```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Net.Mail;
using System.Text;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.Diagnostics;
using System.Media;
namespace Drowsy_Driver_Detection_System
{
    public partial class Webcam : Form
    {

        #region Variables
        private Capture capWebCam;
        private HaarCascade haar, haar_eye;
        private HaarCascade _faces;
        private HaarCascade _eyes;
        int xMax;
        int yMax;
        Stopwatch stpWatch, stpWatch2;
        #endregion
        int eye_counter = 0;
        int speed_counter = 0;

        public Webcam()
        {
            InitializeComponent();
            _faces = new HaarCascade("haarcascade_frontalface_alt_tree.xml");
            _eyes = new HaarCascade("haarcascade_eye.xml");
            stpWatch = new Stopwatch();
            stpWatch2 = new Stopwatch();
        }



        private void Webcam_Load(object sender, EventArgs e)
        {
            capWebCam = new Capture(0);
            //capWebCam = new Capture("1.3gp");
            haar = new HaarCascade("haarcascade_frontalface_alt_tree.xml");
            haar_eye = new HaarCascade("parojosG.xml");
          // haar_eye = new HaarCascade("haarcascade_eye.xml");
            timer1.Start();

        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            using (Image<Bgr, Byte> nextFrame = capWebCam.QueryFrame())
            {
              // label2.Visible = false;
               if (nextFrame != null)
```

```
                {
                    Image<Gray, Byte> grayframe = nextFrame.Convert<Gray, Byte>();

                    var faces = grayframe.DetectHaarCascade(haar, 1.4, 4, 
HAAR_DETECTION_TYPE.FIND_BIGGEST_OBJECT, new Size(nextFrame.Width / 8, 
nextFrame.Height / 8))[0];
                    if (faces == null || faces.Length <= 0)
                    {
                        if (stpWatch.IsRunning)
                        {
                            //check if the face has not been detected for 10 secs
                            if (stpWatch.Elapsed.Seconds >= 5)
                            {
                                Console.Beep(1000, 1000);
                                System.Media.SystemSounds.Question.Play();
                                #region Extra Sounds
                                /*
                                 * System.Media.SystemSounds.Beep.Play();
                                    System.Media.SystemSounds.Asterisk.Play();
                                    System.Media.SystemSounds.Exclamation.Play();
                                    System.Media.SystemSounds.Question.Play();
                                    System.Media.SystemSounds.Hand.Play();
                                 * */
                                #endregion
                                //MessageBox.Show("driver is drowsy!");
                               // label2.Visible = true;

                                stpWatch.Reset();
                                stpWatch.Stop();
                            }
                        }
                        else
                        {
                            stpWatch.Start();
                        }
                    }
                    foreach (var face in faces)
                    {
                        try
                        {
                            nextFrame.Draw(face.rect, new Bgr(555, double.MaxValue, 
250), 1);
                            //if the person is attentive stop the stopwatch

                            if (stpWatch.IsRunning)
                            {
                                stpWatch.Reset();
                                stpWatch.Stop();
                            }
                            grayframe.ROI = face.rect;
                            //var eyes = grayframe.DetectHaarCascade(haar_eye, 1.4, 4, 
HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(nextFrame.Width / 8, nextFrame.Height 
/ 8))[0];
                            var eyes = grayframe.DetectHaarCascade(haar_eye, 1.15,4 , 
HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(20, 20))[0];
                         //   MCvAvgComp[][] leftEyesDetected = 
grayFrame.DetectHaarCascade(_eyes, 1.15, 0, 
Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(20, 20));
```

```
                            foreach (var eye in eyes)
                            {



                                 //grayframe.ROI = Rectangle.Empty;
                                 Rectangle rect = new Rectangle(eye.rect.X + face.rect.X,
eye.rect.Y + face.rect.Y, eye.rect.Width, eye.rect.Height);
                                nextFrame.Draw(rect, new Bgr(Color.AliceBlue), 1);
                                 Gray cannyThreshold = new Gray(50);
                                 Gray cannyThresholdLinking = new Gray(10);
                                 grayframe.ROI = rect;
                                grayframe = grayframe.ThresholdBinary(new Gray(40), new
Gray(250)); //CHANGE ACCORDING TO YOUR NEED
                                imageBox1.Image = grayframe;


                                int count = 0;
                                for (int i = 0; i < grayframe.Rows; i++)
                                {
                                    for (int j = 0; j < grayframe.Cols; j++)
                                    {
                                        if (grayframe.Data[i, j, 0] == 0)
                                            count++;
                                        label1.Text = string.Format("{0} black pixels",
count);

                                    }
                                }

                                //count = 20;

                                 //Change this value according to ur camera & environment
brightness
                                if (count <= 100)
                                {
                                    eye_counter++;
                                }
                                else
                                    eye_counter = 0;
                                if ( eye_counter >=5) // NO. of Consecutive Frames
                                {
                                    if (stpWatch2.IsRunning)
                                    {
                                        //check if the face has not been detected for 10
secs

                                        if (stpWatch2.Elapsed.Seconds >= 5)
                                        {
                                            Console.Beep(5000, 1000);
                                            System.Media.SystemSounds.Question.Play();
                                            //SoundPlayer simpleSound = new
SoundPlayer(@"C:\voice.wav");

                                            //simpleSound.Play();
                                            #region Extra Sounds
                                            /*
                                  * System.Media.SystemSounds.Beep.Play();
                                   System.Media.SystemSounds.Asterisk.Play();
                                   System.Media.SystemSounds.Exclamation.Play();
                                   System.Media.SystemSounds.Question.Play();
                                   System.Media.SystemSounds.Hand.Play();
                                  * */
                                            #endregion
                                            //MessageBox.Show("driver is drowsy!");

                                            stpWatch2.Reset();
                                            stpWatch2.Stop();

                                            SmtpClient smtpserver = new SmtpClient();
                                            MailMessage mail = new MailMessage();
                                            smtpserver.Credentials = new
System.Net.NetworkCredential("togale4@gmail.com","tej@12345");
                                            smtpserver.Port = 587;
                                            smtpserver.EnableSsl = true;
                                            smtpserver.Host = "smtp.gmail.com";
                                            mail = new MailMessage();
                                            mail.From = new
MailAddress("togale4@gmail.com");
                                            mail.To.Add("pranetakasbe@gmail.com");
                                            mail.Subject = "Warning";
                                            mail.Body = " Details: driver drowsy detected
" ;
                                            smtpserver.Send(mail);


                                        }
                                    }
                                    else
                                    {
                                        stpWatch2.Start();

                                    }
                                }
                                 //Image<Gray, Byte> cannyEdges = grayframe.Canny(new
Gray(150), new Gray(60));
                                 //imageBox2.Image = cannyEdges;
                                 //CircleF[] circles = cannyEdges.HoughCircles(
                                 //                cannyThreshold,
                                 //                cannyThresholdLinking,
                                 //                1.0, //Resolution of the accumulator
used to detect centers of the circles
                                 //                15.0, //min distance
                                 //                5, //min radius
                                 //                10 //max radius
                                 //                )[0]; //Get the circles from the first
channel
                                 //for (int i = 0; i < circles.Length; i++)
                                 //{
                                 //    circles[i].Center = new PointF(rect.X +
circles[i].Center.X, rect.Y + circles[i].Center.Y);
                                 //    nextFrame.Draw(circles[i], new
Bgr(Color.DarkOliveGreen), 1);
                                 //}



                              }
```

```
                        }
                        catch (Exception ee)
                        { }
                    }
                    webcamBox.Image = nextFrame;
                }

            }

        }


    }

}
```