

数字图像处理综合实验

一、实验名称

油茶果果实目标分割识别方法研究

二、实验器材

MATLAB R2020a 编程软件

三、实验方法

基于不同颜色空间下的不同分量的提取；基于 K-means 的聚类分割研究

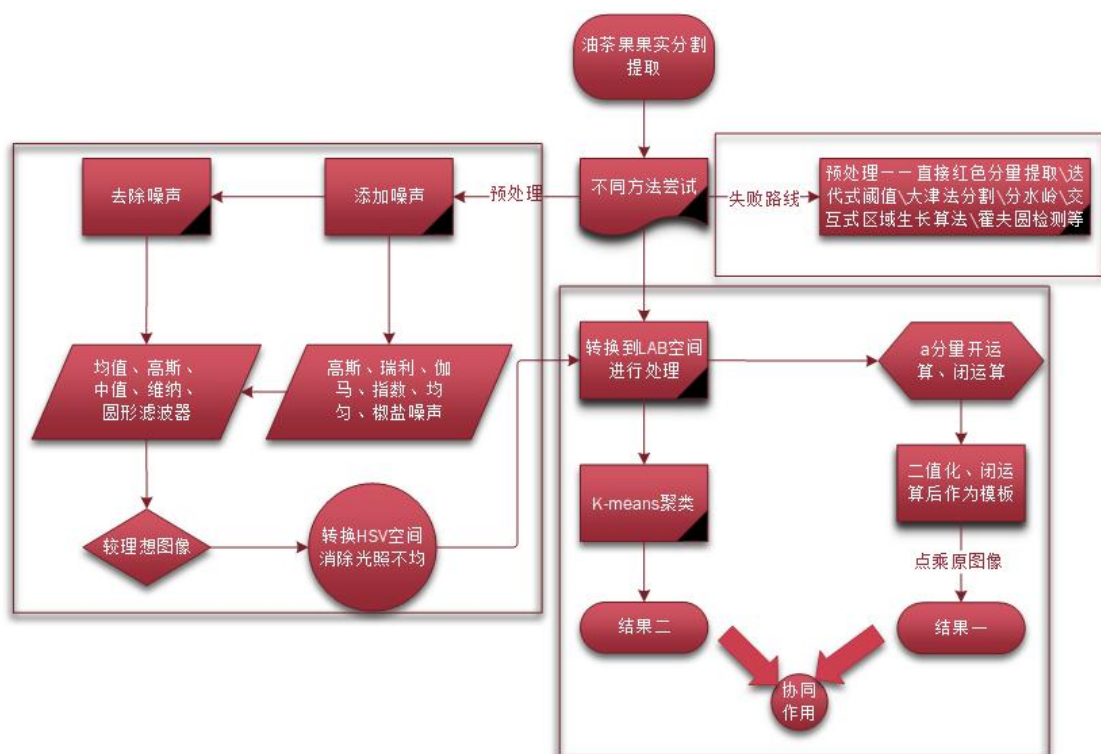
四、实验主要内容

本实验所做的工作主要为两个部分：

(1) 为了获得不同噪声情况下较佳去噪的方法，撰写加噪声函数 `add_noise` 以及相应的滤波函数 `quzao` 对不同种类的噪声以及相应的较佳去噪方法进行研究和实验验证。

(2) 在 RGB 颜色空间直接处理分析后，尝试转换颜色空间进行处理，最终选择了对油茶果颜色更敏感的 Lab 空间的 a 通道分量进行配合开闭运算、`bwareaopen` 等方法直接处理得到结果并利用 K-means 聚类算法进行处理对比。

具体过程如图一。



图一 算法流程图

五、Matlab 具体实现代码展示

1) 第一部分：加噪声、去噪函数。

1. 添加噪声函数：add_noise(imgIn, type, x, y)

```
function imgOut = add_noise(imgIn,type,x,y)
% 函数功能：为图像添加噪声
% 输入：
%     imgIn: 输入图形矩阵，为灰度图像
%     type:字符串，取值随噪声种类而定
%         高斯噪声： gaussian,参数为(x,y)，默认值为(0,10)
%         瑞利噪声： rayleigh,参数为x，默认值为30
%         伽马噪声： gamma,参数为(x,y),默认值为(2,10)
%         指数噪声： exp，参数为x,默认值为15
%         均匀分布： uniform，参数为(x,y)，默认值为(-20,20)
%         椒盐噪声： salt & pepper: 强度为x，默认值为0.02
% 输出：
%     imgOut: 添加噪声后的图像
% 预处理
if ndims(imgIn)>=3
    imgIn=rgb2gray(imgIn);
end
[M,N]=size(imgIn); % 输入图像的大小
% 设置默认噪声类型
if nargin==1
    type='gaussian';
end
% 开始处理
switch lower(type)
    case 'gaussian' % 高斯噪声
        if nargin<4
            y=10;
        end
        if nargin<3
            x=0;
        end
        % 产生高斯分布随机数
        R=normrnd(x,y,M,N);
        imgOut=double(imgIn)+R;
        imgOut=uint8(round(imgOut));
    case 'uniform' % 均匀分布
        if nargin<4
            y=20;
        end
        if nargin<3
            x=-20;
```

```

end
% 产生均匀分布随机数
R=unifrnd(x,y,M,N);
imgOut=double(imgIn)+R;
imgOut=uint8(round(imgOut));
case 'salt & pepper' % 椒盐噪声
    imgOut=imnoise(imgIn,'salt & pepper',x);
    if nargin<3
        x=0.02;
    end
    a1=rand(M,N)<x;
    a2=rand(M,N)<x;
    imgOut=imgIn;
    imgOut(a1)=0;
    imgOut(a2)=255;
case 'rayleigh' % 瑞利噪声
    if nargin<3
        x=30;
    end
    R=raylrnd(x,M,N);
    imgOut=double(imgIn)+R;
    imgOut=uint8(round(imgOut));
case 'exp' % 指数噪声
    if nargin<3
        x=15;
    end
    R=exprnd(x,M,N);
    imgOut=double(imgIn)+R;
    imgOut=uint8(round(imgOut));
case 'gamma' % 伽马噪声
    if nargin<4
        y=10;
    end
    if nargin<3
        x=2;
    end
    R=gamrnd(x,y,M,N);
    imgOut=double(imgIn)+R;
    imgOut=uint8(round(imgOut));
otherwise
    error('Unkown distribution type')
end
end

