**实 验 报 告 四**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **课 程** | 桌面应用程序开发 | **实验项目** | C#中异常的处理 | **成 绩** |  |
| **专业班级** | **软件工程大一班** | **实验日期** | **2019.10.27** | **批阅日期** |  |
| **姓 名** | **李毅** | **学 号** | **201731062133** | **指导教师** | 王世元 |
| **姓 名** | **李文毅** | **学 号** | **201731062208** |  |  |
| **姓 名** | **江镇岐** | **学 号** | **201731062210** |  |  |

**一【实验目的】**

1. 掌握C#中异常的使用。

**二【实验内容】**

**【实验1】图片添加水印软件（软件运行界面截屏及源代码）**

1. 载入原始图片功能
2. 载入水印图片功能
3. 调节水印位置功能
4. 设置水印透明度功能
5. 保存加了水印图片功能
6. 批量载入图片批量加水印功能



一、运行截图：

**运行主界面截图：**



**1、单张图片加水印：**

**载入原始图片截图：**

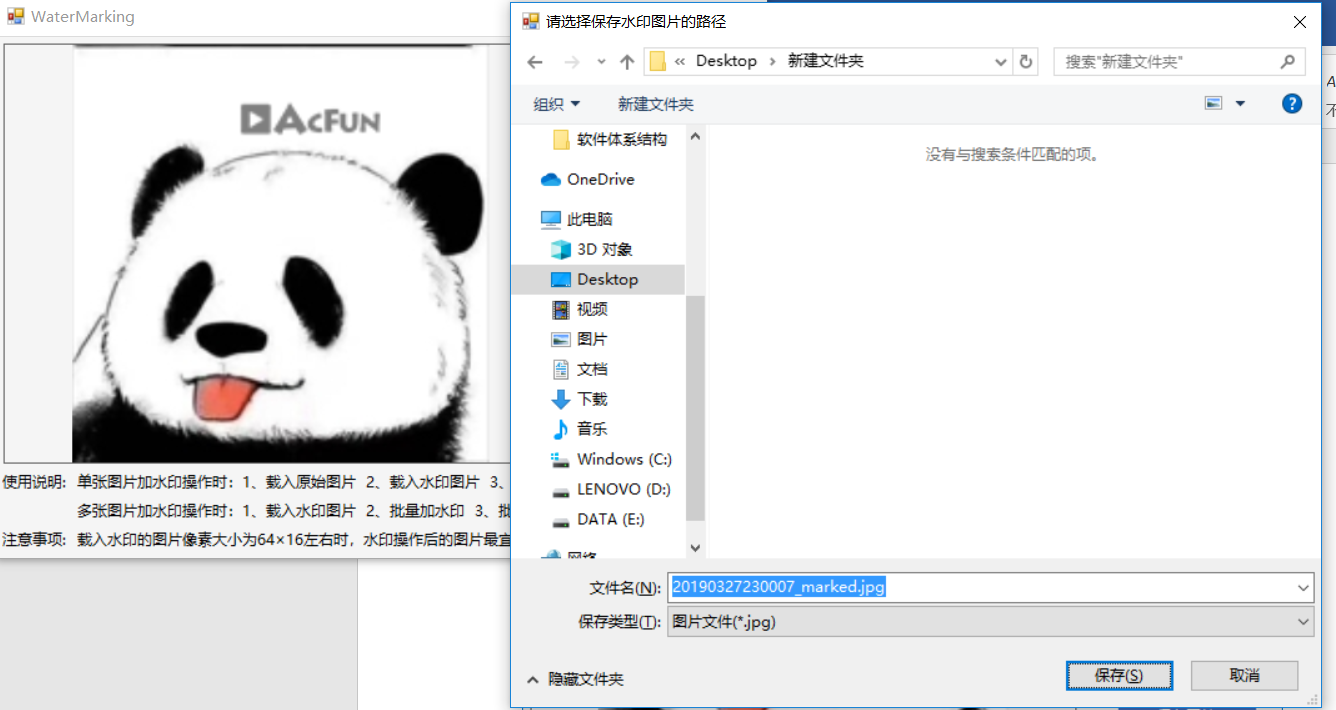




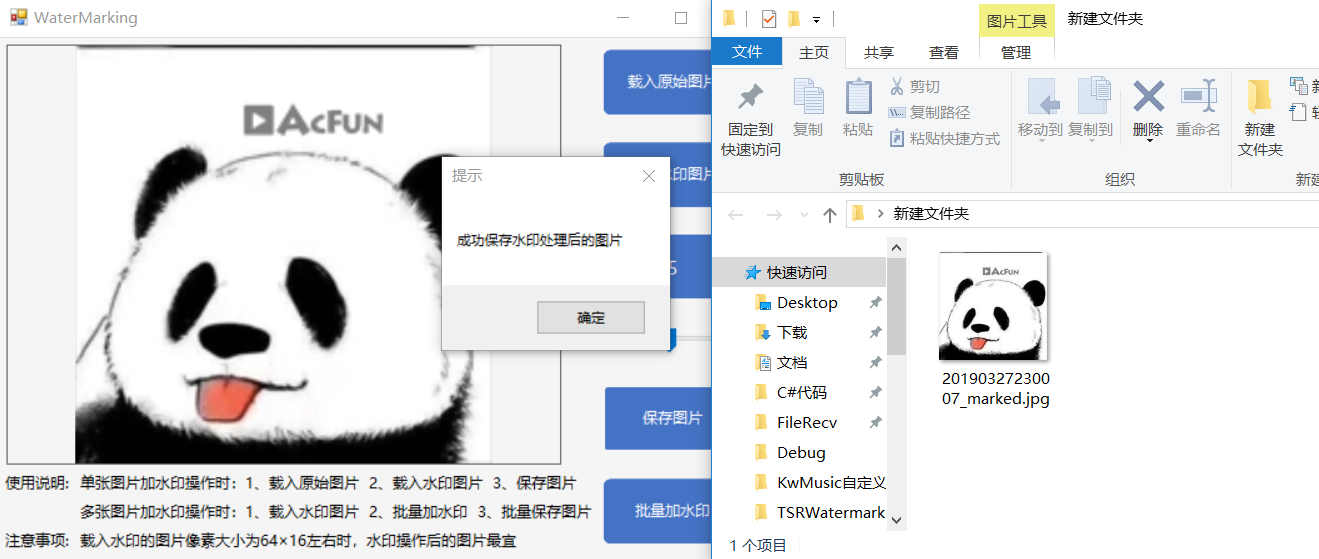
