

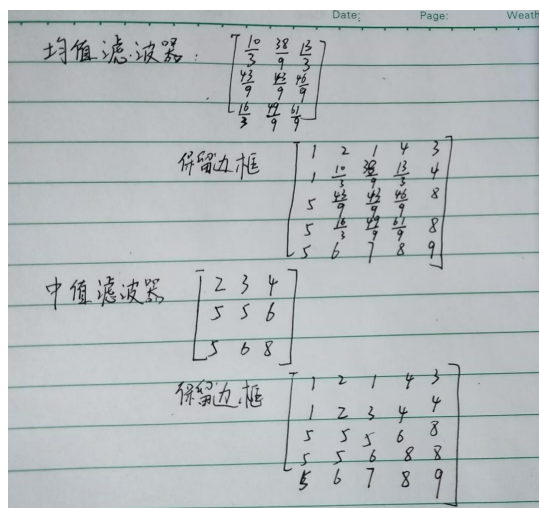
数字图像处理第二次作业

杨鲲

人工智能 2204

U202215264

理论作业：



编程作业：

1.

代码展示：

```
% 读取低照度图像
img = imread('1.jpg');

% 灰度化处理
grayImg = rgb2gray(img);

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

% 计算并显示灰度直方图
figure;
imhist(grayImg);
title('灰度直方图');

% 计算离散傅里叶变换
dftImg = fft2(double(grayImg)); % 计算二维 DFT
dftImgShifted = fftshift(dftImg); % 将零频率分量移到频谱中心
```

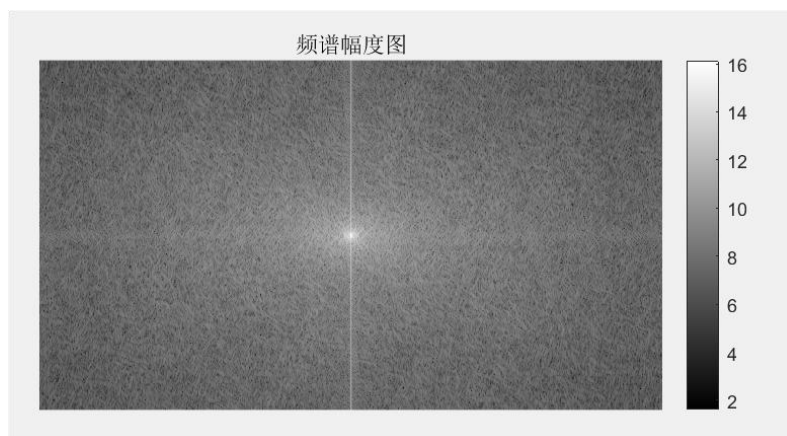
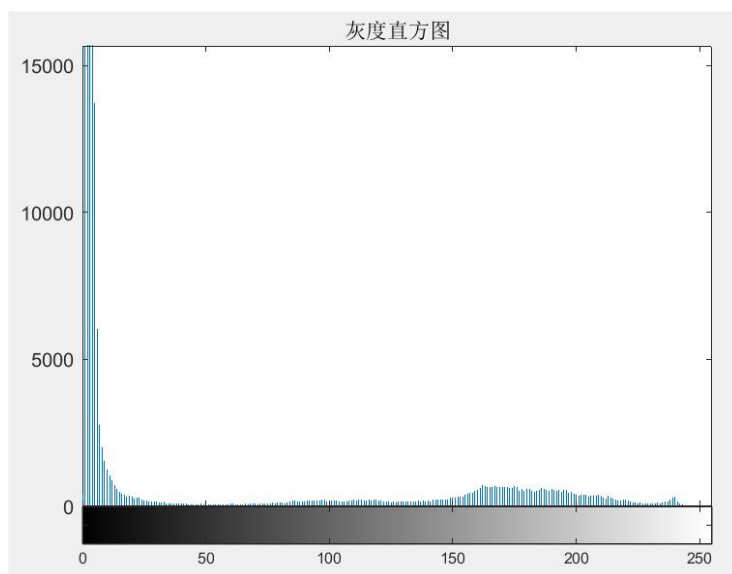
```

% 计算幅度谱
magnitudeSpectrum = log(abs(dftImgShifted) + 1); % 使用 log(1+x)以避免在显示时
出现无穷大

% 显示幅度谱
figure;
imshow(magnitudeSpectrum, []);
title('频谱幅度图');
colormap gray; % 使用灰度色图
colorbar; % 添加颜色条以表示幅度范围

```

结果:



2.

代码展示:

```

% 读取低照度图像
img = imread('1.jpg');
img_gray = rgb2gray(img); % 转为灰度图像

```

```

%% 自定义直方图均衡化
function img_eq = custom_histeq(img)
% 计算直方图
[counts, ~] = imhist(img);
% 归一化直方图
pdf = counts / sum(counts);
% 计算累积分布函数 (CDF)
cdf = cumsum(pdf);
% 将 CDF 归一化到 [0, 255]
cdf_normalized = uint8(255 * cdf);
% 使用 CDF 更新图像
img_eq = cdf_normalized(double(img) + 1); % +1 因为 MATLAB 的索引从 1 开始
end

img_eq = custom_histeq(img_gray);

%% 自定义同态滤波
function img_homo = custom_homomorphic_filter(img)
% 转换为浮点数
img_float = im2double(img);
% 取对数
img_log = log(1 + img_float);
% 创建高通滤波器
[rows, cols] = size(img_log);
[X, Y] = meshgrid(1:cols, 1:rows);
D0 = 30; % 截止频率
D = sqrt((X - cols/2).^2 + (Y - rows/2).^2);
H = 1 - exp(-(D.^2) / (2 * (D0^2))); % 高通滤波器

% 进行傅里叶变换
F = fft2(img_log);
F_shifted = fftshift(F);
% 应用高通滤波器
G_shifted = F_shifted .* H;
% 逆傅里叶变换
G = ifftshift(G_shifted);
img_homo = ifft2(G);
img_homo = exp(real(img_homo)) - 1; % 取指数并减去 1
img_homo = mat2gray(img_homo); % 归一化
end

img_homomorphic = custom_homomorphic_filter(img_gray);

```

```

%% 显示结果
figure;

subplot(1, 3, 1);
imshow(img_gray);
title('原始低照度图像');

subplot(1, 3, 2);
imshow(img_eq);
title('自定义直方图均衡化');

subplot(1, 3, 3);
imshow(img_homomorphic);
title('自定义同态滤波');

% 调整图像显示
sgtitle('低照度图像处理对比');

```

结果:

