

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет информационных технологий

Кафедра «СМАРТ-технологии»

ЛАБОРАТОРНО-ПРАКТИЧЕСКОЕ ЗАДАНИЕ №2

*По дисциплине: «Нейронные сети в задачах технического
зрения и управления:*

Применение персептронов для распознавания изображений

Студент: Сайфудинов Роман Рамезович / _____ / группа 171-311

Преподаватель: Идиатуллов Тимур Тофикович / _____ /

Оценка _____ Дата: _____

МОСКВА – 2019

Цель работы:

Разработать алгоритм обработки зон интереса для классификации изображения на основе нейронной сети

Задачи:

- Подготовить приложение для генерации обучающей выборки из исходного набора изображений;
- Разработать класс, реализующий функциональность нейронной сети на основе персептрона с одним скрытым слоем;
- Реализовать функцию классификации произвольного изображения;
- Разработать функцию обучения с учителем для нейронной сети по подготовленной ранее обучающей выборке.

Ход работы

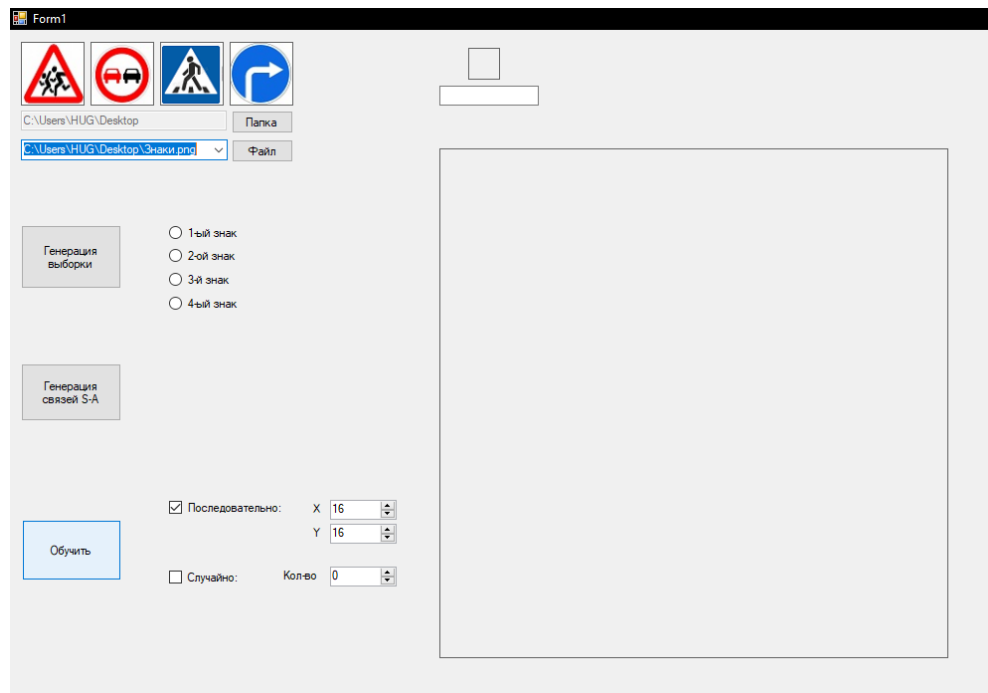


Рис1.открываем папку и файл с изображением знаков

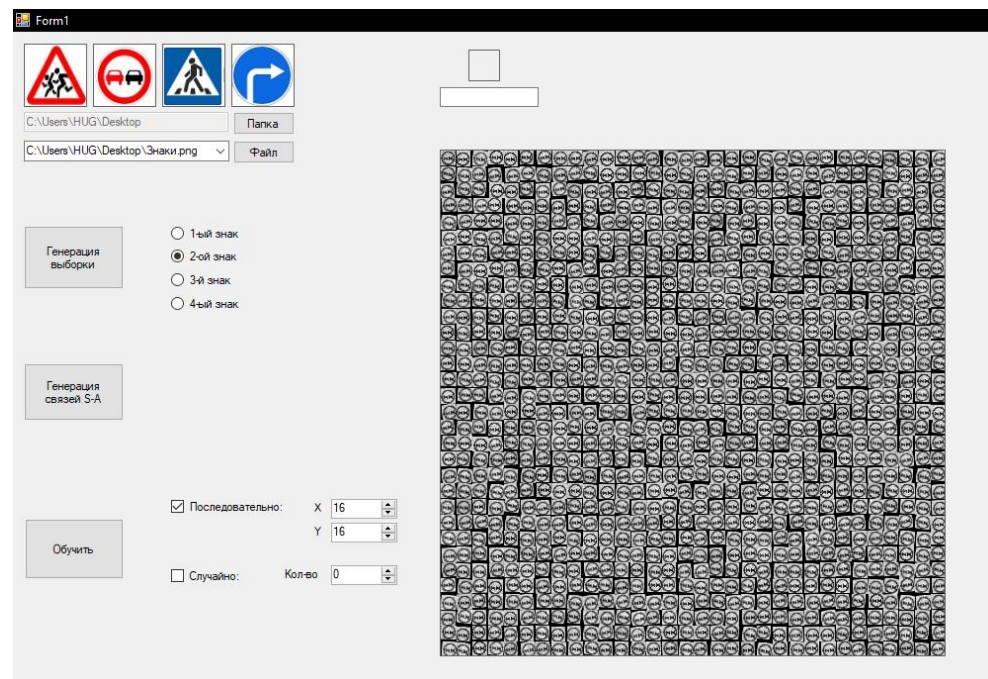


Рис2. Создаем обучающую выборку

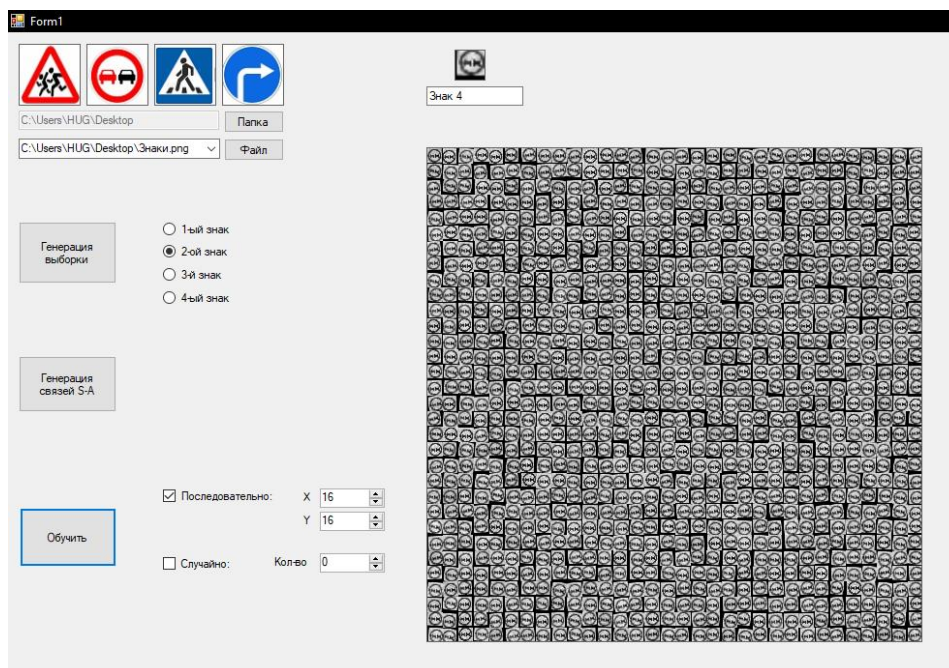


Рис3. Генерируем связь S-A(весы равно нулю, поэтому знак неправильно определился)