**移动水印位置且修改透明度后截图：**



**保存图片界面截图：**

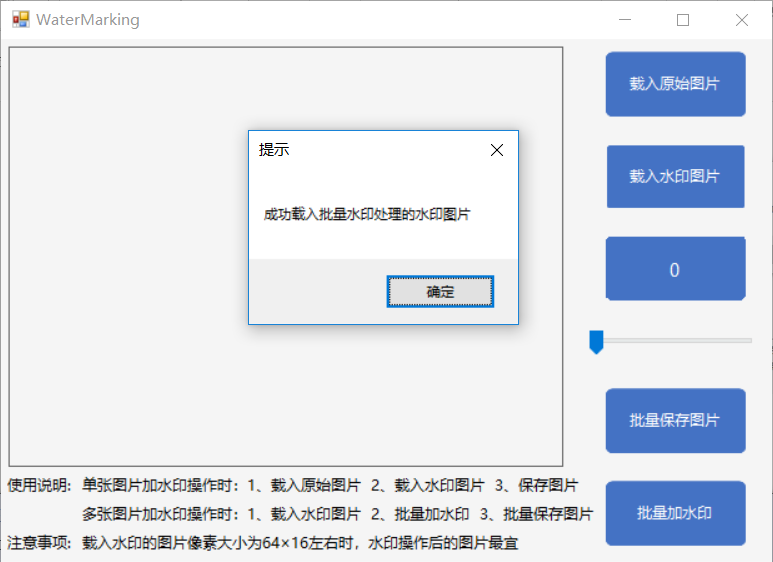


**保存成功后截图：**

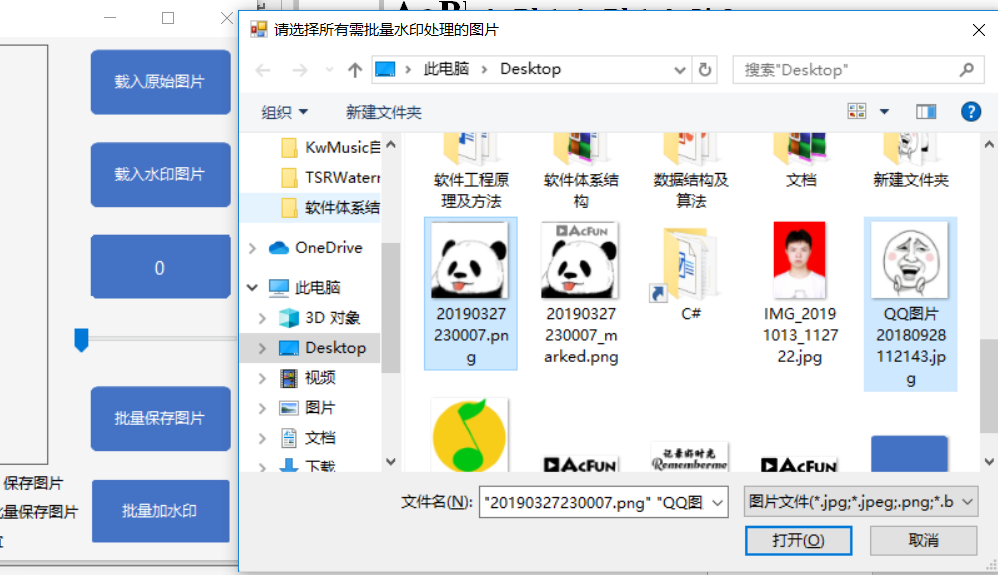


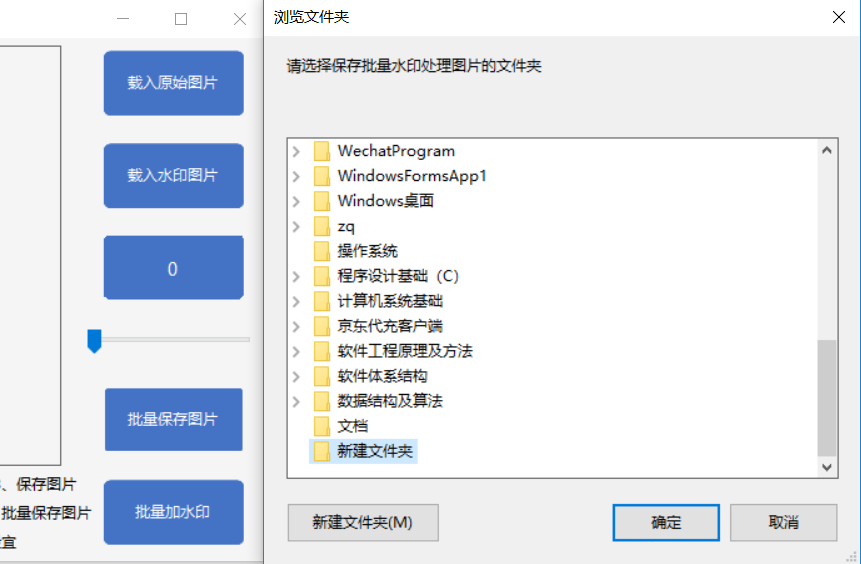
**2、批量加水印操作**

**选择水印：**

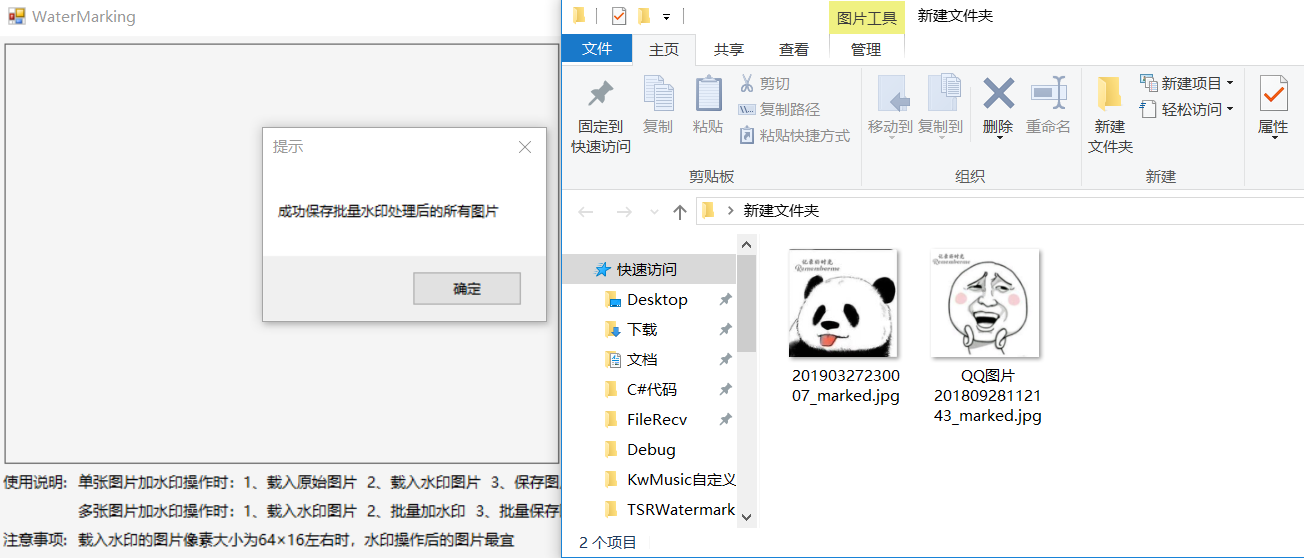


**选择批量水印处理的图片：**



**选择保存批量处理图片的文件夹：** 

**成功保存批量水印处理后的所有图片：**



二、代码：

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Drawing.Imaging;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TSRWatermarkImage

{

public partial class frmWaterMarking : Form

{

public frmWaterMarking()

{

InitializeComponent();

}

private Image imgOriginal;

private Image imgMark;

private Image imgWaterMarked;

private string imgOriginalPath;

private string imgMarkPath;

private int xPosOfMk=10;//默认水印起始位置x坐标

private int yPosOfMk=10;//默认水印起始位置y坐标

private float transParency = 0.5f;//默认水印透明度为半透明

private bool isLoadimgOriginal=false;//是否载入了原始需水印处理的图片

private bool isLoadMark=false;//是否载入了水印图片

private Image Watermark(string imgOriginalPath, string imgMarkPath)

{

imgOriginal = Image.FromFile(imgOriginalPath);

int oglHeight = imgOriginal.Height;

int oglWidth = imgOriginal.Width;

Bitmap bmpOriginal = new Bitmap(oglWidth, oglHeight, PixelFormat.Format24bppRgb);

bmpOriginal.SetResolution(72, 72);//设置分辨率

Graphics gOriginal = Graphics.FromImage(bmpOriginal);

imgMark = new Bitmap(imgMarkPath);

int mkHeight = imgMark.Height;

int mkWidth = imgMark.Width;

gOriginal.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;//消除锯齿

gOriginal.DrawImage(imgOriginal,

new Rectangle(0, 0, oglWidth, oglHeight),

0,

0,

oglWidth,

oglHeight,

GraphicsUnit.Pixel

);

//根据前面修改后的照片创建一个Bitmap。把这个Bitmap载入到一个新的Graphic对象。

Bitmap bmpWaterMarking = new Bitmap(bmpOriginal);

bmpWaterMarking.SetResolution(imgOriginal.HorizontalResolution, imgOriginal.VerticalResolution);//设置分辨率

Graphics gWaterMarking = Graphics.FromImage(bmpWaterMarking);

//通过定义一个ImageAttributes 对象并设置它的两个属性，我们就是实现了两个颜色的处理，以达到半透明的水印效果。

//处理水印图象的第一步是把背景图案变为透明的(Alpha=0, R=0, G=0, B=0)。我们使用一个Colormap 和定义一个RemapTable来做这个。

//就像前面展示的，我的水印被定义为100%绿色背景，我们将搜到这个颜色，然后取代为透明。

ImageAttributes imageAttributes = new ImageAttributes();

ColorMap colorMap = new ColorMap();

colorMap.OldColor = Color.FromArgb(255, 0, 255, 0);

colorMap.NewColor = Color.FromArgb(0, 0, 0, 0);

ColorMap[] remapTable = { colorMap };

//第二个颜色处理用来改变水印的不透明性。

//通过应用包含提供了坐标的RGBA空间的5x5矩阵来做这个。

//通过设定第三行、第三列为0.3f我们就达到了一个不透明的水平。结果是水印会轻微地显示在图象底下一些。

imageAttributes.SetRemapTable(remapTable, ColorAdjustType.Bitmap);

float[][] colorMatrixElements = {

new float[] {1.0f, 0.0f, 0.0f, 0.0f, 0.0f},

new float[] {0.0f, 1.0f, 0.0f, 0.0f, 0.0f},

new float[] {0.0f, 0.0f, 1.0f, 0.0f, 0.0f},

new float[] {0.0f, 0.0f, 0.0f, transParency, 0.0f},

new float[] {0.0f, 0.0f, 0.0f, 0.0f, 1.0f}

};

ColorMatrix wmColorMatrix = new ColorMatrix(colorMatrixElements);

imageAttributes.SetColorMatrix(wmColorMatrix, ColorMatrixFlag.Default, ColorAdjustType.Bitmap);

//两个颜色处理加入到imageAttribute对象

gWaterMarking.DrawImage(imgMark,

new Rectangle(xPosOfMk, yPosOfMk, mkWidth \* (imgOriginal.Width / 180), mkHeight \* (imgOriginal.Height / 180)),

0,

0,

mkWidth,

mkHeight,

GraphicsUnit.Pixel,

imageAttributes

);

gOriginal.Dispose();

gWaterMarking.Dispose();

return bmpWaterMarking;

}

private string oglFileName;

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

{

OpenFileDialog ofdOriginal = new OpenFileDialog();

ofdOriginal.Title = "请选择您想要添加水印的图片";

ofdOriginal.Filter = "图片文件(\*.jpg;\*.jpeg;.png;\*.bmp;\*.ico;)|\*.jpg;\*.jpeg;\*.png;\*.bmp;\*.ico";//文件格式筛选

ofdOriginal.RestoreDirectory = true;//记忆关闭窗口前选定的路径

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

{

isLoadimgOriginal = true;//载入了需打水印的图片

isBatchProcessing = false;//载入了单张水印处理的图片后不可批量加水印处理

oglFileName = System.IO.Path.GetFileName(ofdOriginal.FileName);//文件名和扩展名

imgOriginalPath = ofdOriginal.FileName;//选定图片的路径

imgOriginal = Image.FromFile(ofdOriginal.FileName);

picDisplay.BackgroundImage = imgOriginal;

MessageBox.Show("成功载入原始图片","提示");

}

}

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

{

if (isLoadimgOriginal&&!isBatchProcessing)//选择单张图片水印处理的水印

{

OpenFileDialog ofdMark = new OpenFileDialog();

ofdMark.Title = "请选择您的水印图片";

ofdMark.Filter = "图片文件(\*.jpg;\*.jpeg;.png;\*.bmp;\*.ico;)|\*.jpg;\*.jpeg;\*.png;\*.bmp;\*.ico";//文件格式筛选

ofdMark.RestoreDirectory = true;//记忆关闭窗口前选定的路径

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

{

isLoadMark = true;//载入了水印图片

imgMarkPath = ofdMark.FileName;//水印路径

imgMark = Image.FromFile(ofdMark.FileName);

xPosOfMk = 5;//载入时默认水印起点x轴坐标

yPosOfMk = 5;//载入时默认水印起点y轴坐标

transParency = 1.0f;//水印透明度为不透明

imgWaterMarked = Watermark(imgOriginalPath, imgMarkPath);

picDisplay.BackgroundImage = imgWaterMarked;

MessageBox.Show("成功载入水印图片", "提示");

}

}

else if(isBatchProcessing)//选择批量水印处理的水印图片

{

OpenFileDialog ofdMarkforBatch = new OpenFileDialog();

ofdMarkforBatch.Title = "请选择批量水印处理的水印图片";

ofdMarkforBatch.Filter = "图片文件(\*.jpg;\*.jpeg;.png;\*.bmp;\*.ico;)|\*.jpg;\*.jpeg;\*.png;\*.bmp;\*.ico";//文件格式筛选

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

{

isLoadMark = true;

imgMarkPath = ofdMarkforBatch.FileName;

MessageBox.Show("成功载入批量水印处理的水印图片", "提示");

}

}

}

bool isDown;

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

{

if (isDown && isLoadimgOriginal && isLoadMark)

{

int imgWidth = imgOriginal.Width;//图片宽度

int imgHeight = imgOriginal.Height;//图片高度

PropertyInfo rectangleProperty = this.picDisplay.GetType().GetProperty("ImageRectangle", BindingFlags.Instance | BindingFlags.NonPublic);

Rectangle rectangle = (Rectangle)rectangleProperty.GetValue(this.picDisplay, null);

int currentWidth = rectangle.Width;//缩放后宽度

int currentHeight = rectangle.Height;//缩放后高度

double rateWidth = (double)currentWidth / (double)imgWidth;//宽缩放比例

double rateHeight = (double)currentHeight / (double)imgHeight;//高缩放比例

int blackLeftWidth = (currentWidth == this.picDisplay.Width) ? 0 : (550 - currentWidth) / 2;//左边及右边空白部分宽度

int blackTopHeight = (currentHeight == this.picDisplay.Height) ? 0 : (420 - currentHeight) / 2;//顶部及底部空白部分高度

xPosOfMk = Convert.ToInt32((e.X - blackLeftWidth) / rateWidth);//画水印起点x坐标

yPosOfMk = Convert.ToInt32((e.Y - blackTopHeight) / rateHeight);//画水印起点y坐标

//对水印起点xPosOfMk，yPosOfMk的限制

if (xPosOfMk <= 0)

{

xPosOfMk = 0;

}

if (xPosOfMk >= imgWidth - ((blackLeftWidth + imgMark.Width) / rateWidth))

{

xPosOfMk = Convert.ToInt32(imgWidth - ((blackLeftWidth + imgMark.Width) / rateWidth));

}

if (yPosOfMk <= 0)

{

yPosOfMk = 0;

}

if (yPosOfMk >= imgHeight - ((blackTopHeight + imgMark.Height) / rateWidth))

{

yPosOfMk = Convert.ToInt32(imgHeight - ((blackTopHeight + imgMark.Height) / rateWidth));

}

imgWaterMarked = Watermark(imgOriginalPath, imgMarkPath);

picDisplay.BackgroundImage = imgWaterMarked;

}

}

private void picDisplay\_MouseDown(object sender, MouseEventArgs e)

{

if (e.Button == System.Windows.Forms.MouseButtons.Left)

{

isDown = true;

}

}

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

{

isDown = false;

}

private void lblTrackBar\_Paint(object sender, PaintEventArgs e)

{

DrawRoundRec(e.Graphics, this.lblTrackBar);//将lblTrackBar圆角矩形化，效果不很好，坐标不太好调配

}

private void DrawRoundRec(Graphics g, Label label)//圆角矩形

{

float X = float.Parse(label.Width.ToString());

float Y = float.Parse(label.Height.ToString()) - 1;

PointF[] points =

{

new PointF(2,-1),

new PointF(X-2,-1),

new PointF(X-1,1),

new PointF(X,2),

new PointF(X-1,Y-3),

new PointF(X-2,Y-2),

new PointF(X-3,Y-1),

new PointF(2,Y),

new PointF(3,Y-1),

new PointF(0,Y-3),

new PointF(0,2),

new PointF(0,2)

};

GraphicsPath path = new GraphicsPath();

path.AddLines(points);

label.Region = new Region(path);

Pen pen = new Pen(Color.FromArgb(150, Color.FromArgb(68, 114, 196)), 1);

pen.DashStyle = DashStyle.Solid;

g.DrawPath(pen, path);

