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 №1

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

Базовые принципы применения нейронных сетей для обработки изображений.

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

Преподаватель: Идиатуллов Тимур Тофикович/_____/

Оценка _____ Дата: _____

МОСКВА – 2019

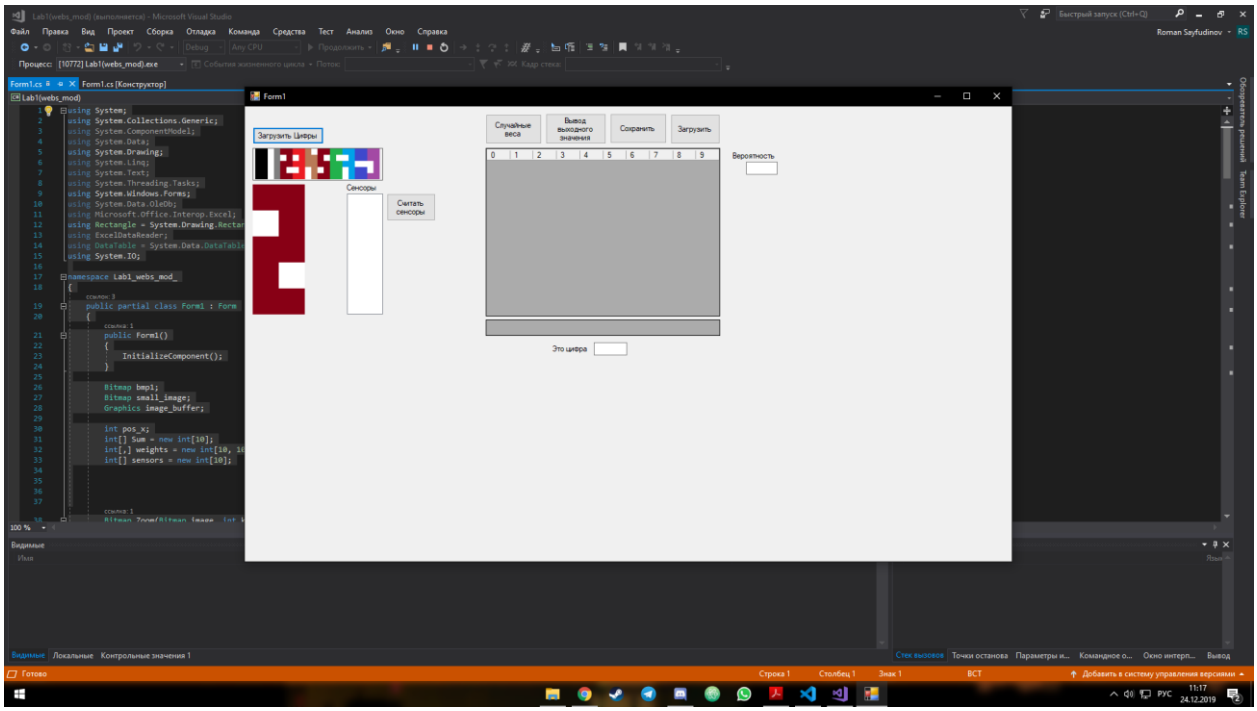
Цель работы:

Разработать алгоритм обработки изображения с использованием простого однослойного перцептрона без обучения.

