

form.cs

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Reflection;
using System.Threading.Tasks;
using System.Windows.Forms;
using openalprnet;
using System.Diagnostics;

namespace final
{
    public partial class Form1 : Form
    {
        String car="";
        public Form1()
        {
            InitializeComponent();
            comboBox1.SelectedIndex = 0;
            radioButton1.Checked = true;
        }

        public static string AssemblyDirectory
        {
            get
            {
                var codeBase =
Assembly.GetExecutingAssembly().CodeBase;
                var uri = new UriBuilder(codeBase);
                var path = Uri.UnescapeDataString(uri.Path);
                return Path.GetDirectoryName(path);
            }
        }
    }
}
```

```

    }
}

public Rectangle boundingRectangle(List<Point> points)
{
    var minX = points.Min(p => p.X);
    var minY = points.Min(p => p.Y);
    var maxX = points.Max(p => p.X);
    var maxY = points.Max(p => p.Y);

    return new Rectangle(new Point(minX, minY), new
Size(maxX - minX, maxY - minY));
}

private static Image cropImage(Image img, Rectangle
cropArea)
{
    var bmpImage = new Bitmap(img);
    return bmpImage.Clone(cropArea,
bmpImage.PixelFormat);
}

public static Bitmap combineImages(List<Image> images)
{
    Bitmap finalImage = null;
    try
    {
        var width = 0;
        var height = 0;

        foreach (var bmp in images)
        {
            width += bmp.Width;
            height = bmp.Height > height ? bmp.Height :
height;

```

```

    }

    finalImage = new Bitmap(width, height);

    using (var g = Graphics.FromImage(finalImage))
    {
        g.Clear(Color.Black);
        var offset = 0;
        foreach (Bitmap image in images)
        {
            g.DrawImage(image,
                new Rectangle(offset, 0,
image.Width, image.Height));
            offset += image.Width;
        }
    }

    return finalImage;
}
catch (Exception ex)
{
    if (finalImage != null)
        finalImage.Dispose();

    throw ex;
}
finally
{
    foreach (var image in images)
    {
        image.Dispose();
    }
}
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    openFileDialog.FilterIndex = 1;
    if (radioButton1.Checked == true)
    {
        if (openFileDialog.ShowDialog(this) ==
DialogResult.OK)
        {
            owner_nameTextBox.Text = null;
            owner_phoneTextBox.Text = null;
            owner_addressTextBox.Text = null;
            car_IDTextBox.Text = null;
            processImageFile(openFileDialog.FileName);
        }
    }
    else
    {
        openFileDialog.FilterIndex = 2;
        if (openFileDialog.ShowDialog(this) ==
DialogResult.OK)
        {
            owner_nameTextBox.Text = null;
            owner_phoneTextBox.Text = null;
            owner_addressTextBox.Text = null;
            car_IDTextBox.Text = null;
            resetControls();
            var region =
comboBox1.SelectedItem.ToString();
            using (Process process = new Process())
            {
                process.StartInfo.FileName =
"alpr.exe";
                process.StartInfo.Arguments = "-c "+
region.ToLower()+" "+openFileDialog.SafeFileName;

```

```

        process.StartInfo.UseShellExecute =
false;

process.StartInfo.RedirectStandardOutput = true;
        process.StartInfo.CreateNoWindow =
true;

        process.Start();

while (process.StandardOutput.EndOfStream != true)
    {

lbxPlates.Items.Add(process.StandardOutput.ReadLine());
        }
        /* String output =
process.StandardOutput.ReadToEnd();
        int Start, End;
        if (output.Contains("Performing
high-contrast edge detection") && output.Contains("Frame: "))
        {
            Start =
output.IndexOf("Performing high-contrast edge detection", 0);
            End = output.IndexOf("Frame: ",
Start);

lbxPlates.Items.Add(output.Substring(Start, End - Start));
        }
        else
        {
            lbxPlates.Items.Add("No plates
found in the video") ;
        }*/
    }
}
}

```

```

this.tableTableAdapter.Fill(this.driverDataSet.Table, car);
    }

    private void processImageFile(string fileName)
    {
        resetControls();
        var region = comboBox1.SelectedItem.ToString();
        String config_file = Path.Combine(AssemblyDirectory,
"D:\\openalpr_64\\openalpr.conf");
        String runtime_data_dir =
Path.Combine(AssemblyDirectory,
"D:\\openalpr_64\\runtime_data");
        using (var alpr = new AlprNet(region.ToLower(),
config_file, runtime_data_dir))
        {
            if (!alpr.IsLoaded())
            {
                lbxPlates.Items.Add("Error initializing
OpenALPR");

                return;
            }
            picOriginal.ImageLocation = fileName;
            picOriginal.Load();

            var results = alpr.Recognize(fileName);

            var images = new
List<Image>(results.Plates.Count());
            var i = 1;
            foreach (var result in results.Plates)
            {
                var rect =
boundingRectangle(result.PlatePoints);

```

```

        var img = Image.FromFile(fileName);
        var cropped = cropImage(img, rect);
        images.Add(cropped);
        car =
result.BestPlate.Characters.ToString();
        lbxPlates.Items.Add("\t\t-- Plate #" + i++ +
" --");

        foreach (var plate in result.TopNPlates)
        {
            lbxPlates.Items.Add(string.Format(@"{0}
{1}% {2}",

plate.Characters.PadRight(12),

plate.OverallConfidence.ToString("N1").PadLeft(8),

plate.MatchesTemplate.ToString().PadLeft(8)));
            if (plate.MatchesTemplate.ToString() ==
"True")

                car = plate.Characters.ToString();

        }
    }

    if (images.Any())
    {
        picLicensePlate.Image =
combineImages(images);
    }
}

private void resetControls()
{
    picOriginal.Image = null;

```

```
        picLicensePlate.Image = null;
        lbxPlates.Items.Clear();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

        resetControls();
    }

}
}
```