```

2. 滤波函数quzao(J)

```

function quzao(J)
%%去噪：输入图像J，自动进行均值、高斯、中值、维纳、圆形滤波器噪声去除。
%%均值滤波：采用MATLAB中的函数filter2对受噪声干扰的图像进行均值滤波
K1=filter2(fspecial('average',3),J)/255;%模板尺寸为3
figure; imshow(K1);title('均值滤波模板为3');
K2=filter2(fspecial('average',5),J)/255;% 模板尺寸为5
figure ;imshow(K2);title('均值滤波模板为5');
%高斯
K3=filter2(fspecial('gaussian',5),J)/255;% n表示模版尺寸，默认值为[3,3]，sigma，单位为像素，默认值0.5。
figure;
imshow(K3);title('高斯滤波模板为5');
%中值滤波
%中值滤波
K4=medfilt2(J,[5,5]);
figure
imshow(K4)
title('中值滤波，模板为5*5')

K5=medfilt2(J,[9,9]);
figure
imshow(K5)
title('中值滤波，模板为9*9')
%维纳滤波
K6=wiener2(J,[3 3]);
figure;
imshow(K6);title('维纳滤波');
%圆形
h=fspecial('disk',5);
K7=imfilter(J,h);
figure;
imshow(K7);
title('圆形滤波，模板为半径为5');
h1=fspecial('disk',9);
K8=imfilter(J,h1);
figure
imshow(K8)
title('圆形滤波，模板为半径为9');
end

```

3. 转换HSV空间消除亮度不均：yuchuli(im)

```

function img=yuchuli(im)
%%在HSV空间中对亮度H进行处理，去除亮度不均的影响。
tic;
% im=imread('4.jpg');

```

```

figure;
imshow(im);
title('原图');
[h,s,v]=rgb2hsv(im);    %转到hsv空间，对亮度h处理
% 高斯滤波
HSIZE= min(size(im,1),size(im,2));%高斯卷积核尺寸
q=sqrt(2);
SIGMA1=15;%论文里面的c
SIGMA2=80;
SIGMA3=250;
F1 = fspecial('gaussian',HSIZE,SIGMA1/q);
F2 = fspecial('gaussian',HSIZE,SIGMA2/q);
F3 = fspecial('gaussian',HSIZE,SIGMA3/q);
gaus1= imfilter(v, F1, 'replicate');
gaus2= imfilter(v, F2, 'replicate');
gaus3= imfilter(v, F3, 'replicate');
gaus=(gaus1+gaus2+gaus3)/3;    %多尺度高斯卷积，加权，权重为1/3
% gaus=(gaus*255);
figure;
imshow(gaus,[]);
title('光照分量');
%二维伽马卷积
m=mean(gaus(:));
[w,height]=size(v);
out=zeros(size(v));
gama=power(0.5,((m-gaus)/m));%根据公式gamma校正处理
out=(power(v,gama));
figure;
imshow(out,[]);
rgb=hsv2rgb(h,s,out);    %转回rgb空间显示
figure;
imshow(rgb);
title('处理结果')
img=rgb;
toc;

```

2) 第二部分：LAB 空间处理

1. lab空间中直接进行处理。

```

function labchuli(l)
% l = imread('4.jpg');
cform = makecform('srgb2lab');%需要输入参数(要转换的类型)，然后应用applycform函数
Lab = applycform(l, cform);
Lab = lab2double(Lab);

```

```

%显示lab颜色空间的三个分量,发现a分量对油茶果的区分效果最明显
% L = Lab(:,:,1);figure;
% subplot(211);
% imshow(L)
% se = strel('disk',5);
% l_opened = imopen(L,se);
% subplot(212);imshow(l_opened,[]);

a = Lab(:,:,2);figure;
subplot(211);imshow(a);title('a');
se = strel('disk',45);
l_opened = imopen(a,se);
subplot(212);imshow(l_opened,[]);title('开运算后结果');

l_closed = imclose(l_opened,se);
thresh=graythresh(l_closed);%确定二值化阈值
B=im2bw(l_closed,thresh);%对图像二值化
figure;imshow(B);title('B');

l_closed11 = imclose(B,strel('disk',75));
figure;imshow(l_closed11)
i1=uint8(l_closed11).*I;
figure;imshow(i1);title('lab闭运算提取结果');
%%
% 消除小连通域问题
bwAreaOpenBW = bwareaopen(l_closed11,22250);
figure;imshow(bwAreaOpenBW);title('bwAreaOpenBW处理模板');
i1=uint8(bwAreaOpenBW).*I;
figure;imshow(i1);title('bwAreaOpenBW提取结果');
se=strel('disk',180);fse=imdilate(bwAreaOpenBW,se);%膨胀
lL=imerode(fse,se);%腐蚀
i1=uint8(lL).*uint8(I);
figure;imshow(fse);
figure;imshow(i1);title('较大的卷积核进行膨胀腐蚀操作结果');