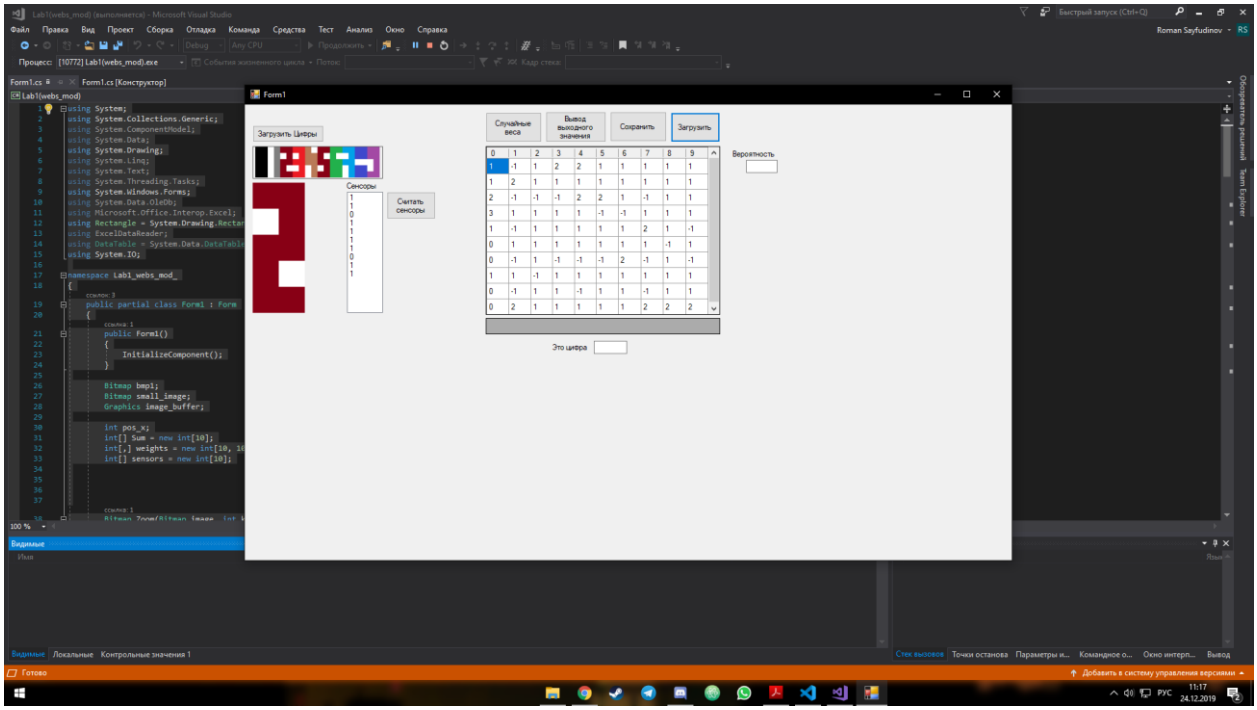
Задачи:

1. Подготовить файл с массивом цифр 0-9 в виде изображения 2×5 пикселей. Итоговое изображение 20×5 в графическом редакторе в формате bmp (bitmap) в элемент picturebox. Реализовать функцию увеличенного показа изображения символа (цифры) в отдельном picturebox 80×200 по клику на символе. Реализовать функцию вывода параметров пикселей в виде текстового списка значений (По красному каналу).
2. Реализовать функцию расчёта результатов распознавания изображения на базе однослойной нейронной сети с 10 рецепторами и 10 нейронами с линейной функцией активации. Хранение параметров реализовать через массив 10×12 элементов (10 весов на каждый нейрон + накопленный сигнал в нейроне + коэффициент функции активации). Реализовать функцию загрузки параметров из файла формата CSV (для удобства лучше хранить параметры в виде целых чисел [параметр×100]). Также реализовать функцию выгрузки параметров сети в файл формата CSV. При загрузке и выгрузке должна обеспечиваться возможность выбора файла через диалоговые окна.
3. Реализовать функцию вычисления выходных и выходных значений и вывода их в виде результирующих значений с выдачей результатов классификации входного объекта. Должно выводиться сообщение, с какой вероятностью (весом) объект относится к одному из 10 заданных классов – цифр.
4. Вручную подобрать параметры нейронной сети для уверенного распознавания символов. Ввести данные значения в качестве параметров нейронной сети, загружаемой из CSV файла.
5. Реализовать функцию обработки произвольного изображения 20×5, загружаемого из файла (с выбором из диалогового окна). Файлы необходимо подготовить заранее.

