

C# 系统应用之ListView实现简单图片浏览器

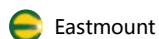
原创 Eastmount 最后发布于2014-10-10 13:27:16 阅读数 12079 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...

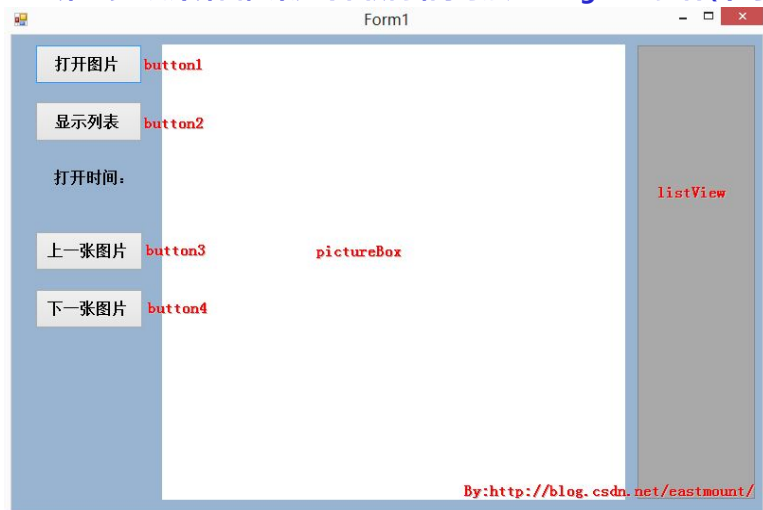


¥9.90

去订阅

最近有同学问我如何使用ListView加载图片列表,前面在"C#系统应用"中TreeView+ListView+ContextMenuStrip控件实现树状图显示磁盘目录,并在ListView中显示文件的详细信息.这里准备简单介绍下给同学讲述的如何使用ListView+ImageList控件实现简单的图片浏览器知识.

第一步 设计界面框架如下图所示,同时添加ImageList控件(不可见)

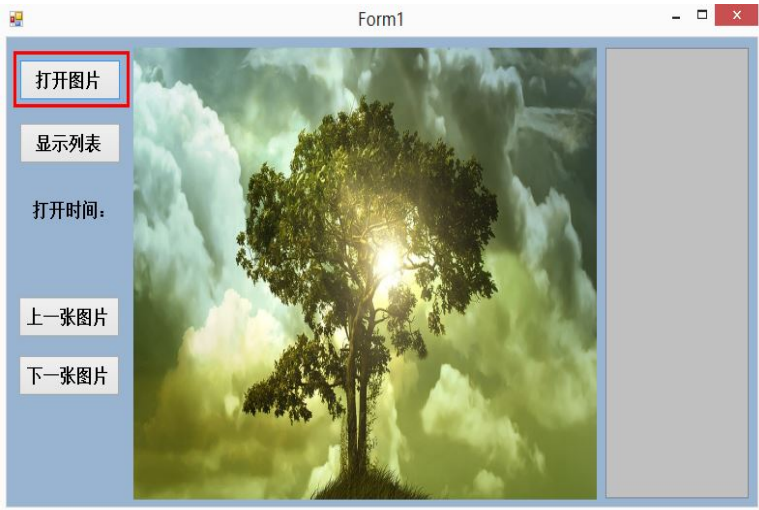


注意:设置ListView控件的Anchor属性为Top,Bottom,Right;设置PictureBox的Anchor属性为上下左右.

第二步 使用OpenFileDialog控件打开显示图片

```
//打开图片
private void button1_Click(object sender, EventArgs e)
{
    // 设置打开文件控件
    OpenFileDialog openfile = new OpenFileDialog();
    openfile.Filter = "JPG(*.JPG;*.JPEG);gif文件(*.GIF);BMP文件(*.BMP);PNG文件(*.PNG)|*.jpg;*.jpeg;*.gif;*.bmp;*.png";
    openfile.FilterIndex = 1; // 当前选定索引    openfile.RestoreDirectory = true;
    openfile.FileName = "";
    // 对话框选择确定按钮
    if (openfile.ShowDialog() == DialogResult.OK)
    {
        //FromFile从指定的文件创建Image
        pictureBox1.Image = Image.FromFile(openfile.FileName);
        // 图片被拉伸或收缩适合pictureBox大小
        pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
    }
}
```

图片显示效果如下图所示,需要注意的是在使用FromFile显示图片,可能图片全屏显示时会出现只出现部分图片现象,我设置图片为可拉伸或收缩StretchImage模式.



