

徐年, 陆文静, 何奕成, 等. 地震应急平台电子传真报警处理系统设计与实现[J]. 华南地震, 2021, 41(4): 80–84. [XU Nian, LU Wenjing, HE Yicheng, et al. Design and Implementation of Electronic Fax Alarm Processing System for Earthquake Emergency Platform [J]. South China journal of seismology, 2021, 41(4): 80–84]

地震应急平台电子传真报警处理系统设计与实现

徐 年, 陆文静, 何奕成, 蒋 成, 杜 航

(江苏省地震局, 南京 210014)

摘要: 设计开发江苏省地震应急平台电子传真报警处理系统, 当省政府领导批示或文件到达时, 电子传真将发出警报提示, 提醒值班人员及时处理, 并自动下载打印输出。该系统旨在解决江苏省应急平台电子传真信息接收系统没有报警功能、信息接收和处理延迟的问题, 对提高江苏省地震应急处置工作的时效性具有重要现实意义。

关键词: 电子传真; 报警; 应急平台

中图分类号: P315.9

文献标识码: A

文章编号: 1001-8662(2021)04-0080-05

DOI: 10.13512/j.hndz.2021.04.10

Design and Implementation of Electronic Fax Alarm Processing System for Earthquake Emergency Platform

XU Nian, LU Wenjing, HE Yicheng, JIANG Cheng, DU Hang

(Jiangsu Earthquake Agency, Nanjing 210014, China)

Abstract: The paper designs and develops the electronic fax alarm processing system for the earthquake emergency platform of Jiangsu province. When the provincial government leaders' instructions or documents arrive, the electronic fax will issue an alarm to remind the on-duty staff to deal with it in time and automatically download the printout. The system is designed to solve the problem that the electronic fax information receiving system of the emergency platform of Jiangsu province has no alarm function, information receiving and processing delay, which has important practical significance for improving the timeliness of the earthquake emergency response work of Jiangsu Province.

Keywords: Electronic fax; Alarm; Emergency platform

0 引言

地震应急值班工作最重要的一条就是信息处

理的时效性。而地震应急值班有效运转最重要的一环就是地震信息的上传下达, 特别是上级领导的批示以及上级部门传达的文件, 及时接受上级

收稿日期: 2021-06-08

基金项目: 江苏省地震局青年基金(201806)

作者简介: 徐年(1982-), 男, 高级工程师, 现主要从事地震应急技术研究工作。

E-mail: xunian0215@163.com

领导的批示以及上级部门传达的文件,就可以最大程度上保证地震事件的有效快速处置,把地震事件所造成的影响降至最低程度。针对江苏省地震应急平台电子传真信息接收系统没有报警功能、信息接收处理延迟的问题,本文开发了一个江苏省地震应急平台电子传真报警处理系统。当系统接收到省政府领导批示或文件时,系统自动报警,提示值班人员立即处理并自动保存到本地并打印输出,提高江苏省地震局应急平台电子传真信息接收和处理的实效性,从而可以更好地开展地震应急处置工作。

1 系统总体设计

本系统主要基于 .net Framework 框架,使用 C# 编程语言进行后台开发。

核心部分在图像对比,图片对比方法采用“感知哈希算法”。该算法将为图片生成一个指纹(字符串格式),两张图片的指纹越相似,说明两张图片就越相似^[1-2]。

其实现原理如下:

- (1)缩小图片尺寸,将图片缩小到8*8的尺寸,总共64个像素。这一步的作用是去除各种图片尺寸和图片比例的差异,只保留结构、明暗等基本信息;
- (2)转为灰度图片,将缩小后的图片,转为64级灰度图片;
- (3)计算灰度平均值,计算图片中所有像素的灰度平均值;
- (4)比较像素的灰度,将每个像素的灰度与平均值进行比较,如果大于或等于平均值记为1,小于平均值记为0;
- (5)计算哈希值,将上一步的比较结果,组合在一起就构成了一个64位的二进制整数,这就是这张图片的指纹;
- (6)对比图片指纹,得到图片的指纹后,就可以对比不同的图片的指纹,计算出64位中有多少位是不一样的。如果不相同的数据位数不超过5,就说明两张图片很相似,如果大于10,说明它们是两张不同的图片。

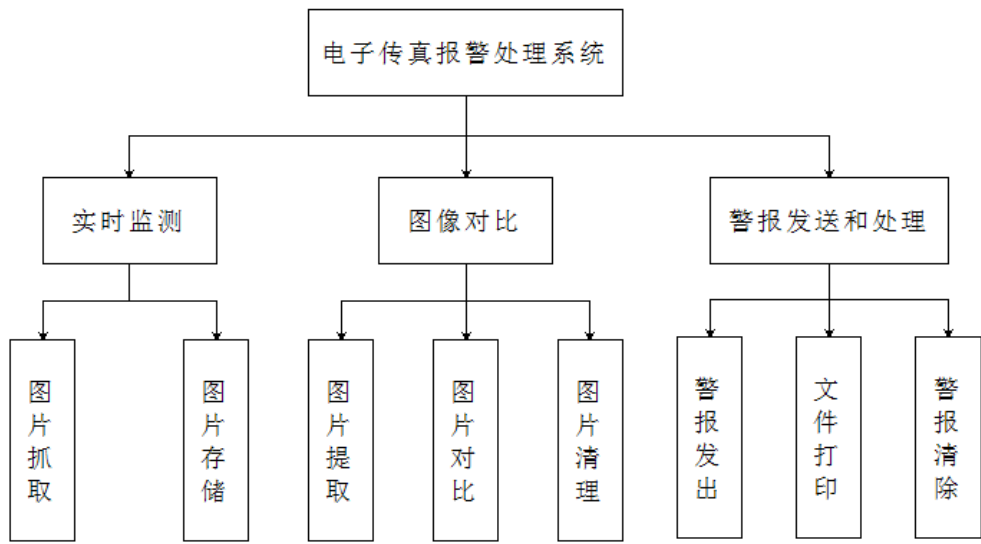


图1 系统功能结构图
Fig.1 The diagram of system function structure

2 系统实现

该系统包括3个模块(如图1所示):分别为实时监测、图像对比分析、警报发送和处理模块^[3],每个模块的功能如下:

2.1 实时监测模块

- (1)抓取整个屏幕图片

每一分钟,对整个屏幕图片进行抓取。核心代码:
//创建一个屏幕大小的标准位图
Graphics gra = Graphics.FromImage(myImage);
//从屏幕到绘图的位块传输
gra.CopyFromScreen (new Point (0, 0) , new Point (0, 0) , new Size (Screen.PrimaryScreen.Bounds.Width, Screen.PrimaryScreen.Bounds.Height));

(2)对截取的全屏屏幕图片进行指定区域截取
每一分钟,对屏幕的指定区域进行截取,截取的图片保存在本地电脑里的指定图像存储文件夹。需要用到辅助工具PS,获取指定区域的像素坐标。核心代码:

//创建作图区域

```
Graphics graphic = Graphics.FromImage
(bitmap);
```

//截取原图相应区域写入作图区

```
graphic.DrawImage(fromImage, 0, 0, new
Rectangle(offsetX, offsetY, width, height),
GraphicsUnit.Pixel);
```

//从作图区生成新图

```
Image saveImage = Image.FromHbitmap(bitmap.
GetHbitmap());
```

2.2 图像对比分析模块

将最近两张图像进行对比,判断是否有差异。每一分钟,从图像存储文件夹内取最近时间节点的两张图片,通过对比两张图片的数据流来判断图片是否一致,并将对比异常结果发送给报警模块,同时反馈给实时监测模块。比对完成后清理文件夹内的多余图片,将前面的图片删除,保留最近时间的一张图片。这里图片对比用到的算法为“感知哈希算法”^[4-5],核心代码:

#region 第一步缩小图片尺寸(8*8, 64个像素)

```
public Image ReduceSize(String path, int width = 8,
int height = 8)
```

```
{
//获取图片
```

```
Image SourceImage = Image.FromFile(path);
```

//将图像缩小为指定的尺寸

```
Image image = SourceImage.GetThumbnailImage
(width, height, () => { return false; }, IntPtr.
Zero);
```

//释放资源

```
SourceImage.Dispose();
```

```
return image;
```

```
}
```

#endregion

#region 第二步将图片转化为64级的灰度图片

```
public Byte[] ReduceColor(Image image)
```

```
{
```

//将图片转化为位图

```
Bitmap bitmap = new Bitmap(image);
```

//存储每个像素的灰度值

```
Byte[] grayValues = new Byte[image.
Width*image.Height];
```

//获取每个像素的灰度值

```
for (int x = 0; x < image.Width; x++)
```

```
{
```

```
for (int y = 0; y < image.Height; y++)
```

```
{
```

//通过像素坐标获取对应像素值

```
Color color = bitmap.GetPixel(x, y);
```

```
byte grayValue = (byte)((color.R * 30 + color.G *
59 + color.B * 11) / 100);
```

```
grayValues[x * image.Width + y] = grayValue;
```

```
}
```

```
}//end for
```

```
return grayValues;
```

```
}
```

#endregion

#region 第三步计算图片中所有像素的灰度平均值

```
public Byte CalcAverage(byte[] values)
```

```
{
```

```
int sum = 0;
```

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

```
{
```

```
sum += (int) values[i];
```

```
}
```

```
return Convert.ToByte(sum/values.Length);
```

```
}
```

#endregion

#region 第四步比较像素的灰度并计算图片的哈希值

```
public String ComputeBits(byte[] values, byte
averageValue)
```

```
{
```

```
char[] result = new char[values.Length];
```

//将每个像素的灰度值与所有像素灰度平均值进行比较

//如果小于灰度平均值记为0,大于等于灰度平均值记为1

//将上一步的比较结果,组合在一起,就构成了一个64位的二进制整数,这就是这张图片的指纹

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

```
{
```

```

if (values[i] < averageValue)
{
    result[i] = '0';
}
else
{
    result[i] = '1';
}
}
return new string (result);
}
#endregion
#region 第五步通过哈希值比较两张图片的相似度
//比较两张图片的哈希值, 64位中有多少位是不一样的, 如果大于10, 则这两张图片是不同的图片
public Int32 CalcSimilarDegree(string a, string b)
{
    //存储两张图片哈希值中不同的位数
    int count = 0;
    //哈希值不等长, 则抛出异常
    if(a.Length != b.Length)
        throw new ArgumentException();
    for (int i = 0; i < a.Length; i++)
    {
        if (a[i] != b[i])
            count++;
    }
    return count;
}
#endregion
#region 第六步判断两张图片是否相同
public bool JudgePicSimilar(int count)
{
    if(count >=1)
        return false;
    else
        return true;
}
#endregion

```

2.3 警报发送和处理模块

2.3.1 警报发送

当收到图片对比异常结果反馈时, 播放声音报警。核心代码:

```

player.Load(); //同步加载声音
player.PlayLooping(); //循环播放模式
鼠标模拟点击
当收到图片对比异常结果反馈时, 开启鼠标模拟功能。核心代码:
//设置鼠标的坐标
SetCursorPos(X, Y);
//鼠标模拟, 单击跳转页面
mouse_event((int)(MouseEventFlags.LeftDown |
MouseEventFlags.Absolute), X, Y, 0, IntPtr.Zero);
mouse_event((int)(MouseEventFlags.LeftUp |
MouseEventFlags.Absolute), X, Y, 0, IntPtr.Zero);

```

2.3.2 打印文件

鼠标模拟结束, 则开始打印鼠标模拟下载好的文件。将桌面文件按时间顺序排序, 获取最新文件路径, 通过该路径打印输出下载好的文件。核心代码:

```

PrinterSettings settings = new PrinterSettings();
PrintDocument pd = new PrintDocument();
settings.PrinterName = pd.PrinterSettings.PrinterName;
settings.PrintToFile = false;
//设置纸张大小(可以不设置, 取默认设置)3.90 in, 8.65 in
PaperSize ps = new PaperSize();
ps.RawKind = 9; //如果是自定义纸张, 就要大于118
PDFPrintSettings pdfPrintSettings = new PDFPrintSettings(settings);
pdfPrintSettings.PaperSize = ps;
pdfPrintSettings.PrinterSettings.Copies = 1;
file.Print(pdfPrintSettings);

```

3 系统测试

系统设计完成后(如图2所示), 联系省应急办工作人员进行联调联试, 均能做到系统接收到通知后, 自动发出报警提示并自动打印输出该文件(如图3所示)。

4 应用效果

2019年2月12日江苏省政府通过应急平台发

出省领导在《值班快报》第23期上的批示，系统可以在接收到该批示后，自动打印输出，从而值班人员可以及时报送到江苏省地震局办公室。2019年4月6日江苏南京溧水2.8级地震，2019年5月23日江苏常州金坛2.8级地震，2019年12月12日江苏盐城东台海域3.5级地震，2020年1月29日江苏苏州昆山2.2级地震以及2020年3月3日江苏南京鼓楼3.0级地震，这些地震后江苏省领导均作了批示，系统在接收到该批示后，自动打印输出，从而值班人员可以及时报送到江苏省地震局局办公室，这大大提升了江苏省地震局信息接收的速度，有利于江苏省地震局及时贯彻江苏省领导批示要求，开展震后相关工作。