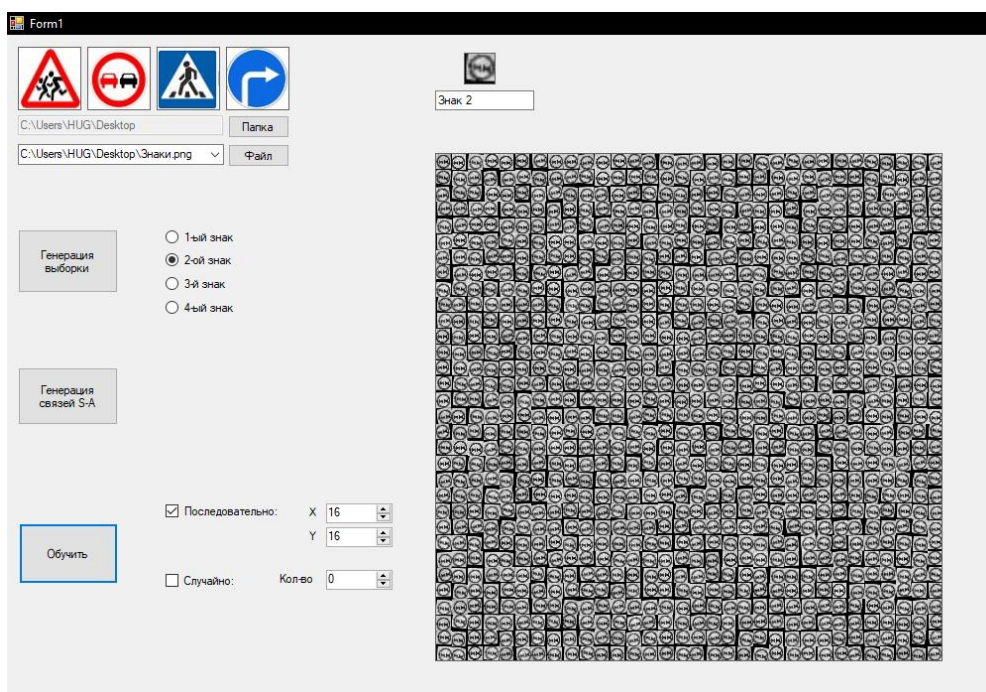


Рис4. Обучаем нейросеть последовательно и проверяем на знаке из выборки

Код:

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

using OpenCvSharp;
using OpenCvSharp.Extensions;

namespace WindowsFormsApp69
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        Bitmap bitmap1;
        Mat input_frame;
        Mat output_frame;
        Mat grid_frame;
        Bitmap[] bmp = new Bitmap[4];
        Bitmap[] bmp_view = new Bitmap[4];
        Bitmap[] bmp_grid = new Bitmap[4];
        string img_path;
        Mat tmp_rm;
        Random rd = new Random();
        int pv_s = 0;
        string path;
        Bitmap base_image;
        Bitmap base_image_view;
        Bitmap eject_image;
        Bitmap input_image;
        Graphics image_buffer;
        int toggle = -1;
        int[,] evaluate_table = new int[512, 16];
        int[,] address_table = new int[512, 3];
        int[,] mass_table = new int[512, 4];
        int[,] image_table = new int[16, 16];
        int point_x = 0;
        int point_y = 0;
        int[] img_mass = new int[4];
        int[] ns = new int[4];
        int cg = 0;
        int certain_symbol;
        int cert_value = 0;

        private void Form1_Load(object sender, EventArgs e)
        {
            checkBox1.Checked = true;
            numericUpDown1.Value = 16;
            numericUpDown2.Value = 16;
        }

        //ВЫБОР ПАПКИ
        private void folder_button_Click(object sender, EventArgs e)
        {
            FolderBrowserDialog fbd = new FolderBrowserDialog();
            fbd.Description = "Select a folder of base images:";
            fbd.ShowNewFolderButton = true;
            fbd.RootFolder = System.Environment.SpecialFolder.MyComputer;
```

```

        if (fbd.ShowDialog() == DialogResult.OK)
        {
            path = fbd.SelectedPath;
            folder_box.Text = path;
        }
    }

    //ВЫБОР ФАЙЛА
    private void open_button_Click(object sender, EventArgs e)
    {
        if (Directory.Exists(path))
        {
            files_list_box.DataSource = Directory.GetFiles(path);
            files_list_box.Enabled = true;
        }
        else
        {
            MessageBox.Show("No files in folder.", "Load", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
    }

    private void files_list_box_SelectedIndexChanged(object sender, EventArgs e)
    {
        try
        {
            img_path = Convert.ToString(files_list_box.SelectedItem); //Чтение файла
            по адресу

            Bitmap test_image = new Bitmap(img_path);
            if (test_image.Width == 256 && test_image.Height == 64)
            {
                base_image_view = new Bitmap(img_path); //Запись изображения в
                буферный объект
                base_image = new Bitmap(base_image_view.Clone() as Bitmap, 64, 16);
                opened_image_clear(); //Сброс изображения
            }
            else
            {
                sbx_0.BackgroundImage = null;
                sbx_1.BackgroundImage = null;
                sbx_2.BackgroundImage = null;
                sbx_3.BackgroundImage = null;
            }
        }
        catch
        {
            ;
        }
    }

