Question 1

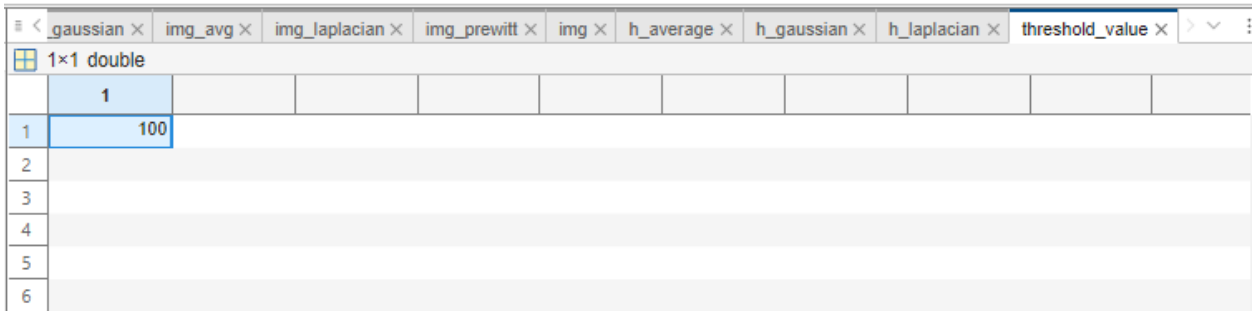
1. Create a binary mask for the region of interest in the image, then apply low-pass filters (Gaussian and Average filters) and high-pass filters (Laplacian and Prewitt filters) in MATLAB.

Matlab Code:

```
% Loading image

img = imread('Bat.jpg');


% Binary mask

gray_img = rgb2gray(img);

threshold_value = 100;

binary_mask = imbinarize(gray_img, threshold_value/255);


% Low pass Filters

% Gaussian filter

h_gaussian = fspecial('gaussian', [5,5], 2);

img_gaussian = imfilter(gray_img, h_gaussian);


% Average filter

h_average = fspecial('average', [5,5]);

img_avg = imfilter(gray_img, h_average);


% High pass Filters

% Laplacian filter

h_laplacian = fspecial('laplacian', 0.2);

img_laplacian = imfilter(gray_img, h_laplacian);
```

```
% Prewitt filter for edge detection
img_prewitt = edge(gray_img, 'prewitt');

% 5. Display the original, binary mask, and filtered images
figure;
subplot(3,2,1), imshow(gray_img), title('Original Grayscale Image');
subplot(3,2,2), imshow(binary_mask), title('Binary Mask');
subplot(3,2,3), imshow(img_gaussian), title('Gaussian Filter');
subplot(3,2,4), imshow(img_avg), title('Average Filter');
subplot(3,2,5), imshow(img_laplacian), title('Laplacian Filter');
subplot(3,2,6), imshow(img_prewitt), title('Prewitt Filter');
```

Threshold Value:



The output for the code:

Original Grayscale Image

Binary Mask

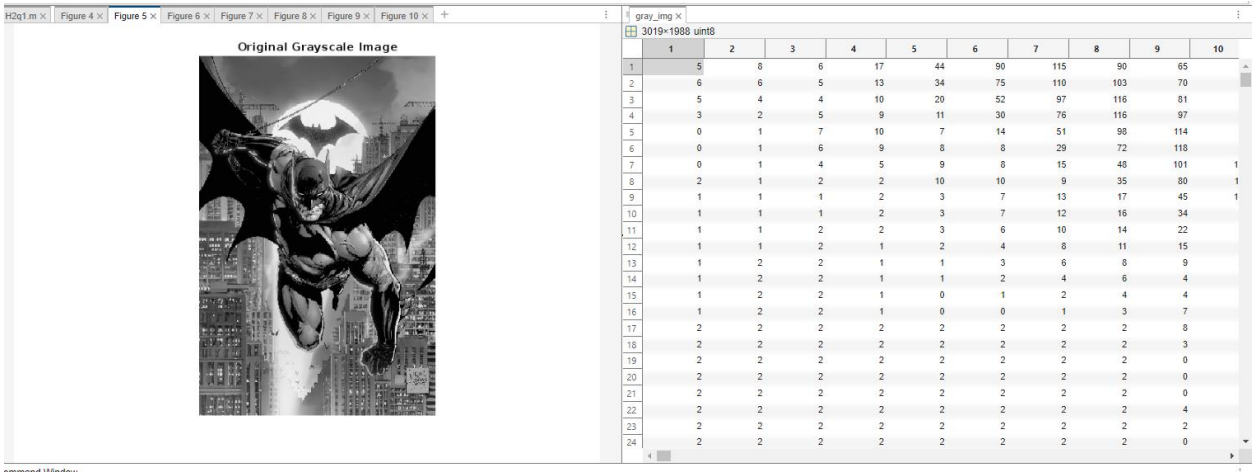
Gaussian Filter

Average Filter

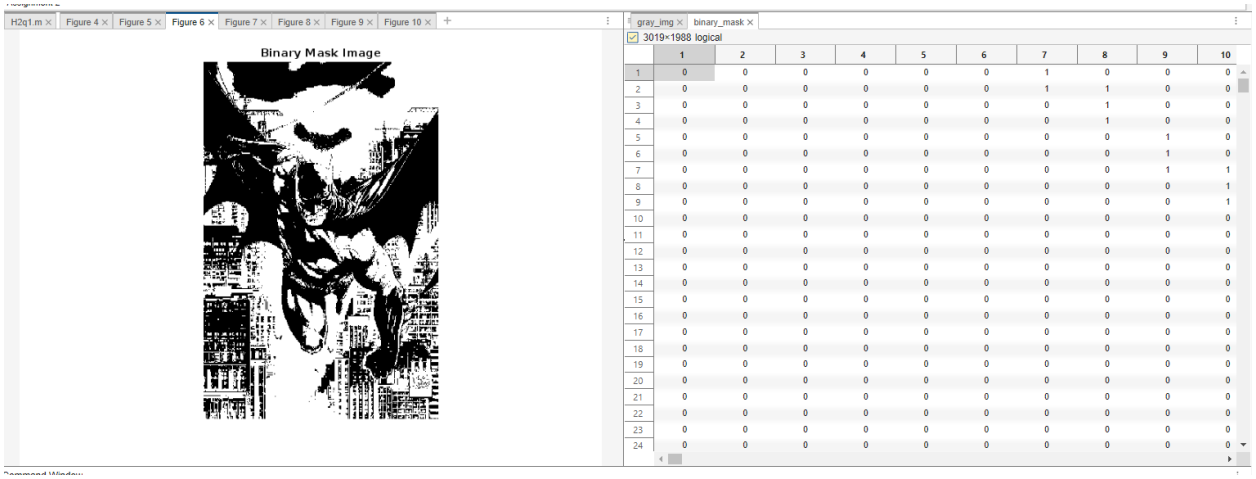
Laplacian Filter

Prewitt Filter

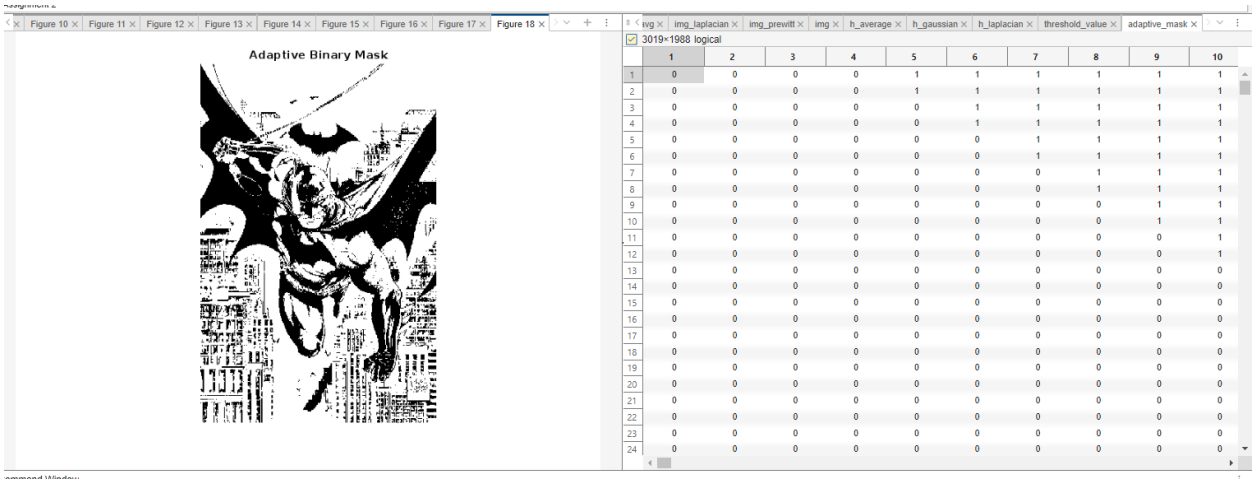
Original Gray Scale image:



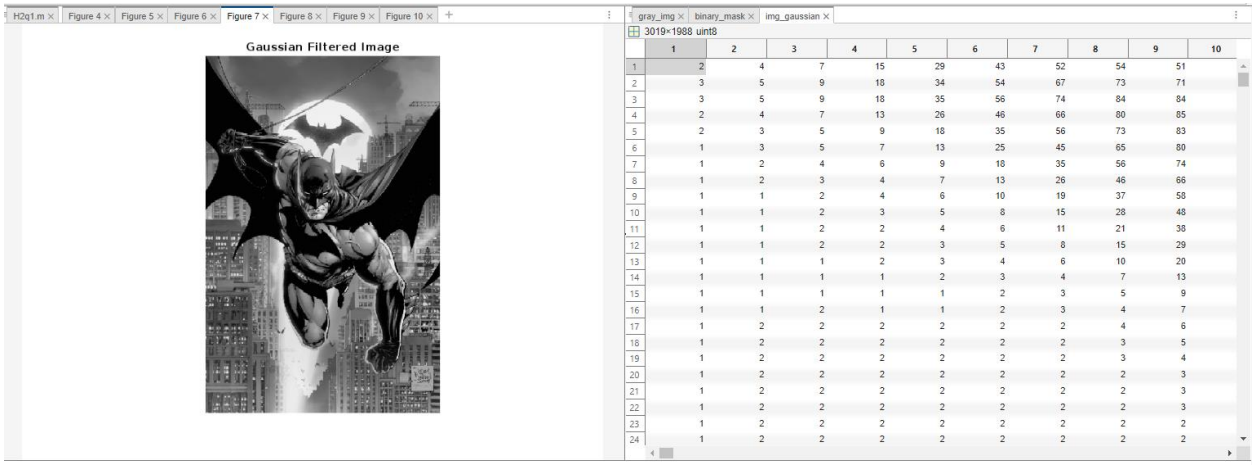
Binary Mask Image:



Adaptive Binary Mask:



Gaussian Filtered Image:

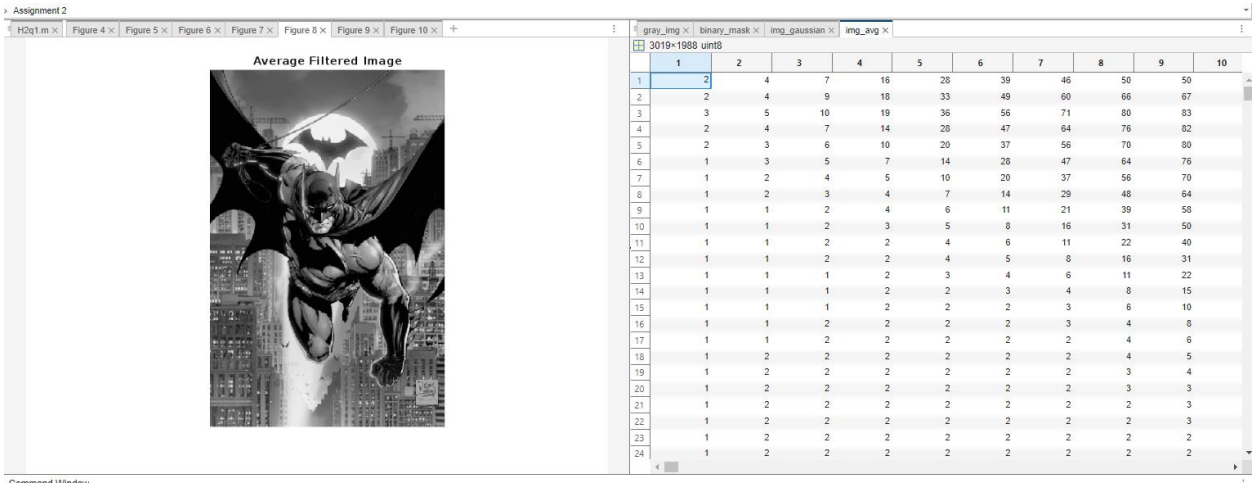


inary_mask × | img_gaussian × | img_avg × | img_laplacian × | img_prewitt × | img × | h_average × | h_gaussian × | h_laplacian ×

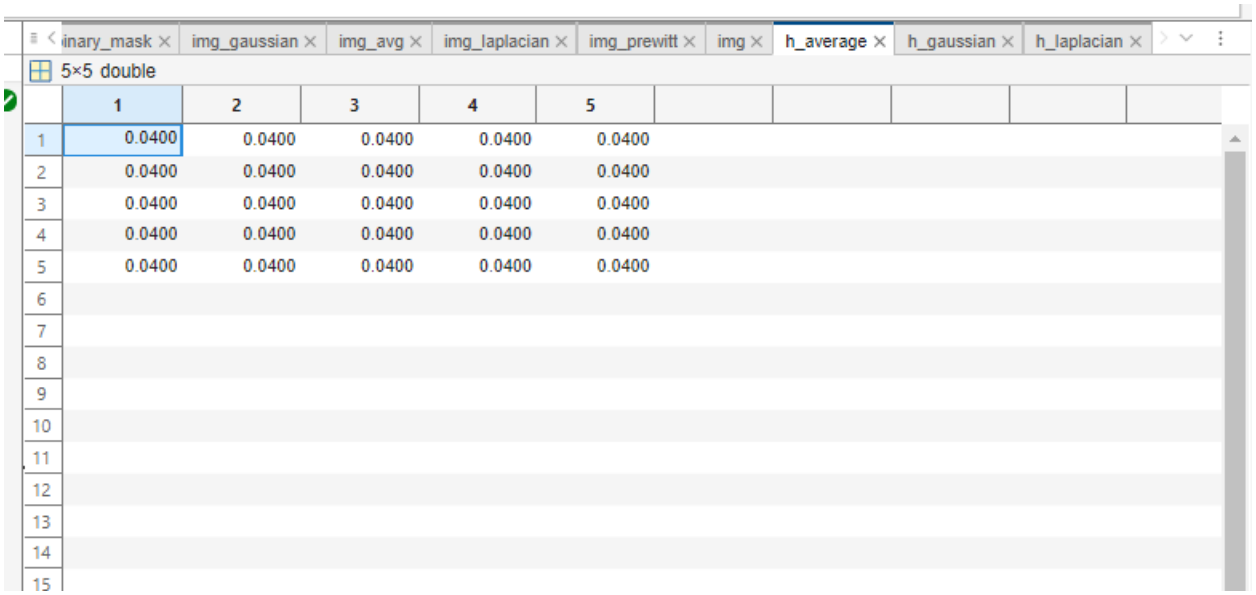
5×5 double

	1	2	3	4	5
1	0.0232	0.0338	0.0383	0.0338	0.0232
2	0.0338	0.0492	0.0558	0.0492	0.0338
3	0.0383	0.0558	0.0632	0.0558	0.0383
4	0.0338	0.0492	0.0558	0.0492	0.0338
5	0.0232	0.0338	0.0383	0.0338	0.0232
6					
7					
8					
9					

Average Filtered Image:

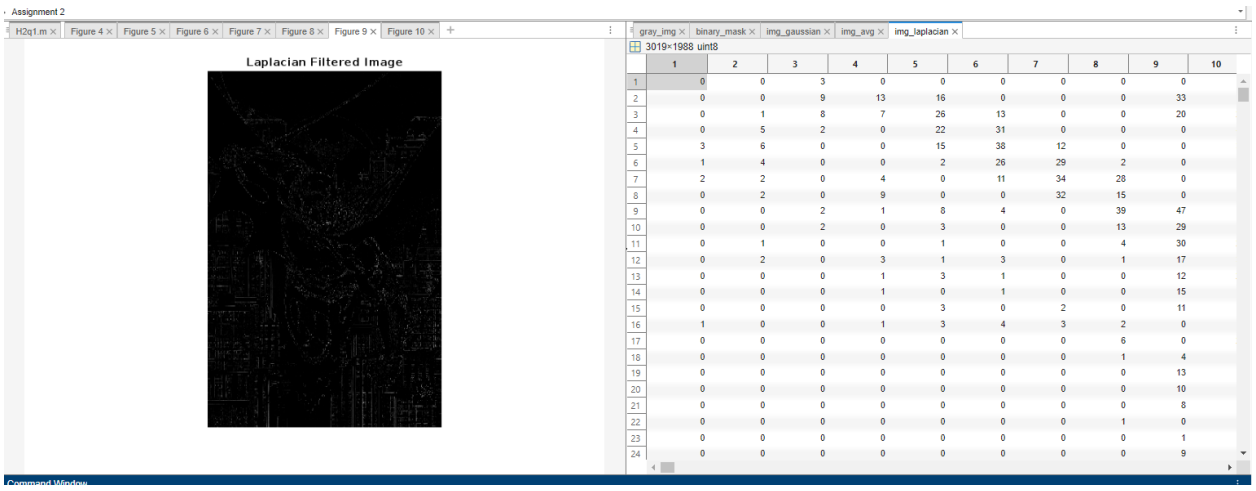


	1	2	3	4	5	6	7	8	9	10
1	2	4	7	16	28	39	46	50	50	
2	2	4	9	18	33	49	60	66	67	
3	3	5	10	19	36	56	71	80	83	
4	2	4	7	14	28	47	64	76	82	
5	2	3	6	10	20	37	56	70	80	
6	1	3	5	7	14	28	47	64	76	
7	1	2	4	5	10	20	37	56	70	
8	1	2	3	4	7	14	29	48	64	
9	1	1	2	4	6	11	21	39	58	
10	1	1	2	3	5	8	16	31	50	
11	1	1	2	2	4	6	11	22	40	
12	1	1	2	2	4	5	8	16	31	
13	1	1	1	2	3	4	6	11	22	
14	1	1	1	2	2	3	4	8	15	
15	1	1	1	2	2	2	3	6	10	
16	1	1	2	2	2	2	3	4	8	
17	1	1	2	2	2	2	2	4	6	
18	1	2	2	2	2	2	2	4	5	
19	1	2	2	2	2	2	2	3	4	
20	1	2	2	2	2	2	2	3	3	
21	1	2	2	2	2	2	2	2	3	
22	1	2	2	2	2	2	2	2	3	
23	1	2	2	2	2	2	2	2	2	
24	1	2	2	2	2	2	2	2	2	



	1	2	3	4	5
1	0.0400	0.0400	0.0400	0.0400	0.0400
2	0.0400	0.0400	0.0400	0.0400	0.0400
3	0.0400	0.0400	0.0400	0.0400	0.0400
4	0.0400	0.0400	0.0400	0.0400	0.0400
5	0.0400	0.0400	0.0400	0.0400	0.0400
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

Laplacian Filtered Image:



	1	2	3	4	5	6	7	8	9	10
1	0	0	3	0	0	0	0	0	0	
2	0	0	9	13	16	0	0	0	33	
3	0	1	8	7	26	13	0	0	20	
4	0	5	2	0	22	31	0	0	0	
5	3	6	0	0	15	38	12	0	0	
6	1	4	0	0	2	26	29	2	0	
7	2	2	0	4	0	11	34	28	0	
8	0	2	0	9	0	0	32	15	0	
9	0	0	2	1	8	4	0	39	47	
10	0	0	2	0	3	0	0	13	29	
11	0	1	0	0	1	0	0	4	30	
12	0	2	0	3	1	3	0	1	17	
13	0	0	0	1	3	1	0	0	12	
14	0	0	0	1	0	1	0	0	15	
15	0	0	0	0	3	0	2	0	11	
16	1	0	0	1	3	4	3	2	0	
17	0	0	0	0	0	0	0	6	0	
18	0	0	0	0	0	0	0	1	4	
19	0	0	0	0	0	0	0	0	13	
20	0	0	0	0	0	0	0	0	10	
21	0	0	0	0	0	0	0	0	8	
22	0	0	0	0	0	0	0	1	0	
23	0	0	0	0	0	0	0	0	1	
24	0	0	0	0	0	0	0	0	9	

