

Лабораторно-практическое задание

№ 2

«Системы технического зрения в автоматизированных системах»

Сайфудинов Р.Р. 171-261

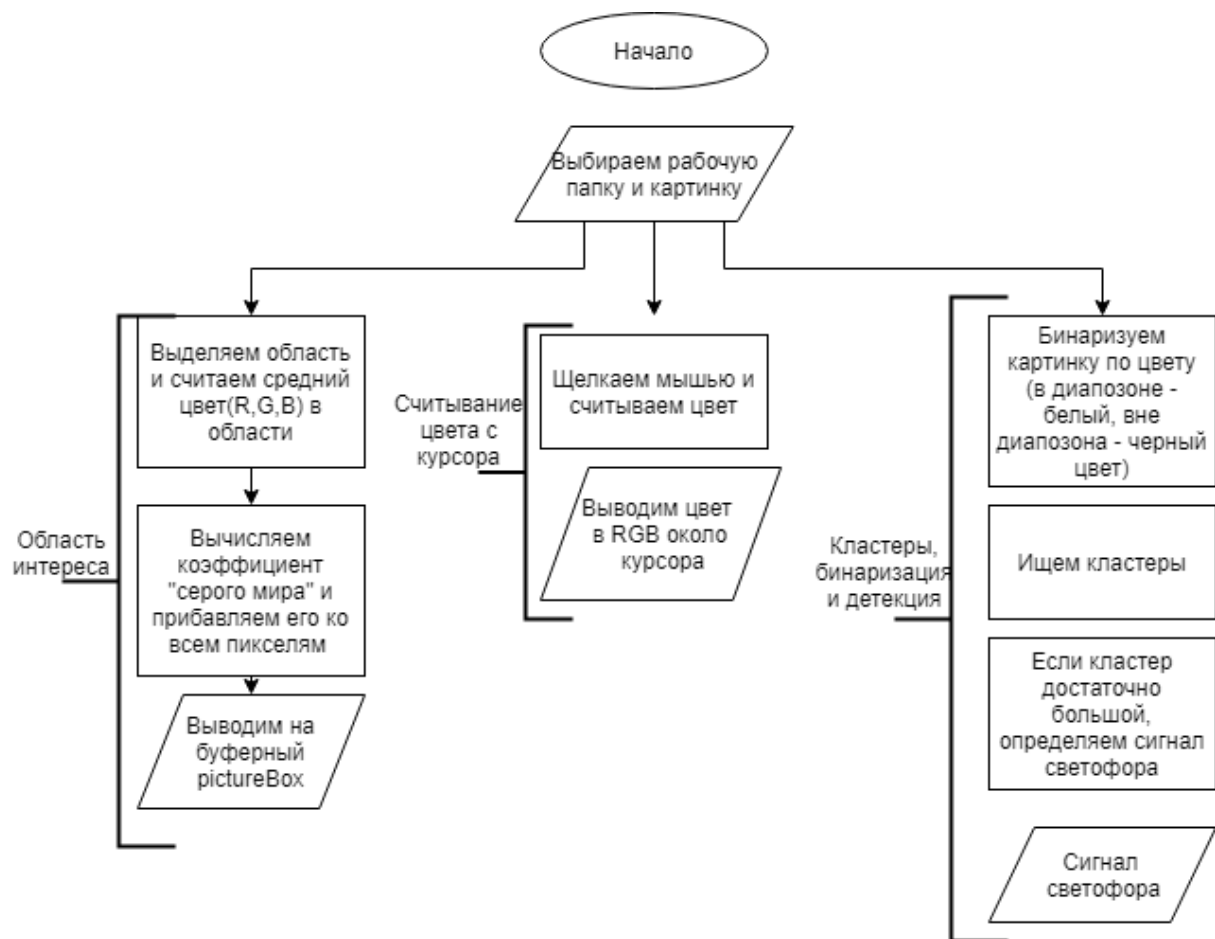
Тема: Базовые принципы обработки изображений и поиска зон интереса.