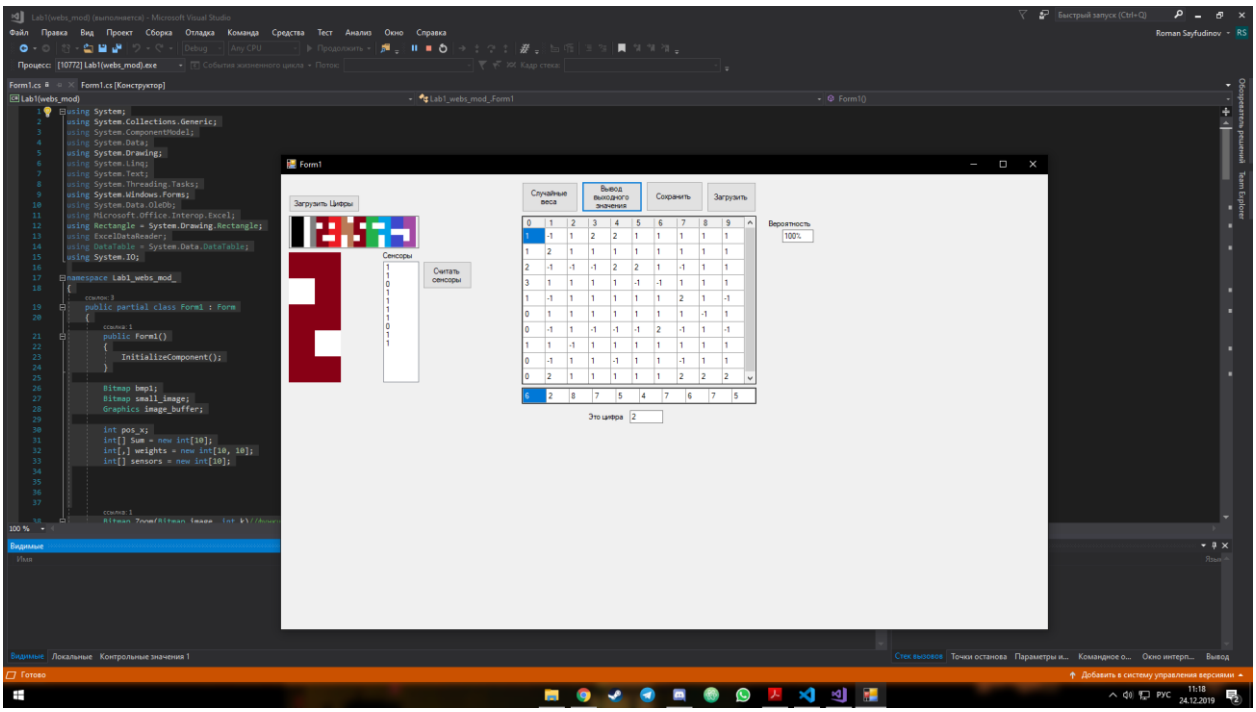
Работа Программы:



Выбор цифры



Определение сенсоров и вывод весов



Определение цифры

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.Data.OleDb;
using Microsoft.Office.Interop.Excel;
using Rectangle = System.Drawing.Rectangle;
using ExcelDataReader;
using DataTable = System.Data.DataTable;
using System.IO;

namespace Lab1_webs_mod_
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      Bitmap bmp1;
      Bitmap small_image;
      Graphics image_buffer;

      int pos_x;
      int[] Sum = new int[10];
      int[,] weights = new int[10, 10];
      int[] sensors = new int[10];




      Bitmap Zoom(Bitmap image, int k)//функция увеличения области битмапа
      {
         if (k <= 1)
            return image;
         Bitmap img = new Bitmap(image);
         int width = img.Width;
         int height = img.Height;
         Bitmap zoomImg = new Bitmap(width * k, height * k);
         Graphics g = Graphics.FromImage(zoomImg);

         for (int i = 0; i < width; i++)
```

```
            for (int j = 0; j < height; j++)
            {
                Color color = img.GetPixel(i, j);
                g.FillRectangle(new SolidBrush(color), i * k, j * k, k, k);//заполненные прямоугольники
            }

        return zoomImg;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        OpenFileDialog ofd = new OpenFileDialog();
        if (ofd.ShowDialog() == DialogResult.OK)
        {

            bmp1 = new Bitmap(ofd.FileName);

        }
        pictureBox1.Image = bmp1;
    }

    //УВЕЛИЧЕННЫЙ ПОКАЗ ЦИФРЫ
    private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
    {
        int pointx;

        pos_x = e.Location.X;
        if (pos_x > -1 && pos_x <200)
        {
            pointx = ((pos_x /20) *20);
            Rectangle pos_rect = new Rectangle(pointx, 0, 20, 50);
            small_image = new Bitmap(20, 50);
            image_buffer = Graphics.FromImage(small_image);
            image_buffer.DrawImage(bmp1, 0, 0, pos_rect, GraphicsUnit.Pixel);//заносим обрезанное
изобр в битмап
            pictureBox2.Image = Zoom(small_image,4);//увеличиваем битмап в 20 раз
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        int k = 0;
        List<int> PixelValue = new List<int>();
        listBox1.Items.Clear();
        for (int y = 0; y < 50; y+=10)
        {
            for (int x = 0; x < 20; x+=10)
            {
                Color color = small_image.GetPixel(x, y);
                if (color.R == 255 || color.G == 255 || color.B == 255)
```

