

# 智能人流量检测报警系统

姓名及贡献：软件 6 班 乔翱（服务器端编写，程序、报告汇总）

软件 6 班 白阜玉（客户端应用程序编写）

软件 6 班 张璇（客户端应用程序编写）

软件 6 班 刘耀鹏（服务器端应用程序编写）

软件 6 班 安琦（程序测试修改）

软件 6 班 赵晓晴（报告编写，程序测试）

课程名称：计算机网络原理（三级项目）

指导教师：穆运峰 郝晓冰 余扬 刘佳新

2021 年 5 月



# 智能人流量检测报警系统

乔翱、白阜玉、张璇、刘耀鹏、安琪、赵晓晴

（燕山大学 软件学院）

## 摘要

本项目为智能人流量检测报警系统，采用 C# 中的 winform 和 socket 编程实现，客户端通过无人机捕捉信息，将人流数据发送给服务端，服务端接受数据，发出命令控制无人机去人流量较大的地方发出警报。本报告展示了本小组完成计算机网络三级项目的整个过程，包括应用需求、项目设计、项目实施、项目测试、总结五个方面。

**关键词：**人流量检测、报警系统、winform、socket 编程

# 目录

|                 |    |
|-----------------|----|
| 1 应用需求.....     | 5  |
| 2 项目设计.....     | 5  |
| 2.1 系统组成.....   | 5  |
| 2.2 硬件设计.....   | 6  |
| 2.3 软件设计.....   | 7  |
| 2.3.1 服务器端..... | 7  |
| 2.3.2 客户端.....  | 7  |
| 3 项目实施.....     | 8  |
| 4 项目测试.....     | 16 |
| 5 总结.....       | 19 |
| 6 参考文献.....     | 19 |

## 1 应用需求

人流量也被称作客流量，是商场、道路、车站等公共场所管理不可缺少的数据，通过它可以获得这些场所完整的运行状况，有效地组织运营工作。随着现代社会经济的发展和城市化进程的加快，人口密度越来越大，在一些公共场所由于人群拥挤极易发生危险事故，所以对商场、车站等公共场所进行人流监控和分析具有重要的意义。

另外、由于当前疫情的影响，在很多的方面由于疫情防控的要求都必须严格遵守防控规则，“戴口罩，不聚集”就是疫情防控非常重要的一点。基于无人机的智能人流量检测报警系统可以对商场等场所进行不间断的自动监控、分析和报警，具有着重大的现实意义和应用前景。

例如在学校中，想去里仁楼前的操场跑跑步，奈何不知道人流多少。如果有一个无人机可以实时监测当下操场的人流量可谓极好。另外，疫情当下，人员最好不要聚集，如果无人机发现里仁楼前有人聚集的情况，则可以向聚集地发出警报，勒令人员疏散。

于是本小组设计了一款智能人流量检测报警系统，利用无人机实时监测人流密度，把人流信息实时传给总控台，另外总控平台可以查看人流分布图，并且发布预警信息，命令无人机去人流聚集点发布预警信息，驱散人群。