Цель работы: Разработать алгоритм обработки и коррекции изображения для поиска зон интереса с использованием цветового и структурного ключей

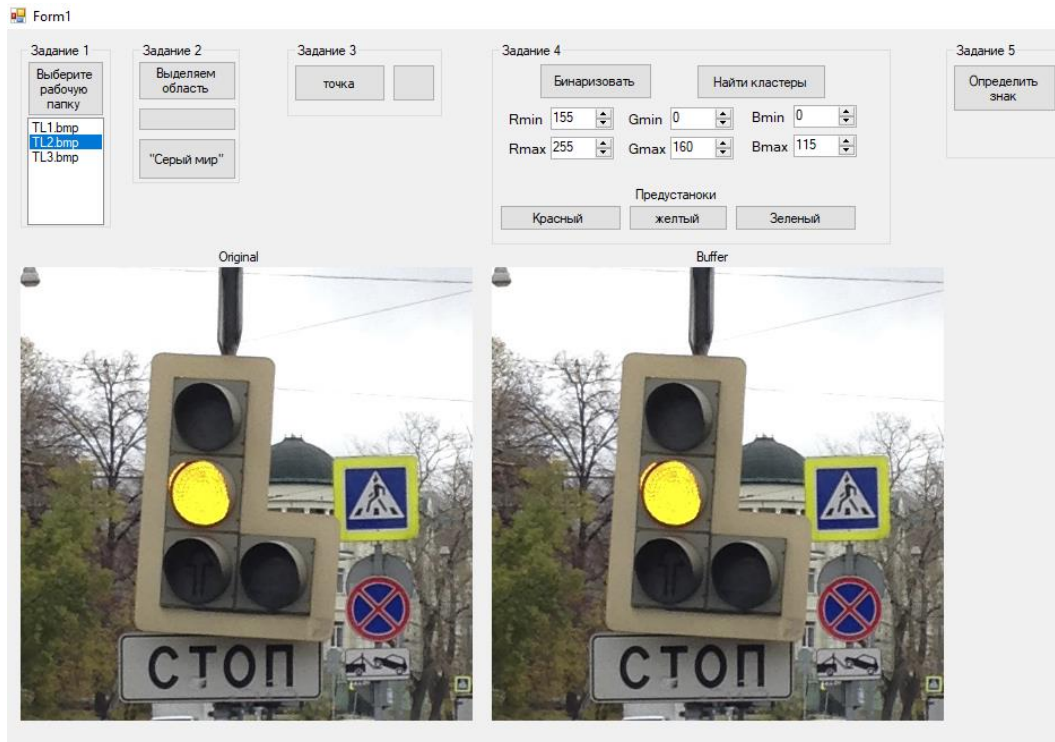
Задачи:

1. Подготовить приложение для показа изображений из рабочей папки с выбором изображения из списка;
2. Разработать функцию задания области интереса указателем мыши с использованием буфера для хранения исходного изображения;
3. Разработать функцию вычисления параметров цветокоррекции по заданной области интереса и выполнения цветокоррекции;
4. Разработать функцию считывания параметров пикселя и вывода значений на изображении;
5. Разработать функцию фильтрации изображения с использованием цветового ключа по всем компонентам RGB;
6. Разработать функцию расчёта координат центра кластера выделенных фильтром пикселей в области модальных значений и отсекающих по удалённости точек;
7. Разработать программу детектирования работы светофора.

