**ПРИЛОЖЕНИЕ Б**

**Листинг модуля «Обнаружение пешеходов на видео»**

using MainForm.Constants;

using Emgu.CV;

using Emgu.CV.Structure;

using System;

using System.Drawing;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace MainForm.Forms

{

public partial class Detect\_pedestrian : Form

{

private VideoCapture capture = null; // отвечает за видео

private double frames; // кадры видео (общее количество)

private double framesCounter; // текущий кадр видео

private double fps; // частота кадров

private bool play;

public Detect\_pedestrian()

{

InitializeComponent();

}

private Image<Bgr, byte> FindPedestrian(Image<Bgr, byte> image)

{

MCvObjectDetection[] regions;

using (HOGDescriptor descriptor = new HOGDescriptor())

{

descriptor.SetSVMDetector(HOGDescriptor.GetDefaultPeopleDetector());

regions = descriptor.DetectMultiScale(image);

}

foreach(MCvObjectDetection pesh in regions)

{

image.Draw(pesh.Rect, new Bgr(Color.Red), 3);

CvInvoke.PutText(image, "Pedestrian", new Point(pesh.Rect.X, pesh.Rect.Y), Emgu.CV.CvEnum.FontFace.HersheyPlain, 1, new MCvScalar(255, 255, 255), 2);

}

return image;

}

private void LoadTheme()

{

foreach (Control btns in this.Controls)

{

if (btns.GetType() == typeof(Button))

{

Button btn = (Button)btns;

btn.BackColor = ThemeColor.PrimaryColor;

btn.ForeColor = FormConstants.BUTTON\_ACTIVATE\_FORE\_COLOR;

btn.FlatAppearance.BorderColor = ThemeColor.SecondaryColor;

}

}

}

private void Detect\_pedestrian\_Load(object sender, EventArgs e)

{

LoadTheme();

}

private async void ReadFrames()

{

Mat m = new Mat();

while(play && framesCounter < frames)

{

framesCounter += 1;

capture.SetCaptureProperty(Emgu.CV.CvEnum.CapProp.PosFrames, framesCounter);

capture.Read(m);

pictureBoxDetectPedestrian.Image = m.Bitmap;

pictureBoxDetectPedestrianResult.Image = FindPedestrian(m.ToImage<Bgr, byte>()).Bitmap;

labelFrames.Text = $"{framesCounter} / {frames}"; //счетчик кадров

await Task.Delay(1000 / Convert.ToInt16(fps));

if (framesCounter < 0)

{

framesCounter = 0;

}

else if (framesCounter == frames - 1)

{

framesCounter = 0;

play = true;

}

}

}

private void buttonFileOpen\_Click(object sender, EventArgs e)

{

try

{

DialogResult res = openFileDialog1.ShowDialog();

if (res == DialogResult.OK)

{

capture = new VideoCapture(openFileDialog1.FileName);

Mat m = new Mat();

capture.Read(m);

pictureBoxDetectPedestrian.Image = m.Bitmap;

fps = capture.GetCaptureProperty(Emgu.CV.CvEnum.CapProp.Fps);

frames = capture.GetCaptureProperty(Emgu.CV.CvEnum.CapProp.FrameCount);

framesCounter = 1;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message,"Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonPedestrianDetect\_Click(object sender, EventArgs e)

{

try

{

if(capture == null)

{

throw new Exception("Видео не выбрано");

}

else

{

play = true;

ReadFrames();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnbtnPauseVideo\_Click(object sender, EventArgs e)

{

try

{

play = false;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRewindBack\_Click(object sender, EventArgs e)

{

try

{

framesCounter -= Convert.ToDouble(numericUpDownFrames.Value);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRewindForward\_Click(object sender, EventArgs e)

{

try

{

framesCounter += Convert.ToDouble(numericUpDownFrames.Value);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void Detect\_pedestrian\_FormClosed(object sender, FormClosedEventArgs e)

{

play = false;

}

private void button1\_Click(object sender, EventArgs e)

{

play = false;

pictureBoxDetectPedestrian.Image = null;

pictureBoxDetectPedestrianResult.Image = null;

labelFrames.Text = String.Empty;

capture = null;

}

}

}