```

2. K-means算法进行聚类分析

```

function kmeanstry(I)
[m, n, p] = size(I);
%%
DBI=log(im2double(I))+1;%%变换双精度
IO=strel('disk',3);
l_rgb=imclose(DBI,IO);%闭运算去斑驳色块和内部毛刺
figure();imshow(l_rgb);title('图像：同态滤波增强图像+除斑驳色');
%三通道去噪

```

```

filter=ones(5,5);
filter=filter/sum(filter(:));
denoised_r=conv2(l_rgb(:,1),filter,'same');
denoised_g=conv2(l_rgb(:,2),filter,'same');
denoised_b=conv2(l_rgb(:,3),filter,'same');
denoised_rgb=cat(3, denoised_r, denoised_g, denoised_b);%链接矩阵
figure();imshow(denoised_rgb);title('去噪后图像');%去噪后的结果

```

%测试后选择将彩色图像从RGB转化到lab彩色空间(尝试了HSV、YcbCr-color space)

cform = makecform('srgb2lab');%这是matlab封装好的方法，只需要输入参数(要转换的类型)，然后应用applycform函数即可

```

Lab = applycform(I, cform);
figure; imshow(Lab);title('lab');

```

```

Lab = lab2double(Lab);

```

```

figure; imshow(Lab);title('lab1');

```

%a表示黄—蓝通道，b表示红—绿通道

%进行K-mean聚类将图像分割

```

ab =double(l_lab(:,2:3)); %取出lab空间的a分量和b分量

```

```

nrows= size(ab,1);

```

```

ncols= size(ab,2);

```

```

ab =reshape(ab,nrows*ncols,2);

```

```

nColors= 2; %分割的区域个数为2

```

```

[cluster_idx,cluster_center] =kmeans(ab,nColors,'distance','sqEuclidean','Replicates',100); %重复聚类100次

```

```

pixel_labels= reshape(cluster_idx,nrows,ncols);

```

%显示分割后的各个区域

```

segmented_images= cell(1,2);

```

```

rgb_label= repmat(pixel_labels,[1 1 3]);

```

```

for k= 1:nColors

```

```

color = l_rgb;

```

```

color(rgb_label ~= k) = 0;

```

```

segmented_images{k} = color;

```

```

end

```

```

figure(),imshow(segmented_images{1}),title('分割结果1');

```

```

figure(),imshow(segmented_images{2}),title('分割结果2');%显示分割后的图

```

```

end

```

3. 主函数

%%可取消注释自由选择图片。

```

% [fileName,pathName] = uigetfile('*.','Please select an image');%油茶果图片，选择文件

```

```

% if(fileName)

```

```

%     fileName = strcat(pathName,fileName);

```

```

%     fileName = lower(fileName);%一致的小写字母形式

```

```

% else

```

```

%     msgbox('Please select an image');

```

```

%     return; %退出程序

```

```

% end
% %读入图像后进行预处理
% I = imread(fileName);
I=imread('4.jpg');%简化输入图像过程
%% 加噪声和去噪的实验，用以选择最佳的去噪手段。
% %保存指定图片到指定位置并展示图片,因为矩阵运算和句柄不是很熟悉，程序不够简洁。
Fig_num = [1 2 3 4 5 6];
str=strings(1,6);
f=strings(1,6);
path="D:\Matlab 2018a (64bit)\rjazz (2)\油茶果检测\";
% 用strcat函数生成 "path"+"自动编号的图片名" + "文件格式后缀"
for i=1:6
str(i)=string(strcat(path,'Fig',num2str(Fig_num(i)),'.jpg'));
f(i)=string(strcat('y',num2str(i)));
end
% 添加噪声
x=imread('4.jpg');
imshow(x);title('原图');
y1=figure;
Y1=add_noise(x,'gaussian',0,10);
subplot(2,1,1);imshow(Y1);title('高斯')
subplot(2,1,2);hist(double(Y1(:)),200);
saveas(y1,str(1));
% 去除噪声
quzao(Y1);

% 添加噪声
y2=figure;
Y2=add_noise(x,'uniform',-20,20);
subplot(2,1,1);imshow(Y2);title('均匀')
subplot(2,1,2);hist(double(Y2(:)),200);
saveas(y2,str(2));
% 去除噪声
quzao(Y2);

% 添加噪声
y3=figure;
Y3=add_noise(x,'salt & pepper',0.05);
subplot(2,1,1);imshow(Y3);title('椒盐')
subplot(2,1,2);hist(double(Y3(:)),200);
saveas(y3,str(3));
% 去除噪声
quzao(Y3);

