LAB:12

NAME: DHRUV PATEL

ROLL NO: CE111

SUBJECT: NIS

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.

♦ **Source Code**

```cpp
#include <bits/stdc++.h>
#include <vector>
using namespace std;

int convertBintoDec(string bin)
{
    int dec = 0;
    for (int i = bin.length() - 1, j = 0; i >= 0; i--, j++)
    {
        if (bin[i] == '1')
        {
            dec += pow(double(2), double(j));
        }
    }
    return dec;
}
string convertDectoBin(int dec)
{
    string bin;
    while (dec != 0)
    {
        bin += to_string(dec % 2);
        dec /= 2;
    }
    reverse(bin.begin(), bin.end());
    return bin;
}
vector<vector<int>> embeded(vector<vector<int>> cover, string msg)
{
    vector<vector<int>> stego_obj;
    int index = 0;
    for (int i = 0; i < cover.size(); i++)
    {
        vector<int> row;
        for (int j = 0; j < cover[0].size(); j++)
```

```cpp
            {
                string bin = convertDectoBin(cover[i][j]);
                bin[bin.length() - 1] = msg[index];
                cover[i][j] = convertBintoDec(bin);
                row.push_back(cover[i][j]);
                index++;
            }
            stego_obj.push_back(row);
        }
    return stego_obj;
}
string extraction(vector<vector<int>> embeded_cover)
{
    string extracted_msg = "";
    for (int i = 0; i < embeded_cover.size(); i++)
    {
        for (int j = 0; j < embeded_cover[0].size(); j++)
        {
            string bin = convertDectoBin(embeded_cover[i][j]);
            extracted_msg += bin[bin.length() - 1];
        }
    }
    return extracted_msg;
}
double MSE(vector<vector<int>> i1, vector<vector<int>> i2)
{

    double mse = 0;
    for (int i = 0; i < i1.size(); i++)
    {
        for (int j = 0; j < i1[0].size(); j++)
        {
            mse += pow(double(i1[i][j] - i2[i][j]), double(2));
        }
    }
    mse /= i1.size() * i1[0].size();
    return mse;
}
double PSNR(double MSE)
{
    double r = 255;
    double psnr = 10 * (log10(r * r / MSE));
    return psnr;
}
int isBinary(string input)
{
    for (int i = 0; input[i]; ++i)
        if (input[i] != '0' && input[i] != '1')
            return 0;
```

```cpp
    return 1;
}
int main()
{

    string msg;
    cout << "Enter Message (in binary) :" << endl;
    cin >> msg;
    if (isBinary(msg) == 0)
    {
        cout << "Enter valid binary number" << endl;
        return 0;
    }
    int n = sqrt(msg.length());
    if (n * n != msg.length())
    {
        cout << "Enter Binary Mesage with legth whose whole square root is
possible " << endl;
        return 0;
    }
    cout << "------------------------------------------------------ " << endl;
    cout<< "Enter Cover (Dimensions " << n << " * " << n << " ) : " << endl;
    vector<vector<int>>cover;
    for (int i = 0; i < n; i++)
    {
        vector<int> row;
        for (int j = 0; j < n; j++)
        {
            int val;
            cin >> val;
            row.push_back(val);
        }
        cover.push_back(row);
    }
    cout << "------------------------------------------------------ " << endl;
    cout<< "Stego Object : " << endl;
    vector<vector<int>> stego_obj = embeded(cover, msg);
    for (int i = 0; i < stego_obj.size(); i++)
    {
        for (int j = 0; j < stego_obj[0].size(); j++)
        {
            cout << stego_obj[i][j] << " ";
        }
        cout << endl;
    }
    cout << "------------------------------------------------------ " << endl;
    string extraction_msg = extraction(stego_obj);
    cout << "Extracted message : " << extraction_msg << endl;
    cout << "------------------------------------------------------ " << endl;
```

```
    double mse = MSE(cover, stego_obj);
    cout << "Min Square Error :" << mse << endl;
    cout << "--------------------------------------------------- " << endl;
    cout<< "Peak Signal to Noise Ratio (PSNR) : " << PSNR(mse)<< endl;
    return 0;
}
```

♦ **Input Output**

```
C:\Users\Dhruv j Patel\Desktop\NIS\github\NIS\Lab12>a
Enter Message (in binary) :
1111111111111111
---------------------------------------------------
Enter Cover (Dimensions 4 * 4 ) :
10 11 12 13
14 15 16 17
18 19 20 21
22 23 24 25
---------------------------------------------------
Stego Object :
11 11 13 13
15 15 17 17
19 19 21 21
23 23 25 25
---------------------------------------------------
Extracted message : 1111111111111111
---------------------------------------------------
Min Square Error :0.5
---------------------------------------------------
Peak Signal to Noise Ratio (PSNR) : 51.1411
```

```
C:\Users\Dhruv j Patel\Desktop\NIS\github\NIS\Lab12>a
Enter Message (in binary) :
11110000111100001111000011110000111100001111000011110000
---------------------------------------------------
Enter Cover (Dimensions 8 * 8 ) :
90 30 60 20 10 40 70 80
91 31 61 21 11 41 71 81
92 32 62 22 12 42 72 82
93 33 63 23 13 43 73 83
94 34 64 24 14 44 74 84
95 35 65 25 15 45 75 85
96 36 66 26 16 46 76 86
55 55 55 55 55 55 55 55
---------------------------------------------------
Stego Object :
91 31 61 21 10 40 70 80
91 31 61 21 10 40 70 80
93 33 63 23 12 42 72 82
93 33 63 23 12 42 72 82
95 35 65 25 14 44 74 84
95 35 65 25 14 44 74 84
97 37 67 27 16 46 76 86
55 55 55 55 54 54 54 54
---------------------------------------------------
Extracted message : 11110000111100001111000011110000111100001111000011110000
---------------------------------------------------
Min Square Error :0.5
---------------------------------------------------
Peak Signal to Noise Ratio (PSNR) : 51.1411
```