## 2 项目设计

### 2.1 系统组成

智能人流量检测报警系统主要由服务器和无人机两部分组成，通过以太网进行数据传输，。系统结构图如下图所示：

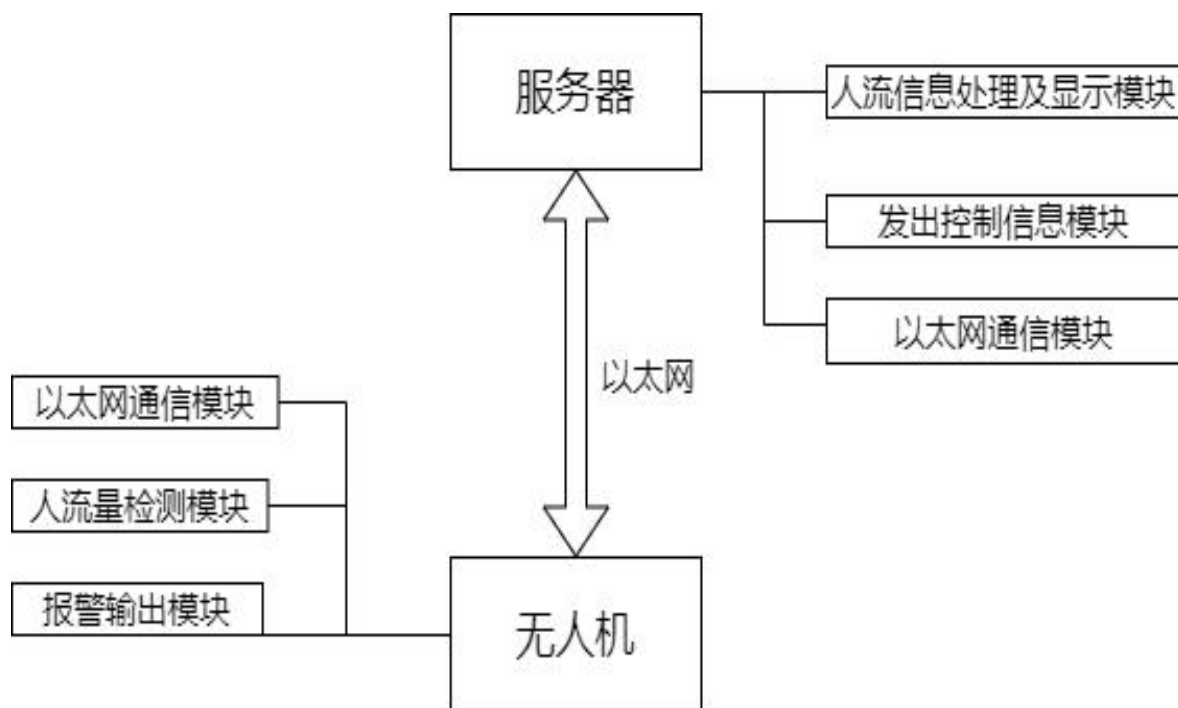


图 2-1 系统结构图

无人机实时检测监控范围内的人流量，通过摄像头进行图像捕捉，并且将图像数据发送给服务器。无人机主要由以太网通信模块、人流量检测模块、报警输出模块组成。

服务器端对无人机发来的人流量图像利用计算机视觉方面一定的算法（人脸识别算法）统计出人流量，并且绘制人流量热力图，另外服务器还可以发出控制信息控制无人机去某个区域发出警报信息。服务器端主要由人流信息处理及显示模块、发出控制信息模块、以太网通信模块组成。

## 2.2 硬件设计

对于系统硬件的设计，主要的硬件是无人机和总控台。总控台只是需要一台 PC 即可，在后期也可以开发出 APP 或者微信小程序，那么总控台也可以使用一款手机。而对于无人机来说，无人机需要配有高清摄像头以及嵌入通信设备，另外还需要有一个警示灯嵌入到无人机上面。除此之外，无人机还需要配有一个小音箱来播报警报信息。