    //ЗАГРУЗКА ЗНАКОВ И РАЗДЕЛЕНИЕ НА 4 picture box
    void opened_image_clear()
    {
        //Сброс изображения
        int pos_x;
        Rectangle pos_rect;
        for (int i = 0; i < 4; i++)
        {
            pos_x = i * 16;
            pos_rect = new Rectangle(pos_x, 0, 16, 16);
            eject_image = new Bitmap(16, 16);
            image_buffer = Graphics.FromImage(eject_image);
            image_buffer.DrawImage(base_image, 0, 0, pos_rect, GraphicsUnit.Pixel);
            bmp[i] = eject_image;
            pos_x = i * 64;
            pos_rect = new Rectangle(pos_x, 0, 64, 64);
            eject_image = new Bitmap(64, 64);
            image_buffer = Graphics.FromImage(eject_image);
            image_buffer.DrawImage(base_image_view, 0, 0, pos_rect,
                GraphicsUnit.Pixel);
        }
    }

```

```

        bmp_view[i] = eject_image;
    }

    sbbox_0.BackgroundImage = bmp_view[0];
    sbbox_1.BackgroundImage = bmp_view[1];
    sbbox_2.BackgroundImage = bmp_view[2];
    sbbox_3.BackgroundImage = bmp_view[3];
}

//СОЗДАНИЕ ВЫБОРКИ ИЗ ИЗМЕНЕННЫХ ЗНАКОВ
private void button2_Click(object sender, EventArgs e)
{
    input_frame = new Mat();
    for (int g = 0; g < 4; g++)
    {
        grid_frame = new Mat(new OpenCvSharp.Size(512, 512), MatType.CV_8U, new
Scalar(0, 0, 0));
        for (int i1 = 0; i1 < 32; i1++)
        {
            for (int i2 = 0; i2 < 32; i2++)
            {
                input_frame = bmp[g].ToMat();
                image_view();
                OpenCvSharp.Rect copy_rect = new OpenCvSharp.Rect(0, 0, 16, 16);
                OpenCvSharp.Rect paste_rect = new OpenCvSharp.Rect(i1 * 16, i2 *
16, 16, 16);

                output_frame.SubMat(copy_rect).CopyTo(grid_frame.SubMat(paste_rect));
            }
            bmp_grid[g] = BitmapConverter.ToBitmap(grid_frame,
PixelFormat.Format8bppIndexed);
        }

        picture_view_selector();
    }
    //ИЗМЕНЕНИЕ ИЗОБРАЖЕНИЯ ДЛЯ ВЫБОРКИ
    void image_view()
    {
        output_frame = new Mat();
        image_brightness();
        input_frame = output_frame;
        image_noise();
        input_frame = output_frame;
        image_rotate();
        input_frame = output_frame;
        image_shift();
    }
    void image_brightness()
    {
        input_frame.ConvertTo(output_frame, -1, 1, rd.Next(-50, 51));
        pictureBox2.BackgroundImage = BitmapConverter.ToBitmap(output_frame);
    }
    void image_noise()
    {
        Cv2.CvtColor(input_frame, input_frame, ColorConversionCodes.RGB2GRAY);
        Mat noise = new Mat(input_frame.Size(), MatType.CV_64F);
        Mat result = new Mat();
        Cv2.Normalize(input_frame, result, 0.0, 1.0, NormTypes.MinMax,
MatType.CV_64F);
        Cv2.Randn(noise, 0, 0.1);
        result = result + noise;
        Cv2.Normalize(result, result, 0.0, 1.0, NormTypes.MinMax, MatType.CV_64F);
        result.ConvertTo(output_frame, MatType.CV_8U, 255, 0);
    }
    void image_rotate()
    {
        tmp_rm = Cv2.GetRotationMatrix2D(new Point2f(8, 8), rd.Next(-15, 15), 1);
        Cv2.WarpAffine(input_frame, output_frame, tmp_rm, new OpenCvSharp.Size(16,
16), InterpolationFlags.Linear);
    }
}

```