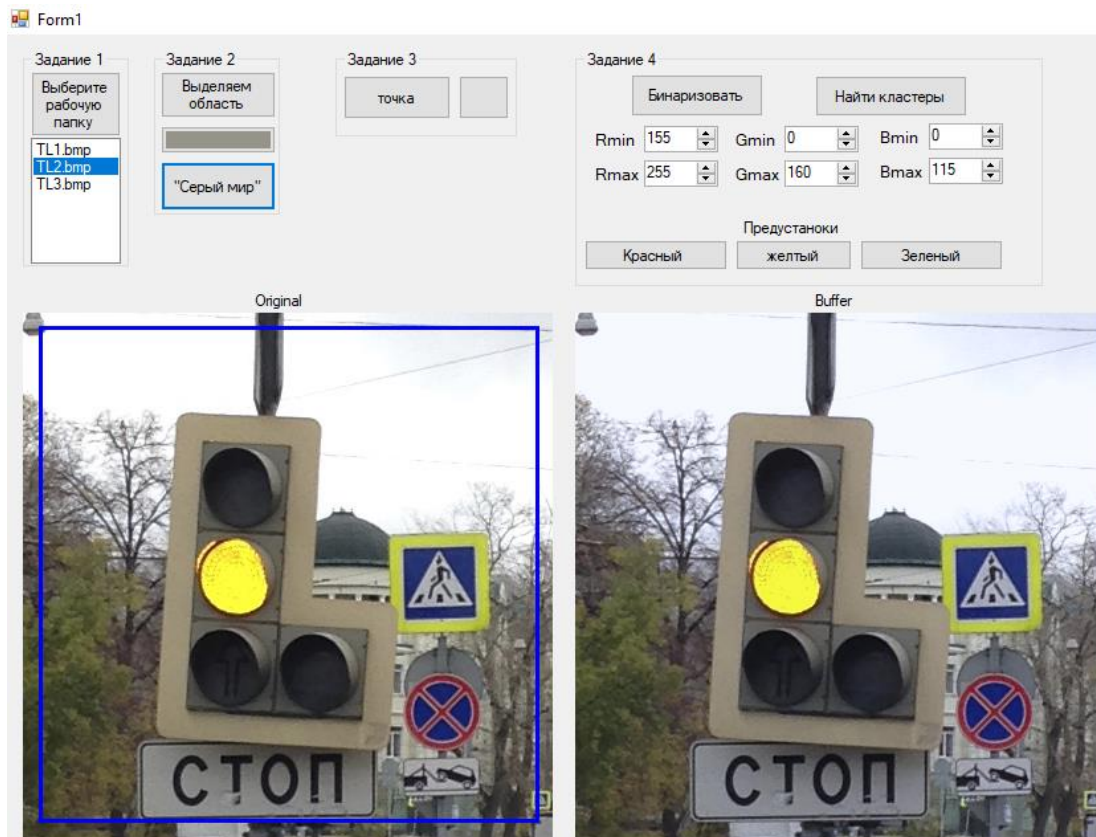
Блок схема



Интерфейс



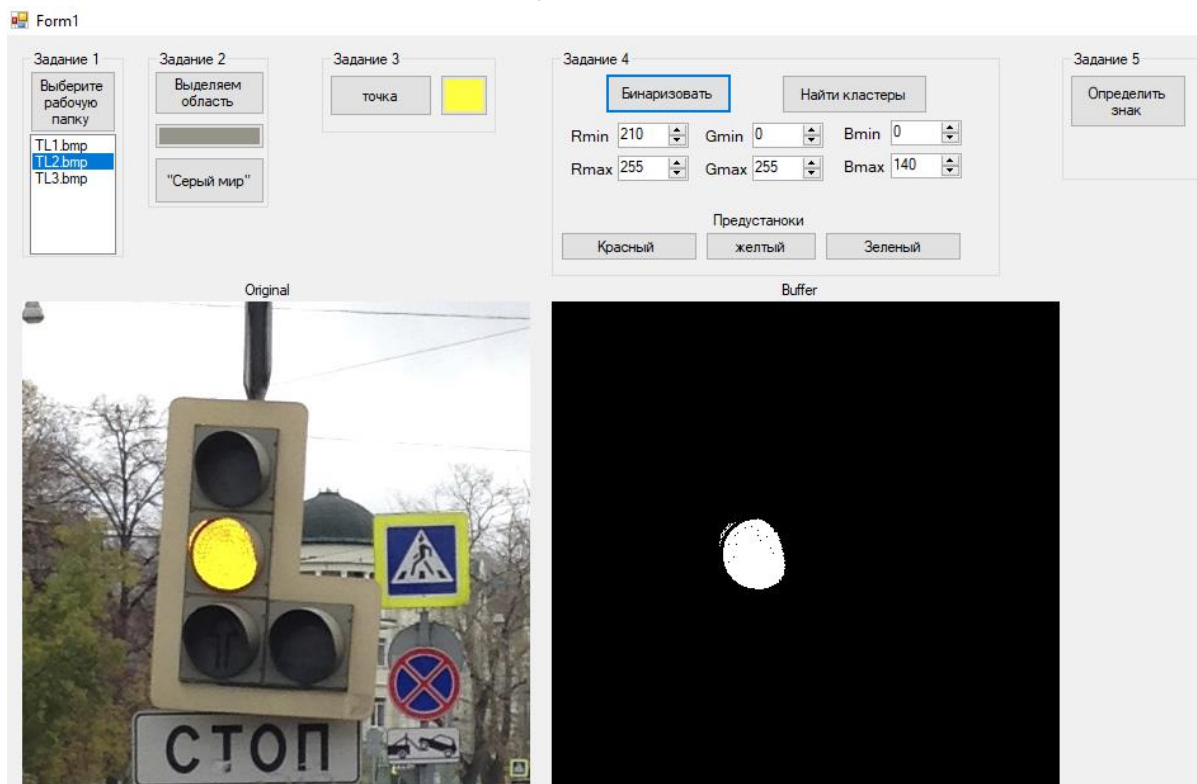
Выбираем рабочую папку и выбираем картинку



