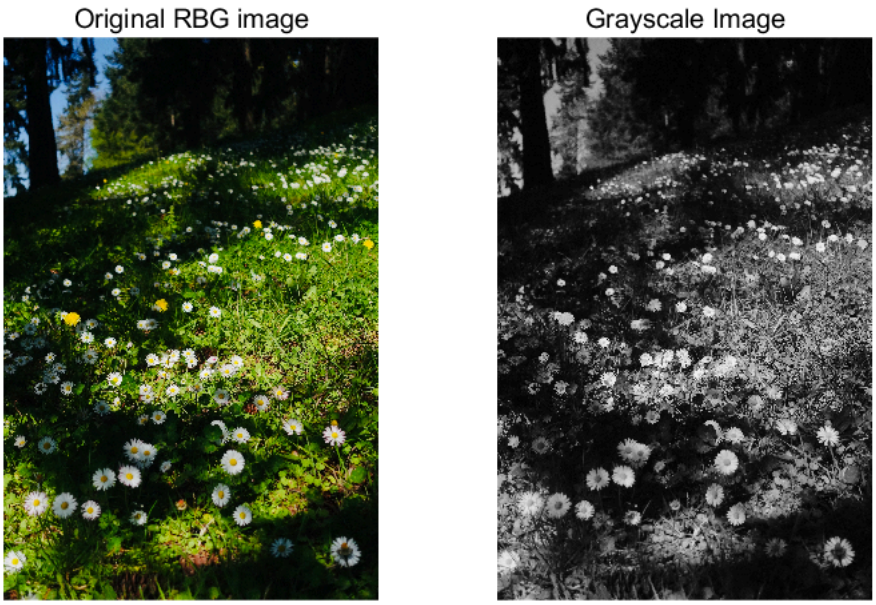
```
%problem 2
clear all; close all; clc;

A = imread('C:\Users\shuaizhouWang\Desktop\6030\img\en.jpg');
size(A)
```

```
ans = 1×3
        3888        2592           3
```

```


figure;
subplot(1,2,1)
imshow(A);
title('Original RBG image', 'FontSize', 18)

Abw = rgb2gray(A); % turn the RGB image into a grayscale image
size(Abw)
```

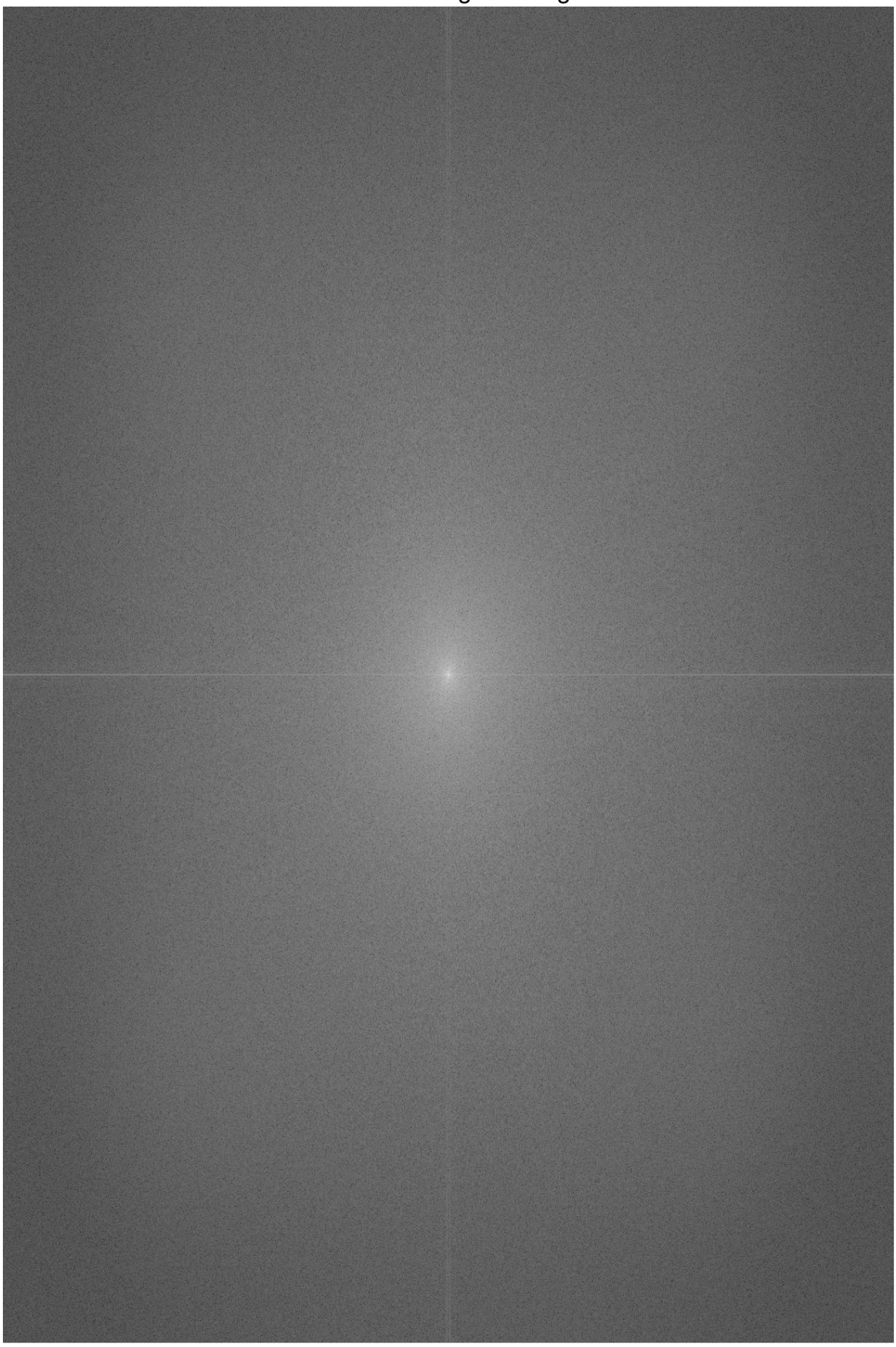
```
ans = 1×2
        3888        2592
```

```
subplot(1,2,2)
imshow(Abw)
title('Grayscale Image', 'FontSize', 18)
figure(3);
subplot(2,2,1)
imshow(Abw)
title('Grayscale Image', 'FontSize', 18)


plot_ind    = 1; % used for subplot index tracking
[nx,ny]     = size(Abw); % identify number of rows and columns in A
N_entries   = nx*ny;

A_fft = fft2(Abw); % take the 2D fft of the image
F     = log(abs(fftshift(A_fft))+1); % helpful for visualizing the data in frequency domain
F     = mat2gray(F); % scaling image for visualizing in grayscale

figure;
imshow(F, []);
```