```

void image_shift()
{
    int s_w = rd.Next(-2, 2);
    int s_h = rd.Next(-2, 2);
    output_frame = new Mat(input_frame.Size(), input_frame.Type(), new Scalar(0,
0, 0));
    OpenCvSharp.Rect source_rect = new OpenCvSharp.Rect(Math.Max(-s_w, 0),
Math.Max(-s_h, 0), 16 - Math.Abs(s_w), 16 - Math.Abs(s_h));
    OpenCvSharp.Rect dest_rect = new OpenCvSharp.Rect(Math.Max(s_w, 0),
Math.Max(s_h, 0), 16 - Math.Abs(s_w), 16 - Math.Abs(s_h));
    input_frame.SubMat(source_rect).CopyTo(output_frame.SubMat(dest_rect));
}

//ПРОСМОТР ВЫБОРКИ
void picture_view_selector()
{
    pictureBox2.BackgroundImage = bmp_grid[pv_s];
}
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    pv_s = 0;
    picture_view_selector();
}
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    pv_s = 1;
    picture_view_selector();
}
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    pv_s = 2;
    picture_view_selector();
}
private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
    pv_s = 3;
    picture_view_selector();
}

//ГЕНЕРИРУЕМ S-A СВЯЗИ(АДРЕСА И ВЕС)
private void button3_Click(object sender, EventArgs e)
{
    for (int i1 = 0; i1 < 6; i1++)
    {
        if (toggle < 0)
        {
            for (int i2 = 0; i2 < 512; i2++)
            {
                evaluate_table[i2, i1] = rd.Next(0, 255);
            }
        }
        else
        {
            for (int i2 = 0; i2 < 512; i2++)
            {
                evaluate_table[i2, i1] = rd.Next(-1, 2);
            }
        }
        toggle *= -1;
    }
}

//СОХРАНЕНИЕ И ЗАГРУЗКА АДРЕСОВ И ВЕСОВ
void save_address()
{
    for (int i1 = 0; i1 < 6; i1 += 2)
    {
        for (int i2 = 0; i2 < 512; i2++)
        {
            address_table[i2, (i1 / 2)] = evaluate_table[i2, i1];
        }
    }
}

```



```

    }
}
void load_adress()
{
    for (int i1 = 0; i1 < 3; i1++)
    {
        for (int i2 = 0; i2 < 512; i2++)
        {
            evaluate_table[i2, (i1 * 2)] = address_table[i2, i1];
        }
    }
}

//ВЫТАСКИВАЕМ ИЗОБРАЖЕНИЕ ИЗ ВЫБОРКИ И СЧИТАЕМ ЕГО ВЕСА
void image_import()
{
    save_address();

    OpenCvSharp.Rect copy_rect = new OpenCvSharp.Rect(point_x, point_y, 16, 16);
    OpenCvSharp.Rect paste_rect = new OpenCvSharp.Rect(0, 0, 16, 16);
    Mat copy_frame = new Mat(new OpenCvSharp.Size(512, 512), MatType.CV_8U);
    Mat paste_frame = new Mat(new OpenCvSharp.Size(16, 16), MatType.CV_8U);
    copy_frame = bmp_grid[cg].ToMat();
    copy_frame.SubMat(copy_rect).CopyTo(paste_frame.SubMat(paste_rect));

    for (int i1 = 0; i1 < 512; i1++)
    {
        for (int i2 = 6; i2 < 8; i2++)
        {
            evaluate_table[i1, i2] = 0;
        }
    }

    input_image = BitmapConverter.ToBitmap(paste_frame);

    int a = 0;
    for (int y = 0; y < 16; y++)
    {
        for (int x = 0; x < 16; x++)
        {
            for (int i1 = 0; i1 < 6; i1 += 2)
            {
                for (int i2 = 0; i2 < 512; i2++)
                {
                    if (evaluate_table[i2, i1] == a)
                    {
                        evaluate_table[i2, i1] = -(255 - input_image.GetPixel(x,
y).G);
                    }
                }
            }
            image_table[x, y] = 255 - input_image.GetPixel(x, y).G;
            a++;
        }
    }

    for (int i1 = 0; i1 < 6; i1 += 2)
    {
        for (int i2 = 0; i2 < 512; i2++)
        {
            evaluate_table[i2, i1] = Math.Abs(evaluate_table[i2, i1]);
        }
    }

    for (int i = 0; i < 512; i++)
    {
        evaluate_table[i, 6] = evaluate_table[i, 0] * evaluate_table[i, 1] +
            evaluate_table[i, 2] * evaluate_table[i, 3] +
            evaluate_table[i, 4] * evaluate_table[i, 5];
    }
}

```

