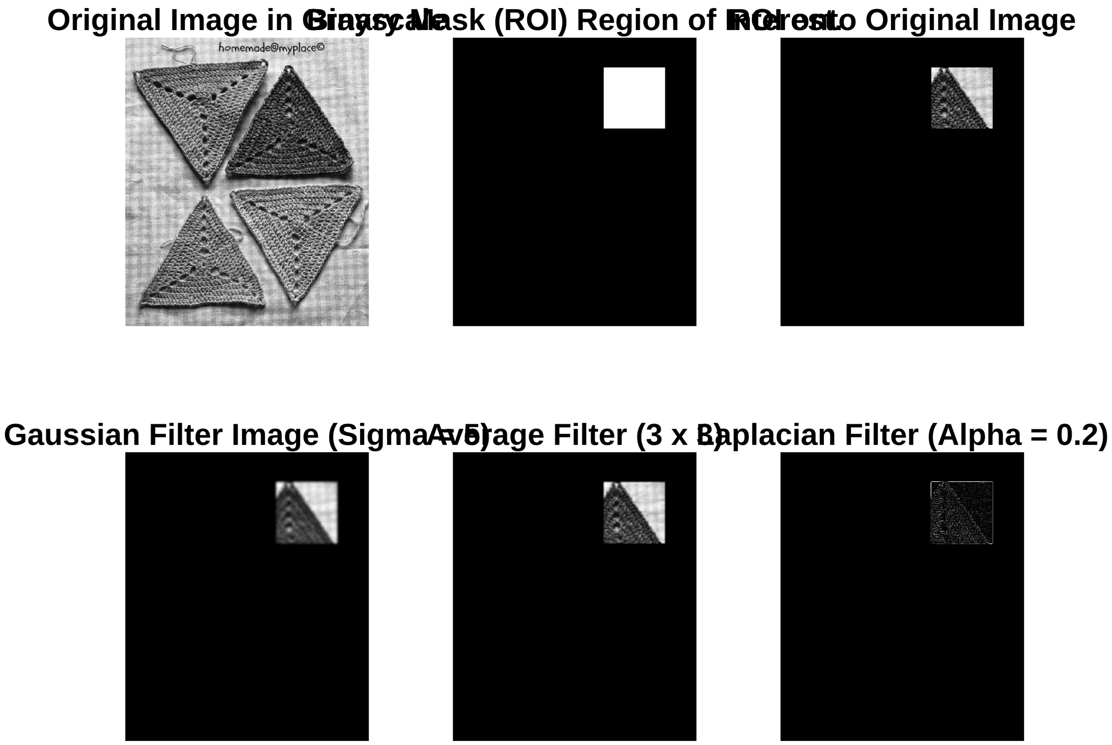
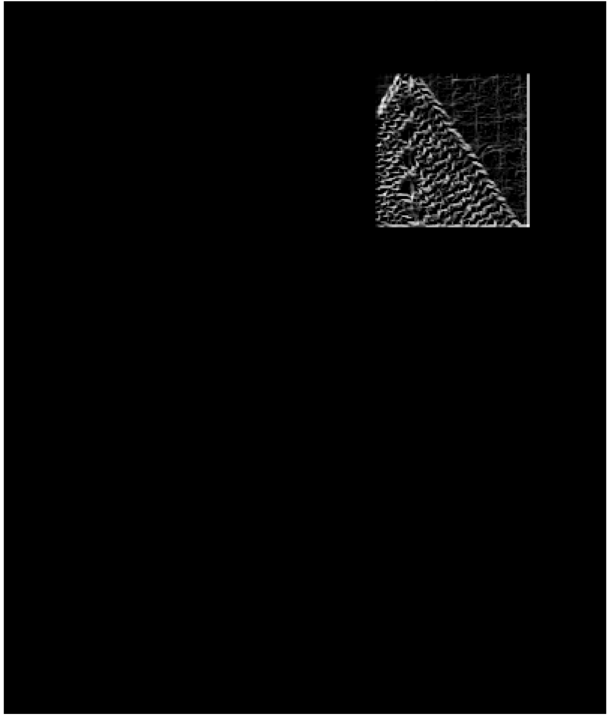
```
% Loading the image using imread().
img = imread('triangles.jpeg');
rgb_gray = rgb2gray(img); % convert the image from rgb to grayscale to get
more uniformed image.


binary_roi = false(size(rgb_gray)); % Binary mask on the ROI (Region of
interest).
binary_roi(50:150, 250:350) = true; % The region of interst(ROI) coordinates.


roi_ori_img = rgb_gray .* uint8(binary_roi); % Mapping the binary mask image
coordinates onto our original image and extracting that ROI.


gauss_filt = imgaussfilt(roi_ori_img, 2);  % Applying the gaussian filter
using imgaussfilt() on our "ROI" image, The sigma value can be adjusted
according to the intensity of the filter, higher or lower.



avg_filter = fspecial('average', [3, 3]);  % Using fspecial() and 3x3
averaging filter to smooth out the image.
avg_filter_img = imfilter(roi_ori_img, avg_filter); % imfilter() is used to
apply the filter onto our image.


lap_filter = fspecial('laplacian', 0.5);  % Laplacian filter is used
to enhance the edges in an image based on the values from 0 to
1,                                          % with 0 being slight emphasis
on edges and 1 being more weight on center pixels making the edges look
sharper.
lap_filter_img = imfilter(roi_ori_img, lap_filter);


prewitt_horizontal = imfilter(roi_ori_img, fspecial('prewitt')); % To detect
horizontal edges in an image.
prewitt_vertical = imfilter(roi_ori_img, fspecial('prewitt')'); % To detect
vertical edges in an image.
prewitt_img = sqrt(double(prewitt_horizontal).^2 +
double(prewitt_vertical).^2); % Combine both the functions to get overall
results of edge detection using sqrt().


figure;

subplot(2, 3, 1);
imshow(rgb_gray);
title('Original Image in Grayscale');
```

```
subplot(2, 3, 2);
imshow(binary_roi);
title('Binary Mask (ROI) Region of Interest.');

subplot(2, 3, 3);
imshow(roi_ori_img);
title('ROI onto Original Image');

subplot(2, 3, 4);
imshow(gauss_filt, []);
title('Gaussian Filter Image (Sigma = 5)');

subplot(2, 3, 5);
imshow(avg_filter_img, []);
title('Average Filter (3 x 3)');

subplot(2, 3, 6);
imshow(lap_filter_img, []);
title('Laplacian Filter (Alpha = 0.2)');
```


Original Image in Grayscale
Binary Mask (ROI) Region of Interest.
ROI onto Original Image
homemade@myplace©
Gaussian Filter Image (Sigma = 5)
Average Filter (3 x 3)
Laplacian Filter (Alpha = 0.2)

```
figure;
imshow(prewitt_img, []);
title('Prewitt Filtered Image');
```



Prewitt Filtered Image

```
% Conclusion:

% Gaussian Filter is used to smooth out the image by putting more emphasis
on the ceter pixels,
% while Averaging filter is more focused on treating all the neighboring
pixels equally for a more unifrom bluring effect.

% Laplacian filter is focused more on putting emphasis on detecting edges
uniformly with out focussing on their direction,
% where as, Prewitt filter, can be used in specifc use cases, where
detecting edges in a specifc direction is the priority.
```