

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);
```

```

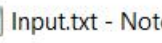
Image stego(N, vector<unsigned long>(N));
int k = 0;
for (int i = 0; i < N; ++i)
    for (int j = 0; j < N; ++j)
    {
        bitset<M> pixel(image[i][j]);
        pixel[0] = msg_binary[k++];
        stego[i][j] = pixel.to_ulong();
    }
return stego;
}

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;
}

```

- **For 4x4 matrix**

- 
- Input.txt - Notepad
- File Edit Format View Help
- ```
50 25 49 79
78 23 78 80
49 52 90 201
100 59 70 75
46725
```

- **For 16x16 matrix**

```
-----Original Image-----
50 25 49 79 50 25 49 79
49 52 90 201 78 23 78 80
100 59 70 75 78 23 78 80
49 52 90 201 100 59 70 75
50 25 49 79 100 59 70 75
78 23 78 80 78 23 78 80
49 52 90 201 50 25 49 79
100 59 70 75 50 25 49 79

-----Stego Image-----
51 24 49 78 50 24 49 78
49 53 90 201 79 22 79 81
100 59 70 74 78 22 79 80
49 53 90 201 101 58 71 74
50 24 48 78 100 58 70 74
78 22 78 80 78 22 78 80
48 52 90 200 50 24 48 78
100 58 70 74 50 24 48 78

Extracted Msg in Binary: 0000000000000000000000000000000001011011010000101101101101000101
Original Msg: 1531108165
Extracted Msg: 1531108165
mean squared error: 0.546875
Peak signal to Noise Ratio: 50.751923
```