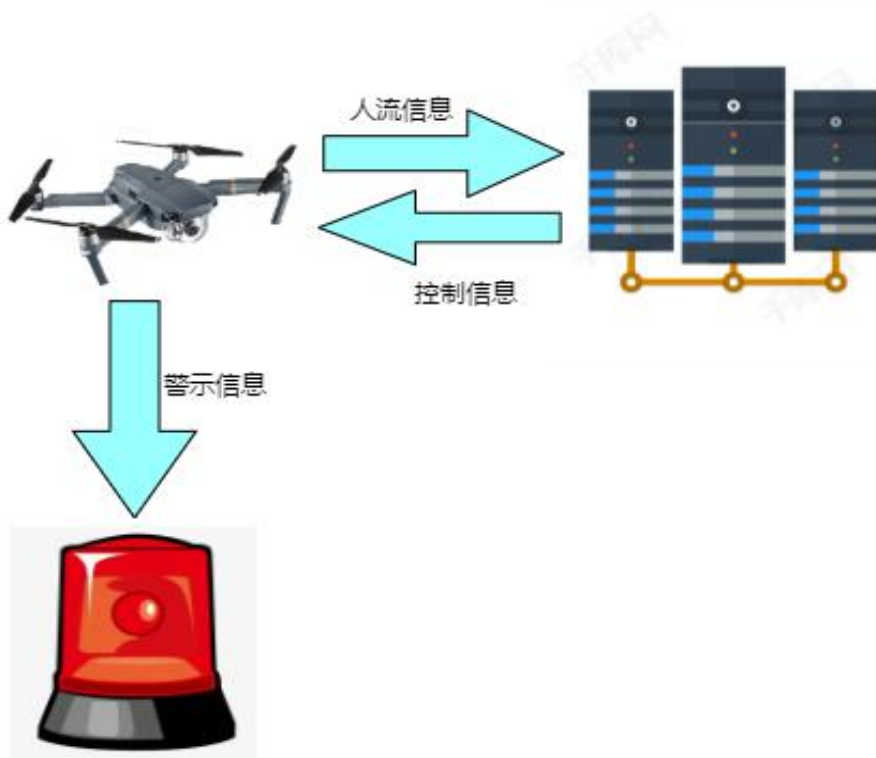


图 2-2 硬件设计

由于无人机是通过摄像头捕捉人流图像信息进行传输，所以对于无人机的摄像头有一定的要求，需要无人机配有超清镜头，另外无人机需要稳定飞行，所以无人机还需要有高速电机以及大容量电池来支持无人机长时间飞行，无人机的飞行是没有人控制的，其飞行通过超声波的视觉定位系统来飞行，此外无人机的飞行路线也是通过视觉定位系统进行选择。无人机需要向服务器端传送自己的位置，所以无人机配有雷达系统，可以知道无人机所处的位置。无人机在接受到服务器端发出的控制信息后，需要飞往指定区域发出警报信息，所以无人机还需要嵌入警报灯和小音箱。本小组设定的无人机结构图如下：



图 2-3 无人机结构图

由于此次三级项目重点在于对计算机网络原理的学习与理解，所以设计到的硬件系统都采用窗体模拟的方式来实现。利用随机数生成点来模拟人群，无人机也采用图片模拟的方式，而对于人流数量的计算也采用无人机所在区域有多少点来模拟。

## 2.3 软件设计

由于硬件部分实现起来较为困难，所以本小组对硬件部分采用窗体动画进行模拟，其中对于人群的模拟采用随机数在地图中生成点进行模拟，而对于无人机的飞行，也是采用图片移动动画的形式模拟这个过程。

### 2.3.1 服务器端

在服务器端，也就是总控平台显示客户端和服务端通信信息，以及无人机探测区域的人数分布情况，并用热力值图显示。服务器端可以控制无人机飞向的区域，当某一区域出现聚集情况时，服务器端可以控制无人机对该区域发送警告，以达到驱散人群，防控疫情的目的。

### 2.3.2 客户端

对于客户端也就是无人机，采用窗体进行模拟，其中地图、无人机及其飞行路线我们使用 C#窗体中的 pictureBox 实现，地图上人的位置我们使用红点表示。利用两个 timer 修改无人机的飞行路线和人群定位，每隔一定的时间，人群位置发生变化，模拟现实中人群的移动。无人机对某区域的人群聚集情况进行探测，利用 winsocket 通信，使无人机为客户端，与服务器端进行通信，返回某区域的人数情况，以此判断某区域是否发生人群聚集情况。

本项目重点内容是服务器和客户端的通信，在大多数编程语言中都采用 socket（套接字）进行网络编程，完成服务器和客户端数据通信，此过程有一个固定的流程，实现较为容易，socket 通信示意图如下图所示：

