

```

% Load the Image and Convert to Grayscale
% Load the image
img = imread('bookpage.jpg'); % Replace 'your_image.png' with the actual image
path

% Convert to grayscale if the image is RGB
if size(img, 3) == 3
    img_gray = rgb2gray(img);
else
    img_gray = img;
end

% Display the original grayscale image
figure;
imshow(img_gray);
title('Original Grayscale Image');

```

Original Grayscale Image



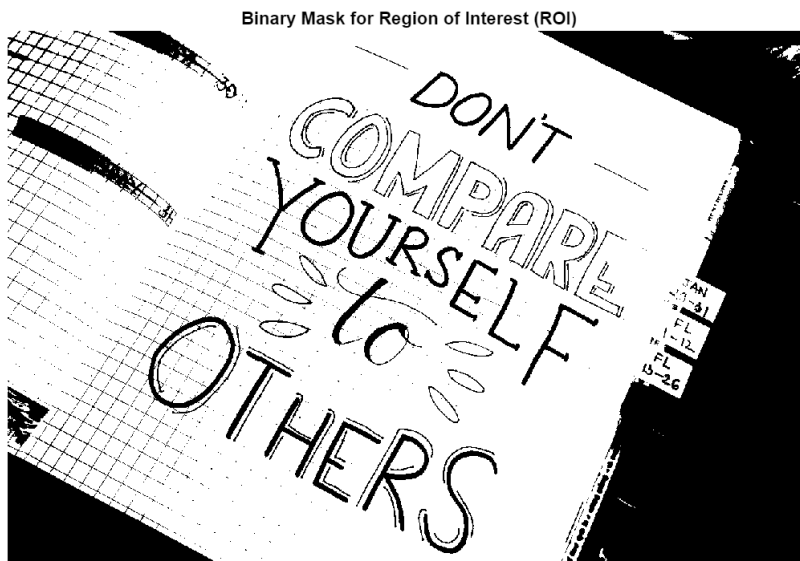
```

% Automatically Create a Binary Mask via Thresholding
% We will use Otsu's method for automatic thresholding to create the binary
mask
% for the region of interest.
% Apply Otsu's thresholding to create a binary mask
threshold_level = graythresh(img_gray); % Automatic threshold using Otsu's
method
binary_mask = imbinarize(img_gray, threshold_level);

% Display the binary mask

```

```
figure;  
imshow(binary_mask);  
title('Binary Mask for Region of Interest (ROI)');
```



```
% Apply Low-Pass Gaussian filter for smoothing  
gaussian_filtered = imgaussfilt(img_gray, 2); % Standard deviation = 2 for  
Gaussian filter  
  
% Display the Gaussian filtered image  
figure;  
imshow(gaussian_filtered);  
title('Gaussian Filtered Image (Low-Pass)');
```

Gaussian Filtered Image (Low-Pass)



```
% Apply average filter (also known as box filter)
average_filter = fspecial('average', [5 5]); % Create a 5x5 averaging filter
average_filtered = imfilter(img_gray, average_filter);

% Display the average filtered image
figure;
imshow(average_filtered);
title('Average Filtered Image (Low-Pass)');
```

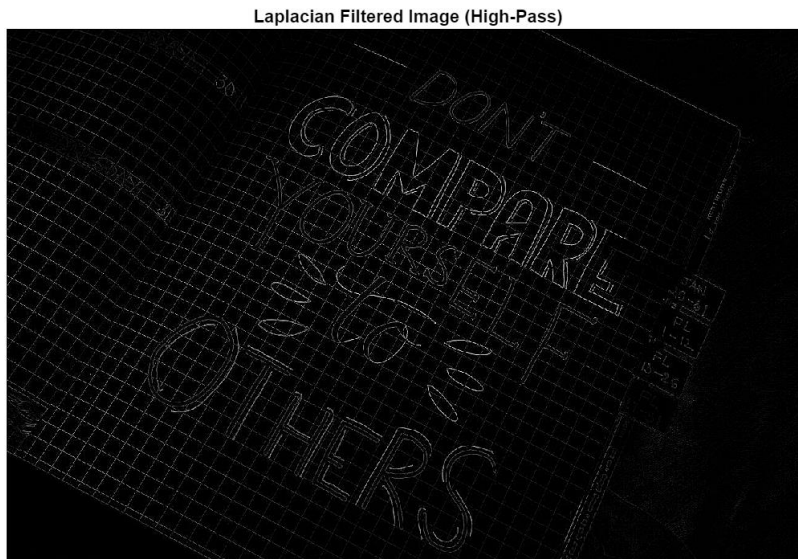
Average Filtered Image (Low-Pass)



```
% Apply High-Pass Laplacian filter for edge detection
```

```
laplacian_filter = fspecial('laplacian', 0.2); % Alpha = 0.2 controls the
sensitivity of the filter
laplacian_filtered = imfilter(img_gray, laplacian_filter);

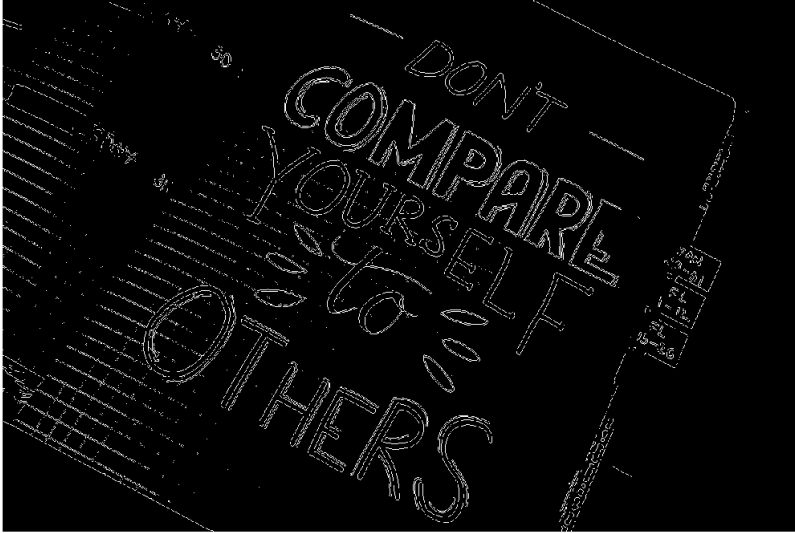
% Display the Laplacian filtered image
figure;
imshow(laplacian_filtered, []);
title('Laplacian Filtered Image (High-Pass)');
```



```
% Apply Prewitt filter for edge detection
prewitt_filtered = edge(img_gray, 'Prewitt');

% Display the Prewitt filtered image
figure;
imshow(prewitt_filtered, []);
title('Prewitt Filtered Image (High-Pass)');
```

Prewitt Filtered Image (High-Pass)



```
% Apply the Filters Only on the Region of Interest
% We want to apply the filters only on the region of interest (ROI) defined by
the binary mask,
% so we will multiply the binary mask by the filtered image.

% Apply Gaussian filter on the ROI only
gaussian_roi = gaussian_filtered .* uint8(binary_mask);

% Apply Average filter on the ROI only
average_roi = average_filtered .* uint8(binary_mask);

% Apply Laplacian filter on the ROI only
laplacian_roi = laplacian_filtered .* uint8(binary_mask);
% Convert to double to handle negative values

% Apply Prewitt filter on the ROI only
prewitt_roi = prewitt_filtered .* binary_mask;

% Display filtered images on the ROI
figure;
subplot(2, 2, 1), imshow(gaussian_roi, []), title('Gaussian Filter on ROI');
subplot(2, 2, 2), imshow(average_roi, []), title('Average Filter on ROI');
subplot(2, 2, 3), imshow(laplacian_roi, []), title('Laplacian Filter on ROI');
subplot(2, 2, 4), imshow(prewitt_roi, []), title('Prewitt Filter on ROI');
```

Gaussian Filter on ROI



Average Filter on ROI



Laplacian Filter on ROI



Prewitt Filter on ROI