```



```

% 添加噪声
y4=figure;
Y4=add_noise(x,'rayleigh',30);
subplot(2,1,1);imshow(Y4);title('瑞利')
subplot(2,1,2);hist(double(Y4(:)),200);
saveas(y4,str(4));
% 去除噪声
quzao(Y4);

% 添加噪声
y5=figure;
Y5=add_noise(x,'exp',15);
subplot(2,1,1);imshow(Y5);title('指数')
subplot(2,1,2);hist(double(Y5(:)),200);
saveas(y5,str(5));
% 去除噪声
quzao(Y5);

% 添加噪声
y6=figure;
Y6=add_noise(x,'gamma',2,10);
subplot(2,1,1);imshow(Y6);title('伽马')
subplot(2,1,2);hist(double(Y6(:)),100);
saveas(y6,str(6));
% 去除噪声
quzao(Y6);
%%
%此处直接选择原图像进行预处理。
%HSV光照不均预处理
I1=yuchuli(I);%在HSV空间中对H分量进行处理消除光照不均;
%效果较差方法：直接对彩色图像处理，本代码仅展示这一种效果。
f=otsumethod(I1);%迭代阈值法与天津法进行阈值选择分割处理。输出为迭代阈值法结果
%%
%%lab空间分割油茶果
labchuli(I);

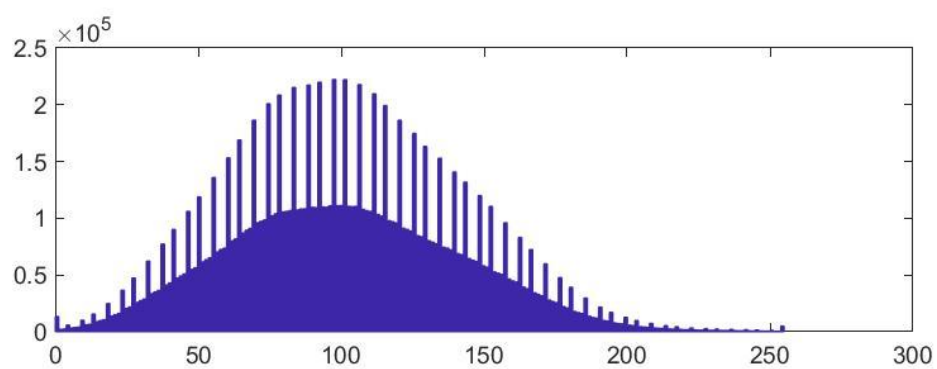
%%kmeans聚类算法分割油茶果
kmeanstry(I1);

```

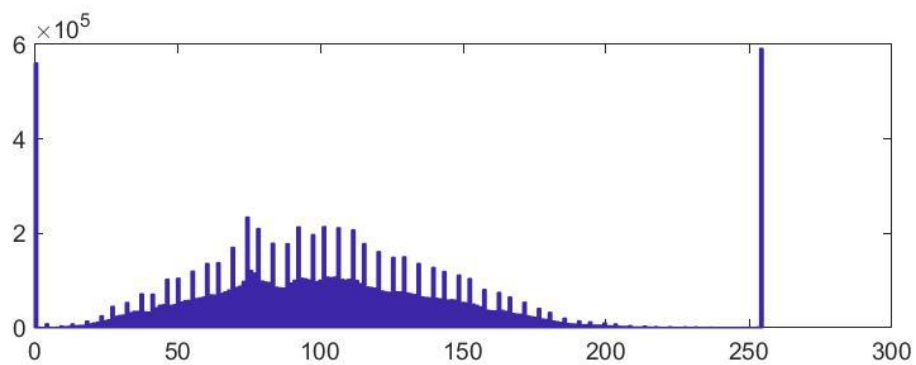
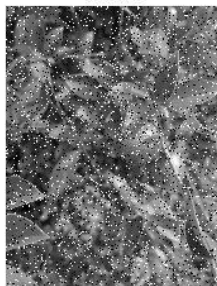
五、实验结果

加噪声处理结果，如下图所示。

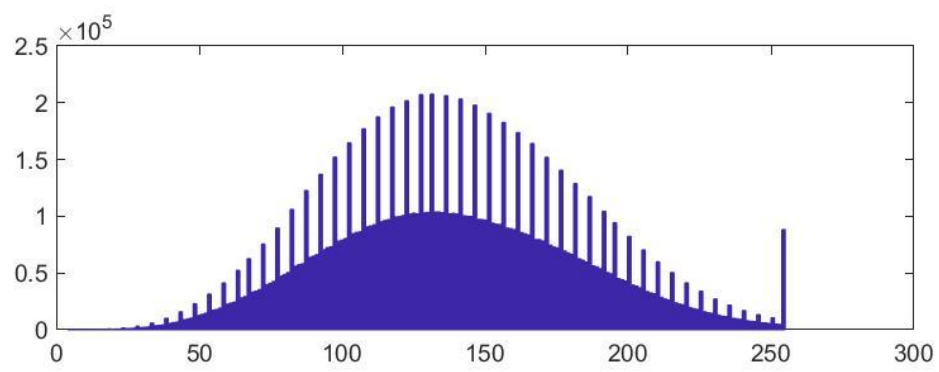
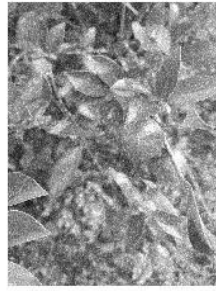
高斯