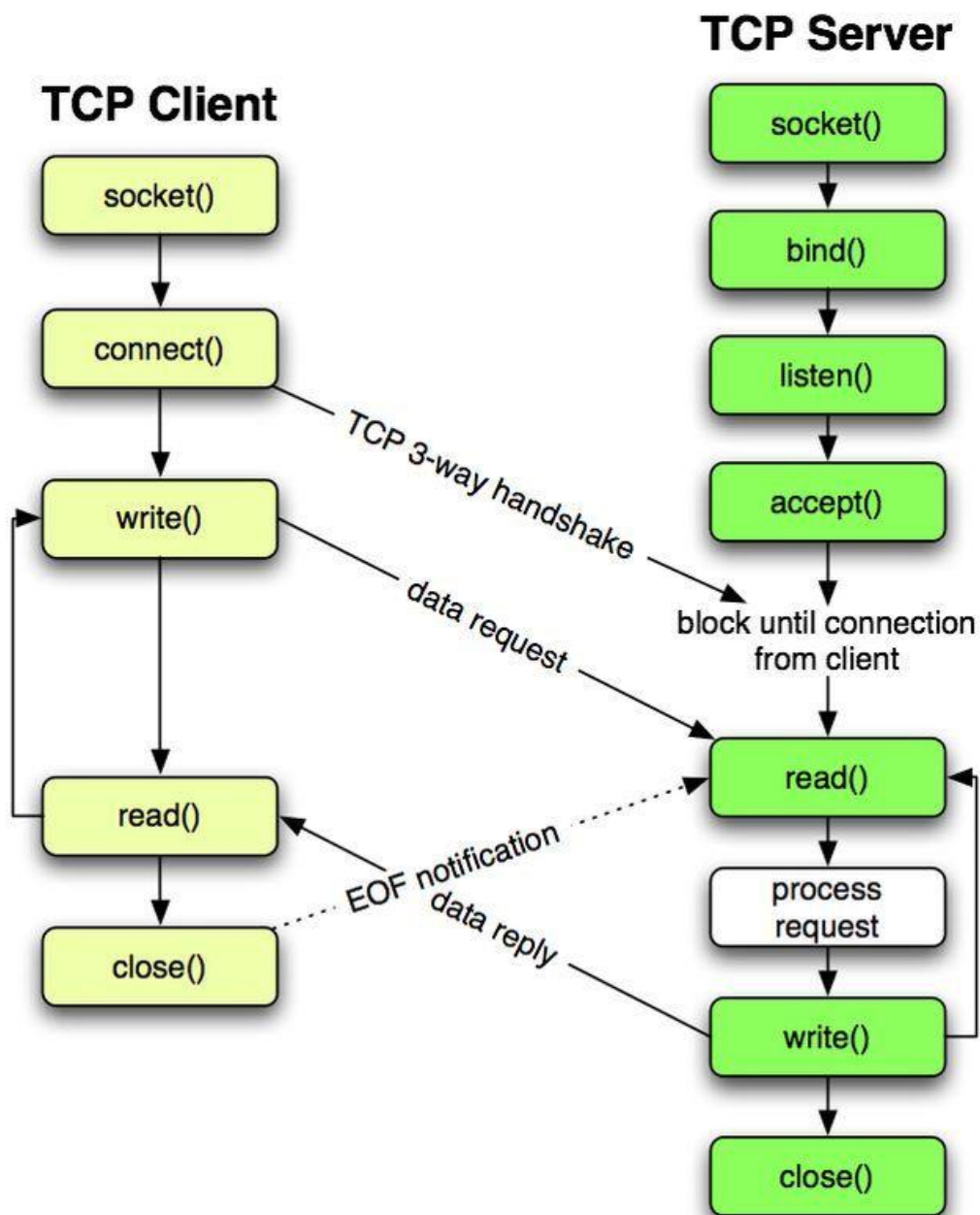


图 2-3 socket 通信

### 3 项目实施

服务器端建立 socket，并与端口绑定开始监听，核心代码如下：

```
IPAddress ip = IPAddress.Parse("127.0.0.1");  
IPEndPoint point = new IPEndPoint(ip, int.Parse("8090"));  
Socket socket = new Socket(AddressFamily.InterNetwork,
```



```

SocketType.Stream, ProtocolType.Tcp);
    try
    {
        socket.Bind(point);
        socket.Listen(10);
        ShowMsg("服务器开始监听");
        Thread thread = new Thread(AcceptInfo);
        thread.IsBackground = true;
        thread.Start(socket);
    }
    catch (Exception ex)
    {
        ShowMsg(ex.Message);
    }
}