inary_mask × | img_gaussian × | img_avg × | img_laplacian × | img_prewitt × | img × | h_average × | h_gaussian × | h_laplacian ×

3×3 double

	1	2	3
1	0.1667	0.6667	0.1667
2	0.6667	-3.3333	0.6667
3	0.1667	0.6667	0.1667
4			
5			
6			
7			
8			
9			

Prewitt Filtered Image:

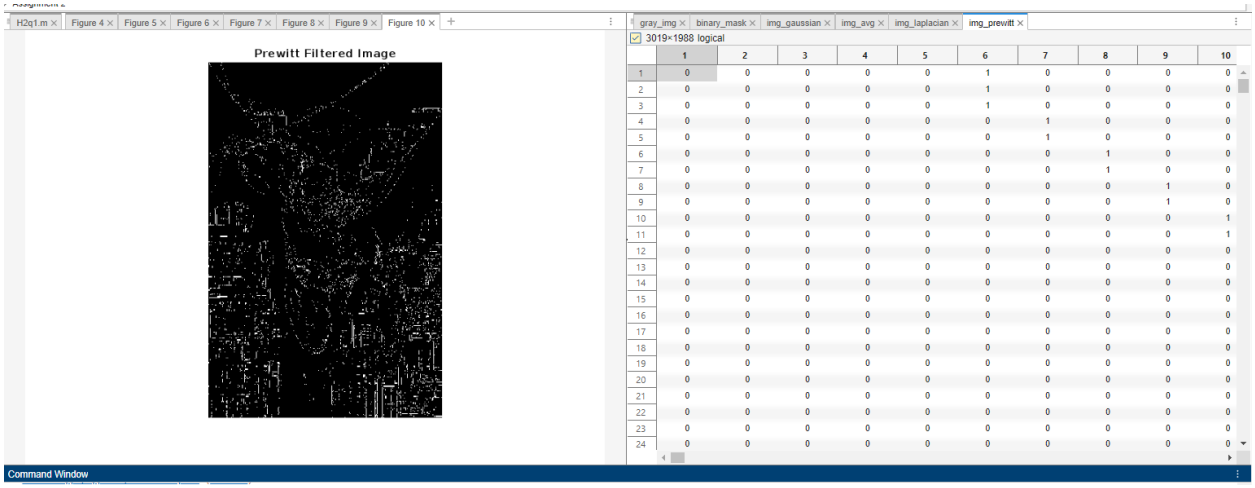
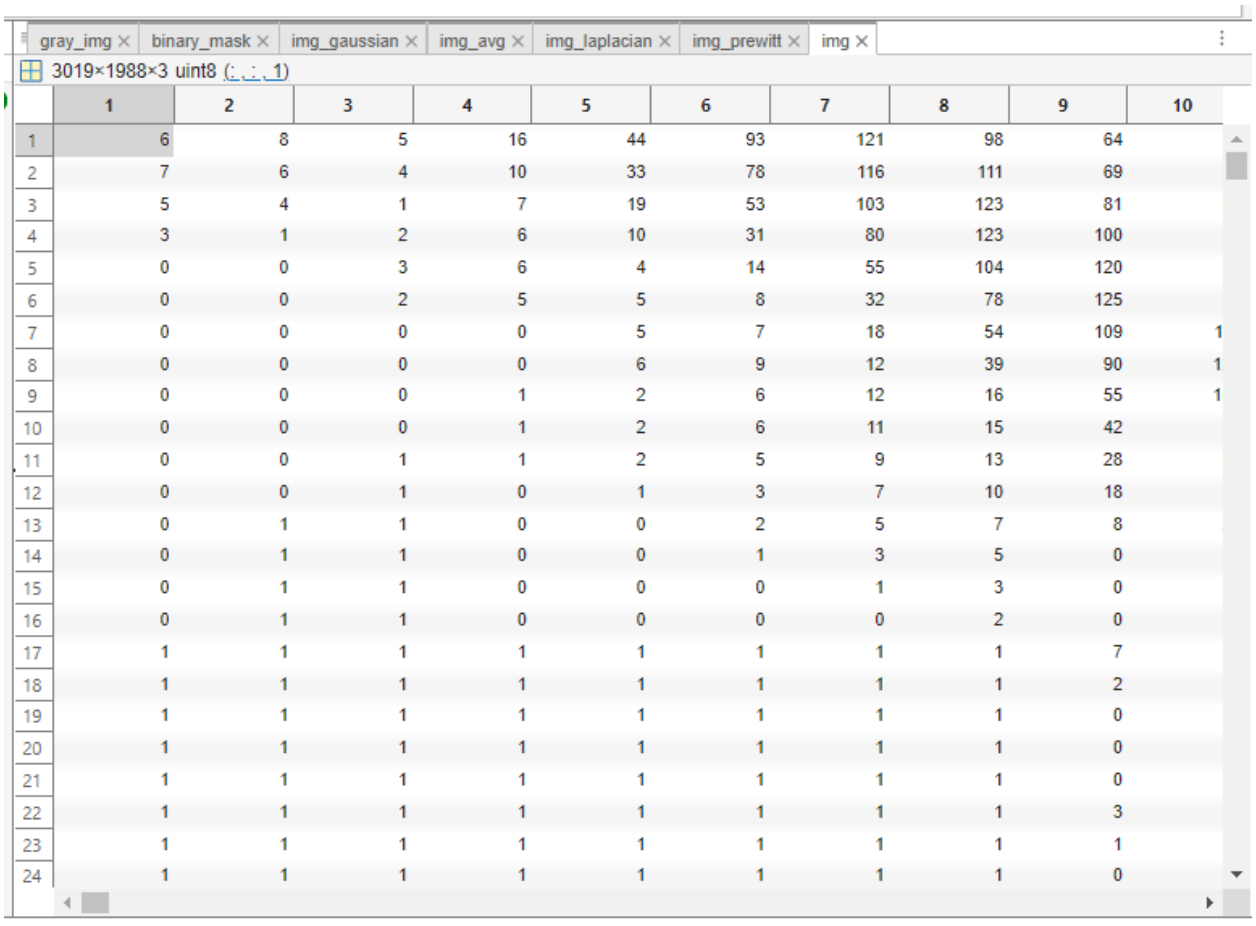