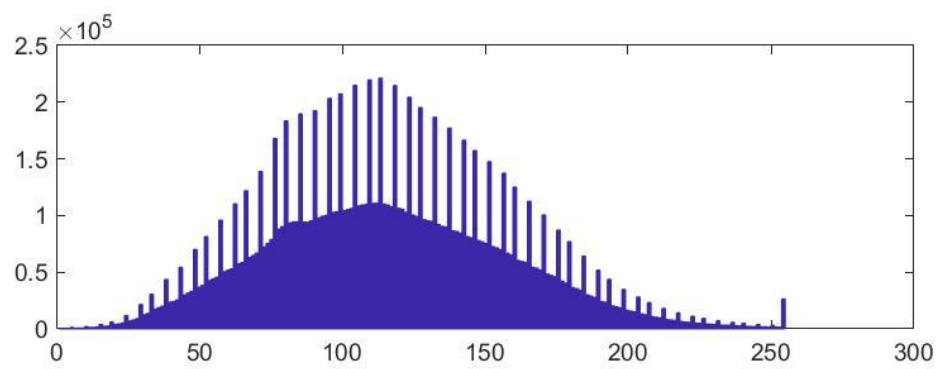
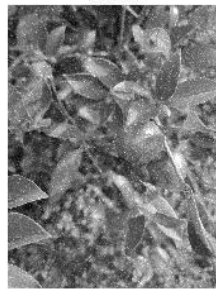
椒盐



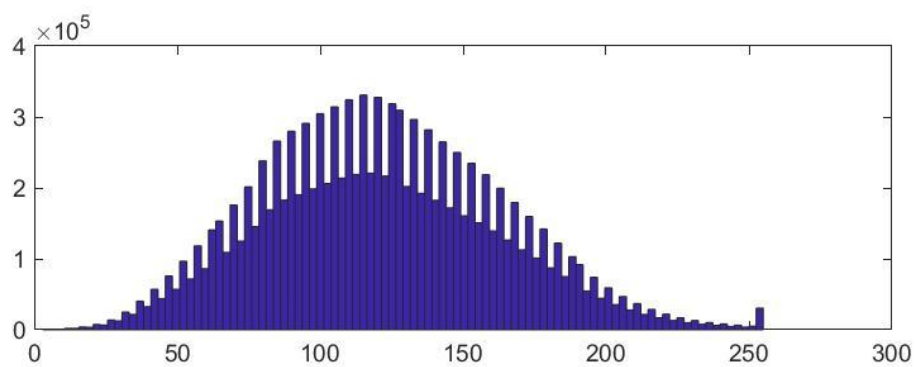
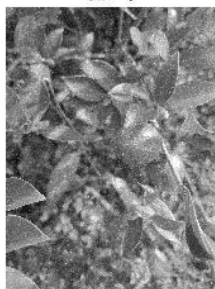
瑞利



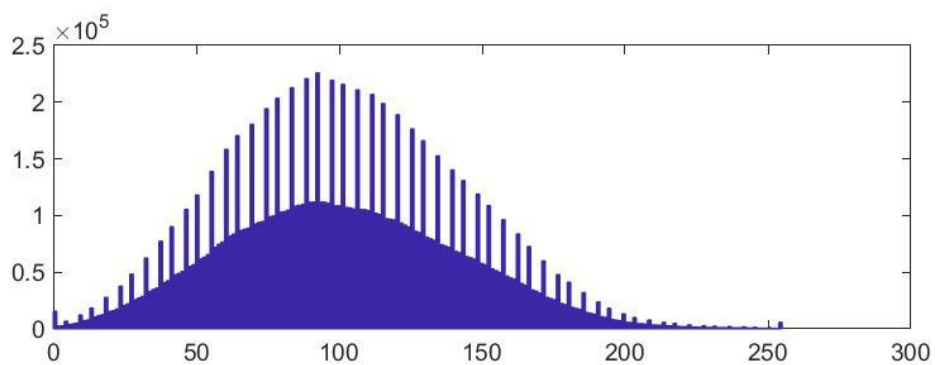
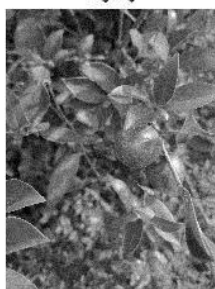
指数



伽马

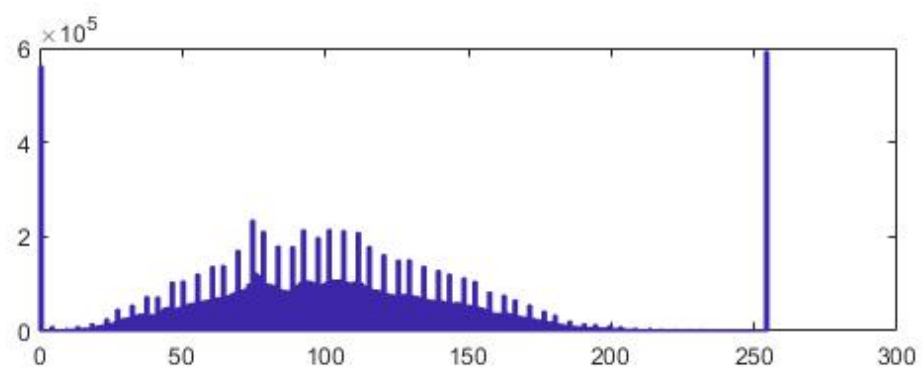
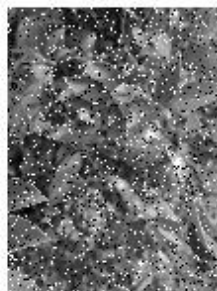


