# 《数字图像处理课程设计》

# 实习报告

学　院:　　遥感信息工程学院

班　级:　　2202　2203　2208

实习地点:　遥感5号楼220机房

指导教师:　杨　　代　　琴

组　员:　　2022302131030 贾羽佳

2022302131215 刘怿欣

2022302142025 张欣蕊

2022302131291 王俊烨

2024年1月10日

# 灰度共生矩阵特征提取与分析

## 1. 研究内容与目标

### 1.1 研究内容

在日常生活中，我们应该会遇到需要提取图像中特定地物特定特征的问题，通常会提取图像的纹理特征来获得图像特征。本次应用我们将会在图像中提取出林地，利用灰度共生矩阵能够较好的解决这个问题。

灰度共生矩阵（GLDM）是一种研究图像灰度的空间相关性来描述纹理的方法，其定义为像素对的联合分布概率，是一个对称矩阵，它不仅反映图像灰度在相邻的方向、相邻间隔、变化幅度的综合信息，也反映了相同灰度级像素之间的位置分布特征。在研究过程中，由于灰度共生矩阵的数据量较大，一般不直接作为区分纹理的特征，而是基于它构建的一些统计量作为纹理分类特征，例如能量、熵、对比度、均匀性、相关性、方差等共计十四种。在本次应用中，我们选取了能量、对比度、熵、逆差矩、相关5个二次统计量体现图像的纹理特征。

一幅图像有4个灰度共生矩阵。设一点为(x,y)，步长为n，使该点遍历图像中每个像素，分别统计该点与(x+n, y)，(x+n, y+n)，(x, y+n)，(x-n, y+n)，即得到0度、45度、90度、135度四个灰度共生矩阵，再分别利用这四个灰度共生矩阵求得二次统计量，得到平均值。通过比较二次统计量，获得特定地物（例如林地）的特征值，不同的地物会有不同的特征值，从而可以从一幅图像中提取出我们所需要的林地。

### 1.2 研究目标

① 统计灰度共生矩阵，求出四个方向（0°、45°、90°和135°）的灰度共生矩阵。求取灰度共生矩阵的能量、对比度、熵、逆差矩、相关五个统计量,得到特征值图像，为下一步的提取纹理特征、物体识别等应用打好基础；

② 选择灰度共生矩阵比较常见的能量、熵、对比度、相关和逆差距五种特征进行提取和分析，通过分析这些特征去认识图像灰度的灰度、间隔、变化幅度等综合信息以及相同灰度级像素的位置分布特征。

③ 对比不同类型的图像的灰度共生矩阵特征值，探索由特征值进行地物分类的方法；提取图像的二次统计量作为特征值进一步提取出特征图像，辅助图像纹理分析。

## 2. 算法原理

### 2.1 灰度共生矩阵的基本原理与定义

共生矩阵法是一种基于灰度联合概率矩阵的方法，能够通过分析图像中不同灰度级之间的关系来反映各个灰度级的像素之间的位置分布特性。而纹理是由灰度在空间位置上反复出现而形成的，因而在图像空间中某距离的两像素之间会存在一定的灰度关系，即图像中灰度的空间相关特性。灰度共生矩阵就是一种通过研究灰度的空间相关特性来描述纹理的常用方法。因此它有助于图像的纹理分析。

集体来说，就是从灰度级为i的像素点出发，与该像素点距离为（Dx，Dy）（也就是d），方向为θ的另一个像素点的灰度级为j，灰度共生矩阵$\boldsymbol{P(i,j|d,\theta)}$的定义也就是这两个灰度级在整幅图像中的发生概率，用数学表示就是：$\boldsymbol{P(i,j,d,\theta) = 集合\{(x,y)|f(x,y) = i, f(x+Dx, y+Dy) = j; x,y = 0,1,2,\dots,N-1\}}$。若两点的距离为d，与坐标横轴的夹角为θ，那么所有的值就可以表示成一个矩阵的形式，由此得到各种间距及角度的灰度共生矩阵。

在图像中，首先选择两个像素之间的距离（d由(Dx,Dy)构成），然后有四个独立的方向可选择：0°、45°、90°、135°，在每一个方向都会产生一个灰度共生矩阵，且显然，这个灰度共生矩阵是一个对称矩阵，其阶数由图像中的灰度层数决定。它是距离和方向的函数，在规定的计算窗口或图像区域内统计符合条件的像素元对数。

### 2.2 灰度共生矩阵特征

灰度共生矩阵虽然不能直接提供纹理信息，但能够反映了图像灰度关于方向、相邻间隔、变化幅度的综合信息,是分析图像的局部特征和排列规律的基础。对于由相似灰度值的像素块构成的纹理图像，其灰度共生矩阵对角线上的数值较大。纹理的变化越快，则对角线上的数值越小，而对角线两侧的值增大。同时，灰度共生矩阵实际上是两个像素点的联合直方图，对于图像中细而规则的纹理，成对像素点的二维直方图倾向于均匀分布；对于粗而规则的纹理，则倾向于最对角分布。

### 2.3 矩阵优化

一幅图像的灰度级数一般是256级，这样级数太多会导致计算灰度共生矩阵大，计算量大，计算延时长等缺点，严重影响了计算的实时性。为了解决这一问题，在求灰度共生矩阵之前，常常将图像的灰度级压缩至16级。因此，在提取特征之前，需对灰度共生矩阵作正规化处理。

## 2.4 灰度共生矩阵二次统计量

因为灰度共生矩阵并不能直接提供纹理信息，为了能描述纹理的状况，需要在灰度共生矩阵的基础上再提取能综合表现灰度共生矩阵状况的纹理特征量，这些特征量被称为二次统计量。

**① 二阶矩(能量)**

$$ASM = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} p^2(i,j,\delta,\theta)$$

二阶矩是灰度共生矩阵各元素的平方和，又称能量，反映了图像灰度分均匀程度和纹理粗细度。能量大时纹理粗；能量小时纹理细。

**② 对比度（惯性矩）**

$$CON = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}(i-j)^2 p^2(i,j,\delta,\theta)$$

对比度可以理解为图像的清晰度。纹理的纹沟越深，对比度越大，效果清晰；反之则对比度越小，效果模糊。

**③ 逆差矩**

$$L = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\frac{p(i,j,\delta,\theta)}{1+(i-j)^2}$$

逆差矩反映了图像纹理的同质性，度量图像纹理局部变化的多少。其值大说明图像纹理的不同区域间缺少变化，局部非常均匀。

**④ 熵**

$$ENT = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} p(i,j,\delta,\theta)\log_2 p(i,j,\delta,\theta)$$

熵反映图像中纹理的复杂程度或非均匀度。若纹理复杂，熵具有极大值；反之，若图像没有任何纹理，灰度均匀，则熵较小。

**⑤ 相关**

$$COR = \frac{\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} ijp(i,j,\delta,\theta) - u_1 u_2}{\sigma_1^2 \sigma_2^2}$$

其中，$u_1 u_2 \sigma_1^2 \sigma_2^2$分别定义为

$$u_1 = \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} p(i,j,\delta,\theta)$$

$$u_2 = \sum_{j=0}^{L-1} j \sum_{i=0}^{L-1} p(i,j,\delta,\theta)$$

$$\sigma_1^2 = \sum_{i=0}^{L-1} (i-u_1)^2 \sum_{j=0}^{L-1} p(i,j,\delta,\theta)$$

$$\sigma_2^2 = \sum_{j=0}^{L-1} (j-u_2)^2 \sum_{i=0}^{L-1} p(i,j,\delta,\theta)$$

