

## Digital Image Processing Mini Project #3

# Unsharp Masking

Seyed Hesamoddin Hosseini

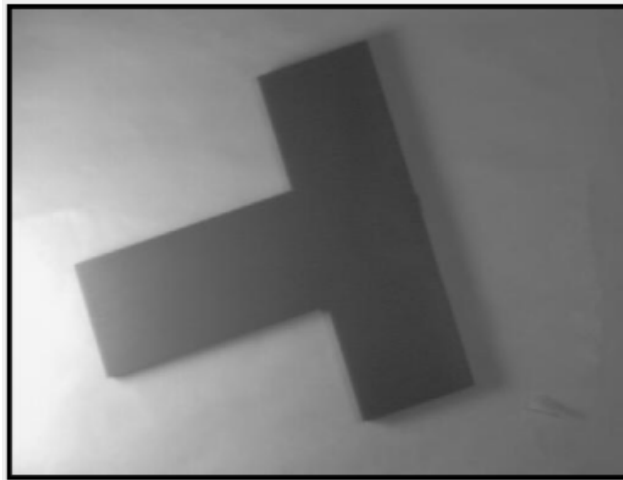
### خلاصه

بناست در این مینی پروژه با استفاده از Box Filter مناسب، برای تعدادی تصویر خاکستری (grayscale) لبه ها را تقویت کنیم. بدین منظور از دو الگوریتم نقاب گذاری غیر تیز و لاپلاسیین استفاده شده است. در الگوریتم نقاب گذاری غیر تیز از روش های فیلتر هموار ساز میانگین ساده، فیلتر هموار ساز میانگین وزن دار و فیلتر میانه با اندازه هایی همچون  $3 \times 3$ ،  $5 \times 5$ ،  $7 \times 7$  و  $9 \times 9$  استفاده شده است. در روش لاپلاسیین به دو روش غیر وفقی با یک کلیشه و وفقی با 8 کلیشه، سعی در تیز کردن لبه های سه تصویر خاکستری (grayscale) داریم. برای پیاده سازی این پروژه از Matlab استفاده می نماییم.

### شرح تکنیکال:

#### مرحله اول:

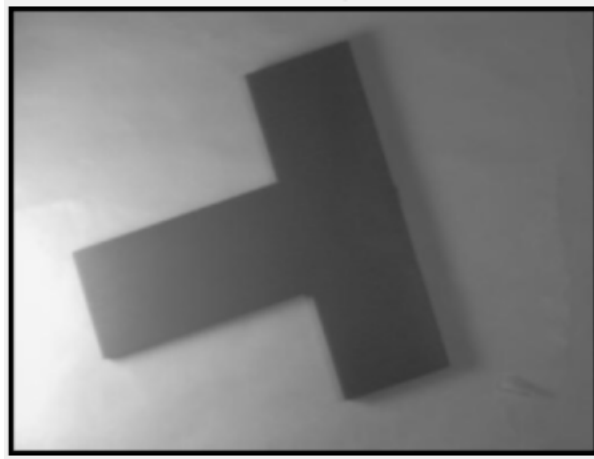
به کمک استفاده از یک فیلتر هموارساز میانگین  $3 \times 3$  از نوع Box Filter، نقاب گذاری غیرتیز را به تصویر شماره (1) اعمال می کنیم. برای این کار ابتدا به تصویر مورد نظر zero padding اعمال می نماییم تا برای محاسبه ماسک روی لبه های تصویر، دچار مشکل نشویم. سپس ماسکی مانند شکل شماره (2) طراحی می کنیم. سپس آن را بر روی تصویر اصلی حرکت می دهیم. نتیجه به صورت شکل شماره (3) می باشد. لبه ها تار شده اند و وضوح آن ها کاهش یافته است.



