**Lab-12**

**Name: Jenil J Gandhi**

**Roll-No: CE047**

**Subject: Network information and security**

**Q-1)**

**Aim:** Write a program to demonstrate Image Steganography operations: Embed and Extract

Hide 1 bit per pixel. Compute MSE (Mean Squared Error) and PSNR (Peak Signal to Noise Ratio) values.

**Ans**

*Code:*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*Written by:  Jenil J Gandhi*

*Subject:     Network and information security*

*Lab-no:      Lab-12*

*Description: Image steganography*

*Guidance by: Prof Mrudang T Mehta*

*Date:        10/03/2022*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

#include <bits/stdc++.h>

using namespace std;

int IMG\_SIZE = 4;

vector<vector<int>> getSteagnoObj(vector<vector<int>> image, vector<bool> message)

{

    vector<vector<int>> steanoImage;

    int c = 0;

    for (auto x : image)

    {

        vector<int> temp;

        for (auto pixel : x)

        {

            int px = pixel;

            bool messageBit = message[c++];

            if (messageBit == 1 && pixel % 2 == 0)

            {

                px += 1;

            }

            else if (messageBit == 0 && pixel % 2 == 1)

            {

                px -= 1;

            }

            temp.push\_back(px);

        }

        steanoImage.push\_back(temp);

    }

    return steanoImage;

}

vector<int> getOriginalMessage(vector<vector<int>> stegoObj)

{

    vector<int> decodedMessage;

    for (auto x : stegoObj)

    {

        for (auto pixel : x)

        {

            decodedMessage.push\_back(pixel % 2);

        }

    }

    return decodedMessage;

}

double getLoss(vector<vector<int>> image, vector<vector<int>> stegoImage)

{

    double loss = 0;

    for (int i = 0; i < IMG\_SIZE; i++)

    {

        for (int j = 0; j < IMG\_SIZE; j++)

        {

            loss += ((image[i][j] - stegoImage[i][j]) \* (image[i][j] - stegoImage[i][j]));

        }

    }

    loss /= (16);

    return loss;

}

double getPSNR(int r, double MSE)

{

    return 10 \* log10(double(r \* r) / MSE);

}

int main()

{

    int tt;

    cin >> tt;

    while (tt--)

    {

        vector<vector<int>> image{{50, 25, 49, 79}, {78, 23, 78, 80}, {49, 52, 90, 201}, {100, 59, 70, 75}};

        vector<bool> message;

        for (int i = 0; i < 16; i++)

        {

            bool tmp;

            cin >> tmp;

            message.push\_back(tmp);

        }

        vector<vector<int>> stegoImage = getSteagnoObj(image, message);

        vector<int> decodedMessage = getOriginalMessage(stegoImage);

        cout << "Stego Image:\n";

        for (auto x : stegoImage)

        {

            for (auto pixel : x)

            {

                cout << pixel << " ";

            }

            cout << endl;

        }

        cout << "Decoded Message\n";

        for (auto x : decodedMessage)

        {

            cout << x << " ";

        }

        cout << endl;

        double loss = getLoss(image, stegoImage);

        cout << "MSE Loss: " << loss << endl;

        cout << "PSNR:" << getPSNR(255, loss);

    }

    return 0;

}

**Outputs:**

**Input.txt:**

3

1 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1

1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

**Output.txt**

--------------------------------------------------------

Case #1

Stego Image:

51 24 49 79

78 23 79 80

49 52 90 200

100 59 70 75

Decoded Message

1 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1

MSE Loss: 0.25

PSNR:54.1514

--------------------------------------------------------

Case #2

Stego Image:

51 24 49 78

79 22 79 80

49 52 91 200

101 58 71 74

Decoded Message

1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

MSE Loss: 0.75

PSNR:49.3802

--------------------------------------------------------

Case #3

Stego Image:

51 25 49 79

79 23 79 81

49 53 91 201

101 59 71 75

Decoded Message

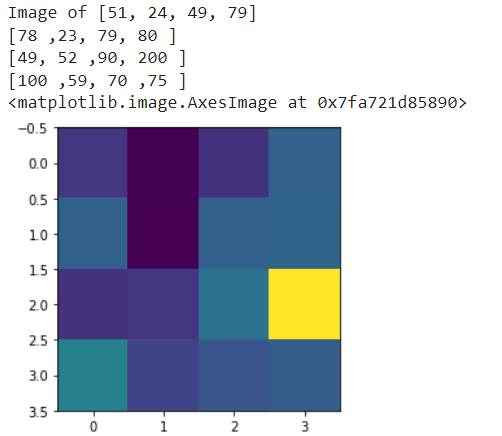
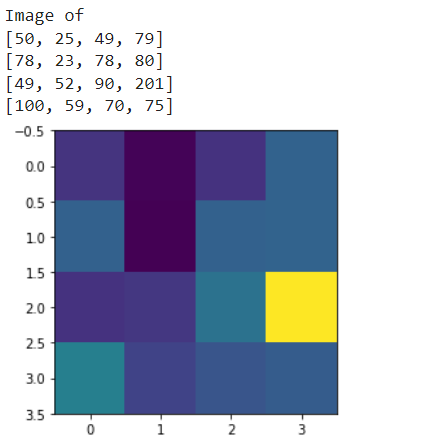
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

MSE Loss: 0.5

PSNR:51.1411

--------------------------------------------------------

**Visualizing example-1**

**  
 Input Image Stego image**