```

服务器连接客户端并且接收信息，核心代码如下：

```

void AcceptInfo(object o)
{
    Socket socket = o as Socket;
    while (true)
    {
        try
        {
            tSocket = socket.Accept();
            string point = tSocket.RemoteEndPoint.ToString();
            ShowMsg(point + "连接成功! ");
            Thread th = new Thread(ReceiveMsg);
            th.IsBackground = true;
            th.Start(tSocket);
        }
        catch (Exception ex)
        {
            ShowMsg(ex.Message);
            break;
        }
    }
}

void ReceiveMsg(object o)
{
    Socket client = o as Socket;
    while (true)
    {
        try
        {

```

```

        byte[] buffer = new byte[1024 * 1024];
        int n = client.Receive(buffer);
        string words = Encoding.UTF8.GetString(buffer, 0, n);
        int index = int.Parse(words.Substring(1, 1));
        int num = int.Parse(words.Substring(9));
        area[index - 1] = num;
        heat();
        ShowMsg("客户端" + client.RemoteEndPoint.ToString() + ":" +
+ words);
    }
    catch (Exception ex)
    {
        ShowMsg(ex.Message);
        break;
    }
}
}

```

服务器端向无人机发送预警信息，核心代码如下：

```

private void btnSend_Click(object sender, EventArgs e)
{
    try
    {
        string index = comboBox1.SelectedItem.ToString();

        string msg = index + " " + txtMsg.Text;
        ShowMsg(msg);
        byte[] buffer = Encoding.UTF8.GetBytes(msg);
        tSocket.Send(buffer);
    }
    catch (Exception ex)
    {
        ShowMsg(ex.Message);
    }
}

```

服务器端生成人流热力图，核心代码如下：

```

void heat()
{
    for (int i = 0; i < area.Length; i++)
    {
        if (area[i] > 25)
        {
            col[i] = Colorlist[0];
        }
        else if (area[i] > 20)

```

```

        {
            col[i] = Colorlist[1];
        }
        else if (area[i] > 15)
        {
            col[i] = Colorlist[2];
        }
        else
        {
            col[i] = Colorlist[3];
        }
        if (area[i] == 0)
        {
            col[i] = Color.White;
        }
    }
    pictureBox1.BackColor = col[0];
    pictureBox2.BackColor = col[1];
    pictureBox3.BackColor = col[2];
    pictureBox4.BackColor = col[3];
    pictureBox5.BackColor = col[4];
    pictureBox6.BackColor = col[5];
    pictureBox7.BackColor = col[6];
    pictureBox8.BackColor = col[7];
    pictureBox9.BackColor = col[8];
}

```

```

public static List<Color> GetSingleColorList(Color srcColor, Color
desColor, int count)
{
    List<Color> colorFactorList = new List<Color>();
    int redSpan = desColor.R - srcColor.R;
    int greenSpan = desColor.G - srcColor.G;
    int blueSpan = desColor.B - srcColor.B;
    for (int i = 0; i < count; i++)
    {
        Color color = Color.FromArgb(
            srcColor.R + (int)((double)i / count * redSpan),
            srcColor.G + (int)((double)i / count * greenSpan),
            srcColor.B + (int)((double)i / count * blueSpan)
        );

        colorFactorList.Add(color);
    }
}

```

```

        return colorFactorList;
    }

    public static List<Color> GetFullColorList(int totalCount)
    {
        List<Color> colorList = new List<Color>();
        if (totalCount > 0)
        {
            colorList.AddRange(GetSingleColorList(Color.Red,
            Color.Yellow, totalCount / 5 + (totalCount % 5 > 0 ? 1 : 0)));
            colorList.AddRange(GetSingleColorList(Color.Yellow,
            Color.Lime, totalCount / 5 + (totalCount % 5 > 1 ? 1 : 0)));
            colorList.AddRange(GetSingleColorList(Color.Lime,
            Color.Cyan, totalCount / 5 + (totalCount % 5 > 2 ? 1 : 0)));
            colorList.AddRange(GetSingleColorList(Color.Cyan,
            Color.Blue, totalCount / 5 + (totalCount % 5 > 3 ? 1 : 0)));
            colorList.AddRange(GetSingleColorList(Color.Blue,
            Color.Magenta, totalCount / 5 + (totalCount % 5 > 4 ? 1 : 0)));
        }
        return colorList;
    }
}