شکل 1 - تصویر اصلی

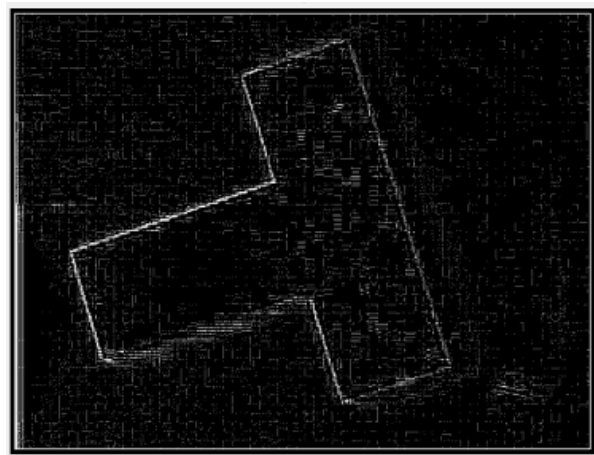
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

شکل 2- فیلتر هموار ساز میانگین بدون وزن



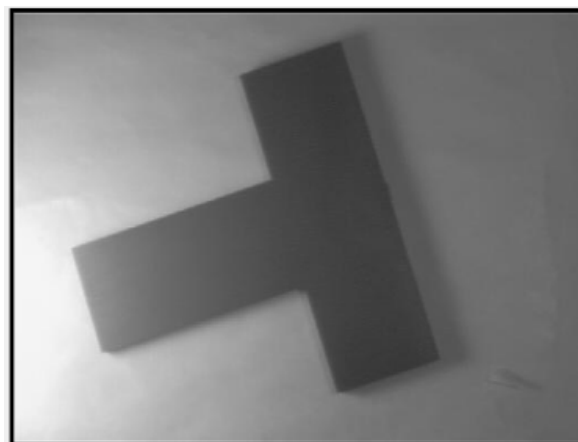
شکل 3- تصویر با لبه های تار شده

حال تصویر هموار شده را از تصویر اصلی کم می نماییم. تصویر حاصل شماره (4) است، که شامل لبه هاست. برای مشاهده بهتر تصویر خروجی، آن را در 40 ضرب کردیم.



شکل 4- لبه ها

حال تصویر لبه را با تصویر اصلی جمع می کنیم. تصویر حاصل شماره (5) است، که لبه ها در آن تقویت شده است.



شکل 5- تصویر با لبه های تیز شده

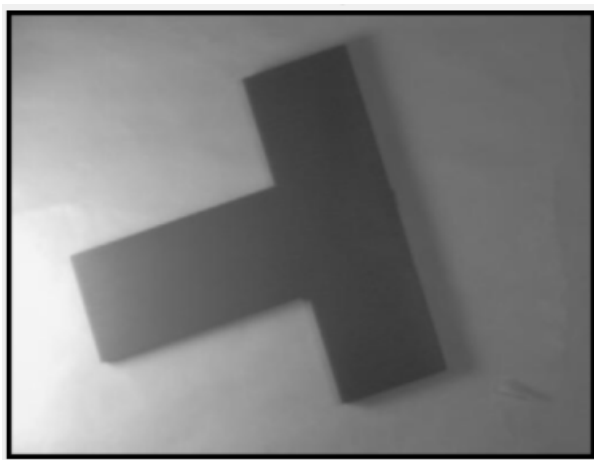
## مرحله دوم:

نقاب‌گذاری غیرتیز را با فیلتر هموار ساز میانگین  $3 \times 3$  از نوع Weighted Average بر روی تصویر شماره (1) پیاده‌سازی می کنیم. و در انتها نتیجه را با نتیجه قبل مقایسه می کنیم.

برای این کار ابتدا به تصویر مورد نظر zero padding اعمال می نماییم تا برای محاسبه ماسک روی لبه های تصویر، دچار مشکل نشویم. سپس ماسکی مانند شکل شماره (6) طراحی می کنیم. سپس آن را بر روی تصویر اصلی حرکت می دهیم. نتیجه به صورت شکل شماره (7) می باشد. لبه ها تار شده اند و وضوح آن ها کاهش یافته است.

