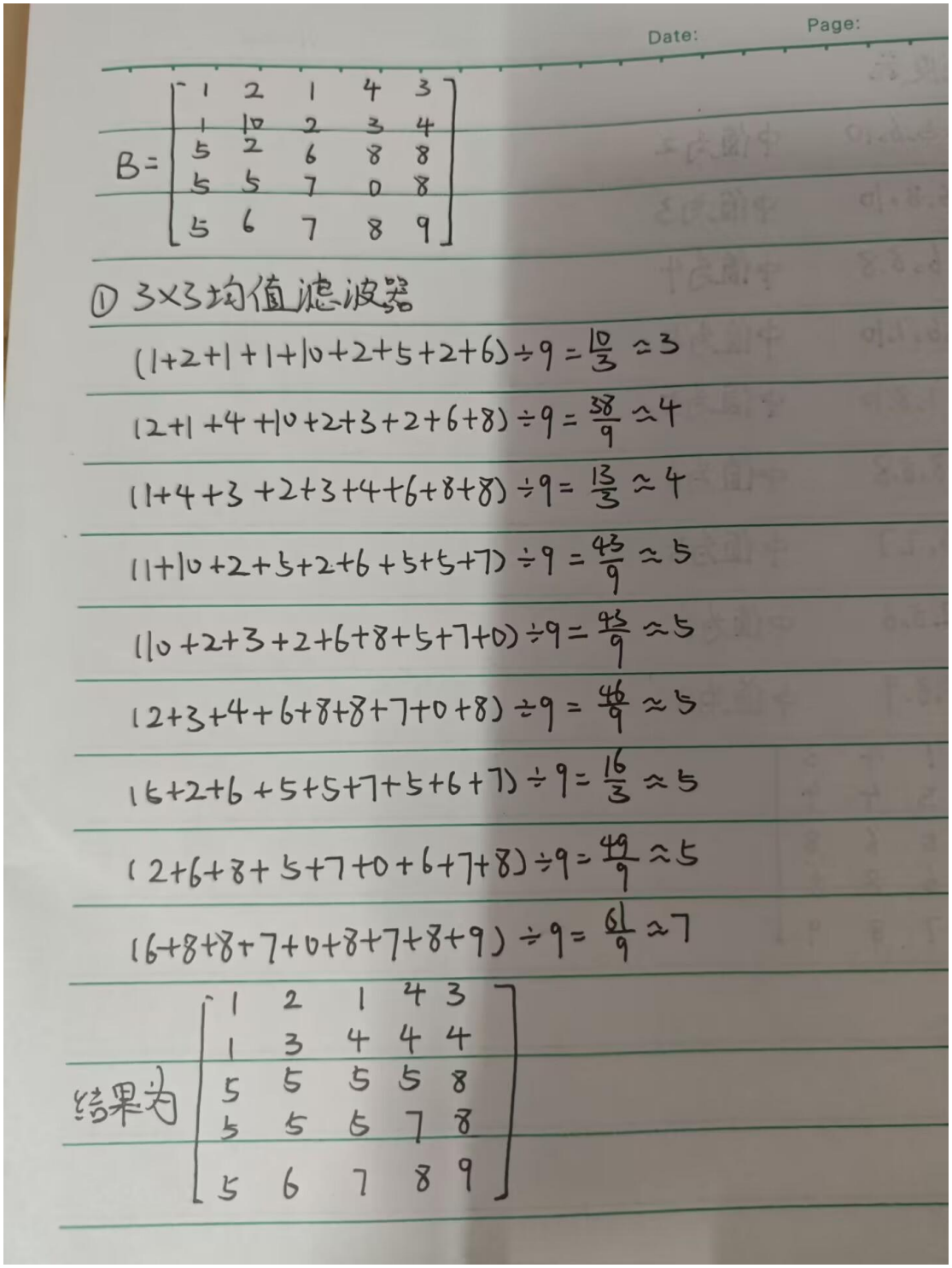
