding

If the input image is RGB then access each pixel on the gap of the key Extract each channel from the pixel and convert each channel into 8-bit binary string append the LSB of each channel into a resultant string then convert this resultant string into a character string.

If the input image is not RGB then access each pixel on the gap of key, extract each channel from pixel convert each pixel binary string and append LSB of each pixel into a resultant string then convert this string into char string.

Output :

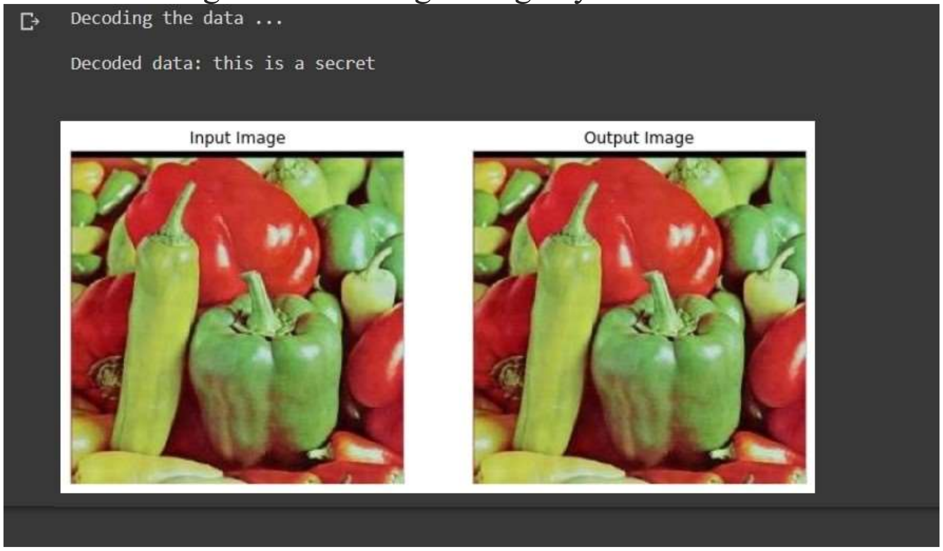
Encode message in Grayscale image using keyset.



```
Decoding the data ...

Decoded data: this is also a secret
```

Encode message in RGB image using keyset.



```
Decoding the data ...

Decoded data: this is a secret
```

Link to code:
https://colab.research.google.com/drive/1FNDOt8ZsX2htVk2LBSW_ym7WJ1gLrbHA?usp=sharing