用来度量图像的灰度级在行或列方向上的相似程度，因此值的大小反应了局部灰度相关性，值越大，相关性也越大。

### 2.5 纹理特征影像提取

通过滑动窗口，利用纹理特征计算程序计算该窗口的子影像的灰度共生矩阵和纹理特征值，然后将得到的纹理特征值的均值赋给窗口的中心点；一个滑动窗口计算结束后，该窗口就可以移动一个像素点，形成另一个小窗口图像，重复进行上一步的计算，生成新窗口图像的共生矩阵和纹理特征值；以此类推，滑动窗口遍历完所有的图像像素点后，整个图像就形成了一个由纹理特征值构成的一个纹理特征值矩阵。

### 2.6 利用纹理特征值与林地的纹理特征值进行比对

#### 2.6.1 方法一

根据每种纹理的灰度共生矩阵的特征值不同，我们可以利用此特性将其与林地进行区分。首先，我们可以先在图中选取一块标准林地，该林地的特征值作为识别林地的标准；其次，求出林地的5种特征值，运用概率论有关知识确定每一种林地特征值的范围（此部分在后续将详细阐述），将林地中5个都符合条件的对应像素值标记为白，否则为黑，得到一张二值图像；再次，将要进行识别的图像重复上述过程，得到目标图像的二值化图像（以刚才确定的每一种林地特征值的范围作为阈值）；最后统计白色像素在两张图像中出现的概率，如果林地出现的概率更大或二者相近（经调试选择误差容错率为0.03），则判定这一片区域为林地，否则不是。

### 2.6.2 方法二

选取一片林地的典型图像,选取17*17的模板进行不重叠遍历,每个模板中进行灰度矩阵的运算,并获得5个特征值。计算所有17*17模板特征向量返回。之后读取需要提取林地的图像,用17*17模板重叠遍历，并计算每个模板的特征向量与上一步得出的特征向量进行对比，并计算余弦相似度。当余弦相似度大于0.998时认为高度相似，判断为林地。

### 2.7 方法一中确定每一种林地特征值的范围的方法及合理性

在得到5种特征值的对应矩阵后，计算每种特征值的均值和标准差，经过调试选择在1σ作为范围。这里用到的核心思想是概率论中的大数定律，在假设林地选取的范围无穷大时，同分布的随机变量（在这里为林地的每个像素对应的5中特征值）服从正态分布。所以理论上选取每个特征值1σ范围内对应的概率应为68%，5个特征值同时落在1σ范围内的概率约为15%（也就是白色像素占总像素的概率约为17%）。所以在选择要比对的图像时，通过5组特征值的细微差别的乘积也能清晰地将林地与其他类型的图像分开。经测试，林地的白色像素出现的概率为39%，高于应该的17%，这是由于此时选取的林地范围不能看成是近似于无穷大的，所以有误差存在也正常。

## 3. 流程设计与实现

### 3.1 程序流程设计

程序整体设计采用交互式设计，用户可以通过选择不同的二次统计量，获得相应的纹理特征影像。提取特征图像分析需要逐个像素处理，遍历以该像素中心为中心的13x13邻域图像元。首先，需要先压缩图像灰度级，然后统计出四个方向的灰度共生矩阵并做正规化处理，之后提取出对应的五个二次统计量（每个二次统计量为四个灰度共生矩阵分别求特征值再求平均），将得到的二次统计量赋给窗口的中心点，滑动窗口遍历完所有的图像像素点后，整个图像就形成了一个由纹理特征值构成的一个纹理特征值矩阵，再做灰度线性变换作为其像素值，最终提取出图像的特征统计量。

### 3.2 程序代码实现

#### 3.2.1 灰度级压缩代码

首先获取图像的高和宽，然后统计出原图像的最大灰度值，可知最大灰度级要比最大灰度值大1，便可计算出得到最大灰度级，再按照公式:$New = Old * nNewPixel / nMaxPixel$将原图像压缩到指定灰度级16级。

```
unsigned char* ptr = my_image.data;
int height = my_image.rows;
int width = my_image.cols;
//计算图像最大灰度级
int pixel = 0;
double nMaxPixel = 0;
for (int i = 0; i < height; i++)
{
    for (int j = 0; j < width; j++)
    {
        if (ptr[i * width + j] > nMaxPixel)
        {
            nMaxPixel = ptr[i * width + j];
        }
    }
}
//最大灰度级比最大灰度值大1
nMaxPixel = nMaxPixel + 1;

//将灰度级压缩到16级
for (int i = 0; i < height; i++)
{
    for (int j = 0; j < width; j++)
    {
        pixel = ptr[i * width + j];
        pixel = int(pixel * 16 / nMaxPixel);

        ptr[i * width + j] = pixel;
    }
}
```

### 3.2.2 求解灰度共生矩阵并正规化

创建四个二维数组p0、p45、p90、p135表示四个方向：0°、45°、90°、135°的灰度共生矩阵。求p0数组时，由灰度共生矩阵的定义可知：位于第i行第j列、灰度为pixel的像素，其0°方向的像素坐标应为第i行第（j+1）列，得其灰度为nPixel_0，则p0[nPixelNow][nPixel_0]++以求每个灰度级像素对的个数，同样，反过来，第i行第（j-1）列也是0°方向的像素坐标，所以p0[nPixel_0][nPixelNow]++；求p45数组时，相邻像素为第（i+1）行第（j+1）列与第（i-1）行第（j-1）列；求p90数组时，相邻像素为第（i+1）行第j列或第（i-1）第j列；求p135数组时，相邻像素为第（i+1）行第（j-1）列或第（i-1）行第（j+1）列，通过求其相邻像素的灰度，并计算其个数，最终就可以获得每个方向中，不同灰度级像素对的个数，最终求得灰度共生矩阵。特别需要注意的是越界问题，因此每次处理前都要用一个if语句来判断是否超出了图像边界。

```
for (int i = 0; i < height; i++)
{
    for (int j = 0; j < width; j++)
    {
        pixel = int(ptr[i * width + j]);
        //防止越界
        if (j < width - 1)
        {
            pixel_0 = ptr[i * width + j + 1];
            p0[pixel][pixel_0]++;
            p0[pixel_0][pixel]++;
        }
        if (j < width - 1 && i < height - 1)
        {
            pixel_45 = ptr[(i + 1) * width + j + 1];

            p45[pixel][pixel_45]++;
            p45[pixel_45][pixel]++;
        }
        if (i < height - 1)
        {
            pixel_90 = ptr[(i + 1) * width + j];

            p90[pixel][pixel_90]++;
            p90[pixel_90][pixel]++;
        }
        if (j > 0 && i < height - 1)
        {
            pixel_135 = ptr[(i + 1) * width + j];

            p135[pixel][pixel_135]++;
            p135[pixel_135][pixel]++;
        }
    }
```

在进行求解二次统计量之前要对灰度共生矩阵进行正规化处理，使元素之和为1，也就可以得到每个灰度级像素对出现的概率。公式为：p(i,j)=p(i,j)/sum，sum为归一化常数，表示的是在该方向上相邻像素对的总个数。

```
//正规化处理
int sum0 = 2 * height * (width - 1);
int sum45 = 2 * (height - 1) * (width - 1);
int sum90 = 2 * (height - 1) * width;
int sum135 = 2 * (height - 1) * (width - 1);
for (int i = 0; i < 16; i++)
```

```
{
    for (int j = 0; j < 16; j++)
    {
        p0[i][j] = p0[i][j] / sum0;
        p45[i][j] = p45[i][j] / sum45;
        p90[i][j] = p90[i][j] / sum90;
        p135[i][j] = p135[i][j] / sum135;
    }
}
```

