**LAB – 13**

**Name :** Shubham Pareshbhai Shingala

**Roll no. :** CE146

**College ID:** 19CEUOS159

**Aim:** Write a program to demonstrate Image Steganography operations: Embed and Extract Hide 2 bits per pixel. Hide 3 bits per pixel. Compute MSE (Mean Squared Error) and PSNR (Peak Signal to Noise Ratio) values.

* **Source Code:**

#include <bits/stdc++.h>

using namespace std;

#define BITS 2                  // hide bits per pixel

#define DIMS 4                  // image dimensions

#define MSG DIMS \*DIMS \*BITS    // input number size(in bits)

#define v1D vector<int>         //1d vector

#define v2D vector<v1D>         //2d vector

v2D lsb\_embed(v2D img, bitset<MSG> num)

{

    for (int i = 0, k = 0; i < DIMS; i++)

    {

        for (int j = 0; j < DIMS; j++)

        {

            bitset<8> t(img[i][j]); // convert decimal to binary

            for (int l = BITS - 1; l >= 0; l--, k++)

            {

                t[l] = num[MSG - 1 - k];

            }

            string t1 = t.to\_string();

            img[i][j] = stoi(t1, 0, 2); // convert binary to decimal

        }

    }

    return img;

}

float MSE(v2D img, v2D stegoImg)

{

    float sum = 0;

    for (int i = 0; i < DIMS; i++)

    {

        for (int j = 0; j < DIMS; j++)

        {

            sum += pow(img[i][j] - stegoImg[i][j], 2);

        }

    }

    return (sum / (DIMS \* DIMS));

}

float PSNR(float mse)

{

    float temp = pow(255, 2) / mse;

    return (10 \* log10(temp));

}

long long extract(v2D stegoImg)

{

    string msg = "";

    for (int i = 0; i < DIMS; i++)

    {

        for (int j = 0; j < DIMS; j++)

        {

            bitset<8> t(stegoImg[i][j]);

            for (int l = BITS - 1; l >= 0; l--)

                msg += to\_string(t[l]);

        }

    }

    return stoll(msg, 0, 2);

}

int main()

{

    long long n;

    cout << "Enter number in decimal:";

    cin >> n;

    bitset<MSG> num(n);

    cout << "Enter cover Image:\n";

    auto img = v2D(DIMS, v1D(DIMS));

    for (int i = 0; i < DIMS; i++)

        for (int j = 0; j < DIMS; j++)

            cin >> img[i][j];

    v2D stegoImg = lsb\_embed(img, num);

    cout << "\nStego Image:\n";

    for (v1D vect1D : stegoImg)

    {

        for (int pix : vect1D)

        {

            cout << pix << " ";

        }

        cout << endl;

    }

    cout << "\nMSE:";

    float mse = MSE(img, stegoImg);

    cout << mse;

    cout << "\nPSNR:";

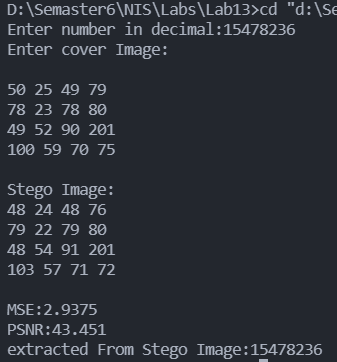
    cout << PSNR(mse);

    cout << "\nextracted From Stego Image:";

    cout << extract(stegoImg);

}

* **Test Case – 1 (Hide 2 bits per pixel):**

****

#include <bits/stdc++.h>

using namespace std;

#define BITS 3                  // hide bits per pixel

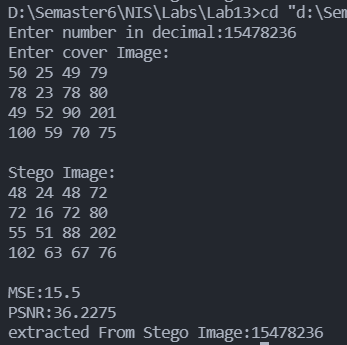
#define DIMS 4                  // image dimensions

#define MSG DIMS \*DIMS \*BITS    // input number size(in bits)

#define v1D vector<int>         //1d vector

#define v2D vector<v1D>         //2d vector

* **Test Case – 2 (Hide 3 bits per pixel):**

****