第三步 显示图片列表至ListView控件中

主要通过控件FolderBrowserDialog控件打开文件夹,同时获取文件夹的路径;在通过GetFiles("*.jpg")函数获取jpg格式图片,并获取文件夹中文件增加至ImageList中,设置ListView的View属性格式为LargeIcon大图标格式显示.

```
// 添加命名空间
using System.IO;                //Directory目录
using System.Diagnostics;        //Stopwatch显示时间

// 定义变量
private string folderDirPath;    // 图片文件夹地址
private string picDirPath = null; // 图片路径
private List<string> imagePathList = new List<string>(); // 获取列表图片路径
private int index;              // 获取选中列表图片序号

//ListView和imageList显示图片列表
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        // 打开选择文件夹对话框
        FolderBrowserDialog folderBrowserDialog = new FolderBrowserDialog();
        DialogResult result = folderBrowserDialog.ShowDialog();
        if (result == DialogResult.OK)
        {
            // 获取用户选择的文件夹路径
            this.folderDirPath = folderBrowserDialog.SelectedPath;
            // 调用自定义函数显示图片列表至ListView控件
            ShowPicture();
        }
        else if (result == DialogResult.Cancel)
        {
            MessageBox.Show("取消显示图片列表");
        }
    }
    catch (Exception msg)
    {
        // 报错提示 未将对象引用设置到对象的实例
        throw msg;
    }
}

// 显示图片列表至ListView控件
private void ShowPicture()
```

```

{
    // 提供一种方法测试运行时间 开始计算
    // 参考资料:http://www.cnblogs.com/newstart/archive/2012/09/21/2696884.html
    Stopwatch sw = new Stopwatch();
    sw.Start();

    // 获取目录与子目录
    DirectoryInfo dir = new DirectoryInfo(folderDirPath);
    // 获取当前目录JPG文件列表 GetFiles 获取指定目录中文件的名称(包括其路径)
    FileInfo[] fileInfo = dir.GetFiles("*.jpg");
    // 防止图片失真
    // 参考资料:http://blog.csdn.net/cdefg198/article/details/7821891 (博客中引用)
    this.imageList1.ColorDepth = ColorDepth.Depth32Bit;
    for (int i = 0; i < fileInfo.Length; i++)
    {
        // 获取文件完整目录
        picDirPath = fileInfo[i].FullName;
        // 记录图片源路径 双击显示图片时使用
        imagePathList.Add(picDirPath);
        // 图片加载到ImageList控件和imageList 图片列表
        this.imageList1.Images.Add(Image.FromFile(picDirPath));
    }

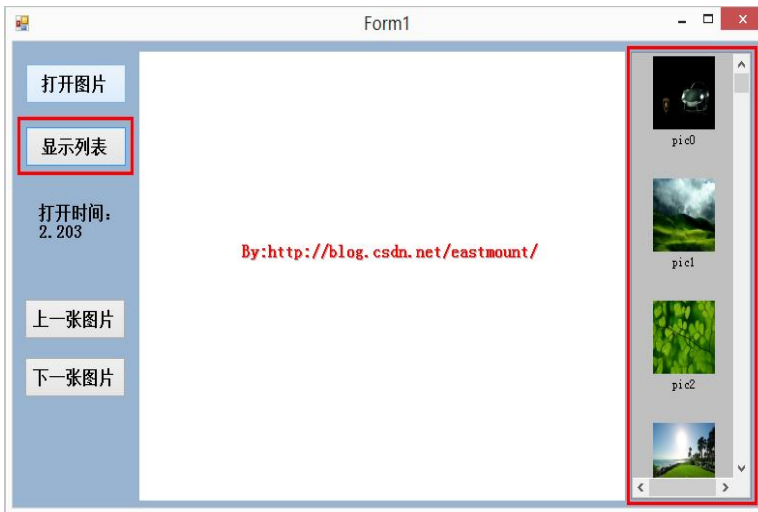
    // 显示文件列表
    this.listView1.Items.Clear();
    this.listView1.LargeImageList = this.imageList1;
    this.listView1.View = View.LargeIcon; // 大图标显示
    // imageList1.ImageSize = new Size(40, 40); // 不能设置ImageList的图像大小 属性处更改

    // 开始绑定
    this.listView1.BeginUpdate();
    // 增加图片至ListView控件中
    for (int i = 0; i < imageList1.Images.Count; i++)
    {
        ListViewItem lvi = new ListViewItem();
        lvi.ImageIndex = i;
        lvi.Text = "pic" + i;
        this.listView1.Items.Add(lvi);
    }
    this.listView1.EndUpdate();

    // 显示打开图片列表所需时间
    sw.Stop();
    long seconds = sw.ElapsedMilliseconds; // 毫秒单位
    label1.Text += '\n' + (Convert.ToDouble(seconds) / 1000).ToString(); // 转换为秒
}

```

显示结果如下图所示:



需要注意的是:

1.使用ListView加载信息的几个步骤: 获取文件夹路径 -> DirectoryInfo获取目录 -> GetFiles获取文件 -> Add图片至ImageList -> Add图片至ListView.