### 3.2.3 计算二次统计量

计算二次统计量即按照公式，将灰度共生矩阵和矩阵边长（即压缩后的灰度级）传入函数当中计算，回传数值。

#### ① 二阶矩(能量)

```
double Get_SecMoment(int length, double(*p)[16]) {//传入灰度共生矩阵、矩阵边长
    double SecMoment = 0.0;//定义角二阶矩
    for (int i = 0; i < length; i++) {
        for (int j = 0; j < length; j++) {
            SecMoment += p[i][j] * p[i][j];
        }
    }
    return SecMoment;
}
```

#### ② 对比度（惯性矩）

```
double Get_InaMoment(int length, double(*p)[16])
{
    //传入灰度共生矩阵、矩阵边长
    double InaMoment = 0.0;//定义惯性矩
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length; j++)
        {
            InaMoment += (double)(j - i) * (j - i) * p[i][j];
        }
    }
    return InaMoment;
}
```

#### ③ 逆差矩

```
double Get_inverse(int length, double(*p)[16])
{
    double inverse = 0;
    for (int i = 0; i < length; i++)
        for (int j = 0; j < length; j++)
        {
            inverse = p[i][j] / (1 + (double)(i - j) * (i - j)) + inverse;
        }
    return inverse;
}
```

④ 熵

```
double entropy(int length, double(*p)[16])
{
    double entropy = 0;
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length; j++)
        {
            if (p[i][j] <= 0) continue;
            entropy = entropy - p[i][j] * (log(p[i][j]) / log(2));
        }
    }
    return entropy;
}
```

⑤ 相关

```
double Get_relativity(int length, double(*p)[16]) {
    double u1 = 0, u2 = 0, delta1 = 0, delta2 = 0;
    double s1 = 0, temp = 0;
    for (int i = 0; i < length; i++) {
        temp = 0;
        for (int j = 0; j < length; j++)
        {
            temp += p[i][j];
        }
        u1 += temp * i;
    }
    temp = 0;
    for (int j = 0; j < length; j++) {
        temp = 0;
        for (int i = 0; i < length; i++)
        {
            temp = temp + p[i][j];
        }
        u2 += (temp * j);
    }
    temp = 0;
    for (int i = 0; i < length; i++) {
        temp = 0;
        for (int j = 0; j < length; j++)
        {
            temp = temp + p[i][j];
        }
        delta1 += (i - u1) * (i - u1) * temp;
    }
    temp = 0;
    for (int j = 0; j < length; j++)
    {
        temp = 0;
        for (int i = 0; i < length; i++)
        {
            temp = temp + p[i][j];
```

```
            }
            delta2 += (j - u2) * (j - u2) * temp;
        }
        temp = 0;
        for (int i = 0; i < length; i++)
        {
            for (int j = 0; j < length; j++)
            {
                temp += i * j * p[i][j];
            }
        }
        double relativity = (temp - u1 * u2) / delta1 / d
        return relativity;
    }
```

### 3.2.4 提取特征图像

提取特征图像按照类似于滤波的方式，遍历整幅图像除了四周的每个像素，像素与其邻域一起构成（2*pos+1）*（2*pos+1）的图像元，得到一个新的临时矩阵，对此矩阵求灰度共生矩阵和二次统计量，二次统计量的均值赋给图像元的中心像素作为像素值。一个滑动窗口计算结束后，该窗口就可以移动一个像素点，并重复进行上一步的计算，遍历完所有的图像像素点后，整个图像就形成了一个由纹理特征值构成的一个纹理特征值矩阵。需要注意的是，二次统计量由于值一般较小或者为小数，需要乘以255做线性变换，将其灰度值变换到0-255之间，改善特征图像的视觉效果。

```
Mat GetFeature(Mat img, int choice, int pos)
{
    int height = img.rows;
    int width = img.cols;
    Mat new_img(img.rows - 2 * pos, img.cols - 2 * pos, CV_64FC1, Scalar(0.0));
    Mat tmp_img = (Mat_<double>(2 * pos + 1, 2 * pos + 1));
    for (int i = pos; i < height - pos; i++) {
        for (int j = pos; j < width - pos; j++) {
            for (int m = 0; m < tmp_img.rows; m++) {
                for (int n = 0; n < tmp_img.cols; n++) {
                    tmp_img.at<double>(m, n) = img.at<uchar>(i - pos + m, j - pos + n);
                }
            }
            double feature = 0.0;
            GreyMat(tmp_img, feature, choice);
            new_img.at<double>(i - pos, j - pos) = feature * 255;
        }
    }
    return new_img;
}
```

### 3.2.5 图像中林地区域的提取

#### 3.2.5.1 方法一

先找一块全部是林地的图（即main函数中定义的standard，大小为250*250）。

```
Mat standard = my_image(Range(4350, 4600), Range(5050, 5300));
```

以13*13大小的模板在上边遍历卷积操作6，每个模板内计算4个方向的灰度共生矩阵及的5个特征值（每个特征值是四个方向的灰度共生矩阵分别求特征值再求平均），将5个特征值分别放在对应的p[]层中，得到5个储存standard的特征值的矩阵。求出每一层p的均值，标准差，确定林地的特征值范围。

```
for (int i = 0; i < 5; i++) {
    p[i] = GetFeature(m_image, i + 1, pos);
    //下面是用来计算5个特征值的平均数、标准差的
    for (int m = 0; m < p[i].rows; m++) {
        for (int n = 0; n < p[i].cols; n++) {
            average += p[i].at<double>(m, n);
        }
    }
    average = average / p[i].rows / p[i].cols;
    for (int m = 0; m < p[i].rows; m++) {
        for (int n = 0; n < p[i].cols; n++) {
            standard_minus += (average - p[i].at<double>(m, n)) * (average - p[i].at<double>(m, n));
        }
    }
    standard_minus = sqrt(standard_minus / p[i].rows / p[i].cols);
    cout << average << " " << standard_minus << endl;
    average = 0.0;
    standard_minus = 0.0;

}
```

对林地图像像素进行遍历，若这个像素对应的5个p值都在刚才确定的林地的特征值范围内，就把这个像素标记为1，否则为0，得到林地的二值化图像。

```
for (int i = 0; i < fin.rows; i++)
{
    for (int j = 0; j < fin.cols; j++)
    {
        //这里的阈值都由平均数加减标准差得到
        if (p[0].at<double>(i, j) >= 0.2081962 && p[0].at<double>(i, j) <= 0.2773078) {
            sum++;
        }
        if (p[1].at<double>(i, j) >= 20.0572 && p[1].at<double>(i, j) <= 41.3266) {
            sum++;
        }
        if (p[2].at<double>(i, j) >= 0.4175424 && p[2].at<double>(i, j) <= 0.5004156) {
            sum++;
        }
        if (p[3].at<double>(i, j) <= -0.00790553 && p[3].at<double>(i, j) >= -0.01379007) {
```

```
            sum++;
        }
        if (p[4].at<double>(i, j) <= 2.907726 && p[4].at<double>(i, j) >= 2.471334) {
            sum++;
        }
        if (sum > 4) {
            fin.at<uchar>(i, j) = 255;
        }
        sum = 0;
```