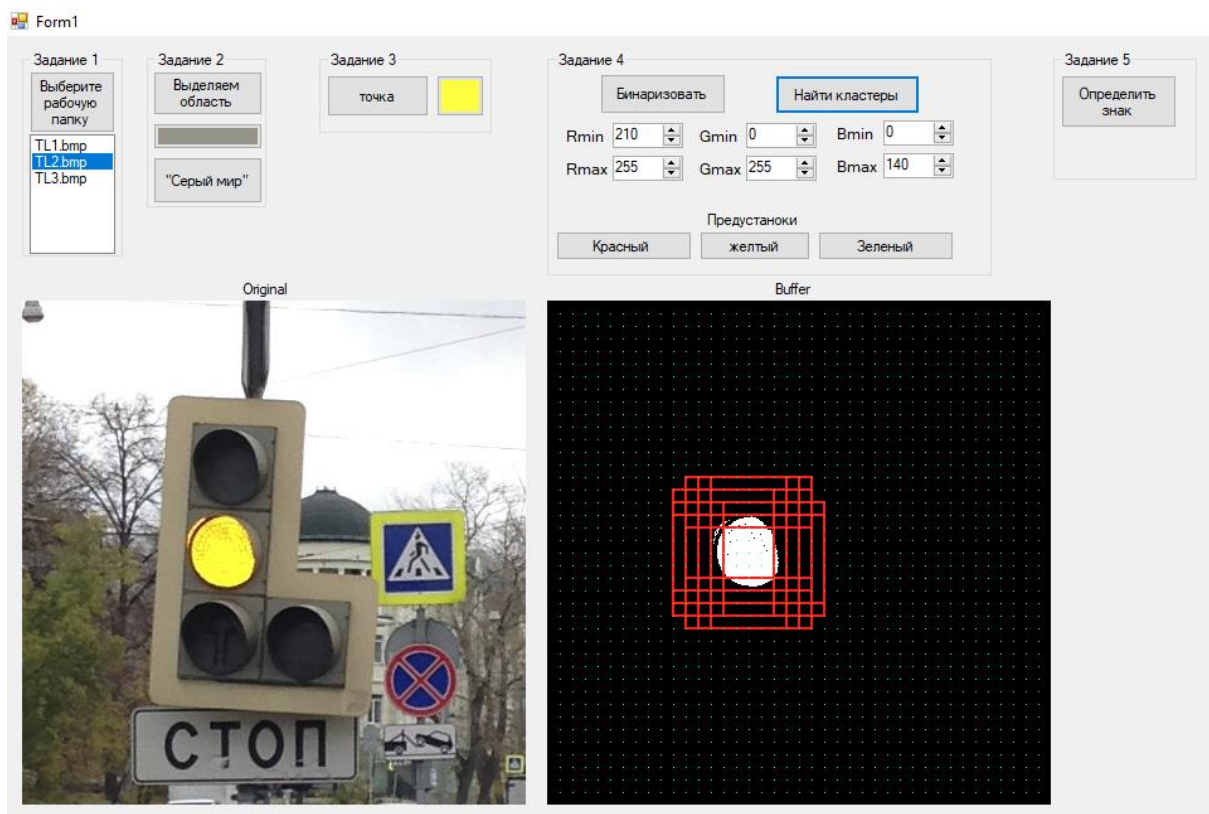
Выбираем область интереса и корректируем по методу "серый мир"