```

        for (int i = 0; i < 512; i++)
        {
            (((evaluate_table[i, 6] >= 2) && (evaluate_table[i, 1] +
evaluate_table[i, 3] + evaluate_table[i, 5] > 2)) || ((evaluate_table[i, 6] >= 1) &&
(evaluate_table[i, 1] + evaluate_table[i, 3] + evaluate_table[i, 5] <= 2)))
            (((evaluate_table[i, 6] >= 3) && (evaluate_table[i, 1] +
evaluate_table[i, 3] + evaluate_table[i, 5] == 3)) || ((evaluate_table[i, 6] < 3) &&
(evaluate_table[i, 1] + evaluate_table[i, 3] + evaluate_table[i, 5] < 3)))
            if (evaluate_table[i, 6] >= 1)
            {
                evaluate_table[i, 7] = 1;
            }
            if (evaluate_table[i, 6] <= -1)
            {
                evaluate_table[i, 7] = -1;
            }
        }

        load_adress();
    }

    //СЧИТАЕМ ВЕС ИЗОБРАЖЕНИЯ
    void mass_counter()
    {
        for (int i1 = 0; i1 < 512; i1++)
        {
            for (int i2 = 9; i2 < 16; i2 += 2)
            {
                evaluate_table[i1, i2] = evaluate_table[i1, 7] * evaluate_table[i1,
(i2 - 1)];
            }
        }

        for (int i1 = 0; i1 < 512; i1++)
        {
            for (int i2 = 9; i2 < 16; i2 += 2)
            {
                img_mass[(i2 - 9) / 2] += evaluate_table[i1, i2];
            }
        }
    }

    //ВЫБОР ИЗОБРАЖЕНИЯ ПО КЛИКУ
    private void pictureBox2_MouseDown(object sender, MouseEventArgs e)
    {
        point_x = e.Location.X;
        point_y = e.Location.Y;

        if ((point_x > -1 && point_x < 512) && (point_y > -1 && point_y < 512))
        {
            cg = pv_s;
            point_x /= 16;
            point_y /= 16;
            point_x *= 16;
            point_y *= 16;
            image_import();
            mass_counter();
            int max_mass = img_mass.Max();
            certain_symbol = Array.IndexOf(img_mass, max_mass);
            textBox5.Text = string.Format("Знак {0}", certain_symbol + 1);
            pictureBox3.BackgroundImage = input_image;
            img_mass = new int[4];
        }
    }

    //ОБУЧЕНИЕ НЕЙРОСЕТИ
    private void button4_Click(object sender, EventArgs e)
    {
        if (checkBox1.Checked == true)
        {
            int baypass_max_x = Convert.ToInt32(numericUpDown1.Value) * 16;

```

```

int baypass_max_y = Convert.ToInt32(numericUpDown2.Value) * 16;
for (int cs = 0; cs < 4; cs++)
{
    cert_value = cs;
    cg = cs;
    for (int i1 = 0; i1 < baypass_max_x; i1 += 16)
    {
        for (int i2 = 0; i2 < baypass_max_y; i2 += 16)
        {
            point_x = i1;
            point_y = i2;
            image_import();
            evaluate_teach();
            img_mass = new int[4];
        }
    }
}
else
{
    int baypass_count = Convert.ToInt32(numericUpDown3.Value);
    for (int i = 0; i < baypass_count; i++)
    {
        point_x = rd.Next(0, 32) * 16;
        point_y = rd.Next(0, 32) * 16;
        cert_value = rd.Next(0, 4);
        cg = cert_value;
        image_import();
        evaluate_teach();
        img_mass = new int[4];
    }
}
}
void evaluate_teach()
{
    int n = (cert_value * 2) + 8;
    for (int i = 0; i < 512; i++)
    {
        if (evaluate_table[i, 7] == 1)
        {
            evaluate_table[i, n]++;
        }
        if (evaluate_table[i, 7] == -1)
        {
            evaluate_table[i, n]--;
        }
    }
}
}
}

```