```

客户端模拟人群，利用随机数模拟人的分布，核心代码如下：

```

public void draw()
{
    for (int i = 0; i < 9; i++)
    {
        area[i] = 0;
    }

    int iHeight = map.Height;
    int iWidth = map.Width;
    Random rand = new Random(); //随机器
    int maxn = 200;
    Bitmap bmp = new Bitmap(map.Width, map.Height);
    Graphics g = Graphics.FromImage(bmp); //建立 Graphics 对象
    Pen p = new Pen(Color.Red); //---创建画笔
    for (int i = 0; i < maxn; i++)
    {
        int x = rand.Next(iWidth); //随机生成 x 轴上的点
        int y = rand.Next(iHeight); //随机生成 y 轴上的点
        cal(x, y);
        g.FillEllipse(Brushes.Red, x, y, 5, 5); ;
        //在画布坐标 (x, y) 上画一个大小只有 1 的圆（就是点）
    }
}

```

```

    }

    g.Save();
    g.Dispose(); //把 Graphics 释放掉
    map.Image = bmp;
}

```

无人机向服务端发送每个区域的人流信息，核心代码如下：

```

public void sendMsg(int index)
{
    string msg = "";
    msg += "第" + index.ToString() + "个区域的人数: " + area[index - 1]
+ "\n";
    if (client != null)
    {
        try
        {
            byte[] buffer = Encoding.UTF8.GetBytes(msg);
            client.Send(buffer);
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }
}

```

无人机接收服务端发来的信息，核心代码如下：

```

void ReceiveMsg()
{
    while (true)
    {
        try
        {
            byte[] buffer = new byte[1024 * 1024];
            int n = client.Receive(buffer);
            string s = Encoding.UTF8.GetString(buffer, 0, n);
            char num ;
            num = s[0];

            s = s.Substring(2);
            dia.Text =s ;
            if (num == '1')
            {
                this.plane.Left = 50;
                this.plane.Top = 25;
            }
        }
    }
}

```

```

    }
    else if (num == '2')
    {
        this.plane.Left = 350;
        this.plane.Top = 25;
    }
    else if (num == '3' )
    {
        this.plane.Left = 650;
        this.plane.Top = 25;
    }
    else if (num == '4')
    {
        this.plane.Left = 50;
        this.plane.Top = 225;
    }
    else if (num == '5')
    {
        this.plane.Left = 350;
        this.plane.Top = 225;
    }
    else if (num == '6')
    {
        this.plane.Left = 650;
        this.plane.Top = 225;
    }
    else if (num == '7')
    {
        this.plane.Left = 50;
        this.plane.Top = 425;
    }
    else if (num == '8')
    {
        this.plane.Left = 350;
        this.plane.Top = 425;
    }
    else if (num == '9')
    {
        this.plane.Left = 650;
        this.plane.Top = 425;
    }
    lastnum = int.Parse(num.ToString());

    flag = false;

```

```

    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        break;
    }
}
}

```

模拟飞机的移动，核心代码如下：

```

public void draw_plane()
{
    int num = Judge_area(plane.Left, plane.Top);
    if (!flag)
    {
        flag = true;
        timer2.Interval = 2000;
    }
    else
    {
        dia.Text = "";
        if (lastnum != num)
        {
            timer2.Interval = 2000;
            sendMsg(lastnum);
        }
        else
        {
            timer2.Interval = 100;
        }
    }

    if (num == 1 || num == 2 || num == 7 || num == 8)
        this.plane.Left += 10;
    else if (num == 3 || num == 4)
        this.plane.Top += 10;
    else if (num == 5 || num == 6)
        this.plane.Left -= 10;
    else if (num == 9)
    {
        if (this.plane.Left == 650)
        {
            this.plane.Left += 10;
        }
    }
}

