

















using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace EyeiresBased\_AttendanceSystem

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

Home h = new Home();

h.Show();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace EyeiresBased\_AttendanceSystem

{

public partial class Home : Form

{

public Home()

{

InitializeComponent();

}

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

{

NewUser n = new NewUser();

n.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Account aa = new Account();

aa.Show();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using Luxand;

using System.Data.SqlClient;

using System.IO;

namespace EyeiresBased\_AttendanceSystem

{

public partial class NewUser : Form

{

enum ProgramState { psRemember, psRecognize }

ProgramState programState = ProgramState.psRecognize;

String cameraName;

bool needClose = false;

string userName;

String TrackerMemoryFile;

int mouseX = 0;

int mouseY = 0;

// WinAPI procedure to release HBITMAP handles returned by FSDKCam.GrabFrame

[DllImport("gdi32.dll")]

static extern bool DeleteObject(IntPtr hObject);

public NewUser()

{

InitializeComponent();

}

private void NewUser\_Load(object sender, EventArgs e)

{

TrackerMemoryFile = Path.GetDirectoryName(Application.ExecutablePath).ToString() + "\\tracker.dat";

if (FSDK.FSDKE\_OK != FSDK.ActivateLibrary("E9Y/X8y+5kZjK89OUhC7MyuZfGdkDlTVZjBh4Zt/rhPnkYNrZZC/JI1jeYxFVBfVq8fwiOjaKhmwTYi+bbP+E4blf0pkCmGxSNB/3uI9GvjMwAGK+MlK2+o5HBCgLLOCvMMq6VPKIV/M8VXKrTAIR0FwOx8NmuAd29qfpX76xjU="))

{

MessageBox.Show("", "Error activating FaceSDK", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

FSDK.InitializeLibrary();

FSDKCam.InitializeCapturing();

string[] cameraList;

int count;

FSDKCam.GetCameraList(out cameraList, out count);

if (0 == count)

{

MessageBox.Show("Please attach a camera", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

cameraName = cameraList[0];

FSDKCam.VideoFormatInfo[] formatList;

FSDKCam.GetVideoFormatList(ref cameraName, out formatList, out count);

int VideoFormat = 0; // choose a video format

//pictureBox1.Width = formatList[VideoFormat].Width;

//pictureBox1.Height = formatList[VideoFormat].Height;

//this.Width = formatList[VideoFormat].Width + 48;

//this.Height = formatList[VideoFormat].Height + 96;

}

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

{

this.button1.Enabled = false;

int cameraHandle = 0;

int r = FSDKCam.OpenVideoCamera(ref cameraName, ref cameraHandle);

if (r != FSDK.FSDKE\_OK)

{

MessageBox.Show("Error opening the first camera", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

int tracker = 0; // creating a Tracker

if (FSDK.FSDKE\_OK != FSDK.LoadTrackerMemoryFromFile(ref tracker, TrackerMemoryFile)) // try to load saved tracker state

FSDK.CreateTracker(ref tracker); // if could not be loaded, create a new tracker

int err = 0; // set realtime face detection parameters

FSDK.SetTrackerMultipleParameters(tracker, "HandleArbitraryRotations=false; DetermineFaceRotationAngle=false; InternalResizeWidth=100; FaceDetectionThreshold=5;", ref err);

while (!needClose)

{

Int32 imageHandle = 0;

if (FSDK.FSDKE\_OK != FSDKCam.GrabFrame(cameraHandle, ref imageHandle)) // grab the current frame from the camera

{

Application.DoEvents();

continue;

}

FSDK.CImage image = new FSDK.CImage(imageHandle);

long[] IDs;

long faceCount = 0;

FSDK.FeedFrame(tracker, 0, image.ImageHandle, ref faceCount, out IDs, sizeof(long) \* 256); // maximum of 256 faces detected

Array.Resize(ref IDs, (int)faceCount);

// make UI controls accessible (to find if the user clicked on a face)

Application.DoEvents();

Image frameImage = image.ToCLRImage();

Graphics gr = Graphics.FromImage(frameImage);

for (int i = 0; i < IDs.Length; ++i)