2.在设置ListView中图片的大小时,使用imageList1.ImageSize = new Size(40, 40)赋值失败,我是通过修改ImageList1的ImageSize属性为(64,64)实现的.

3.显示ListView中图片,通常会出现失真的情况,主要原因参考博客:[ListView显示图片失真](#).

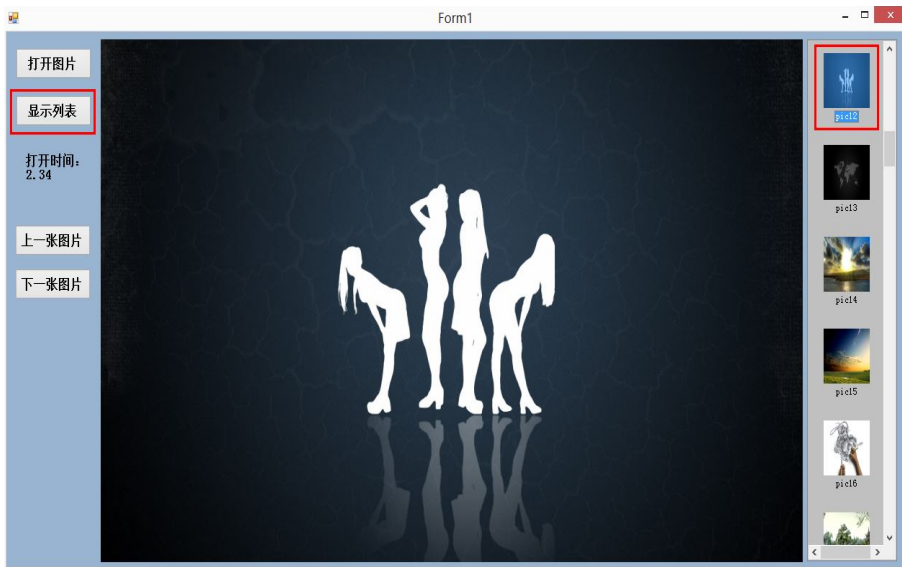
主要概括为ImageList里面图片颜色失真和图片大小失真,其中图片颜色失真原因是Design-Time在VS.NET中添加图片时没有使用用户指定的ColorDepth(如Depth32Bit),而用了ImageList.ColorDepth的默认值(Depth8Bit).因此需要先设置图片颜色深度,在再往ImageList中添加图片,而图片大小统一的都等于ImageList.ImageSize.

第四步 通过listView1_DoubleClick函数双击打开图片

在Form1.cs[设计]中ListView属性页为其添加DoubleClick双击事件,并通过Image.FromFile显示图片.

```
// 增加双击ListView事件 显示图片至PictureBox
private void listView1_DoubleClick(object sender, EventArgs e)
{
    if (this.listView1.SelectedItems.Count == 0)
        return;
    // 采用索引方式 imagePathList记录图片真实路径
    index = this.listView1.SelectedItems[0].Index;
    // 显示图片
    this.pictureBox1.Image = Image.FromFile(imagePathList[index]);
    // 图片被拉伸或收缩适合pictureBox大小
    pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
}
```

双击列表中不同图片的显示效果如下图所示:



其中需要注意的是,我在列表中显示图片重命名为"pic+数字",同时定义变量记录文件夹中图片真实路径与其一一对应.`private List<string> imagePathList = new List<string>()`.这里使用index显示对应图片即可,同样显示上一张\下一张相同.

第五步 显示上一张\下一张

```
// 显示上一张图片
private void button3_Click(object sender, EventArgs e)
{
    if (pictureBox1.Image != null)
    {
        if (index > 0)
        {
            index--;
            // 显示图片
            this.pictureBox1.Image = Image.FromFile(imagePathList[index]);
            pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
        }
        else if (index == 0)
        {
            index = imagePathList.Count;
            index--;
            // 显示图片
            this.pictureBox1.Image = Image.FromFile(imagePathList[index]);
            pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
        }
    }
}

// 显示下一张图片
private void button4_Click(object sender, EventArgs e)
{
    if (pictureBox1.Image != null)
    {
        if (index == imagePathList.Count - 1) // 最后一张图片
        {
            index = 0;
            // 显示图片
            this.pictureBox1.Image = Image.FromFile(imagePathList[index]);
            pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
        }
        else
        {
            index++;
        }
    }
}
```

```

        // 显示图片
        this.pictureBox1.Image = Image.FromFile(imagePathList[index]);
        pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
    }
}
}