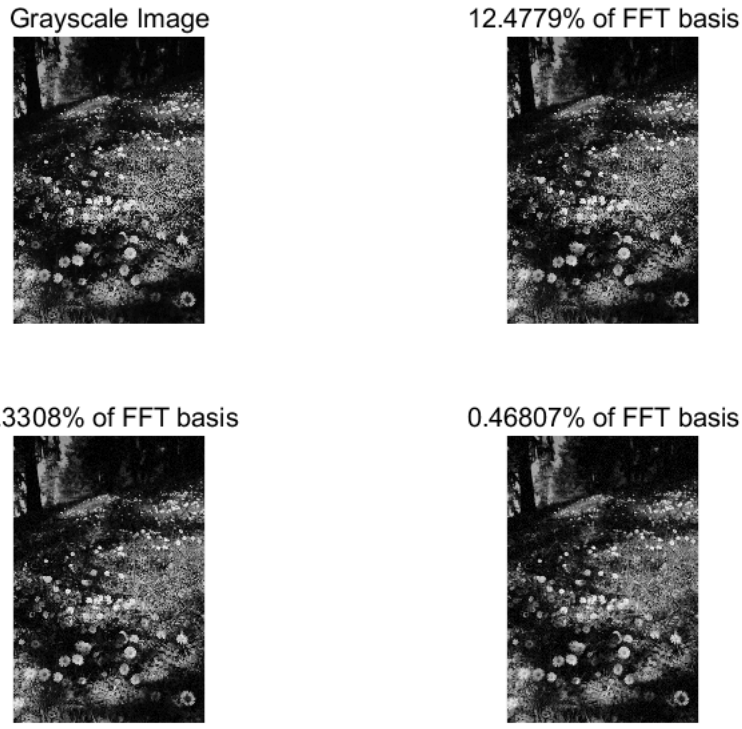



```
title('2D FFT of original image', 'FontSize', 18)
```

2D FFT of original image



```
for thresh = 0.01*[0.01 0.05 0.1]*max(abs(A_fft(:))) % loop over 3 levels of compression (filte
    plot_ind        = plot_ind + 1; % this is just used to identify where in our subplot to plo
    inds            = abs(A_fft)>thresh; % identify which indices correspond to magnitudes of t
    A_fft_filter    = A_fft.*inds; % set all such corresponding entries to zero
    filt_count      = N_entries - nnz(A_fft_filter); % count the number of entries filtered out
    filt_percent    = 100 - filt_count/N_entries*100; % compute the percentage of filtering per

    % Transform the filtered data back to physical (image) space using the iFFT
    A_filter        = uint8(ifft2(A_fft_filter)); % the uint8 just puts the data in a format th


    % Plot the filtered image in the subplot
    figure(3);
    subplot(2,2,plot_ind)
    imshow(A_filter)
    drawnow; % show the image as the code is running
    title([num2str(filt_percent) '% of FFT basis'], 'FontSize', 18)
end
```




```
plot_ind    = 1; % used for subplot index tracking
[nx,ny]     = size(Abw); % identify number of rows and columns in A
N_entries   = nx*ny;


for thresh = 0.01*[0.0001 0.0005 0.001]*max(abs(A_fft(:))) % loop over 3 levels of compression
    plot_ind        = plot_ind + 1; % this is just used to identify where in our subplot to plo
```

```
    inds            = abs(A_fft)>thresh; % identify which indices correspond to magnitudes of t
    A_fft_filter    = A_fft.*inds; % set all such corresponding entries to zero
    filt_count      = N_entries - nnz(A_fft_filter); % count the number of entries filtered out
    filt_percent    = 100 - filt_count/N_entries*100; % compute the percentage of filtering per

    % Transform the filtered data back to physical (image) space using the iFFT
    A_filter        = uint8(ifft2(A_fft_filter)); % the uint8 just puts the data in a format th


    % Plot the filtered image in the subplot
    figure(4);
    subplot(2,2,plot_ind)
    imshow(A_filter)
    drawnow; % show the image as the code is running
    title([num2str(filt_percent) '% of FFT basis'], 'FontSize', 18)
end
```