均匀



选出椒盐噪声去噪结果展示：

椒盐



均值滤波模板为3



均值滤波模板为5



高斯滤波模板为5



中值滤波，模板为 5×5



圆形滤波，模板为半径为5



圆形滤波，模板为半径为9



维纳滤波



中值滤波，模板为9*9



HSV 空间去除亮度不均。

光照分量





处理结果



迭代式阈值选择法、大津法。



原图



迭代式阈值选择法



Otsu算法

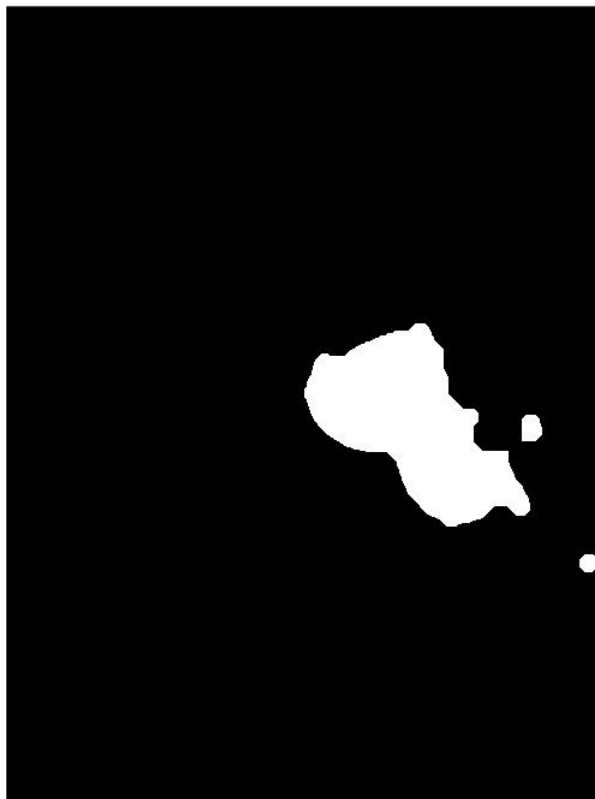
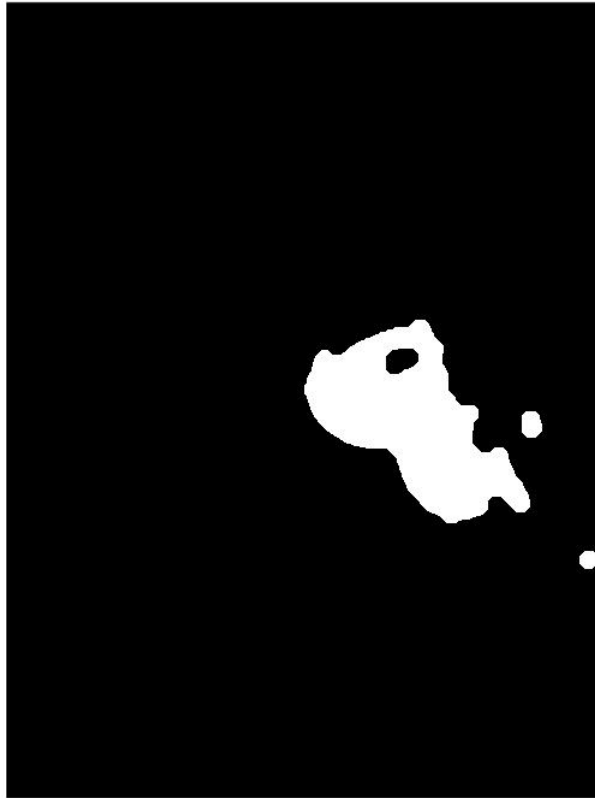
Lab 空间中直接对 a 通道进行处理。