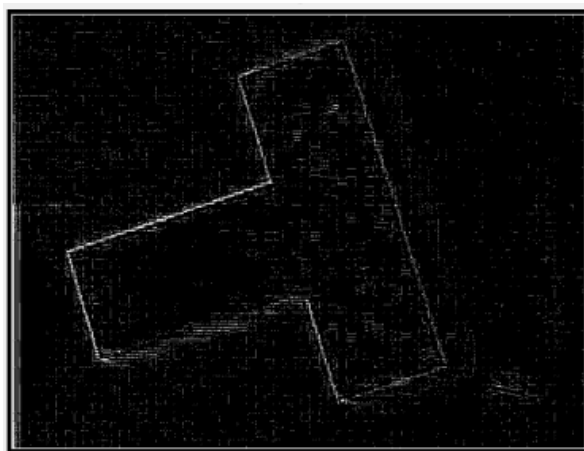
1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

شکل 6- فیلتر میانگین وزن دار



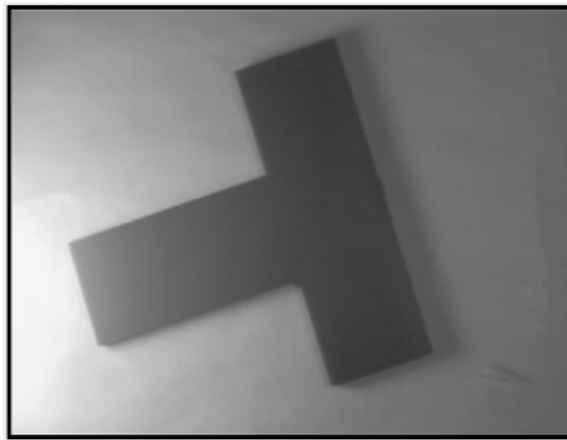
شکل 7 - تصویر با لبه های مات شده در اثر فیلتر وزن دار

حال تصویر هموار شده را از تصویر اصلی کم می نماییم. تصویر حاصل شماره (8) است، که شامل لبه هاست. برای مشاهده بهتر تصویر خروجی، آن را در 40 ضرب کردیم.



شکل 8- لبه ها

حال تصویر لبه را با تصویر اصلی جمع می کنیم. تصویر حاصل شماره (9) است، که لبه ها در آن تقویت شده است.



شکل 9- تصویر با لبه های تیز شده

از مقایسه نتایج مرحله اول و مرحله دوم به این نتیجه میرسیم که چون در حالت دوم، از فیلتر وزن دار استفاده کردیم، مات شدگی لبه به صورت بهتری رخ داد و در نتیجه لبه های دقیق تری داشتیم و در تصویر نهایی، لبه ها دقیق تر شارپ شده اند.

### مرحله سوم:

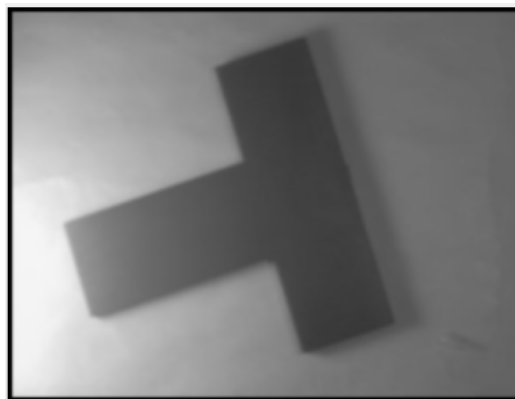
تجربه مرحله اول را با فیلترهای با اندازه بزرگتر  $5 \times 5$ ،  $7 \times 7$  و  $9 \times 9$  تکرار می کنیم. و در انتها بر روی تاثیر بزرگ کردن فیلتر بحث می کنیم.

ابتدا یک فیلتر  $5 \times 5$  میانگین بدون وزن طراحی می کنیم.