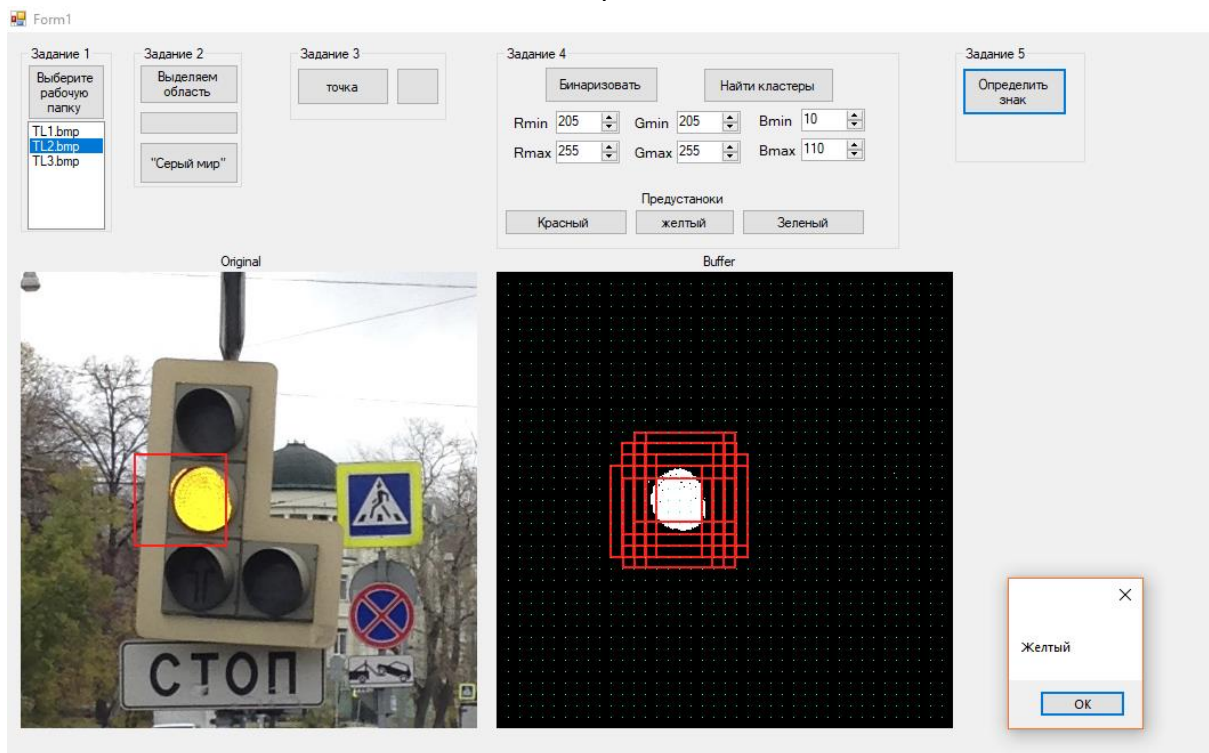
Считываем цвет по нажатию мышки



Бинаризация



Кластеризация



Детектирование

Код программы

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.IO;
using System.Media;

namespace lab2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            groupBox2.Visible = false;
            groupBox3.Visible = false;
            groupBox4.Visible = false;
            groupBox5.Visible = false;
        }

        Bitmap buffer;
        Graphics paint;
        Graphics rectangle;

        //ВЫБОР ПАПКИ
        private void button1_Click_1(object sender, EventArgs e)
        {
            FolderBrowserDialog FDB = new FolderBrowserDialog();
            FDB.SelectedPath =
@"C:\Users\desgr\source\repos\lab2\TrafficLights\";
            if (FDB.ShowDialog() == DialogResult.OK) //если нажали ок то
                выбрали папку
            {

```

```

        listBox1.Items.Clear();
        string[] files =
Directory.GetFiles(FDB.SelectedPath); //массив из файлов
        foreach (string file in files)
        {
            listBox1.Items.Add(Path.GetFileName(file));
        }
    }