```

文章写到处基本内容完成,内容比较简单,但也构成了一个完整的图片浏览器.同时很多时候我们需要上传缩略图,可以调用下面函数(在线笔记):

```

// 添加命名空间
using System.Drawing.Drawing2D;    //CompositingQuality.HighQuality
using System.Drawing.Imaging;      //EncoderParameter

/// <SUMMARY>
/// 图片无损缩放 自定义函数生成缩略图
/// </SUMMARY>
/// <PARAM name="sourceFile">图片源路径</PARAM>
/// <PARAM name="destFile">缩放后图片输出路径</PARAM>
/// <PARAM name="destHeight">缩放后图片高度</PARAM>
/// <PARAM name="destWidth">缩放后图片宽度</PARAM>
/// <RETURNS></RETURNS>
public static bool GetThumbnail(string sourceFile, string destFile, int destHeight, int destWidth)
{
    System.Drawing.Image imgSource = System.Drawing.Image.FromFile(sourceFile);
    System.Drawing.Imaging.ImageFormat thisFormat = imgSource.RawFormat;
    int sW = 0, sH = 0;

    // 按比例缩放
    int sWidth = imgSource.Width;
    int sHeight = imgSource.Height;

    if (sHeight > destHeight || sWidth > destWidth)
    {
        if ((sWidth * destHeight) > (sHeight * destWidth))
        {
            sW = destWidth;
            sH = (destWidth * sHeight) / sWidth;
        }
        else
        {
            sH = destHeight;
            sW = (sWidth * destHeight) / sHeight;
        }
    }
    else
    {
        sW = sWidth;
        sH = sHeight;
    }

    // 新建一个bmp图片
    Bitmap outBmp = new Bitmap(destWidth, destHeight);
    // 新建一个画板
    Graphics g = Graphics.FromImage(outBmp);
    // 清空画布并以透明背景色填充 Color.Black黑色填充
    g.Clear(System.Drawing.Color.Transparent);
    // 设置画布的描绘质量
    g.CompositingQuality = CompositingQuality.HighQuality;
    // 设置高质量,低速度呈现平滑程度
    g.SmoothingMode = SmoothingMode.HighQuality;

```

```

// 设置高质量插值法
        g.InterpolationMode = InterpolationMode.HighQualityBicubic;
// 在指定位置并且按指定大小绘制原图片的指定部分
        g.DrawImage(imgSource, new Rectangle((destWidth - sW) / 2, (destHeight - sH) / 2, sW, sH), 0, 0,
imgSource.Width, imgSource.Height, GraphicsUnit.Pixel);
        g.Dispose();
// 以下代码为保存图片时 设置压缩质量
        EncoderParameters encoderParams = new EncoderParameters();
        long[] quality = new long[1];
        quality[0] = 100;
        EncoderParameter encoderParam = new EncoderParameter(System.Drawing.Imaging.Encoder.Quality, quality);
        encoderParams.Param[0] = encoderParam;

        try
        {
            // 获得包含有关内置图像编解码器的信息的ImageCodecInfo对象
            ImageCodecInfo[] arrayICI = ImageCodecInfo.GetImageEncoders();
            ImageCodecInfo jpegICI = null;
            for (int x = 0; x < arrayICI.Length; x++)
            {
                if (arrayICI[x].FormatDescription.Equals("JPEG"))
                {
                    jpegICI = arrayICI[x]; // 设置JPEG编码
                    break;
                }
            }
            // 保存为JPG格式图片
            if (jpegICI != null)
            {
                outBmp.Save(destFile, jpegICI, encoderParams);
            }
            else
            {
                outBmp.Save(destFile, thisFormat);
            }

            return true;
        }
        catch (Exception e)
        {
            throw e;
        }
        finally
        {
            imgSource.Dispose();
            outBmp.Dispose();
        }
    }
}

```

总结：本文主要是根据给同学讲解ListView控件显示图片写的一篇文章,同时存在一个缺点图片可能被扯拉变形,而且代码中打开ListView图片时有个"打开时间",主要是通过Stopwatch记录批量打开图片所需时间,如果打开大量图片时我希望使用并行的方法实现,与其进行时间对比.同时如果对图片处理感兴趣的同学(C++通过Bitmap打开变换)自己可以去研究.我希望的显示效果想Google Picasa一样快速批量显示(研究ing).


下载地址:<http://download.csdn.net/detail/eastmount/8021077>

最后希望文章对大家有所帮助,如果有错误或不足之处,请海涵~

(By:Eastmount 2014-10-10 中午13点 原创CSDN <http://blog.csdn.net/eastmount/>)

👍 点赞 5 ☆ 收藏 ➦ 分享 ...



Eastmount  博客专家

发布了450 篇原创文章 · 获赞 6227 · 访问量 496万+

他的留言板

关注