再选择一张局部图片，重复上述步骤，得到选择的图像的二值化图像。

```
Mat m_image = my_image(Range(3500, 3700), Range(3000, 3200));
```

进入判断，若这个图的二值化图像中白色的像素出现的频率比选为standard的图像中白色像素出现的频率大或接近（经测试选择容错率0.03），则这个图为林地图，否则不是。

```
double probility1 = 0.0;
for (int m = 0; m < standard.rows; m++) {
    for (int n = 0; n < standard.cols; n++) {
        if (standard.at<uchar>(m, n) == 255) {
            probility1++;
        }
    }
}
probility1 = probility1 / standard.rows / standard.cols;

double probility2 = 0.0;
for (int m = 0; m < image.rows; m++) {
    for (int n = 0; n < image.cols; n++) {
        if (image.at<uchar>(m, n) == 255) {
            probility2++;
        }
    }
}
probility2 = probility2 / image.rows / image.cols;

if (probility1 - probility2 < 0.03) {
    return true;
}
else {
    return false;
}
```

### 3.2.5.2 方法二

余弦相似度的计算部分则先计算两个待比较特征向量的的点乘与各自范数，进而求出余弦相似度的值（-1至1之间）。

```
double dotProduct(const double* vec1, const double* vec2, int size) {
    double result = 0.0;
    for (int i = 0; i < size; i++) {
        result += vec1[i] * vec2[i];
    }
    return result;
}//向量点乘

double calculateNorm(const double* vec, int size) {
    double result = 0.0;
    for (int i = 0; i < size; i++) {
        result += vec[i] * vec[i];
    }
    return std::sqrt(result);
}//范数计算

double calculateCosineSimilarity(const double* vec1, const double* vec2, int size)
{
    double dot = dotProduct(vec1, vec2, size);
    double norm1 = calculateNorm(vec1, size);
    double norm2 = calculateNorm(vec2, size);

    return dot / (norm1 * norm2);
}//计算余弦相似度
```

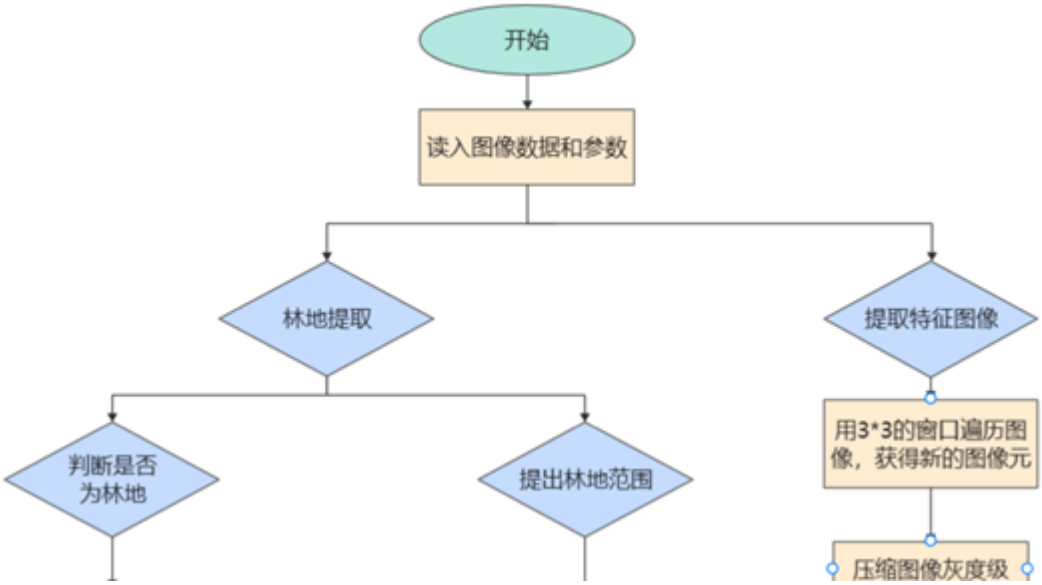
在得到林地区域灰度共生矩阵二次统计量的特征向量基础上，读取需要提取林地的图像，用17*17模板重叠遍历，并计算每个模板的特征向量与上一步得出的特征向量进行对比，并计算余弦相似度。当余弦相似度大于0.998时认为高度相似，判断为林地；否则认为是非林地区域，将该模板范围灰度数值置为0 。

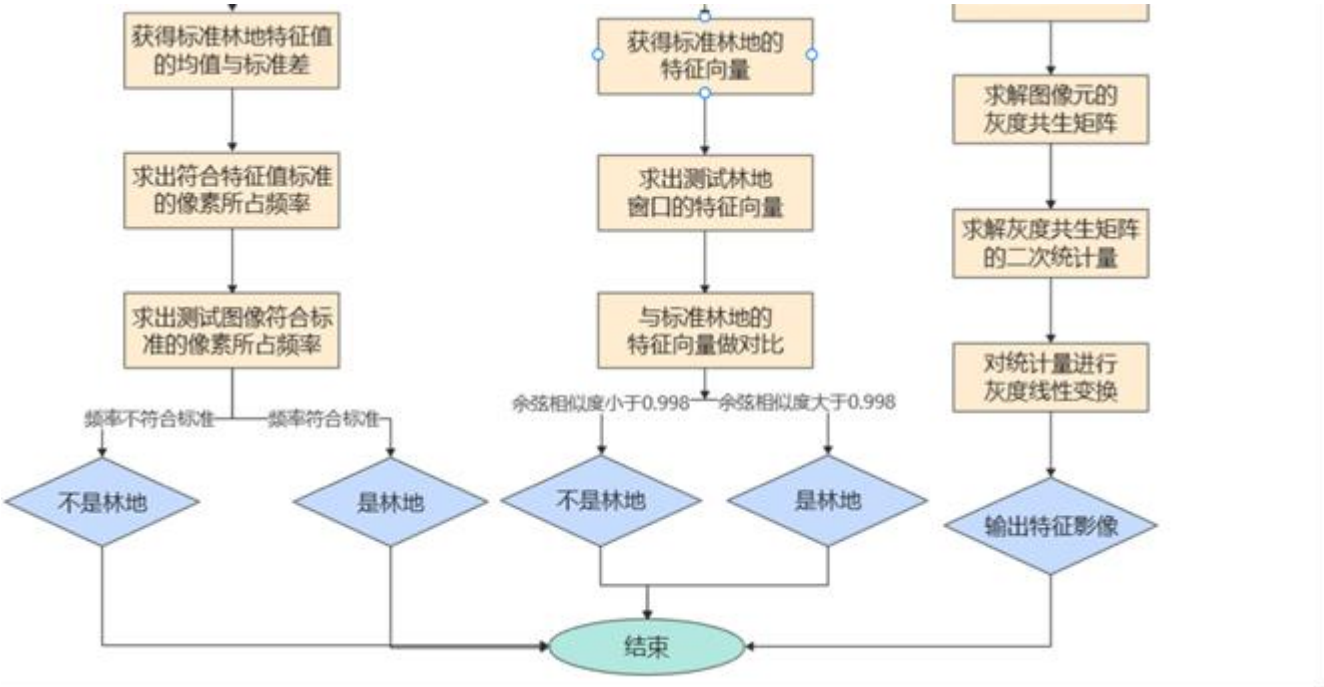
```
Mat GetForest(Mat img, int pos, double feature_aver[5])
{
    int height = img.rows;
    int width = img.cols;
    Mat newimg = img.clone();
    Mat tmp_img = Mat::zeros(2 * pos + 1, 2 * pos + 1, CV_64FC1);
    int num_x = 0, num_y = 0;
    for (int i = pos; i < height - pos; i = i  + 1) //中心位置移动
    {
        for (int j = pos; j < width - pos; j = j + 1)
        {
            for (int m = 0; m < tmp_img.rows; m++)
            {
                for (int n = 0; n < tmp_img.cols; n++)
                {
                    tmp_img.at<double>(m, n) = img.at<uchar>(i - pos + m, j - pos + n);
```

```
                }
            }
            double feature[5] = { 0.0 };
            for (int t = 0; t < 5; t++)
            {
                double p = 0.0;
                GreyMat(tmp_img, p, t + 1);//返回单位图像灰度共生矩阵特征值
                feature[t] = p;
            }
            double com = calculateCosineSimilarity(feature, feature_aver, 5);
            cout << com << endl;
            if (com < 0.998)
            {
                for (int a = 0; a < tmp_img.rows; a++)
                {
                    for (int b = 0; b < tmp_img.cols; b++)
                        newimg.at<uchar>(i - pos + a, j - pos + b) = 0;
                }
            }
        }
    }
    namedWindow("final", WINDOW_NORMAL);
    imshow("final", newimg);
    cv::waitKey();
    return newimg;
}
```

