華中科技大學
人工智能与自动化学院
SCHOOL OF ARTIFICIAL INTELLIGENCE AND AUTOMATION.HUST

数字图像处理

第二次作业报告

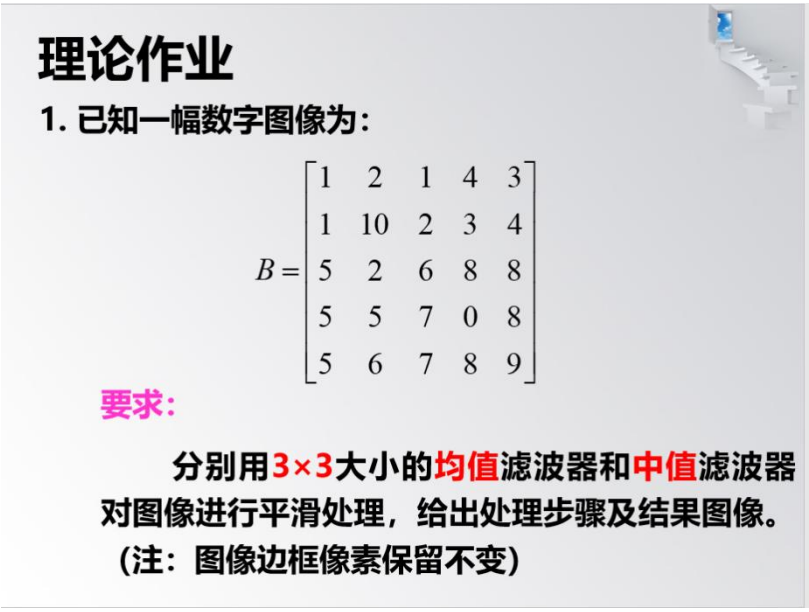
班级：人工智能 2204 班

姓名：黄佳颖

学号：U202215167

一、问题一

- **题目：**



- **问题分析：**

►均值滤波器：一种直接在空间域上进行平滑处理的技术，用一定范围内所有灰度值的均值去代替原本点位置的灰度值，以达到去除噪声的效果。

Eg：设有一幅 N×N 的图像 f(x,y)，若平滑图像为 g(x,y)，则有：

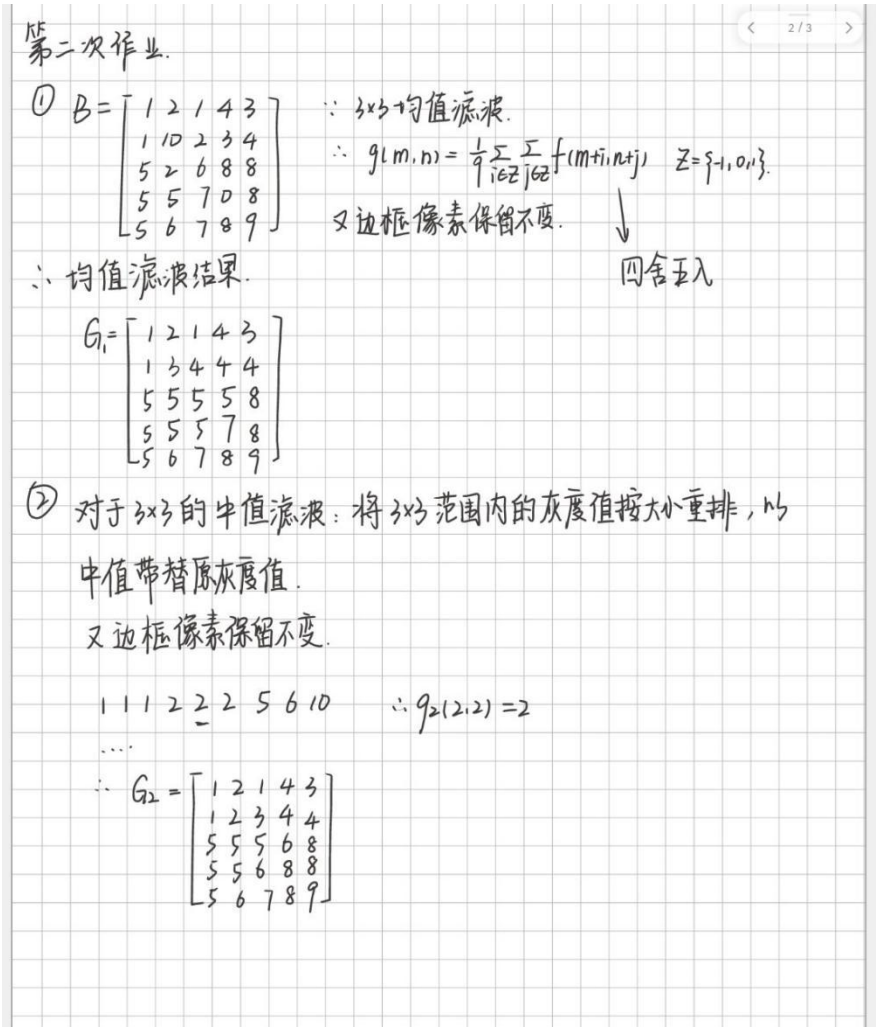
$$g(x,y)=\frac{1}{M}\sum_{(i,j)\in s} f(i,j)$$

