**Lab-12**

**Name : Divan MunafSha Salimsha**

**Roll No : CE035**

**Subject : NIS**

**Student ID: 19CEUBG006**

**Aim: Write a program to implement Steganography using 1-LSB.**

* **Embedding**
* **Extraction**
* **MSE**
* **PSNR**

**Source Code:**

**Programming Language : C++**

**Ans:**

#include "../functions.h"

#define N 8

#define M 64

#define Image vector<vector<unsigned long>>

using namespace std;

double mean\_squared\_error(Image img1, Image img2)

{

    double mse = 0;

    for (int i = 0; i < N; ++i)

    {

        for (int j = 0; j < N; ++j)

        {

            int m = img1[i][j];

            int n = img2[i][j];

            int diff = abs(int(m - n));

            mse += (diff \* diff \* 1.0 / (N \* N));

        }

    }

    cout << fixed << "mean squared error: " << mse << endl;

    return mse;

}

unsigned long extract(Image stego)

{

    bitset<M> msg(0);

    int k = 0;

    for (int i = 0; i < N; ++i)

    {

        for (int j = 0; j < N; ++j)

        {

            bitset<M> pixel(stego[i][j]);

            msg[k++] = pixel[0];

        }

    }

    cout << "\nExtracted Msg in Binary: " << msg << endl;

    return msg.to\_ulong();

}

Image embed(int msg, Image image)

{

    bitset<M> msg\_binary(msg);

    Image stego(N, vector<unsigned long>(N));

    int k = 0;

    for (int i = 0; i < N; ++i)

        for (int j = 0; j < N; ++j)

        {

            bitset<M> pixel(image[i][j]);

            pixel[0] = msg\_binary[k++];

            stego[i][j] = pixel.to\_ulong();

        }

    return stego;

}

void peak\_signal\_to\_noise\_ratio(Image img1, Image img2, double mse)

{

    double psnr = 10 \* log10(255 \* 255.0 / mse);

    cout << "Peak signal to Noise Ratio: " << psnr << endl;

}

int main()

{

    ifstream input("Input.txt");

    Image image(N, vector<unsigned long>(N));

    for (int i = 0; i < N; ++i)

        for (int j = 0; j < N; ++j)

            input >> image[i][j];

    int\_fast64\_t msg;

    input >> msg;

    Image stego\_image = embed(msg, image);

    cout << "\n-------------------------Original Image--------------------------------" << endl;

    for (int i = 0; i < N; ++i)

    {

        for (int j = 0; j < N; ++j)

        {

            cout << image[i][j] << " ";

        }

        cout << endl;

    }

    cout << "-------------------------Stego Image--------------------------------" << endl;

    for (int i = 0; i < N; ++i)

    {

        for (int j = 0; j < N; ++j)

        {

            cout << stego\_image[i][j] << " ";

        }

        cout << endl;

    }

    unsigned long extracted = extract(stego\_image);

    cout << "Original Msg: " << msg << endl;

    cout << "Extracted Msg: " << extracted << endl;

    double mse = mean\_squared\_error(image, stego\_image);

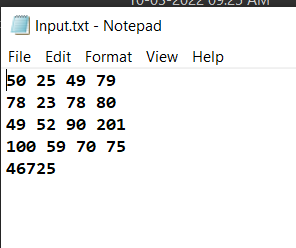
    peak\_signal\_to\_noise\_ratio(image, stego\_image, mse);

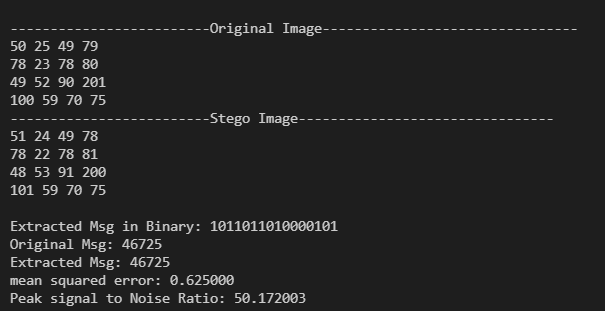
    return 0;

}

**Input & Output Screenshots:**

* ***For 4x4 matrix***





* ***For 16x16 matrix***