{

FSDK.TFacePosition facePosition = new FSDK.TFacePosition();

FSDK.GetTrackerFacePosition(tracker, 0, IDs[i], ref facePosition);

int left = facePosition.xc - (int)(facePosition.w \* 0.6);

int top = facePosition.yc - (int)(facePosition.w \* 0.5);

int w = (int)(facePosition.w \* 1.2);

String name;

int res = FSDK.GetAllNames(tracker, IDs[i], out name, 65536); // maximum of 65536 characters

if (FSDK.FSDKE\_OK == res && name.Length > 0)

{ // draw name

StringFormat format = new StringFormat();

format.Alignment = StringAlignment.Center;

gr.DrawString("", new System.Drawing.Font("Arial", 16),

new System.Drawing.SolidBrush(System.Drawing.Color.LightGreen),

facePosition.xc, top + w + 5, format);

}

Pen pen = Pens.LightGreen;

if (mouseX >= left && mouseX <= left + w && mouseY >= top && mouseY <= top + w)

{

pen = Pens.Blue;

if (ProgramState.psRemember == programState)

{

if (FSDK.FSDKE\_OK == FSDK.LockID(tracker, IDs[i]))

{

// get the user name

InputName inputName = new InputName();

inputName.bmp = new Bitmap(pictureBox1.Image);

if (DialogResult.OK == inputName.ShowDialog())

{

userName = inputName.userName;

if (userName == null || userName.Length <= 0)

{

String s = "";

FSDK.SetName(tracker, IDs[i], "");

FSDK.PurgeID(tracker, IDs[i]);

}

else

{

FSDK.SetName(tracker, IDs[i], userName);

}

FSDK.UnlockID(tracker, IDs[i]);

}

}

}

}

gr.DrawRectangle(pen, left, top, w, w);

}

programState = ProgramState.psRecognize;

// display current frame

pictureBox1.Image = frameImage;

GC.Collect(); // collect the garbage after the deletion

FSDK.SaveTrackerMemoryToFile(tracker, TrackerMemoryFile);

}

FSDK.SaveTrackerMemoryToFile(tracker, TrackerMemoryFile);

FSDK.FreeTracker(tracker);

FSDKCam.CloseVideoCamera(cameraHandle);

FSDKCam.FinalizeCapturing();

}

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

programState = ProgramState.psRemember;

}

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

mouseX = e.X;

mouseY = e.Y;

}

private void pictureBox1\_MouseLeave(object sender, EventArgs e)

{

mouseX = 0;

mouseY = 0;

}

private void button2\_Click(object sender, EventArgs e)

{

needClose = true;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using Emgu.CV.UI;

using Emgu.CV;

using Emgu.CV.Structure;

using Emgu.CV.CvEnum;

using System.IO;

using System.Diagnostics;

using System.Data.SqlClient;

namespace EyeiresBased\_AttendanceSystem

{

public partial class InputName : Form

{

public string userName;

public Bitmap bmp;

string path;

Random rnd;

int count = 0;

Image<Bgr, Byte> currentFrame, faceframe;

public HaarCascade Face = new HaarCascade(Application.StartupPath + "/Cascades/haarcascade\_frontalface\_default.xml");

//public HaarCascade eye = new HaarCascade(Application.StartupPath + "/Cascades/haarcascade\_eye.xml");

Capture grabber;

Image<Gray, byte> result = null;

Image<Gray, byte> gray\_frame = null;

MCvFont font = new MCvFont(FONT.CV\_FONT\_HERSHEY\_COMPLEX, 0.5, 0.5);

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=G:\New Project 2016\BDU\Facesplit\Split.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

public InputName()

{

InitializeComponent();

}

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

{

string gender;

if (radioButton1.Checked == true)

{

gender = radioButton1.Text;

}

else

{

gender = radioButton2.Text;

}

cmd = new SqlCommand("insert into regtb values('" + textBox1.Text + "','" + gender + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + textBox7.Text + "','" + textBox8.Text + "')", con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

this.Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

userName = textBox1.Text;