Image:



gray_img × | binary_mask × | img_gaussian × | img_avg × | img_laplacian × | img_prewitt × | img ×

3019×1988×3 uint8 (:, :, 1)

	1	2	3	4	5	6	7	8	9	10
1	6	8	5	16	44	93	121	98	64	
2	7	6	4	10	33	78	116	111	69	
3	5	4	1	7	19	53	103	123	81	
4	3	1	2	6	10	31	80	123	100	
5	0	0	3	6	4	14	55	104	120	
6	0	0	2	5	5	8	32	78	125	
7	0	0	0	0	5	7	18	54	109	1
8	0	0	0	0	6	9	12	39	90	1
9	0	0	0	1	2	6	12	16	55	1
10	0	0	0	1	2	6	11	15	42	
11	0	0	1	1	2	5	9	13	28	
12	0	0	1	0	1	3	7	10	18	
13	0	1	1	0	0	2	5	7	8	
14	0	1	1	0	0	1	3	5	0	
15	0	1	1	0	0	0	1	3	0	
16	0	1	1	0	0	0	0	2	0	
17	1	1	1	1	1	1	1	1	7	
18	1	1	1	1	1	1	1	1	2	
19	1	1	1	1	1	1	1	1	0	
20	1	1	1	1	1	1	1	1	0	
21	1	1	1	1	1	1	1	1	0	
22	1	1	1	1	1	1	1	1	3	
23	1	1	1	1	1	1	1	1	1	
24	1	1	1	1	1	1	1	1	0	