Department of Electrical and Electronic Engineering

আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম
الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

Project Report on **EEE-3604**
Digital Signal Processing Sessional

Project Title: Steganography - Concealing and Retrieving Messages behind an image using a simple image processing technique.

Instructor:
Fazle Rabbi
Adjunct Faculty, Dept. of EEE, IIUC

Submitted by:

Mohammad Zakaria
mohammad.zakaria@ieee.org
Student ID: ET193025
Section: 6A

Date of Submission: 18 December, 2024

Project Title: Steganography - Concealing and Retrieving Messages behind an image using a simple image processing technique.

Objectives:

i. To learn about image steganography.
ii. To limit unauthorized access and provide better security during message transmission.
iii. To embed secret messages in a digital image by application of suitable encryption and decryption algorithm.

Problem Statement:

In today's scenario, the security of data is a very big challenge in any communication. Digital Image Steganography is the science of hiding sensitive information in another transmission medium to achieve secure and secret communication.

Description of Functions:

uigetfile()	It returns the file name and path to the file when the user clicks open.
strcat()	It returns horizontal concatenation of two strings.
imread()	It reads the image from the file specified by filename.
imwrite()	It writes image data to the file specified by filename. It creates the new file in our current folder.
imshow()	It displays the image stored in the graphics file specified by filename.
rgb2gray()	The rgb2gray function converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.
imbinarize()	It creates a binary image from grayscale image by replacing all values above a globally determined threshold with 1's and setting all other values to 0's.
imresize()	It returns a message image that is scaled times the size of the image base.
bitset(New,1, Msg)	It sets the bit in the given matrix at position 1 of New to the value Msg, which must be either 0 or 1.
bitget(Im,1)	It returns the value of the bit from all the values of a given matrix at position 1 in Im.
logical(A)	It converts A into an array of logical values. Any nonzero element of A is converted to logical 1 (true) and zeros are converted to logical 0 (false).

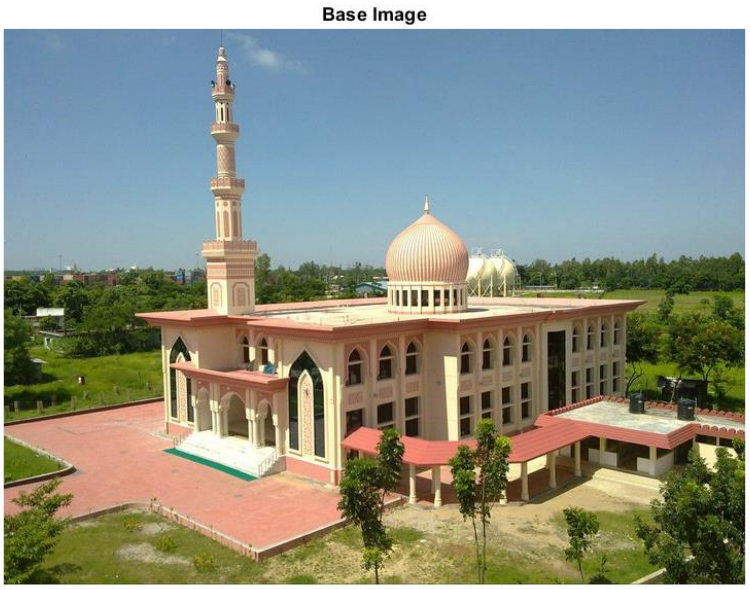
Program-1 and Result:

```
clc;
clear all;
```

Import the Base Image

```
[fname, path]=uigetfile('*.*','Please select a Base file where you want to hide message:');
fname=strcat(path,fname);
[Base]=imread(fname);
```

```
figure
imshow(Base);
title('Base Image')
```



Import the Message Image and convert to Binary Image

```
[fname, path]=uigetfile('*.*','Please select a file which you want to
hide:');
fname=strcat(path,fname);
[Message]=imread(fname);
Msg = imbinarize(rgb2gray(Message));
figure;
imshow(Msg);
title('Hidden message');
```

Resize the message and base image to same size

```
Msg = imresize(Msg,size(Base(:,:,1)));
```

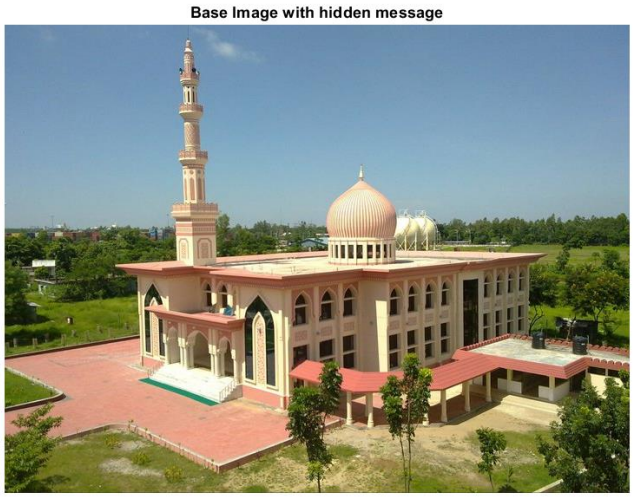
Select a bit plane and change it to our message signal

```
New = Base;
```

```
New(:,:,1) = bitset(New(:,:,1),1,Msg);
```

Save the Image file

```
figure
imshow(New);
title('Base Image with hidden message')
```




```
figure;
subplot(121)
imshow(Base);
title('Base Picture')
subplot(122)
imshow(New)
title('New Picture');
```


```
imwrite(New,'3.1_MsgIm.bmp');
```

Program-2 and Result:

Import the image with hidden image

```
[fname, path]=uigetfile('*.bmp*','Please select a file having message:');
fname=strcat(path,fname);
[Im]=imread(fname);
```

Extract the bitplane of the Messsage Signal

```
MessageImage = bitget(Im(:,:,1),1);
```

Save & Visualize the Message

```
imwrite(logical(MessageImage),'4.1_Decrypt_Message.JPG');
imshow(logical(MessageImage));
```

Analysis:

For program-1:
At first, we input the base image and message image by using uigetfile() respectively. Then convert the message image into a gray image by the rgb2gray function. Again, convert the gray image into a binary image by the imbinarize function. Then resize the message image to the same size as the base image. At this stage, both base and message images are the same size. Now selecting the LSB bit plane of the base image to change some pixels of it and change it to our message signal and get a new image within the hidden message. So, the new image looks like a base image but there embed secret messages. Then save the new image in".bmp" format. It is essential to store an encrypted image in ".bmp" format as it is the only format that doesn't compress images because if we save the image in any other format, it gets compressed and the message may be lost.

For program-2:
Now select the new image to decrypt the secret image. It's very easy to read our secret message by using bitget() function and giving the bit in which it was stored. Here we stored in LSB bit i.e. 1. Then we get the desired secret message and save it on our computer.

Discussion:
Today, in this new era of the internet, Information Security is becoming the biggest challenge for the world due to the rapid growth of internet users day by day. Unauthorized access to secret data can have serious repercussions like financial loss etc.

One of the best techniques for secure communication is Steganography-or covert writing. It is an art of hiding the very existence of communicated message itself. In this open-ended lab report, I present a hiding technique that can encrypt and decrypt messages using image steganography techniques. The main objective of this project is to encrypt a message which is in image format into images without affecting the pixel values of the original image. Every image is made up of three basic colors i.e, Red, Green, and Blue. We can hide our secret messages in these colors. By this method, we can limit unauthorized access and provide better security during message transmission. And we can embed secret messages in a digital image by using image steganography encryption and decryption algorithm.