}

private void InputName\_Load(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

int aa, bb;

string a;

private void button2\_Click(object sender, EventArgs e)

{

a = "";

aa = 0;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using Luxand;

using System.Data.SqlClient;

using System.IO;

using System.Data.SqlClient;

using System.Net;

namespace EyeiresBased\_AttendanceSystem

{

public partial class Login : Form

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=G:\New Project 2016\BDU\Facesplit\Split.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

public string name1, mobile;

public Login()

{

InitializeComponent();

}

enum ProgramState { psRemember, psRecognize }

ProgramState programState = ProgramState.psRecognize;

String cameraName;

bool needClose = false;

string userName;

String TrackerMemoryFile;

int mouseX = 0;

int mouseY = 0;

// WinAPI procedure to release HBITMAP handles returned by FSDKCam.GrabFrame

[DllImport("gdi32.dll")]

static extern bool DeleteObject(IntPtr hObject);

private void Login\_Load(object sender, EventArgs e)

{

TrackerMemoryFile = Path.GetDirectoryName(Application.ExecutablePath).ToString() + "\\tracker.dat";

if (FSDK.FSDKE\_OK != FSDK.ActivateLibrary("E9Y/X8y+5kZjK89OUhC7MyuZfGdkDlTVZjBh4Zt/rhPnkYNrZZC/JI1jeYxFVBfVq8fwiOjaKhmwTYi+bbP+E4blf0pkCmGxSNB/3uI9GvjMwAGK+MlK2+o5HBCgLLOCvMMq6VPKIV/M8VXKrTAIR0FwOx8NmuAd29qfpX76xjU="))

{

MessageBox.Show("", "Error activating FaceSDK", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

FSDK.InitializeLibrary();

FSDKCam.InitializeCapturing();

string[] cameraList;

int count;

FSDKCam.GetCameraList(out cameraList, out count);

if (0 == count)

{

MessageBox.Show("Please attach a camera", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

cameraName = cameraList[0];

FSDKCam.VideoFormatInfo[] formatList;

FSDKCam.GetVideoFormatList(ref cameraName, out formatList, out count);

int VideoFormat = 0; // choose a video format

//pictureBox1.Width = formatList[VideoFormat].Width;

//pictureBox1.Height = formatList[VideoFormat].Height;

//this.Width = formatList[VideoFormat].Width + 48;

//this.Height = formatList[VideoFormat].Height + 96;

}

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

{

this.button1.Enabled = false;

int cameraHandle = 0;

int r = FSDKCam.OpenVideoCamera(ref cameraName, ref cameraHandle);

if (r != FSDK.FSDKE\_OK)

{

MessageBox.Show("Error opening the first camera", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

int tracker = 0; // creating a Tracker

if (FSDK.FSDKE\_OK != FSDK.LoadTrackerMemoryFromFile(ref tracker, TrackerMemoryFile)) // try to load saved tracker state

FSDK.CreateTracker(ref tracker); // if could not be loaded, create a new tracker

int err = 0; // set realtime face detection parameters

FSDK.SetTrackerMultipleParameters(tracker, "HandleArbitraryRotations=false; DetermineFaceRotationAngle=false; InternalResizeWidth=100; FaceDetectionThreshold=5;", ref err);

while (!needClose)

{

Int32 imageHandle = 0;

if (FSDK.FSDKE\_OK != FSDKCam.GrabFrame(cameraHandle, ref imageHandle)) // grab the current frame from the camera

{

Application.DoEvents();

continue;

}

FSDK.CImage image = new FSDK.CImage(imageHandle);

long[] IDs;

long faceCount = 0;

FSDK.FeedFrame(tracker, 0, image.ImageHandle, ref faceCount, out IDs, sizeof(long) \* 256); // maximum of 256 faces detected

Array.Resize(ref IDs, (int)faceCount);

// make UI controls accessible (to find if the user clicked on a face)

Application.DoEvents();

Image frameImage = image.ToCLRImage();

Graphics gr = Graphics.FromImage(frameImage);

for (int i = 0; i < IDs.Length; ++i)

{

FSDK.TFacePosition facePosition = new FSDK.TFacePosition();