5 结语

本文设计开发江苏省地震应急平台电子传真报警系统，是通过截取屏幕进行监控对比来实现的，后期可以进一步优化，如可以直接进行传真文件目录的监控。当有省政府领导批示或文件时，电子传真将有自动报警功能，及时提示值班人员进行处理，同时可自动下载、保存到本地并打印输出。这

对于保证信息畅通、政令畅通和江苏省地震局及时开展地震应急相关处置工作都具有重要意义。

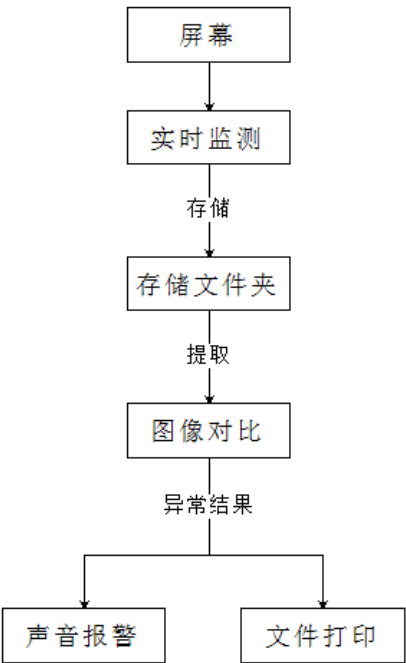


图2 技术路线流程图
Fig.2 The flow chart of technical route

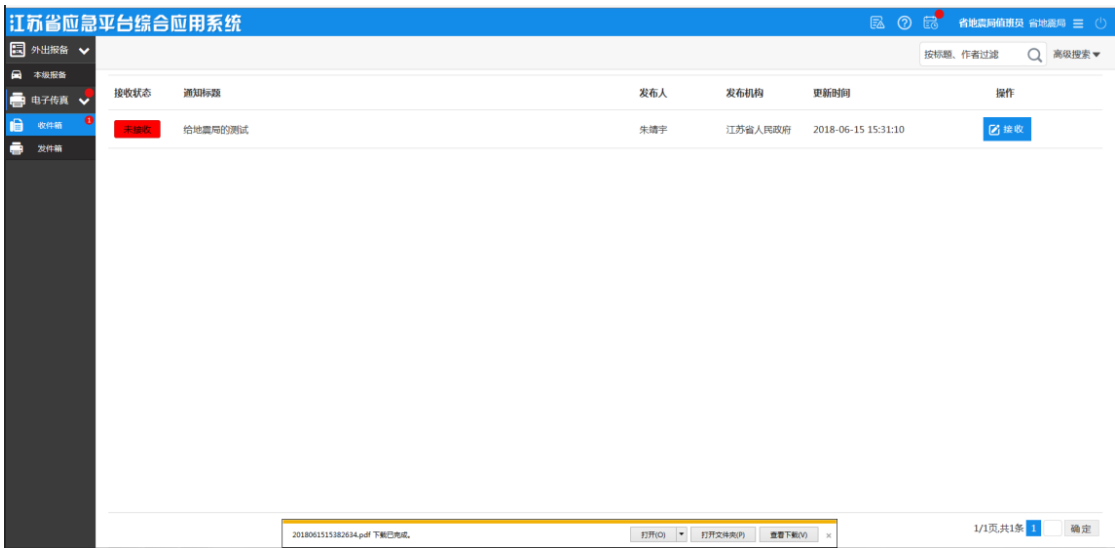


图3 系统测试图
Fig.3 The chart of system test

参考文献

[1] 钱奇,王赞,曹荣辉.电子值班综合报警管理系统的研究[J].中国电力教育,2019(11):251-253.

[2] 俞叶萍,李棋,潘万欣.驾驶室航行值班报警系统的实现[J].上海船舶运输科学研究所学报,2011(12):150-154.

[3] 刘凌,韩林涛,王众全.电子值班系统的设计与实现[J].华中电力,2007(6): 26-29.

[4] 金哲凡,金峰,石教英.统一消息和统一消息邮件的研究和实现[J].计算机工程,2003(13):8-13.

[5] 柳邦声.驾驶室航行值班报警系统(BNWS)的组成和安装[J].航海技术,2010(3):41-43.