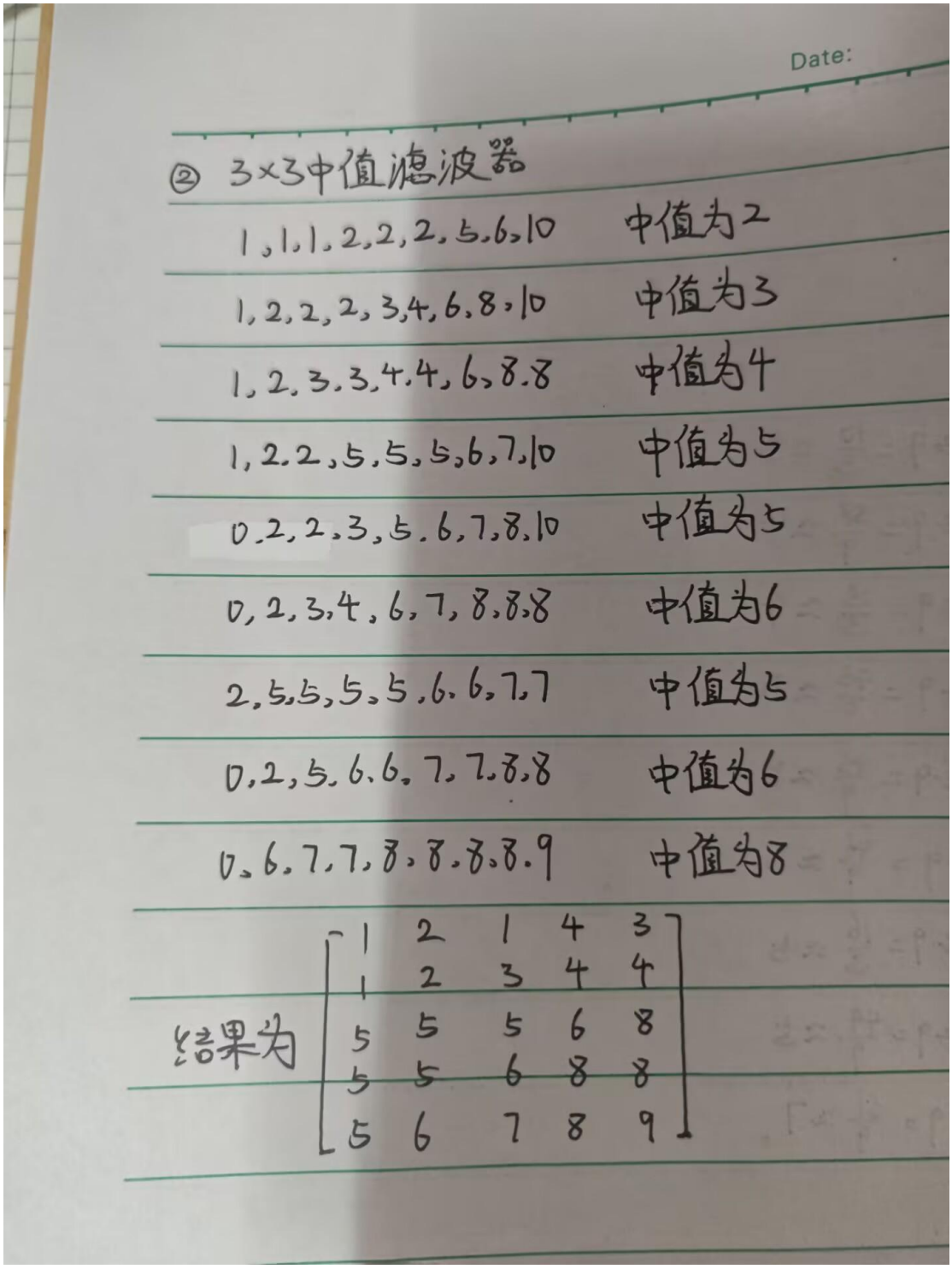
人工智能 2204 班 欧阳旭 U202215172

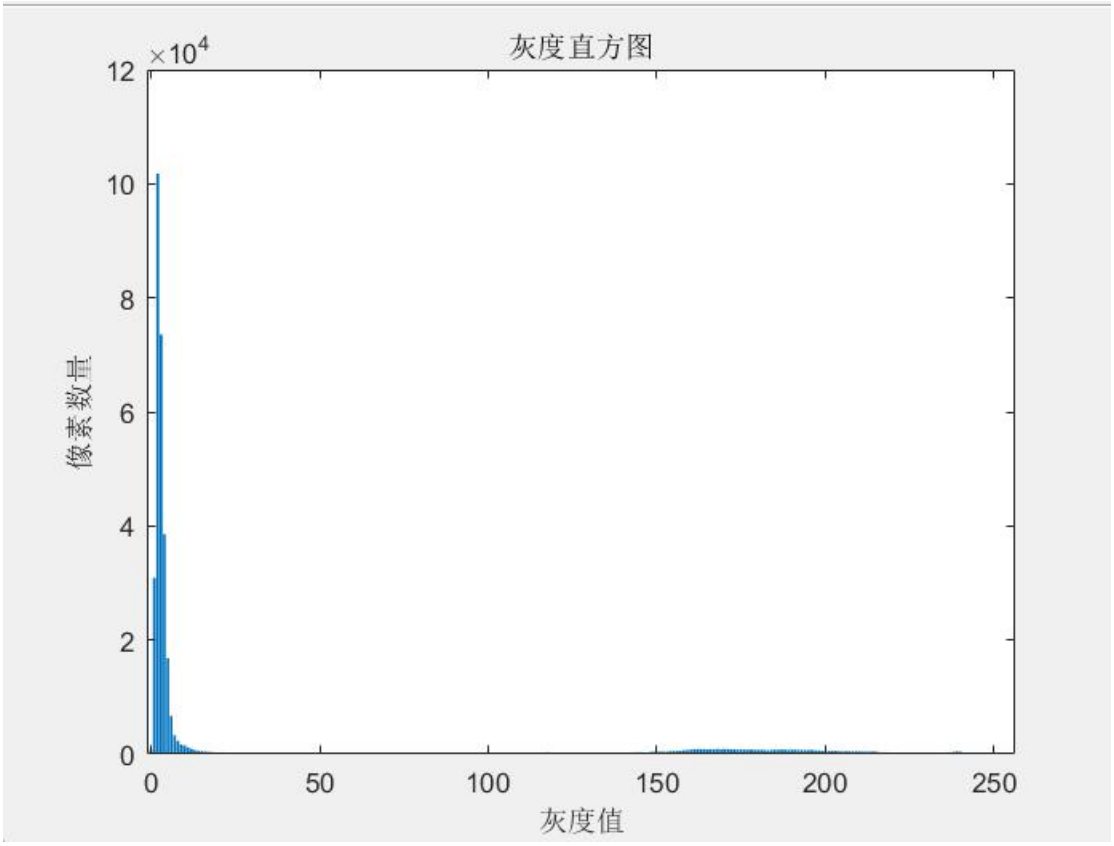




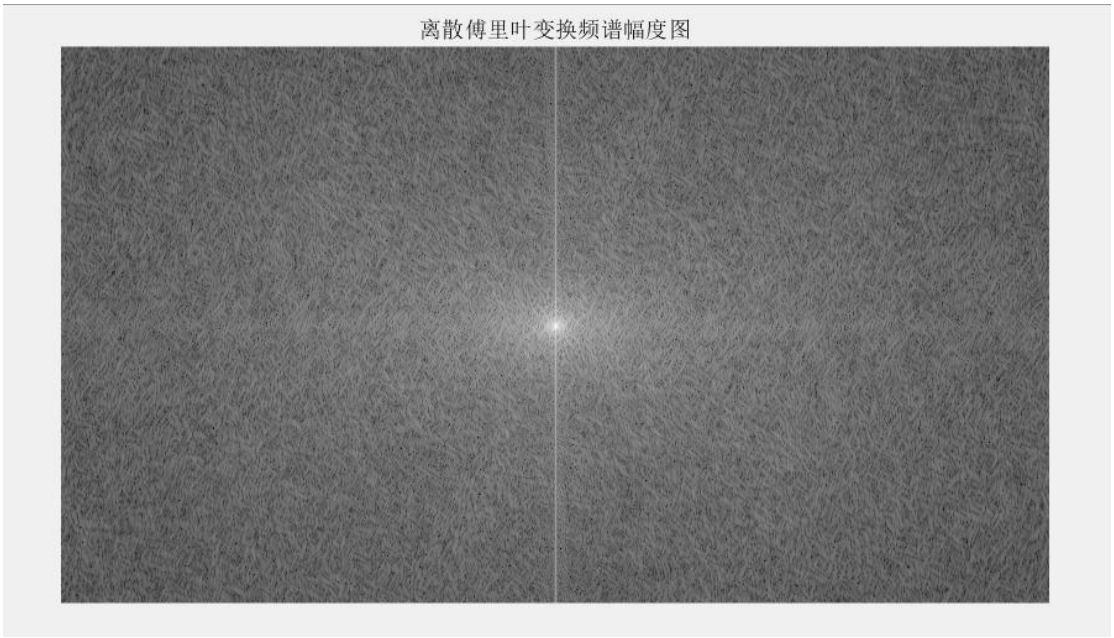
对低照度图片进行灰度化

```
% 读取低照度图像
img = imread('D:\MATLAB代码\低照度图像.jpg');

% 图像灰度化
gray_img = rgb2gray(img);

% 计算并显示灰度直方图
hist = imhist(gray_img);
figure;
bar(0:255, hist);
title('灰度直方图');
xlabel('灰度值');
ylabel('像素数量');

% 计算离散傅里叶变换频谱幅度图
f = fft2(gray_img);
fshift = fftshift(f);
magnitude_spectrum = abs(fshift);
figure;
imshow(log(magnitude_spectrum + 1), []);
title('离散傅里叶变换频谱幅度图');
```

灰度直方图



离散傅里叶变换频谱幅度图



```
% 读取低照度图像
img = imread('D:\MATLAB代码\低照度图像.jpg');
% 图像灰度化
gray_img = rgb2gray(img);

% 计算图像的直方图

% 创建一个长度为256的全零向量histogram来存储图像的直方图
histogram = zeros(1, 256);
% 获取灰度图像gray_img的行数rows和列数cols
[rows, cols] = size(gray_img);
for i = 1:rows
    for j = 1:cols
        % 获取当前像素的灰度值pixel_value
        pixel_value = gray_img(i, j);
        % 灰度值范围为0-255，故将histogram中对应灰度值加1
        histogram(pixel_value + 1) = histogram(pixel_value + 1) + 1;
    end
end

% 计算累积分布函数cdf

% 创建一个长度为256的全零向量cdf来存储累积分布函数
cdf = zeros(1, 256);
% 初始化cdf的第一个值为histogram的第一个值
cdf(1) = histogram(1);
for i = 2:256
    % 通过循环计算累积分布函数
    cdf(i) = cdf(i - 1) + histogram(i);
end
```

```
% 归一化cdf，累积分布函数cdf除以图像的总像素数，使得结果在0到1之间
cdf_normalized = cdf / (rows * cols);

% 进行直方图均衡化
equalized_img = zeros(rows, cols,'like',gray_img);
for i = 1:rows
    for j = 1:cols
        % 获取当前像素的灰度值pixel_value
        pixel_value = gray_img(i, j);
        % 灰度值范围为0-255，故将cdf_normalized中对应灰度值加1
        equalized_img(i, j) = round(cdf_normalized(pixel_value + 1) * 255);
    end
end

% 同态滤波

% 首先进行对数变换
log_img = log(double(gray_img)+1);

% 进行二维离散傅里叶变换，将图像从空间域转换到频域
f = fft2(log_img);

% 设定滤波器参数

% 获取图像的行数和列数
M = size(log_img,1);
N = size(log_img,2);
% 分别创建一个从0到M-1的向量和一个从0到N-1的向量，表示图像的行索引和列索引
u = 0:(M-1);
v = 0:(N-1);
```

```
% 找到大于图像行数一半的索引，将其减去行数，实现中心对称
idx = find(u>M/2);
u(idx) = u(idx)-M;
% 找到大于图像列数一半的索引，将其减去列数，实现中心对称
idy = find(v>N/2);
v(idy) = v(idy)-N;
% 创建二维网格，用于后续计算频率坐标
[V,U] = meshgrid(v,u);
% 计算频率坐标的模
D = sqrt(U.^2 + V.^2);
% 定义同态滤波器函数，用于调整照度和反射率分量
H = (0.5 + 2 * ((D./(D + 0.5))));

% 滤波
% 将滤波器与频域图像相乘，实现滤波操作
filtered_f = H.* f;

% 逆傅里叶变换和指数变换
% 对滤波后的图像进行逆傅里叶变换，将其从频域转换回空间域
% 然后进行指数变换，恢复到原始图像的形式，再减1与前面的对数变换对应
filtered_img = real(exp(ifft2(filtered_f)) - 1);

% 显示直方图均衡化图像和同态滤波图像
figure;
subplot(1,2,1);
imshow(equalized_img);
title('直方图均衡化图像');
subplot(1,2,2);
imshow(filtered_img);
title('同态滤波图像');
```

实验结果

直方图均衡化图像

同态滤波图像

从图像上可以看到，直方图均衡化图像的灰度级更多，能看到更多细节与差距，同态滤波图像灰度级少，图像轮廓明显，对比度更强。