缺点：均值滤波器的缺点是，会使图像变模糊，原因是它对所有的像素点都是同等对待，在将噪声点抑制的同时，将景物的边缘点也抑制了；由此引出了加权平均滤波器。

►中值滤波器：将一定范围内的灰度值从小到大排成一排，取模板中排在中间位置上的像素的灰度值替代待处理像素的灰度值，就可以达到滤除噪声的目的。

缺点：随着窗口尺寸的增大，虽然能更好地滤除噪声，但图像的模糊程度加重；对点、线等细节较多的图像不太合适。

- **结果图片：**



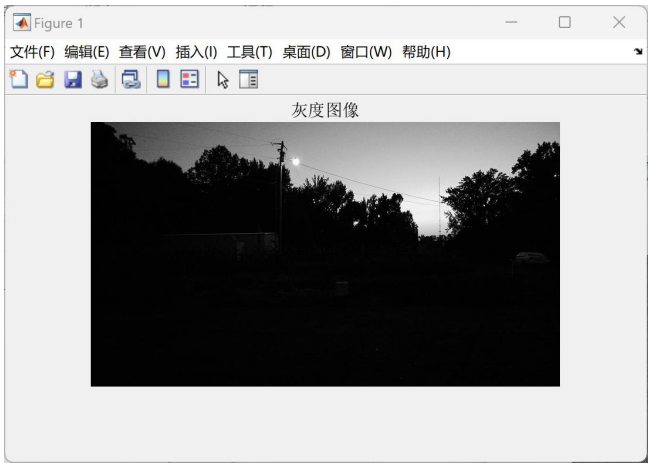
二、问题二

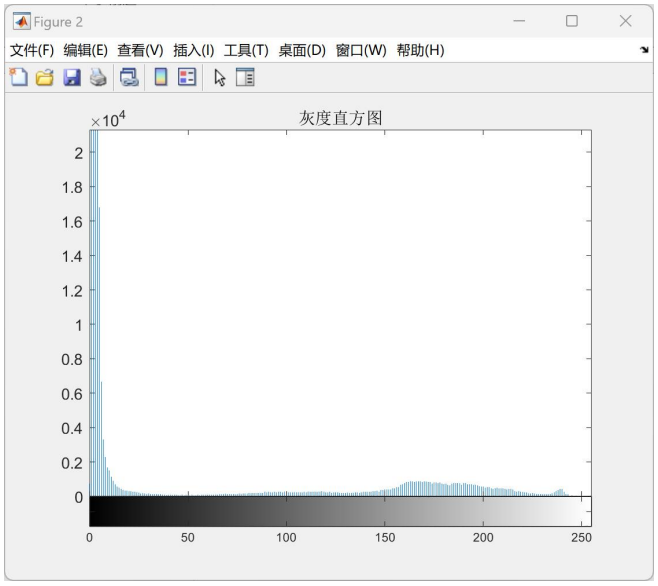
- **题目：对上述低照度图像进行灰度化，计算并显示以上低照度图像的灰度直方图和离散傅里叶变换频谱幅度图；**

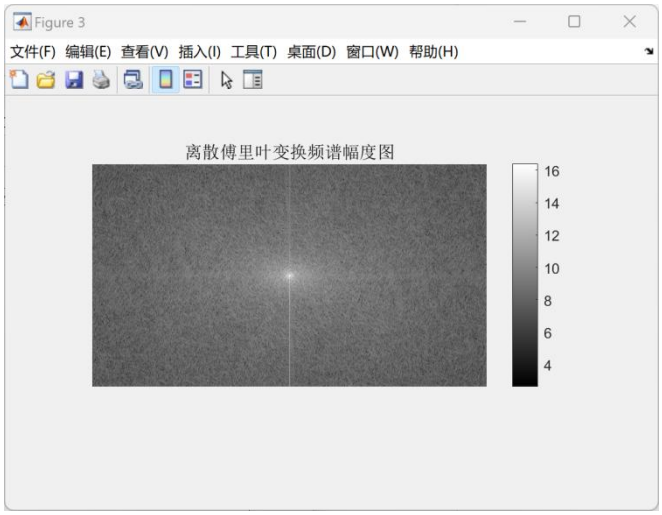
- **问题分析：**

该部分为预处理，比较简单，调用相关函数即可。傅里叶变换在上一次实验中已经说明，灰度直方图可以直观地看到图像灰度值分布情况，并便于后续对低照度图像进行相关处理操作。

- **结果图片：**







分析：由上述图片可见，该低照度图像灰度值普遍在 0 左右，非常暗，需要进行相关处理以使得图像更加清晰。

- **代码：**

```
% 读取图像
img = imread('1.png');

% 将图像转换为灰度图像
grayImg = rgb2gray(img);

% 显示灰度图像
figure;
imshow(grayImg);
title('灰度图像');

% 灰度直方图
figure;
imhist(grayImg);
title('灰度直方图');

% 离散傅里叶变换
F = fft2(double(grayImg));
F_shifted = fftshift(F);    % 将零频率分量移到频谱中心
magnitude_spectrum = abs(F_shifted);  % 计算频谱幅度

% 显示频谱幅度图（对数变换以增强可视化效果
figure;
imshow(log(1 + magnitude_spectrum), []);
title('离散傅里叶变换频谱幅度图');
colormap(gray);
colorbar;
```

三、问题三

- **题目：**对以上低照度图像分别进行直方图均衡化和同态滤波操作，并对两种算法的最终效果进行对比；

- **问题分析：**

  ►直方图均衡化：直方图均衡化是通过对原图像进行某种变换，使原图像的灰度直方图修正为均匀分布的直方图的一种方法。

  步骤：1）求出原图 R 的灰度直方图，设为 H

  2）求出图像 R 的总像素个数，计算每个灰度级的像素个数在整个图

像中所占的百分比。

3）计算图像各灰度级的累积分布 HS

4）求出新图像 P 的灰度值（四舍五入

$$S=\mathrm{int}\left[(L-1)\cdot HS(i)+0.5\right]\quad i=0,1,2,\cdots$$

►**同态滤波：**对一幅图像进行简单建模，把图像亮度 f(x, y)看成是由入射分量 i(x, y)和反射分量 r(x, y)组成：

$$f(x,y)=i(x,y)\cdot r(x,y)$$

同态滤波的基本思想是减少入射分量 i(x, y)，并同时增加反射分量 r(x, y)来改善图像 f(x, y)的显示效果；

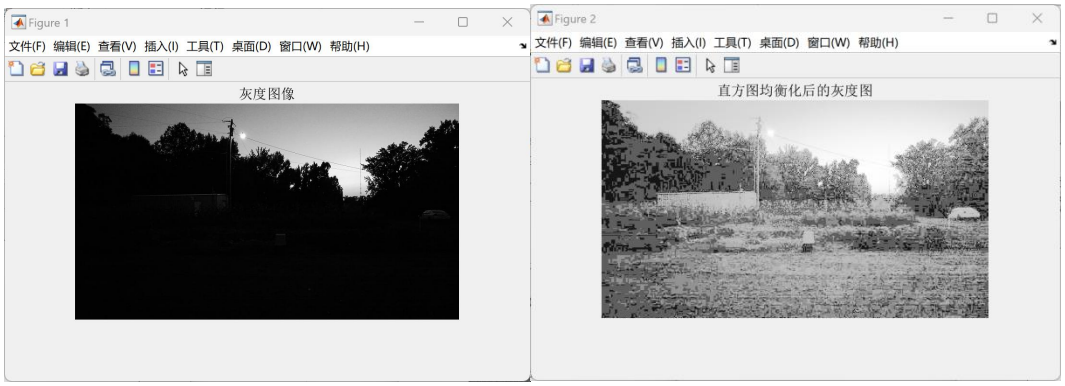
步骤：

- **对成像模型两边取对数：** $\ln f(x,y)=\ln i(x,y)+\ln r(x,y)$
- **两边取傅里叶变换：** $F(u,v)=I(u,v)+R(u,v)$
- **用一频域函数$H(u,v)$处理$F(u,v)$：**

$$H(u,v)F(u,v)=H(u,v)I(u,v)+H(u,v)R(u,v)$$

- **结果图片：**

►**直方图均衡化：**



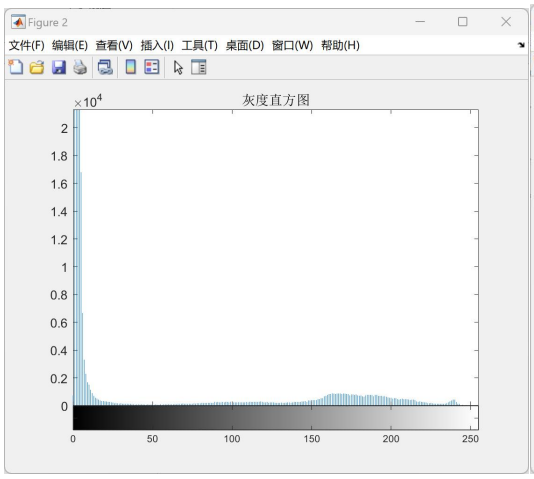


Figure 2
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
灰度直方图
×10^4
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250

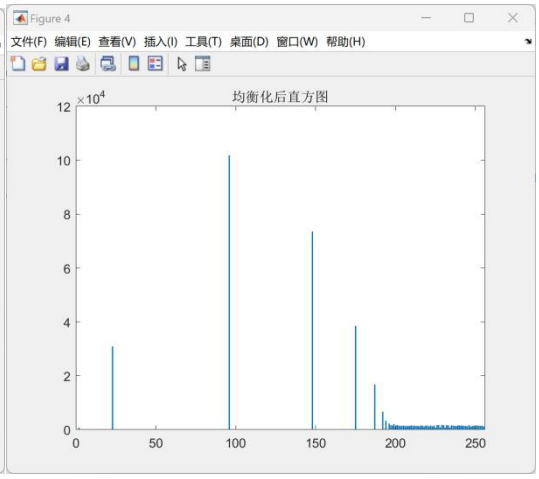


Figure 4
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
均衡化后直方图
×10^4
12
10
8
6
4
2
0
0
50
100
150
200
250

►同态滤波：

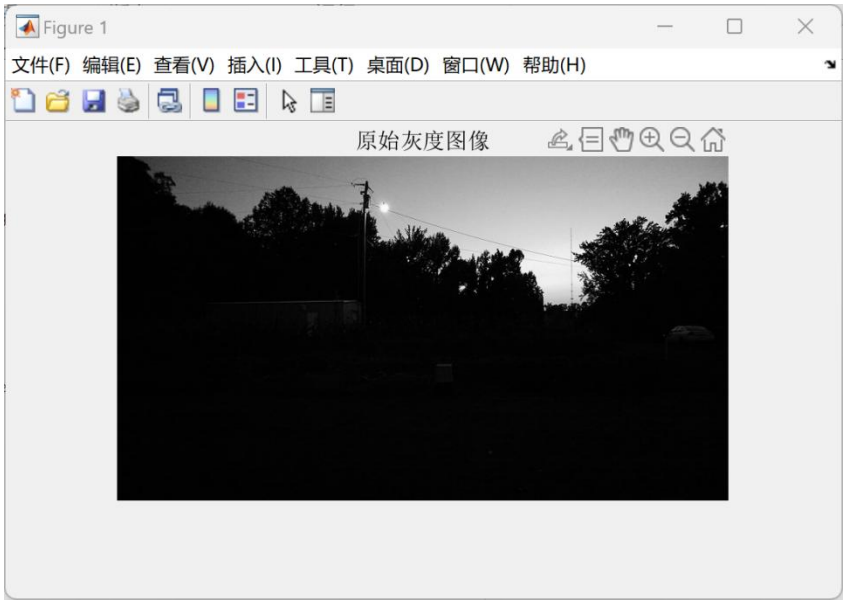


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原始灰度图像


Figure 2
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
同态滤波的图像

分析：相对于直方图均衡化的操作简便，同态滤波涉及到多种参数：从高通滤波器截止频率到对数变换前的缩放因子,还可以添加线性缩放因子、线性偏移量、伽马校正因子等参数，操作相对繁杂，调整合适的参数是一项较为困难的任务。且如果没有取得合适的参数的话，图片增强效果并不理想。