开运算后结果



B



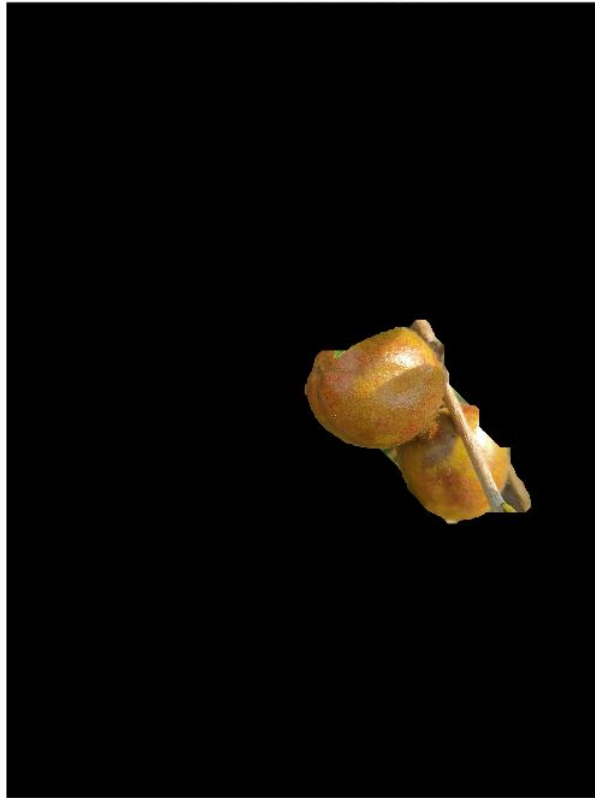
lab提取结果



bwAreaOpenBW提取结果

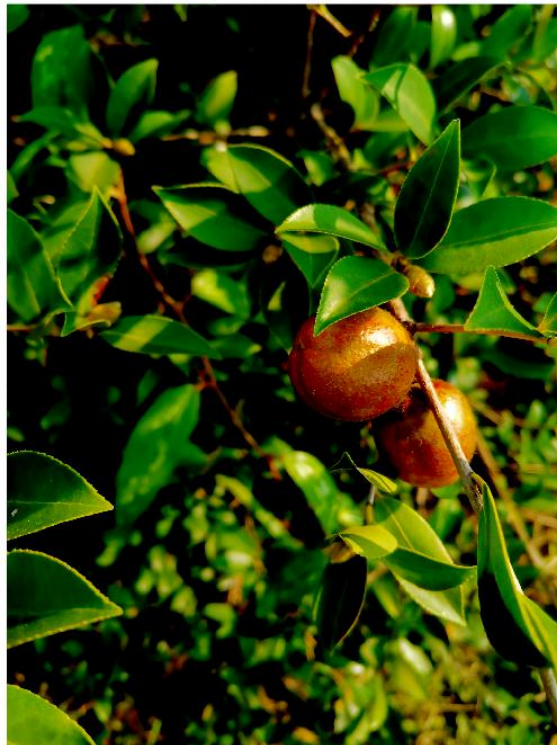


较大的卷积核进行膨胀腐蚀操作结果



K-means 算法结果如下图所示。

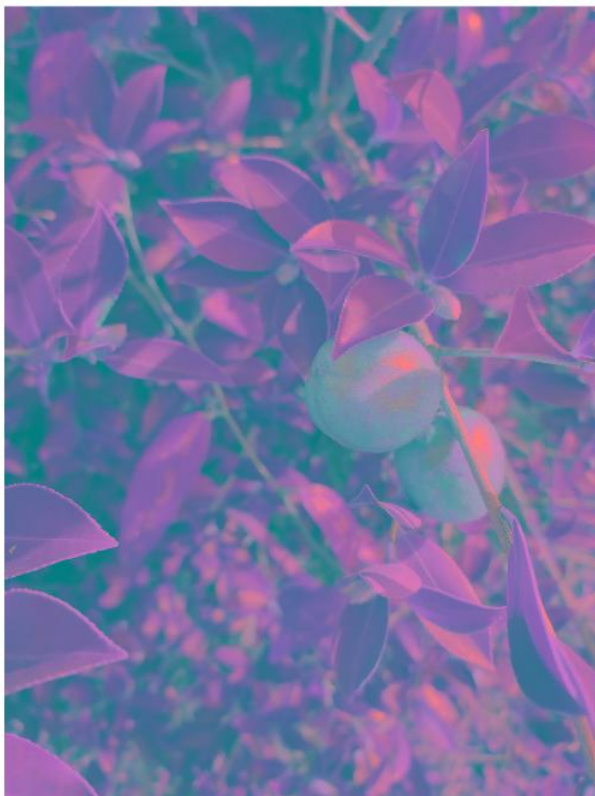
图像：同态滤波增强图像+除斑驳色



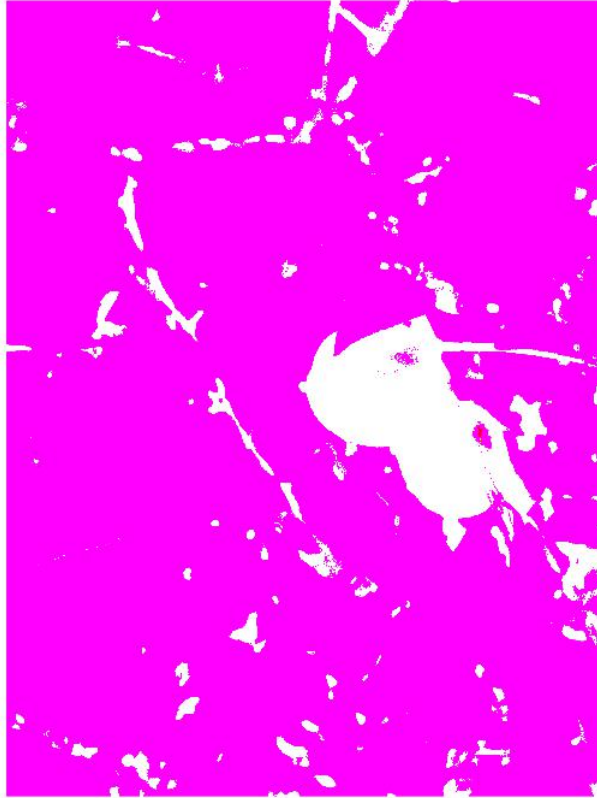
去噪后图像



lab



lab1



分割结果2



分割结果1



六、实验分析

① 根据要求，加入各类随机噪声，对于一些确定的随机噪声，如高斯白噪声，椒盐噪声来说，都有很好的处理方法，分别是高斯平滑滤波和中值滤波法。也对典型的瑞利噪声、伽马噪声、指数噪声、均匀噪声进行了分析，对每一种噪声都利用均值、高斯、中值、维纳、圆形滤波器噪声去除。报告中对典型的噪声和去噪方法进行了实验验证，一些方法取得了较好的效果。