```

```

        else
        {
            this.plane.Left = 50;
            this.plane.Top = 25;
        }
    }
    lastnum = num;
}

```

无人机与服务端建立连接，核心代码如下：

```

private void btn_Connection_Click(object sender, EventArgs e)
{
    IPAddress ip = IPAddress.Parse("127.0.0.1");
    IPEndPoint point = new IPEndPoint(ip, int.Parse("8090"));
    try
    {
        client.Connect(point);
        Thread th = new Thread(ReceiveMsg);
        th.IsBackground = true;
        th.Start();
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
}

```

## 4 项目测试

在编码实现后，本小组对本系统进行了测试，其中包括对各个功能的测试以及各个界面的测试，在测试过程中遇到了一系列问题并且针对问题进行不断完善。例如一个重要的问题是模拟无人机的运行过程，这个过程要是在窗体上模拟，假若只是简单的图片移动会显得非常生硬，视觉效果极差，在经过不断调试后，采用 timer 控件每隔一段时间移动一小步，连起来后可以有动画的效果，虽然效果可能不是特别好，但是相对第一版的图片移动有了很大改善。

在不断调试，不断改善系统性能后对本系统进行测试，相关测试截图如下：



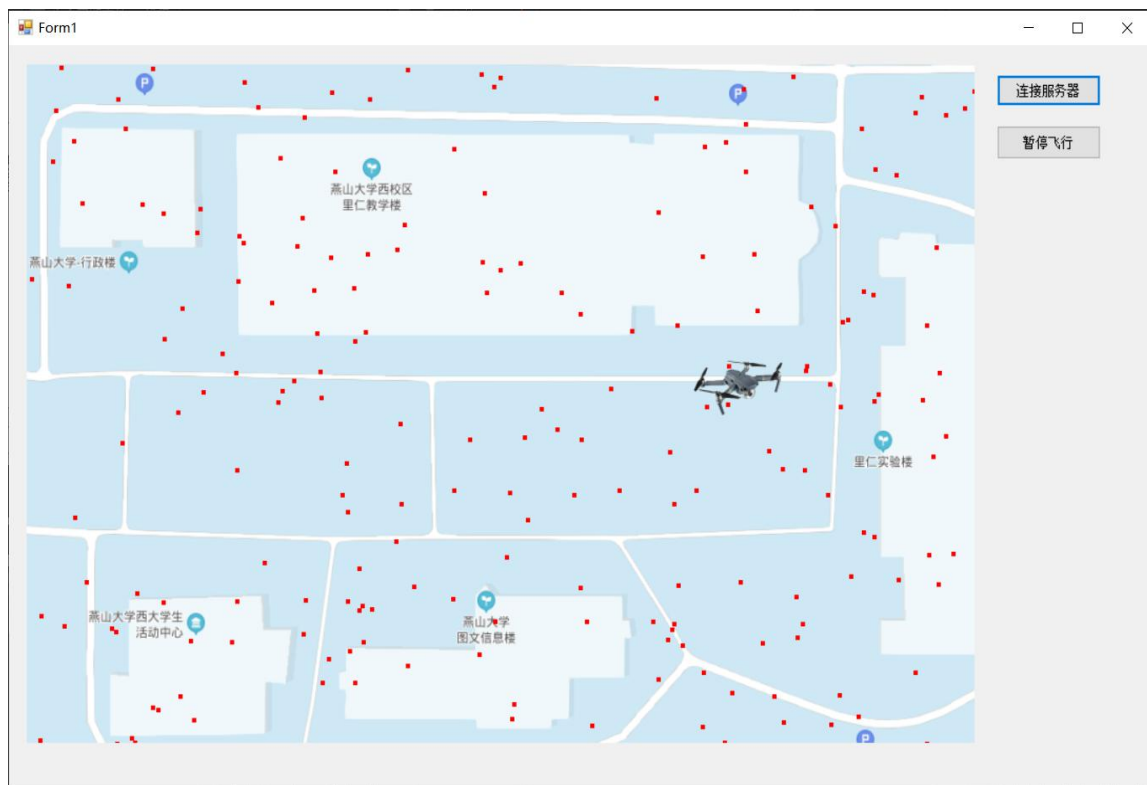


图 4-1 模拟无人机



图 4-2 控制总台收到无人机传来的人流信息



图 4-3 控制台向无人机发送控制信息

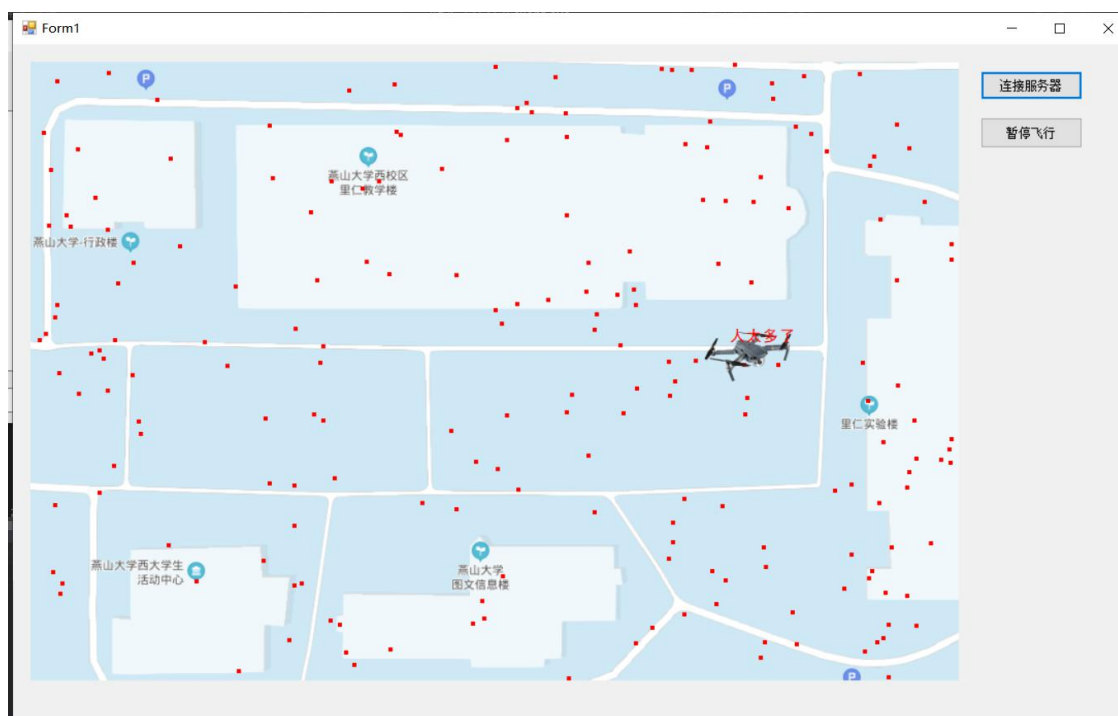


图 4-4 无人机飞到指定地点发出预警信息

## 5 总结

通过这个此次三级项目本小组对计算机网络有了更深刻的理解，并且能更好的知道这些技术的实际运用场景。本小组前后经过了几次讨论，最后明确了项目实行的方向，确定了项目的主题。在项目的实施过程中，小组成员非常积极的配合，非常默契，每个人都在积极参与项目的制作。小组成员分工明确，效率也高。在此次项目的制作过程中，本小组加入了创新元素，使用 timer 进行无人机的动态移动，使界面更加逼真美观。另外在服务器界面展示处也使用了热力图展示的方法，使结果更加清晰可见。最后服务器与客户端完美连接，一个项目便完成了。这次的选题，本小组从实际角度出发，却又不失创新，让全组成员更好的理解了计算机网络在生活和工业中的运用，加深了对计算机网络的兴趣。

## 6 参考文献

[1] 谢希仁. 计算机网络[M]. 北京电子工业出版社, 2017:41-67.