

Название работы: Знакомство с OpenCV

Цель работы: Целью данной работы является проверка остаточных навыков программирования на языке высокого

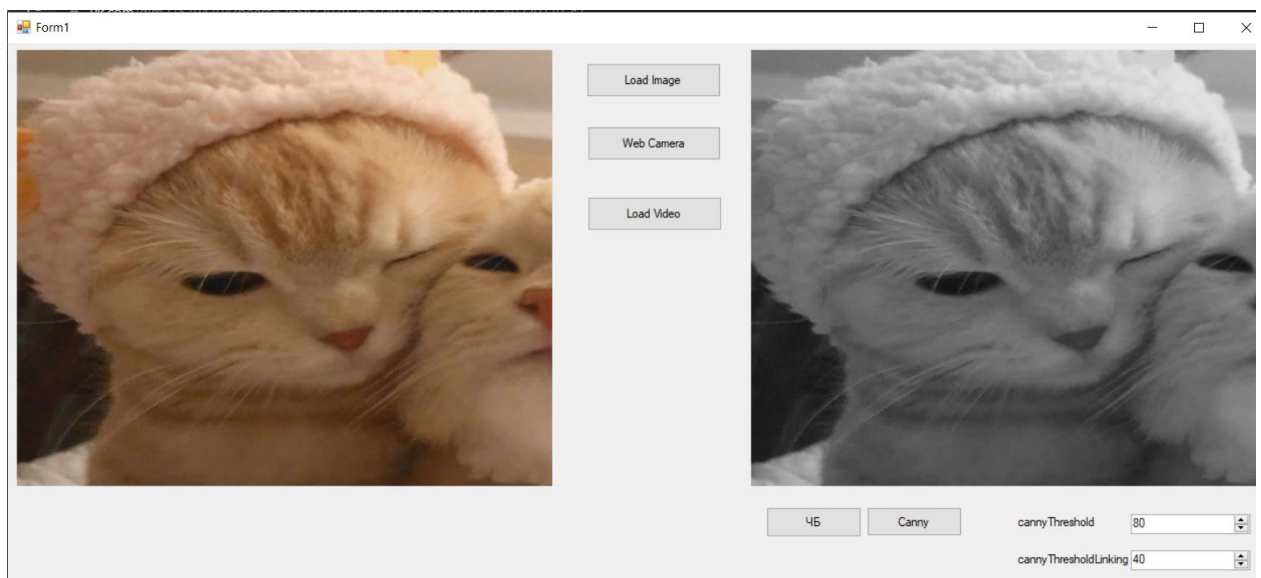
уровня C#, а также навыков использования среды разработки Microsoft Visual Studio.