```
            {
                listBox1.Items.Add(0);
                PixelValue.Add(0);
                sensors[k] = 0;
            }
            else
            {
                listBox1.Items.Add(1);
                PixelValue.Add(1);
                sensors[k] = 1;
            }
            k++;


        }

    }
}

private void button3_Click(object sender, EventArgs e)
{
    Random random = new Random();
    dataGridView1.Rows.Clear();
    for (int col = 0; col < 10; col++)
    {
        for (int line = 0; line < 10; line++)
        {
            weights[col, line] = random.Next(-1, 1);
        }
        dataGridView1.Rows.Add(weights[col, 0], weights[col, 1], weights[col, 2], weights[col, 3],
weights[col, 4], weights[col, 5],
            weights[col, 6], weights[col, 7], weights[col, 8], weights[col, 9]);
    }
}

private void button4_Click(object sender, EventArgs e)
{
    StreamReader sr;
    OpenFileDialog datagrid = new OpenFileDialog();
    if (datagrid.ShowDialog() == DialogResult.OK)
    {
        sr = new StreamReader(datagrid.FileName);
        using (sr)
        {
            dataGridView1.Rows.Clear();
            for (int col = 0; col < 10; col++)
            {
                string data = sr.ReadLine();
                var nums = data.Split(';');
                for (int line = 0; line < 10; line++)
```

```
                {
                    weights[col, line] = Convert.ToInt32(nums[line]);
                }
                dataGridView1.Rows.Add(weights[col, 0], weights[col, 1], weights[col, 2], weights[col, 3],
weights[col, 4], weights[col, 5],
                    weights[col, 6], weights[col, 7], weights[col, 8], weights[col, 9]);
            }

            sr.Close();
        }
    }
}

    private void button5_Click(object sender, EventArgs e)
    {
        using (StreamWriter outfile = new
StreamWriter(@"C:\Users\desgr\source\repos\Lab1(webs_mod)\Random.csv"))
        {
            for (int col = 0; col < 10; col++)
            {
                for (int line = 0; line < 10; line++)
                {
                    outfile.Write(dataGridView1.Rows[col].Cells[line].Value);
                    if (line < dataGridView1.ColumnCount - 1)
                        outfile.Write(";");
                }
                outfile.WriteLine();
            }
            outfile.Close();
        }
    }

    private void button6_Click(object sender, EventArgs e)
    {
        for (int col = 0; col < 10; col++)
        {
            for (int line = 0; line < 10; line++)
            {
                try
                {
                    weights[col,line] = Convert.ToInt32(dataGridView1.Rows[col].Cells[line].Value);
                }
                catch { MessageBox.Show("Input data grid", "Load", MessageBoxButtons.OK,
MessageBoxIcon.Information); }
            }
        }
        for (int col = 0; col < 10; col++)
        {
            int ev_sum = 0;
            for (int line = 0; line < 10; line++)
```

```
            {

                ev_sum += weights[line,col] * sensors[line];
            }
            Sum[col] = ev_sum;

        }
        dataGridView2.Rows.Clear();
        dataGridView2.Rows.Add(Sum[0],Sum[1], Sum[2], Sum[3], Sum[4], Sum[5], Sum[6],
            Sum[7], Sum[8], Sum[9]);
       // dataGridView2.Rows.Add(Sum[0] / 9, Sum[1] / 7, Sum[2] / 8, Sum[3] / 9, Sum[4] / 9, Sum[5] / 9,
Sum[6] / 10,
       //    Sum[7] / 9, Sum[8] / 9, Sum[9] / 9);
        int big_value = -100;
        int probable_value = 0;
        int k = 0;
        int l = 0;
        for (int i = 0; i<10;i++)
        {
          if(Sum[i]>big_value)
          {
              big_value = Sum[i];
              probable_value = i;
          }
        }
        textBox1.Text = probable_value.ToString();

        for (int i = 0; i < 10; i++)
        {
           if (big_value==Sum[i])
           {
              k++;
           }
        }
        l = 100 / k;
        textBox2.Text = l.ToString() + "%";
     }
  }
}
```