LAB12

Name: Suthar Utsav

Roll no: CE153

ID: 19CEUBS025

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:

```
// Image Stagnography

#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++)
        {
            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;
    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;
}
```

Test case 1:

```
PS D:\sem6\NIS\Lab12> ./a.exe
Enter Message (in binary) :
1011011010000101
---------------------------------------------------
Enter Cover (Dimensions 4 * 4 ) :
50 25 49 79
78 23 78 80
49 52 90 201
100 59 70 75
---------------------------------------------------
Stego Object :
51 24 49 79
78 23 79 80
49 52 90 200
100 59 70 75
---------------------------------------------------
Min Square Error :0.25
---------------------------------------------------
Peak Signal to Noise Ratio (PSNR) : 54.1514
```

Test case 2:

```
PS D:\sem6\NIS\Lab12> g++ ImageStagnography.cpp
PS D:\sem6\NIS\Lab12> ./a.exe
Enter Message (in binary) :
10100100110100110010101001100101100101000110001010000101010111 10
--------------------------------------------------
Enter Cover (Dimensions 8 * 8 ) :
25 19 36 28 56 100 27 56
99 43 88 34 82 357 9 47
5 67 22 57 77 34 75 155
93 57 28 66 36 90 77 45
144 34 27 67 86 21 20 85
89 78 77 67 65 55 64 66
34 24 23 35 56 77 83 21
23 78 46 36 47 38 48 99
--------------------------------------------------
Stego Object :
25 18 37 28 56 101 26 56
99 43 88 35 82 356 9 47
4 66 23 56 77 34 75 154
92 57 29 66 36 91 76 45
145 34 26 67 86 21 20 84
88 79 77 66 64 54 65 66
35 24 22 34 56 77 82 21
22 79 46 37 47 39 49 98
--------------------------------------------------
Extracted message : 10100100110100110010101001100101100101000110001010000101010111 10
--------------------------------------------------
Min Square Error :0.53125
--------------------------------------------------
Peak Signal to Noise Ratio (PSNR) : 50.8778
PS D:\sem6\NIS\Lab12>
```

Test case 3:

```
PS D:\sem6\NIS\Lab12> g++ ImageStagnography.cpp
PS D:\sem6\NIS\Lab12> ./a.exe
Enter Message (in binary) :
1000100021001000
Enter valid binary number
PS D:\sem6\NIS\Lab12>
```