    //ВЫБОР СВЕТОФОРА И ЗАЛИВАНИЕ ЕГО НА picture box
    private void listBox1_DoubleClick_1(object sender, EventArgs e)
    {
        groupBox2.Visible = true;
        groupBox3.Visible = true;
        groupBox4.Visible = true;
        groupBox5.Visible = true;
        label1.Visible = true;
        label2.Visible = true;
        try
        {
            if (listBox1.SelectedIndex == 0)
            {
                pictureBox1.Invalidate();

                pictureBox1.Load(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights\TL1.bmp")); // display picture
                buffer = new
                Bitmap(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights\TL1.bmp")); // buffer object
                pictureBox2.Image = buffer;
            }
            if (listBox1.SelectedIndex == 1)
            {
                pictureBox1.Invalidate();

                pictureBox1.Load(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights\TL2.bmp")); // display picture
                buffer = new
                Bitmap(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights\TL2.bmp")); // buffer object
                pictureBox2.Image = buffer;
            }
            if (listBox1.SelectedIndex == 2)
            {
                pictureBox1.Invalidate();

```

```

pictureBox1.Load(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights\TL3.bmp")); // display picture
        buffer = new
Bitmap(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights\TL3.bmp")); // buffer object
        pictureBox2.Image = buffer;
    }
}
catch
{
    MessageBox.Show("ERROR", "ERROR", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
}

//ПЕРЕКЛЮЧАТЕЛЬ ТОЧКИ И ОБЛАСТИ
bool Mode = true;
private void button2_Click(object sender, EventArgs e)
{
    if (Mode == true)
    {
        Mode = true;
    }
    else if (Mode == false) { Mode = true; }
}

private void button5_Click(object sender, EventArgs e)
{
    if (Mode == false)
    {
        Mode = false;
    }
    else if (Mode == true) { Mode = false; }
}

int X1, X2, Y1, Y2 = 0;
int r, g, b = 0;
private void pictureBox1_MouseDown(object sender, MouseEventArgs
e)
{
    X1 = e.X;
    Y1 = e.Y;
    pictureBox1.Invalidate();
}

```



```

//ОТРИСОВКА ПРЯМОУГОЛЬНИКА + ОПР. ЦВЕТА
int width, height = 0;
double S;
private void pictureBox1_MouseUp(object sender, MouseEventArgs
e)
{
    if (Mode == false)
    {
        buffer = new
Bitmap(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights
\" + listBox1.SelectedItem));
        paint = Graphics.FromImage(buffer);

        X2 = e.X;
        Y2 = e.Y;

        int X;
        int Y;

        height = Math.Abs(Y2 - Y1); //берем за высоту и ширину
разность между точками нажатия и отпущения мыши
        width = Math.Abs(X2 - X1);

        if (X2 > X1)
            X = X1;
        else
            X = X1 - width;

        if (Y2 > Y1)
            Y = Y1;
        else
            Y = Y1 - height;

        double k = 0;

        //считываем ср. цвет в прямоугольнике
        for (int i = X; i <= X + width; i++)
        {
            for (int j = Y; j <= Y + height; j++)
            {
                r += buffer.GetPixel(i, j).R;
                g += buffer.GetPixel(i, j).G;
                b += buffer.GetPixel(i, j).B;
                k++;
            }
        }
    }
}

```

```

    }
    }
    paint.DrawRectangle(new Pen(Color.Blue, 3), new
Rectangle(X, Y, width, height));
    pictureBox1.Image = buffer;

    r = (int)Math.Floor((double)r / k);
    g = (int)Math.Floor((double)g / k);
    b = (int)Math.Floor((double)b / k);
    button6.BackColor = Color.FromArgb(r, g, b);
    S = (r + g + b) / 3;

}
else
{
    //Считывание цвета пикселя и отображение
    Graphics gr = pictureBox1.CreateGraphics();
    b = buffer.GetPixel(e.X, e.Y).B;
    g = buffer.GetPixel(e.X, e.Y).G;
    r = buffer.GetPixel(e.X, e.Y).R;
    gr.DrawString("R: " + r + "\r\n" + "G: " + g + "\r\n " +
"B: " + " " + b,
        new Font("Roboto", 12),
        new SolidBrush(Color.Red),
        new PointF(e.X + 8, e.Y - 14));
    button3.BackColor = Color.FromArgb(buffer.GetPixel(e.X,
e.Y).R, buffer.GetPixel(e.X, e.Y).G, buffer.GetPixel(e.X, e.Y).B);
}
}