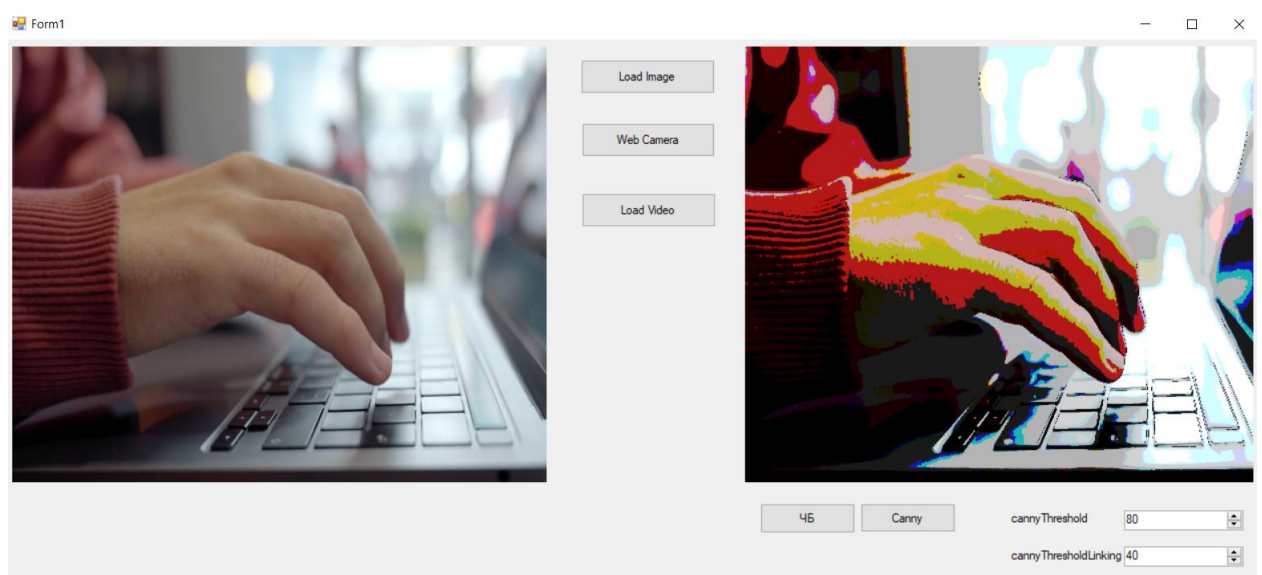
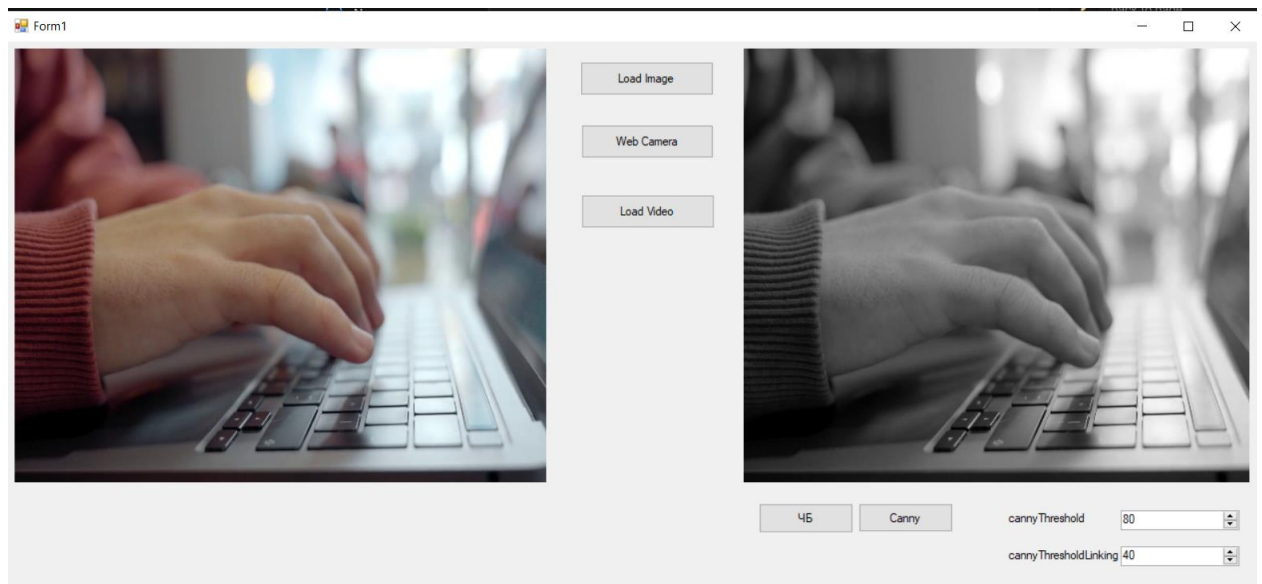
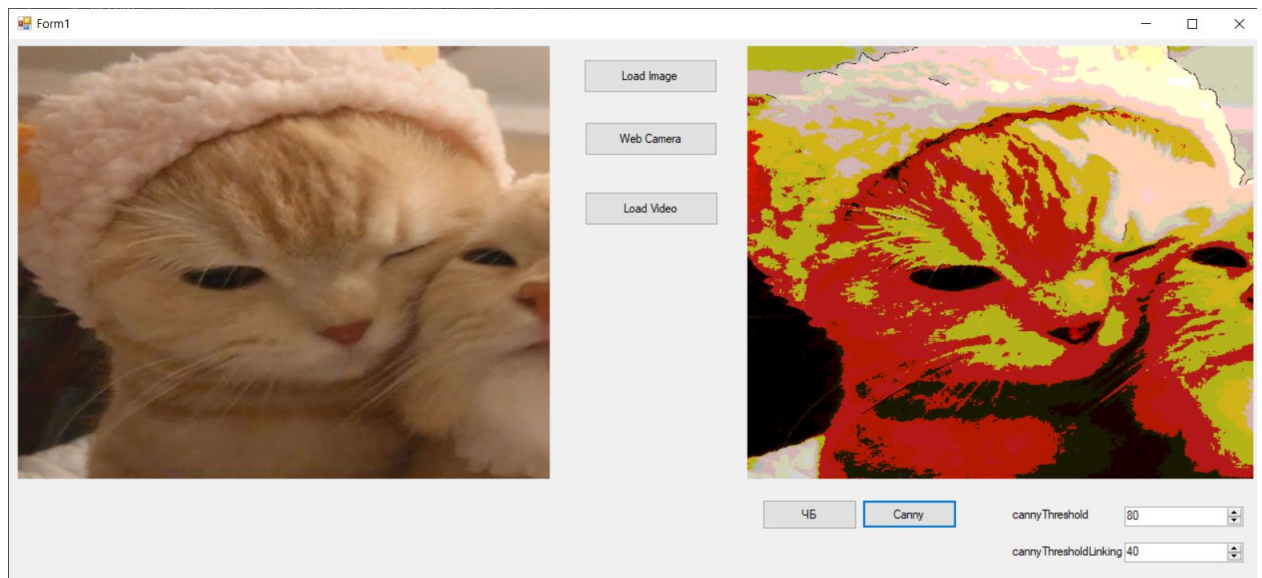
Постановка задачи:

Необходимо разработать приложение WindowsForms, способное:

1. осуществлять выбор и загрузку изображения;
2. отображать изображение на экране;
3. обрабатывать изображение при помощи эффекта Cell Shading.

Листинг программы с комментариями:





using Emgu.CV;

```

using Emgu.CV.Cuda;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;
using Emgu.CV.Util;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Roviaac__1
{
    public partial class Form1 : Form
    {
        public int k = 1;
        private VideoCapture capture;
        bool Pause = false;
        bool Video = false;
        bool CannyPicture = false;
        bool BlackAndWhitePicture = false;
        private Image<Bgr, byte> sourceImage; //глобальная переменная
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            var result = openFileDialog.ShowDialog(); // открытие диалога выбора
            openFileDialog.Filter = "Picture Files (*.jpg, *.png)| *.jpg;*.png";
            if (result == DialogResult.OK) // открытие выбранного файла
            {
                string fileName = openFileDialog.FileName;
                sourceImage = new Image<Bgr, byte>(fileName);
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (k % 2 == 1)
            {
                capture = new VideoCapture();
                capture.ImageGrabbed += ProcessFrame;
                capture.Start(); // начало обработки видеопотока
                k += 1;
            }
            else if (k % 2 == 0)
            {
                capture.Stop();
            }
        }

        private void ProcessFrame(object sender, EventArgs e)
        {
            var frame = new Mat();
            capture.Retrieve(frame); // получение текущего кадра
            imageBox1.Image = frame;
        }
    }
}