### 3.3 代码逻辑图展示

注：这里“提取特征图像”选用3*3窗口是为了当时验证编写提取特征值矩阵的正确性以及输出特征值图的便宜性（我们提取特征值矩阵的函数的窗口大小是可变的）。只有窗口大小比较小时，才能真实反映图像特征值大小随图像像素值灰度及空间位置的变化，更有利于图像的可视化。但最后在图像林地识别与提取的过程中分别使用了13*13和17*17大小的窗口。






# 4. 实验结果与分析

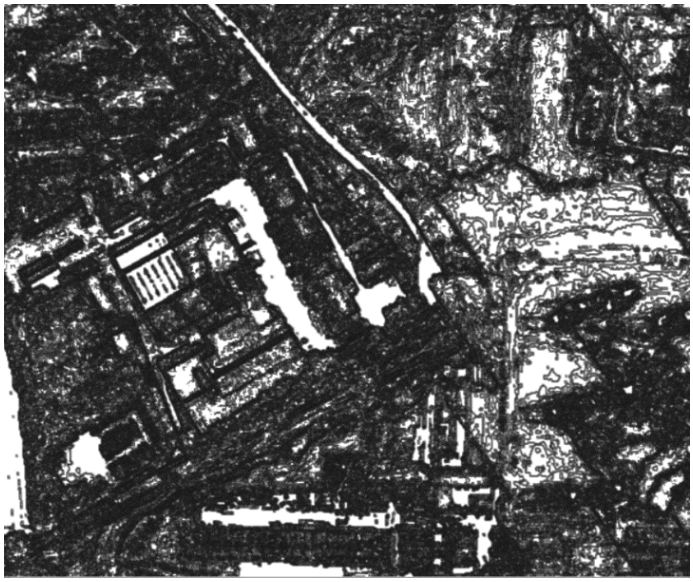
## 4.1 特征图像提取结果

实验采用的图片为一幅局部灰度影像，原图如下，对其提取特征图像。

### 4.1.1 角二阶矩特征影像

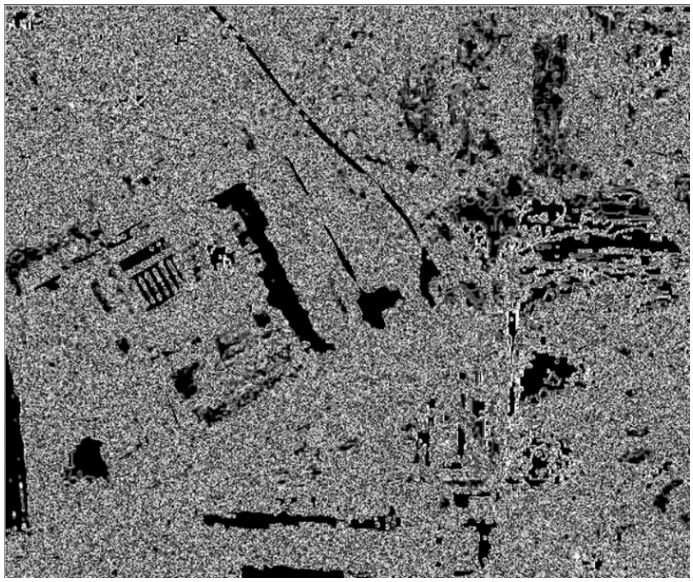
角二阶矩又称为能量，能量是灰度共生矩阵元素值的平方和，反映了图像灰度分布均匀程度和纹理粗细度。若共生矩阵中的所有值较为均匀或纹理较细，则能量值小；若其中的值大小变化或纹理较粗，则能量值大。如角二阶矩影像图中左上角部分亮度较低，则观察源影像中则可发现

该位置的纹理较细，推断为居民区；而角二阶矩影像中的中间部分亮度较大，观察源影像则发现该区域纹理较粗，推断为裸地。



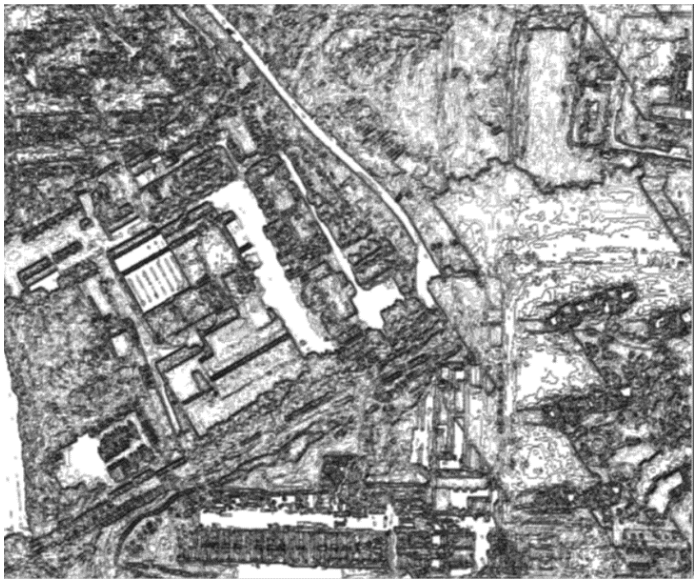
### 4.1.2 惯性矩特征影像

对于对比度（惯性矩）影像而言，多半亮度偏高，则说明这些区域的清晰度较大、纹理的沟纹较深；而其中有类似农田的部分就相对沟纹较浅。它度量了矩阵值的分布情况和图像的局部变化。直接反映了某个像素值及其邻域像素值的亮度的对比情况。从图像上看，亮度较大的地方，其值越大，则表示纹理基元对比越强烈，沟纹越深，图像越清晰，纹理效果越明显，反之亦然。