由上两图比较可知，直方图均衡化后的图更亮（有可能是因为同态滤波调参不够理想导致的），但是由于将很多灰度值归为同一类，导致图像并不平滑，出现了成块的像素；且对比度不够。而同态滤波后的图像，出现的问题则是出现了较多噪声，但相对来说图像细节缺失并不多，并且灰度对比强烈。

- **代码：**

►直方图均衡化：

```
input_image = imread('1.png'); % 读取图片

if size(input_image, 3) == 3  % 如果为 rpg 图像，则转换为灰度图像
    gray_image = rgb2gray(input_image);
else
    gray_image = input_image;
end

[rows, cols] = size(gray_image);% 获取图像的尺寸
[counts, grayLevels] = imhist(gray_image); %获取直方图

pdf = counts / numel(gray_image);  % 直方图归一化 ，numel 函数用来计算像素总数
cdf = cumsum(pdf);% cumsum 函数用来计算累积分布函数
cdf_1 = uint8(255 * cdf);  % 将 CDF 映射到 0-255

equalized_image = zeros(size(gray_image), 'uint8');  %创建一个与 gray_image 一样大小的零矩阵
for i = 1:rows
    for j = 1:cols
        original_gray_level = gray_image(i, j);  %获取原图（i，j）处的像素灰度值
        equalized_gray_level = cdf_1(original_gray_level + 1);  %根据累积分布函数将获得对应的新的灰度值（加 1 是因为数组索引从 1 开始
        equalized_image(i, j) = equalized_gray_level; %将新的灰度值赋予零矩阵中相同的位置（即（i，j）
    end
end
```

```
% 显示原始灰度图像和均衡化后的灰度图
figure;
imshow(gray_image);
title('原始灰度图');

figure;
imshow(equalized_image);
title('直方图均衡化后的灰度图');

% 显示原始灰度图像的直方图
figure;
imhist(gray_image);
title('原始直方图');

% 计算并显示均衡化后图像的直方图
figure;
[counts_equalized, ~] = imhist(equalized_image);
bar(counts_equalized);
title('均衡化后直方图');
xlim([0 256]);
```

►同态滤波：

```
% 读取灰度图像
image = imread('1.png');
if size(image, 3) == 3
    image = rgb2gray(image); % 转换为灰度图像
end
image = double(image); % 转换为双精度类型

scaling_factor=20;
scaled_image = image * scaling_factor; % scaling_factor 是一个大于 1 的常数，用来提升亮度
%还可以添加更多参数来使低照度图更亮

log_image = log1p(scaled_image);
fft_image = fft2(log_image);% 快速傅里叶变换（FFT）
fft_shifted = fftshift(fft_image); % 移到中心

% 创建高通滤波器
[M, N] = size(fft_shifted);
[u, v] = meshgrid(1:N, 1:M);
```

```
center_u = floor(N/2) + 1;
center_v = floor(M/2) + 1;
D0 = 30; % 截止频率
filter_matrix = sqrt((u - center_u).^2 + (v - center_v).^2) > D0;

% 应用滤波器
filtered_image = fft_shifted .* filter_matrix;

% 逆快速傅里叶变换（IFFT）
ifft_shifted = ifftshift(filtered_image);
ifft_image = ifft2(ifft_shifted);
ifft_image = real(ifft_image); % 提取实部

% 指数变换
enhanced_image = expm1(ifft_image);
enhanced_image = uint8(min(max(enhanced_image, 0), 255)); % 裁剪并转换为 uint8 类型

figure;  % 显示原始图像
imshow(uint8(image));
title('原始灰度图像');

figure;% 显示增强后的图像
imshow(enhanced_image);
title('同态滤波的图像');
```

电子签名：