② 在上面的实验中我首先在 RGB 进行了分量分析，即是 RGB 加法三基色空间。由于油茶果的颜色与环境区分并不大，图像质量不是特别高，还存在遮挡以及光照不均衡的影响，所以基于 RGB 空间的分析结果并不理想，尝试了边缘检测、或者直接提取 R 分量等方式，效果非常差；尝试利用大津法和迭代式阈值选择方法进行处理，发现迭代式阈值选择法效果相对较好，但仍然无法完全区分复杂的背景信息。

③ 在消除光照不均的影响时，灰度的方法损失了颜色信息，所以选择在 HSV 自然感觉颜色空间中对不同分量进行观察，由于 V 分量体现了光照信息，查询了论文中一种利用 V 分量进行消除亮度不均的方法：基于二维伽马函数的光照不均匀图像自适应校正算法，处理效果较好。

④ 对于分割的目标，首先尝试观察了几种颜色空间的各个分量，最终发现对于人眼感知均匀的 Lab 空间中的 a、b 通道对于油茶果的分析效果较好。基于此，本文首先直接在 a 通道分量上尝试进行不同卷积核大小的开运算和闭运算，消除了较小的无关背景以及消除了油茶果内部的空洞，实现果实的内部联通。之后采取一定的方式确定了二值化阈值，对图像进行二值化，最后的结果还是保留了一定的背景信息，但根据图形的大小可以较为方便的筛

选掉旁边的背景信息（或者是远处油茶果的信息，远处油茶果采摘不到，可以直接忽略），所以这种处理是有意义的。

⑤ 对于果实周围的背景种两个小连通域，在使用开运算效果不佳，使用 `bwareaopen` 函数进行了处理，消除了小对象。观察到两个油茶果提取出来的边缘不够圆润，所以选择了圆盘结构元素，调高了半径进行处理，使油茶果图像边缘更平滑，代价是两个油茶果相连的部分增大了粘连。

⑥ 在聚类算法的寻找中，K-means 因其实现简单而且在 matlab 中有函数可以调用，作为本文所使用的聚类算法。但是在对 RGB 图像进行聚类过程中，发现由于 matlab 中的 K-means 算法是最初几代的版本，在处理信息量较大的 RGB 图像时性能不佳，处理时长超过了一天，不利于进行参数的调节选择，所以也转换到了 Lab 空间尝试进行处理。本文在转换颜色空间前先将预处理结束后的图像变换为双精度，并进行闭运算去斑驳色块和内部毛刺，再进行同态滤波增强图像，达到除斑驳色的目的。利用 matlab 的矩阵运算和 `filter` 函数进行三通道去噪并拼接矩阵还原为三维图像，之后送入 k-means 算法中进行聚类分析，结果显示分割效果较好，但留下的背景信息较多，这是因为背景中的枝桠等部分颜色与油茶果十分相像，被归为一类，也可以说是提取了更多的细节信息，但此处成为了干扰的背景。

⑦ 将两种方法的结果进行比较，发现 Lab 空间 a 通道直接进行处理后的图保留的背景信息更少，但也损失了一部分果实信息，代价是对于果实的识别具有一定的特异性，不同的果实想要达到同样的效果肯定需要更多的尝试和调制；K-means 算法结果的视觉效果也很好，果实完整一些，但在 K=2 聚类的情况下背景信息保留的更多，效果相对较差，而且处理时间相对长很多，适用性更广。

⑧ 在 matlab 程序的书写过程中，查阅了大量的资料，学习了很多种 matlab 编程的技巧和算法原理并尝试应用在了代码之中。

⑨ 根据实验效果，不同类型的噪声对应不同的最佳处理方法，可以实时调整。在 Lab 颜色空间进行阈值分割和聚类的效果比 RGB 颜色空间更加好，K-means 算法的普适性更好。

七、总结

感谢老师在本次实验中耐心回答问题，帮助我理清了思路，实现了油茶果目标的分割提取。在本次实验中利用不同颜色空间的颜色分量以及 K-means 算法对油茶果果实进行了目标识别结果对比，最终发现在 Lab 空间中提取效果最好。同时，在不断地学习与练习中，不仅提升了我自己的编程能力，我的图像处理知识也得到了很大的提升，对数字图像处理的相关知识和算法学习也有着很浓厚的兴趣和学习动力。

八、参考文献或参考网址

1. [去光照不均匀 matlab 算法和代码放出 baolingq 的博客-CSDN 博客 光照不均匀](#)
2. [K-means 聚类算法 - 止战 - 博客园 \(cnblogs.com\)](#)
3. [MATLAB 图像处理 HSV 与 RGB 颜色空间互转 SCOTT 技术博客-CSDN 博客 matlab rgb 转 hsv](#)
4. [【图像处理】图像滤波去噪声——均值滤波、中值滤波、对称均值滤波 低通滤波 高通滤波 \(opencv\) 开源节流-CSDN 博客 高通滤波去噪](#)
5. [K-Means 聚类算法原理 - 刘建平 Pinard - 博客园 \(cnblogs.com\)](#)

6. <http://bit.edu.cn>
7. [基于 lab 色彩空间的图像分割 ly117626 的博客-CSDN 博客 lab 图像分割](#)
8. [常见噪声的分类与 Matlab 实现 dahua1 的博客-CSDN 博客 常见噪声 matlab](#)
9. [matlab 使用自带 kmeans 聚类思考 huxudong_1012 的博客-CSDN 博客 matlab 自带的 kmeans 效果差](#)
10. [Kmeans 及其改进算法的 Matlab 解析与实现 zhengfuzhi 的博客-CSDN 博客 kmeans 算法改进代码](#)