//СЕРЫЙ МИР
private void button4_Click(object sender, EventArgs e)
{
    double coefB = 0, coefG = 0, coefR = 0;
    Bitmap gray = new
Bitmap(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights
\" + listBox1.SelectedItem));

    //коэффициент серого мира
    coefR = S / r;
    coefG = S / g;
    coefB = S / b;

```

```

        for (int i = 0; i < 400; i++)    //прибавляем ко всем точкам
коэф. серого мира
        {
            for (int j = 0; j < 400; j++)
            {
                if (gray != null)
                {

                    int Rg = gray.GetPixel(i, j).R;
                    int Gg = gray.GetPixel(i, j).G;
                    int Bg = gray.GetPixel(i, j).B;

                    double corR_test = Math.Round(Rg * coefR);
                    int corR = (int)corR_test;
                    if (corR < 0) { corR = 0; }
                    else if (corR > 255) { corR = 255; }

                    double corG_test = Math.Round(Gg * coefG);
                    int corG = (int)corG_test;
                    if (corG < 0) { corG = 0; }
                    else if (corG > 255) { corG = 255; }

                    double corB_test = Math.Round(Bg * coefB);
                    int corB = (int)corB_test;
                    if (corB < 0) { corB = 0; }
                    else if (corB > 255) { corB = 255; }
                    gray.SetPixel(i, j, Color.FromArgb(corR, corG,
corB));
                }
            }
        }
        pictureBox2.Image = gray;
        buffer = gray;
    }

    //предустановки
    private void button9_Click(object sender, EventArgs e) {
Bmin.Value = 0; Bmax.Value = 140; Gmin.Value = 0; Gmax.Value = 255;
Rmin.Value = 210; Rmax.Value = 255; }
    private void button10_Click(object sender, EventArgs e) {
Bmin.Value = 80; Bmax.Value = 255; Gmin.Value = 160; Gmax.Value = 240;
Rmin.Value = 70; Rmax.Value = 150; }
    private void button8_Click(object sender, EventArgs e) {
Bmin.Value = 0; Bmax.Value = 115; Gmin.Value = 0; Gmax.Value = 160;
Rmin.Value = 155; Rmax.Value = 255; }
    //БИНАРИЗАЦИЯ

```

```

Graphics cl_paint;
private void button7_Click(object sender, EventArgs e)
{
    Bitmap binar = new
Bitmap(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights
\" + listBox1.SelectedItem));
    double k = 0;
    for (int i = 0; i < 400; i++)
    {
        for (int j = 0; j < 400; j++)
        {
            int r= binar.GetPixel(i, j).R; //прогоняем все точки
и берем у них цвет
            int g = binar.GetPixel(i, j).G;
            int b = binar.GetPixel(i, j).B;
            //если цвет попал в диапазон, то делаем белым, если
не попал, то черным
            if (!((r <= Rmax.Value) && (r >= Rmin.Value) && (g
<= Gmax.Value) && (g >= Gmin.Value) && (b <= Bmax.Value) && (b >=
Bmin.Value)))
                binar.SetPixel(i, j, Color.FromArgb(0, 0, 0));
            else
            {
                binar.SetPixel(i, j, Color.FromArgb(255, 255,
255));
                k++;
            }
        }
    }
    //если мы уже бинаризовали, то не даём больше
    if (k != 0)
        pictureBox2.Image = binar;
        buffer = binar;
        cl_paint=Graphics.FromImage(buffer);
}

int r_rect, i_rect, j_rect;
bool find = false;
bool fromDetector = false;
List<List<int>> clustersArray = new List<List<int>>();
//КЛАСТЕРИЗАЦИЯ
private void button11_Click(object sender, EventArgs e)
{
    int k = -1;

```