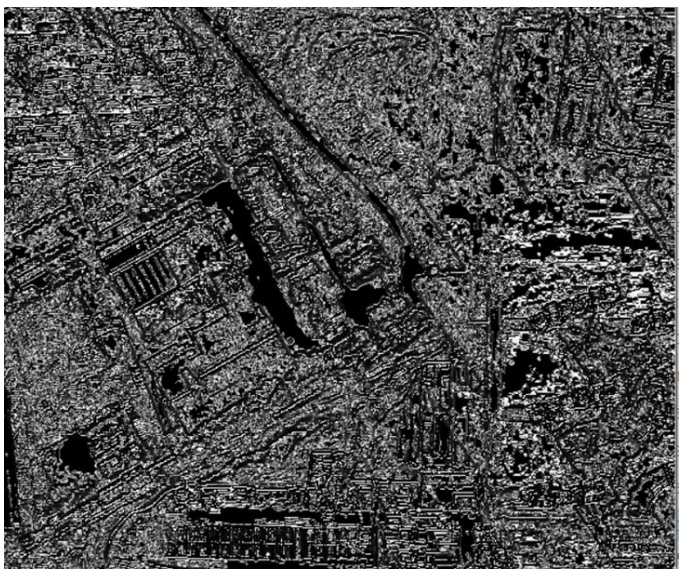
### 4.1.3 逆差矩特征影像

逆差矩反映图像纹理的同质性，度量图像纹理局部变化的多少。如果灰度共生矩阵对角元素有较大值，逆差矩就会取较大的值，说明图像纹理的不同区域间缺少变化，局部非常均匀，即连续灰度的图像会有较大逆差矩值。例如特征图像中间偏右的农田区域，明显发现源影像中相同区域的纹理变化较少，因此局部灰度变化比较均匀，符合农田的纹理特性。



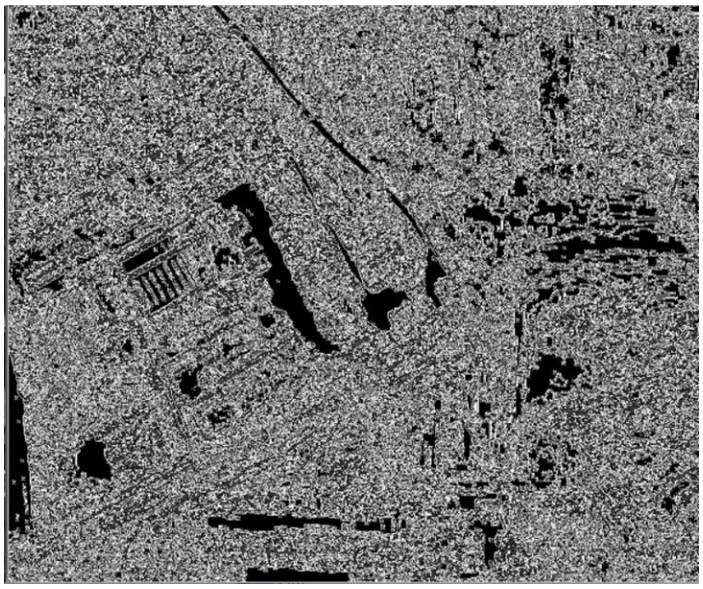
### 4.1.4 相关特征影像

相关反映了图像中局部灰度相关性，即反映了图像纹理的一致性。若图像中有水平方向纹理，则水平方向矩阵的相关值大于其余方向共生矩阵的相关值。它度量空间灰度共生矩阵元素在行或列方向上的相似程度：当矩阵元素值均匀相等时，相关值就大;当矩阵元值相差很大，相关值小。例如下方特征影像中，亮度较大区域，表明其共生矩阵的元素值大小较为均匀，纹理相似程度较高。



### 4.1.5 熵特征影像

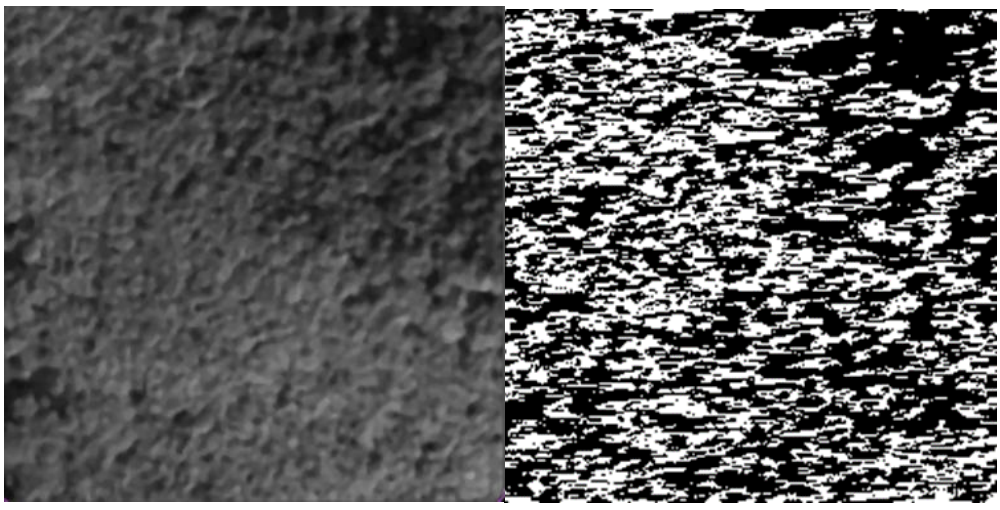
熵是图像所具有的信息量的度量，纹理信息也属于图像的信息，是一个随机性的度量，它表示了图像中纹理的非均匀程度或复杂程度。纹理复杂时，熵的值较大；而纹理较为均匀，共生矩阵中元素大小差异较大，则熵的值较小。上方图像表明，源图像的大部分区域纹理较为复杂，共生矩阵中元素差异较小。