FSDK.GetTrackerFacePosition(tracker, 0, IDs[i], ref facePosition);

int left = facePosition.xc - (int)(facePosition.w \* 0.6);

int top = facePosition.yc - (int)(facePosition.w \* 0.5);

int w = (int)(facePosition.w \* 1.2);

String name;

int res = FSDK.GetAllNames(tracker, IDs[i], out name, 65536); // maximum of 65536 characters

if (FSDK.FSDKE\_OK == res && name.Length > 0)

{ // draw name

StringFormat format = new StringFormat();

format.Alignment = StringAlignment.Center;

gr.DrawString(name, new System.Drawing.Font("Arial", 16),

new System.Drawing.SolidBrush(System.Drawing.Color.LightGreen),

facePosition.xc, top + w + 5, format);

label1.Text = name;

}

else

{

label1.Text = "Wrong Person";

}

con.Open();

con.Close();

Pen pen = Pens.LightGreen;

if (mouseX >= left && mouseX <= left + w && mouseY >= top && mouseY <= top + w)

{

pen = Pens.Blue;

if (ProgramState.psRemember == programState)

{

if (FSDK.FSDKE\_OK == FSDK.LockID(tracker, IDs[i]))

{

// get the user name

InputName1 inputName = new InputName1();

inputName.bmp = new Bitmap(pictureBox1.Image);

inputName.uname = name;

inputName.ShowDialog();

//if (DialogResult.OK == inputName.ShowDialog())

//{

// userName = inputName.userName;

// if (userName == null || userName.Length <= 0)

// {

// String s = "";

// FSDK.SetName(tracker, IDs[i], "");

// FSDK.PurgeID(tracker, IDs[i]);

// }

// else

// {

// FSDK.SetName(tracker, IDs[i], userName);

// }

// FSDK.UnlockID(tracker, IDs[i]);

//}

}

}

}

gr.DrawRectangle(pen, left, top, w, w);

}

programState = ProgramState.psRecognize;

// display current frame

pictureBox1.Image = frameImage;

GC.Collect(); // collect the garbage after the deletion

FSDK.SaveTrackerMemoryToFile(tracker, TrackerMemoryFile);

}

FSDK.SaveTrackerMemoryToFile(tracker, TrackerMemoryFile);

FSDK.FreeTracker(tracker);

FSDKCam.CloseVideoCamera(cameraHandle);

FSDKCam.FinalizeCapturing();

}

private void pictureBox1\_MouseLeave(object sender, EventArgs e)

{

mouseX = 0;

mouseY = 0;

}

private void pictureBox1\_MouseMove(object sender, System.Windows.Forms.MouseEventArgs e)

{

mouseX = e.X;

mouseY = e.Y;

}

private void pictureBox1\_MouseUp(object sender, System.Windows.Forms.MouseEventArgs e)

{

programState = ProgramState.psRemember;

}

public void sendmessage(string targetno, string message)

{

String query = "http://bulksms.mysmsmantra.com:8080/WebSMS/SMSAPI.jsp?username=fantazy&password=1483727597&sendername=fantaz&mobileno=" + targetno + "&message=" + message;

WebClient client = new WebClient();

Stream sin = client.OpenRead(query);

MessageBox.Show(targetno + "Message Send");

}

private void button3\_Click(object sender, EventArgs e)

{

if (name1 ==label1 .Text )

{

MessageBox.Show("Transaction Completed!");

}

else

{

MessageBox.Show("UNKNOWN USER Access You Account");

// sendmessage(mobile, "UNKNOWN USER Access You Account");

}

}

}

}

Regtb:

|  |  |  |
| --- | --- | --- |
| id | bigint | Unchecked |
| UserName | nvarchar(50) | Checked |
| FatherName | nvarchar(50) | Checked |
| Gender | nvarchar(50) | Checked |
| Age | nvarchar(50) | Checked |
| Mobileno | nvarchar(50) | Checked |
| Emailid | nvarchar(50) | Checked |
| BankName | nvarchar(50) | Checked |
| Accno | nvarchar(50) | Checked |
| PinNo | nvarchar(50) | Checked |