```

for (int i = 10; i <= 390; i += 10)
{
    for (int j = 10; j <= 390; j += 10)
    {
        if (buffer.GetPixel(i, j).R != 0)
        {
            //MessageBox.Show("!");

            double div = 1;
            int r = 10;

            double white = 0;
            double black = 0;

            while ((div > 0.3) && (r < 40))
            {
                //MessageBox.Show("!");

                white = 0;
                black = 0;
                r++;
                try
                {
                    for (int m = i-r; m < i + r; m++)
                    {
                        for (int n = j-r; n < j + r; n++)
                        {
                            if (buffer.GetPixel(m, n).R !=
                                0)
                            {
                                white++;
                            }
                            else
                                black++;
                        }
                    }
                }
                catch
                {
                    MessageBox.Show("YOU HAVE TO BINARIZATE
A PICTURE; RESTART THE PROGRAM", "ERROR", MessageBoxButtons.OK,
                    MessageBoxIcon.Information);
                }

                div = white / (white + black);
            }
        }
    }
}

```

```

        k++;

        clustersArray.Add(new List<int>());
        clustersArray[k].Add(i);
        clustersArray[k].Add(j);
        clustersArray[k].Add(r);
    }
}
//MessageBox.Show("Доходит до отрисовки");

for (int i = 0; i < clustersArray.Count; i++)
{
    if ((!fromDetector) && (clustersArray[i].Count != 0))
    {
        cl_paint.DrawRectangle(new Pen(Color.FromArgb(255,
30, 30), 2), clustersArray[i][0] - clustersArray[i][2],
clustersArray[i][1] - clustersArray[i][2], clustersArray[i][2] * 2,
clustersArray[i][2] * 2);
    }
    else
        if ((clustersArray[i].Count != 0) &&
(clustersArray[i][2] > 20) && (find == false) )
        {
            fromDetector = false;
            find = true;
            i_rect = clustersArray[i][0];
            j_rect = clustersArray[i][1];
            r_rect = clustersArray[i][2];

        }
    }

    for (int i = 10; i <= 390; i += 10)
        for (int j = 10; j <= 390; j += 10)
            if (!fromDetector)
                buffer.SetPixel(i, j, Color.FromArgb(0, 255,
130));

    pictureBox2.Image = buffer;
}

//ОПРЕДЕЛЕНИЕ ЗНАКА
private void button12_Click(object sender, EventArgs e)
{

```



```

        Bitmap pictureBox1 = new
Bitmap(Convert.ToString(@"C:\Users\desgr\source\repos\lab2\TrafficLights
\" + listBox1.SelectedItem));
        rectangle = Graphics.FromImage(pictureBox1);
        fromDetector = true;

        Rmax.Value = 255;
        Rmin.Value = 169;

        Gmax.Value = 154;
        Gmin.Value = 30;

        Bmax.Value = 125;
        Bmin.Value = 9;

        button7_Click(sender, e); //вызываем бинаризацию
        button11_Click(sender, e); //вызываем кластеризацию,

        if (find == true)
        {
            rectangle.DrawRectangle(new Pen(Color.FromArgb(255,
30, 30), 2), i_rect - r_rect, j_rect - r_rect, r_rect * 2, r_rect * 2);
            pictureBox1.Image = pictureBox1;
            MessageBox.Show("Красный");
        }

        if (fromDetector == true)
        {

            Rmax.Value = 255;
            Rmin.Value = 205;

            Gmax.Value = 255;
            Gmin.Value = 205;

            Bmax.Value = 110;
            Bmin.Value = 10;

            button7_Click(sender, e); //вызываем бинаризацию
            button11_Click(sender, e); //вызываем кластеризацию,

            if (find == true)
            {

```

```

        rectangle.DrawRectangle(new Pen(Color.FromArgb(255,
30, 30), 2), i_rect - r_rect, j_rect - r_rect, r_rect * 2, r_rect * 2);
        pictureBox1.Image = pictureBox1;
        MessageBox.Show("Желтый");
    }
}
if (fromDetector == true)
{
    Rmax.Value = 187;
    Rmin.Value = 87;

    Gmax.Value = 255;
    Gmin.Value = 191;

    Bmax.Value = 255;
    Bmin.Value = 152;

    button7_Click(sender, e); //вызываем бинаризацию
    button11_Click(sender, e); //вызываем кластеризацию,

    if (find == true)
    {
        rectangle.DrawRectangle(new Pen(Color.FromArgb(255,
30, 30), 2), i_rect - r_rect, j_rect - r_rect, r_rect * 2, r_rect * 2);
        pictureBox1.Image = pictureBox1;
        MessageBox.Show("Зеленый");
    }
}
}
}}
```