## 4.2 纹理图像比对结果

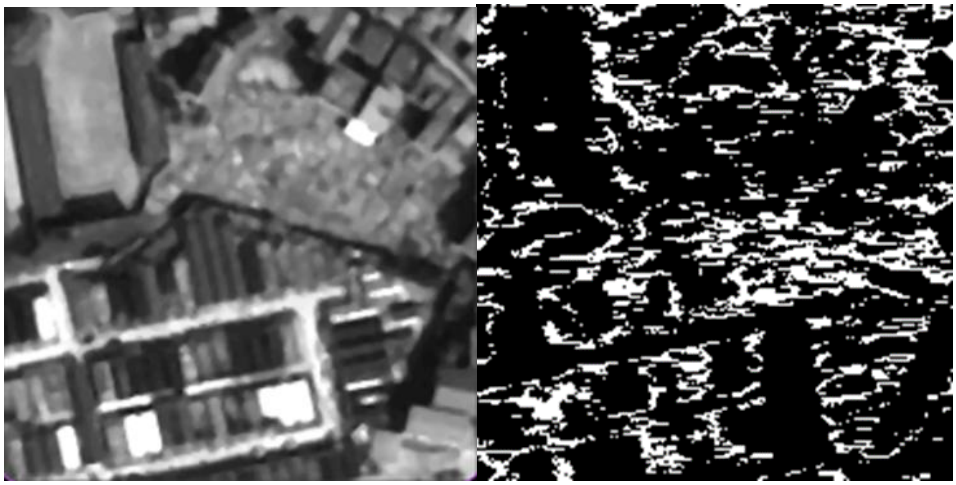
### 4.2.1 方法一

1.林地及其二值化影像



2.选取的局部图片及其二值化影像

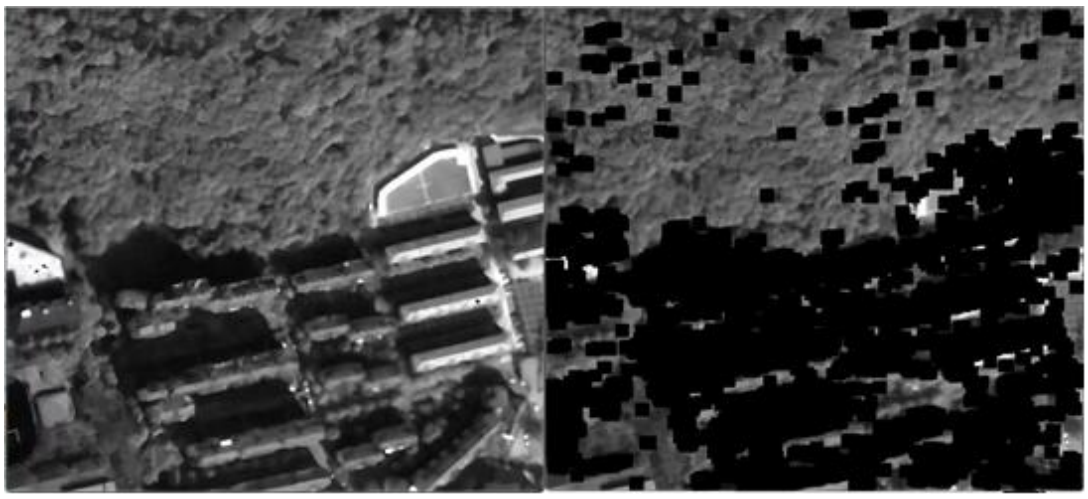
如下图所示，选取了一张包含建筑和耕地的影像，对其进行之前算法中描述的步骤进行处理，得到了一张代表该图特征值匹配度的二值图像（下图右侧）。将该图的二值化图像与上面的林地二值化图像进行比对，发现选取的图像的白色像素明显少于林地的，这也进一步说明选取的图像中符合林地特征值的像素数量少，说明其不是林地，得到了令人满意的实验结果。

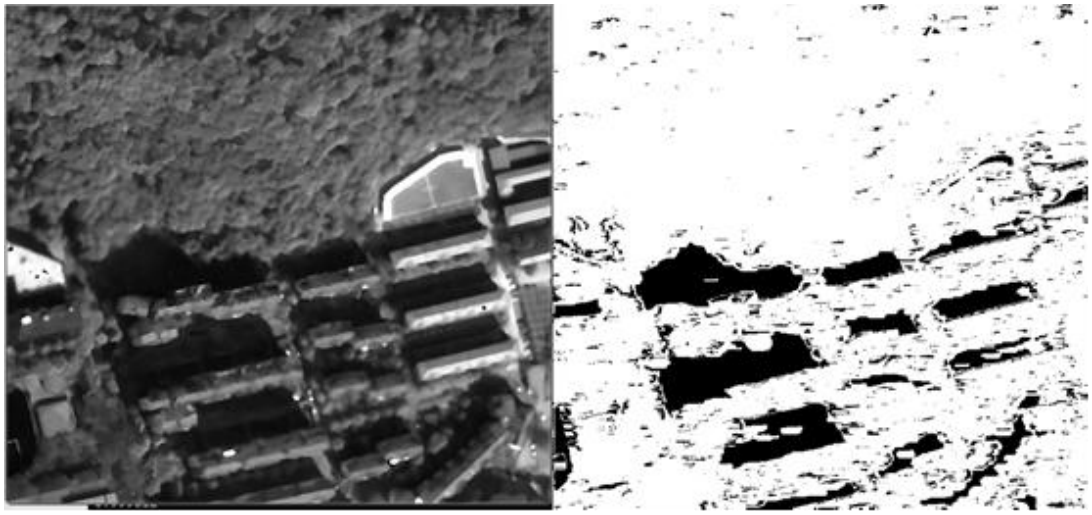


3.原始灰度图像中出现的主要景物对应的5种特征值

| | forest | | lake | | build | | 耕地 | |
|---|---|---|---|---|---|---|---|---|
| | 平均值 | 标准差 | 平均值 | 标准差 | 平均值 | 标准差 | 平均值 | 标准差 |
| 角二阶矩 | 0.2427 | 0.0414 | 0.3369 | 0.00015 | 0.2857 | 0.0587 | 0.2261 | 0.0368 |
| 惯性矩 | 0.0346 | -0.0108 | 0.001 | -0.0059 | 0.0589 | -0.0081 | 0.0265 | -0.0106 |
| 逆差矩 | 30.6919 | 0.0029 | 64.6761 | 1.48E-05 | 56.3622 | 0.004 | 32.7924 | 0.0029 |
| 相关 | 10.6347 | 2.6895 | 0.1727 | 1.9867 | 29.5318 | 2.456 | 10.3626 | 2.7644 |
| 熵 | 0.4590 | 0.2181 | 0.5354 | 0.0219 | 0.4981 | 0.4342 | 0.4341 | 0.1927 |

### 4.2.2 方法二





如图1所示，左侧为随机选取的图像范围，经与林地纹理特征向量对比后，相似度较高的部分判定为林地部分保留，相似度较低的部分用黑色表示，所得出的如右图所示，整片林地以及建筑物间林地部分都被保留。由于部分点位亮度或颗粒大小等影响因素，正片林地中有误判产生的噪点。

对图片进行二值化处理以及先腐蚀再膨胀的开运算后输出二值图像如图2，白色为林地区域。整片林地的提取较好，但由于不同区域间灰度共生矩阵特征量具有一定相似性，因此仍有误差出现。

## 5. 结论

### 5.1 灰度共生矩阵及其特征值能够在一定程度上帮助我们提取地物

(1)角二阶矩反映了图像灰度分布均匀程度和纹理粗细度。当图像纹理均一规则时，能量值（角二阶矩）较大；反之灰度共生矩阵的元素值相近，能量值较小。

(2)熵度量了图像包含信息量的随机性，表现了图像的复杂程度。当共生矩阵中所有值均相等或者像素值表现出最大的随机性时，熵最大。

(3)对比度反应了图像的清晰度和纹理的沟纹深浅。纹理越清晰反差越大对比度也就越大。

(4)逆方差反映了纹理的清晰程度和规则程度，纹理清晰、规律性较强、易于描述的值较大。

(5)相关用来度量图像的灰度级在行或列方向上的相似程度，因此值的大小反应了局部灰度相关性，值越大，相关性也越大。

不同样本的特征值存在一定的不同。在典型林地图像中提取到林地的特征值，再与需要被提取的图像中样本的特征值进行比较即可确定该区域是否为林地。

### 5.2 林地提取

基于灰度共生矩阵二次统计量的特征向量可以作为图像纹理分析的依据。通过分别求取典型林地图像的灰度共生矩阵及其特征值，以17*17的矩阵模板遍历林地图像，每个模板灰度共生矩阵特征值表示模板中点的纹理特征，通过林地图像各像素点纹理五项特征值计算出的均值与方差可以反应出林地的纹理特征。与其他区域如建筑物、湖泊、耕地等地区的纹理特征的均值、方差相比较发现，林地区域纹理的角二阶矩较小、惯性矩较高、逆差矩较低，说明林地区域纹理具有分布相对不均匀，复杂程度较高，但具有规律性，易于描述，与其他区域灰度共生矩阵特征值具有较显著差异。因此图像灰度矩阵的二次统计量可以作为图像纹理分析的依据。