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

lblTrackBar.Text = trackBar1.Value.ToString();

}

private void trackBar1\_ValueChanged(object sender, EventArgs e)

{

if (isLoadimgOriginal && isLoadMark)

{

transParency = Math.Abs((float.Parse(lblTrackBar.Text) / 10) - 1.0f);//透明度

imgWaterMarked = Watermark(imgOriginalPath, imgMarkPath);

picDisplay.BackgroundImage = imgWaterMarked;

}

}

private string getFileName(string oglFileName)//获取原图像名称

{

string str = "";

int a = oglFileName.LastIndexOf('.');

str = oglFileName.Substring(0, a);

return str;

}

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

{

if (isLoadimgOriginal && isLoadMark)//载入了原始图片且载入了水印

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Title = "请选择保存水印图片的路径";

sfd.RestoreDirectory = true;//记忆关闭窗口前选定的路径

sfd.Filter = "图片文件(\*.jpg)|\*.jpg";

sfd.FileName = getFileName(oglFileName) + "\_marked";//文件名为原来的图片名称+\_marked

sfd.DefaultExt = ".jpg";

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

{

imgWaterMarked.Save(sfd.FileName, System.Drawing.Imaging.ImageFormat.Jpeg);//将图片保存

MessageBox.Show("成功保存水印处理后的图片", "提示");

}

}

if (isBatchProcessing && isLoadMark)//允许批量加水印操作且载入了水印

{

FolderBrowserDialog fbd = new FolderBrowserDialog(); //显示对话框下方的创建新文件夹按钮

fbd.Description = "请选择保存批量水印处理图片的文件夹";

fbd.ShowNewFolderButton = true;//允许新建文件夹出现在文件夹浏览器对话框中

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

{

for (int i = 0; i < imgsWaterMarked.Count; i++)

{

imgsWaterMarked[i].Save(fbd.SelectedPath + "\\" + imgFileName[i] + "\_marked.jpg", System.Drawing.Imaging.ImageFormat.Jpeg);

imgsWaterMarked[i].Dispose();//保存后将该资源释放

}

//清空数组中上一次保存的图片或图片路径

Array.Clear(imgFilePath, 0, imgFilePath.Length);

imgFileName.Clear();

imgsWaterMarked.Clear();

isLoadMark = false;//保存批量水印处理的图片后将isLoadMark置为false

isBatchProcessing = true;//保存批量水印处理的图片后将isBatchProcessing置为true

MessageBox.Show("成功保存批量水印处理后的所有图片", "提示");

}

}

}

private string[] imgFilePath;//批量选择时的所有图片路径

private List<string> imgFileName = new List<string>();//批量选择时的所有图片名称,不含后缀名

private List<Image> imgsWaterMarked = new List<Image>();//批量水印处理后保存的所有图片

private bool isBatchProcessing = true;//默认允许批量处理

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

{

if(isBatchProcessing&&isLoadMark)//允许批量操作且载入了水印

{

OpenFileDialog ofdBatch = new OpenFileDialog();

ofdBatch.Multiselect = true;//是否允许选择多个文件

ofdBatch.Title = "请选择所有需批量水印处理的图片";

ofdBatch.Filter = "图片文件(\*.jpg;\*.jpeg;.png;\*.bmp;\*.ico;)|\*.jpg;\*.jpeg;\*.png;\*.bmp;\*.ico";//文件格式筛选

ofdBatch.RestoreDirectory = true;//记忆关闭窗口前选定的路径

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

{

imgFilePath = ofdBatch.FileNames;//将所有的选择的图片路径存于imgFilePath中

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

{

imgOriginalPath = imgFilePath[i];//图片路径

imgFileName.Add(getFileName(System.IO.Path.GetFileName(imgFilePath[i])));//图片名称，不含扩展名

imgOriginal = Image.FromFile(imgOriginalPath);

imgWaterMarked = Watermark(imgOriginalPath, imgMarkPath);

imgsWaterMarked.Add(imgWaterMarked);

}

btnSave.Text = "批量保存图片";

MessageBox.Show("成功载入批量水印处理的图片", "提示");

}

}

}

}

}