```

файла

```

}
private void ProcessFrameVideo(object sender, EventArgs e)
{
    var frame = new Mat();
    capture.Retrieve(frame); // получение текущего кадра
    var picture = frame.ToImage<Bgr, byte>();
    //var picture2 = frame.ToImage<Gray, byte>();
    imageBox1.Image = picture.Resize(640, 480, Inter.Linear);
    if (CannyPicture)
    {
        canny(picture);
    }
    else if (BlackAndWhitePicture)
    {
        BlackAndWhite(picture);
    }
    else
    {
        imageBox2.Image = picture.Resize(640, 480, Inter.Linear);
    }
    //imageBox2.Image = picture2.Resize(640, 480, Inter.Linear);
    //imageBox1.Image = frame.ToImage<Bgr, byte>();
    //imageBox2.Image = frame.ToImage<Gray, byte>();

    Thread.Sleep((int)capture.GetCaptureProperty(Emgu.CV.CvEnum.CapProp.Fps));
}

private void button3_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Video Files (*.mp4, *.flv)| *.mp4;*.flv";
    var result = openFileDialog.ShowDialog(); // открытие диалога выбора
    файла
    if (result == DialogResult.OK) // открытие выбранного файла
    {
        Video = true;
        string fileName = openFileDialog.FileName;
        capture = new VideoCapture(fileName);
        capture.ImageGrabbed += ProcessFrameVideo;
        capture.Start(); // начало обработки видеопотока
    }
}

void canny(Image <Bgr, byte> sourceImageNow)
{
    Image<Gray, byte> grayImage = sourceImageNow.Convert<Gray, byte>();
    var tempImage = grayImage.PyrDown();
    var destImage = tempImage.PyrUp();
    double cannyThreshold = (int)numericUpDown1.Value;
    //double cannyThreshold = 80.0;
    //double cannyThresholdLinking = 40.0;
    double cannyThresholdLinking = (int)numericUpDown2.Value;
    Image<Gray, byte> cannyEdges = destImage.Canny(cannyThreshold,
    cannyThresholdLinking);
    var cannyEdgesBgr = cannyEdges.Convert<Bgr, byte>();
    var resultImage = sourceImageNow.Sub(cannyEdgesBgr); // попиксельное
    //обход по каналам
    for (int channel = 0; channel < resultImage.NumberOfChannels;
    channel++)
    {
        for (int x = 0; x < resultImage.Width; x++)
            for (int y = 0; y < resultImage.Height; y++) // обход по
            пикселям
            {
                // получение цвета пикселя

```

```

        byte color = resultImage.Data[y, x, channel];
        if (color <= 50)
            color = 0;
        else if (color <= 100)
            color = 25;
        else if (color <= 150)
            color = 180;
        else if (color <= 200)
            color = 210;
        else
            color = 255;
        resultImage.Data[y, x, channel] = color; // изменение
цвета пикселя
    }
    imageBox2.Image = resultImage.Resize(640, 480, Inter.Linear);
}

void BlackAndWhite(Image <Bgr,byte> sourceImageNow)
{
    Image<Gray, byte> grayImage = sourceImageNow.Convert<Gray, byte>();
    imageBox2.Image = grayImage.Resize(640, 480, Inter.Linear);
}

private void button4_Click(object sender, EventArgs e)
{
    if (BlackAndWhitePicture == false)
    {
        BlackAndWhitePicture = true;
    }
    else
    {
        BlackAndWhitePicture = false;
    };
    if ((BlackAndWhitePicture) && (Video == false))
    {
        BlackAndWhite(sourceImage);
    }
}

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

private void imageBox1_Click(object sender, EventArgs e)
{
    var k = 0;
    if (capture == null)
    {
        imageBox1.Image = sourceImage.Resize(640, 480, Inter.Linear);
    }
    else
    {
        if ((capture != null) && (k == 0))

```

```

        {
            k = 1;
            capture.Pause();
        }
        else if (k == 1)
        {
            k = 0;
        }
    }
}

private void imageBox2_Click(object sender, EventArgs e)
{
}

private void timer1_Tick(object sender, EventArgs e)
{
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
}

private void label1_Click(object sender, EventArgs e)
{
}
}
}

```