灰度共生矩阵特征向量的比较有助于提取林地区域。对于纹理特征向量，通过对于林地图像特征向量中五个二次统计量均值与方差的观察，可以得出判断像素点是否属于林地的上下阈值，以此对图像进行二值化并且计算林地部分的占比，可以判断出所读图片是否属于林地。或者将任意图片作模板遍历，每个模板的灰度共生矩阵的特征向量与林地的特征向量进行余弦相似度的比较，也可以提取出林地的区域。因此基于多种方法的灰度共生矩阵的特征向量的比较有助于对于纹理的具体分类。

## 6. 小组成员分工说明

贾羽佳：林地提取方法一提出以及代码编写，PPT讲解，实习报告部分撰写

刘怿欣：灰度共生矩阵代码编写，部分特征值求解代码编写，PPT制作，实习报告部分撰写

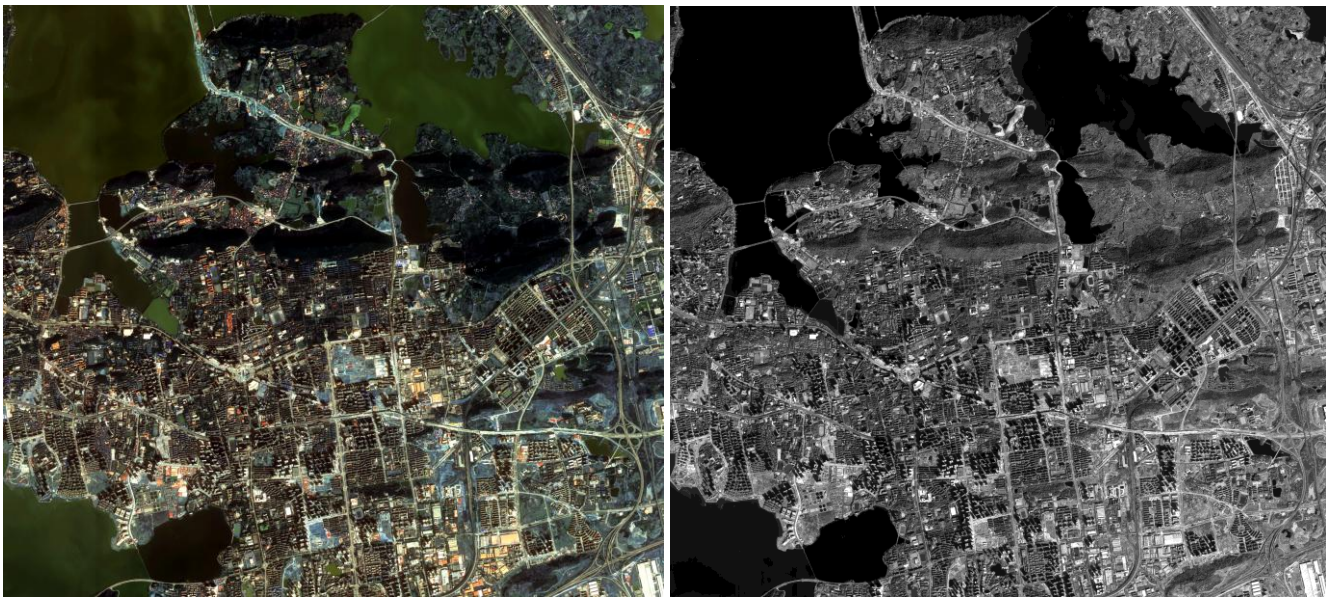
张欣蕊：林地提取方法二提出以及代码编写，PPT讲解，实习报告部分撰写

王俊烨：灰度共生矩阵部分特征值求解代码编写，主函数编写，PPT制作，实习报告部分撰写

## 7. 实习心得与体会

本次实习安排比较紧凑（尤其是在实习结束后无缝衔接期末周的压力十分巨大），在短短6天内学习了OpenCV这一开源计算机视觉库的使用，以及数字图像处理中的灰度线性变换（%2线性拉伸）、图像局部滤波处理、直方图匹配和更为复杂的用于纹理分析的灰度共生矩阵提取和分析算法。在这个过程中提高了我们对信息的处理与分析能力，将实际问题抽象化的能力，同时这也是我第一次独立提出并完成了一个新的算法，这一成就带给我的感触最深。

实习第一天，在老师讲解完实习任务与要求后，我不禁对自己产生了怀疑：我真的能在短短6天内完成3项个人任务，1项小组合作任务吗？我的编码能力比较差，能不能做出符合老师要求的内容呢？带着对未知的害怕与担心，第一天实习（2%线性灰度拉伸）正式开始了。第一天实习并不顺利，在课堂上的一个半小时只完成了对图像中2%和98%处的灰度的计算，而后面的拉伸才是重头戏。下课后，为了不落后于课程进度，我只有熬夜加班加点，但结果仍不尽如人意，在第二天找老师纠错代码后才解决代码的逻辑问题。当天晚上为了得出最后的结果大图，用电脑运行了一晚上的程序（当早晨看到程序跑出来结果超级激动），最终的结果如下图：



虽然起步比别人慢，好在后续的代码基本没出过问题，在第三天结束时完成了个人部分的全部代码（这里又忍不住想放一张我的成果图，但碍于篇幅所限，想想还是算了）。由于我们组的成员有在第二天就完成个人部分的，所以第三天她们就开始进行灰度共生矩阵的提取和特征值的计算，第四天到了，压力给到我头上，我肩负着更加艰难的林地的识别与提取任务。

在林地的提取与识别中，给我最大困扰的就是怎么确定林地的5个特征值取值范围。我一开始想的是通过将林地的5组特征值全部符合取交集，但出来的结果却将耕地也识别成了林地。为了解决林地各特征值间差异大而林地与其他种类的地的特征值差异小的问题，我采取了以下两种方法：①选择更大的模板计算特征值；②选择构造合适的统计量来加强对林地特征的提取与甄别。经过整整两天的计算与调试，最终实现了林地的识别与提取，心中的一块大石头终于落地了。

最后一天代表小组上台进行算法原理的展示。开局不顺，u盘不知道从何处打开，内心惶恐不安；在展示时，声音颤抖，即便代码是自己一个一个敲出来的，仍然抑制不住地紧张，声音都

在发抖，面红耳赤。讲完后感觉自己没有讲得很清楚，拉小组后腿了，心中十分愧疚。看来还是得加强口语表达与演讲能力呀！

虽然实习只持续了短短6天，但这一路走来，有过发现前面全部做错的崩溃与绝望，也有过检查完代码全部正确的欣喜与鼓舞；有过凌晨1点孤灯下与bug的奋战（错误C2084，函数“cv::Mat LaplacianFilter8(cv::Mat)”已有主体），也有过早晨7点因为担心跑不出结果在0.8m宽的床上辗转反侧……。这就是人生吧，潮起潮落，悲欢离合，正因为人生不是一帆风顺的，所以我们能在过程中深刻铭记，在情感上深入体验，否则一潭死水，人也就失去了存在的意义。在结尾特别感谢杨老师与我的小组，没有你们的帮助和信任，我也不可能顺利完成这次的实习。谢谢你们！

**参考文献**

[1] 贾永红，张谦，崔卫红，余卉. 数字图像处理实习教程.武汉：武汉大学出版社, 2016.11

[2] 贾永红.数字图像处理（第三版）.武汉：武汉大学出版社.2019

[3] 冯建辉，杨玉静. 基于灰度共生矩阵提取纹理特征图像的研究.北京测绘，2007年第3期