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

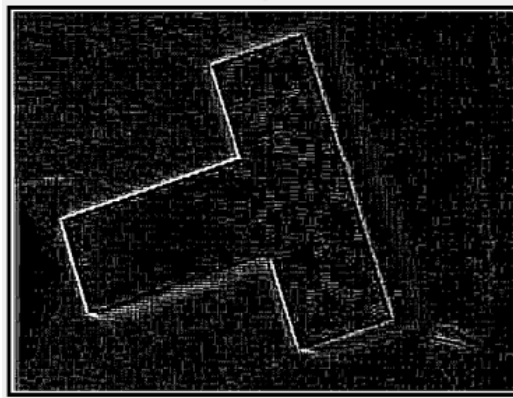
شکل 10- فیلتر  $5 \times 5$  میانگین بدون وزن

سپس آن را بر روی تصویر اصلی حرکت می دهیم. نتیجه به صورت شکل شماره (11) می باشد. لبه ها تار شده اند و وضوح آن ها کاهش یافته است.



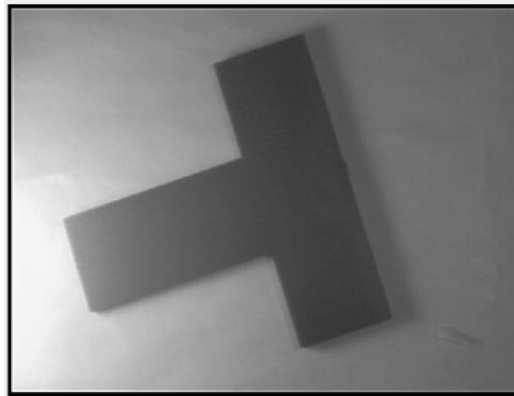
شکل 11- تصویر با لبه های مات شده در اثر فیلتر وزن دار

حال تصویر هموار شده را از تصویر اصلی کم می نماییم. تصویر حاصل شماره (12) است، که شامل لبه هاست. برای مشاهده بهتر تصویر خروجی، آن را در 40 ضرب کردیم.



شکل 12- لبه ها

حال تصویر لبه را با تصویر اصلی جمع می‌کنیم. تصویر حاصل شماره (13) است، که لبه‌ها در آن تقویت شده است.



شکل 13- تصویر با لبه های تیز شده

حال یک فیلتر  $7 \times 7$  میانگین بدون وزن طراحی می‌کنیم.

1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49

شکل 14 - فیلتر  $7 \times 7$  میانگین بدون وزن

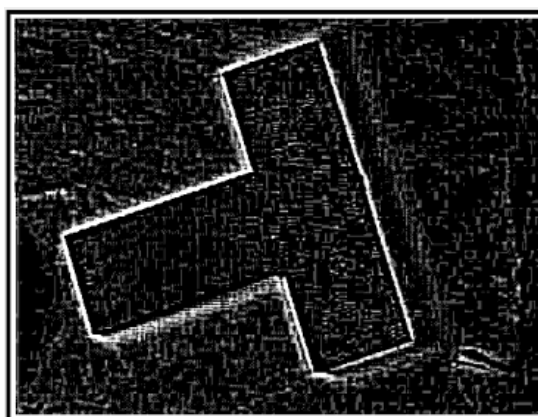


1/81	1/81	1/81	1/81	1/81	1/81	1/81	1/81	1/81
1/81	1/81	1/81	1/81	1/81	1/81	1/81	1/81	1/81
1/81	1/81	1/81	1/81	1/81	1/81	1/81	1/81	1/81

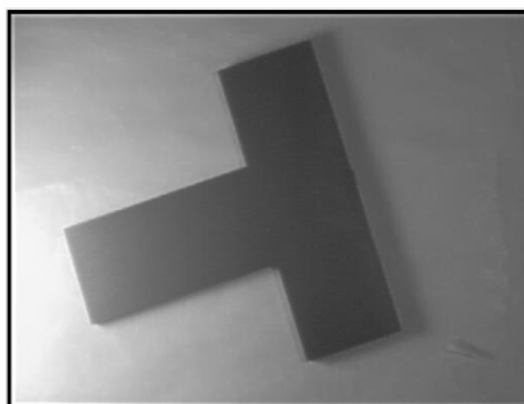
شکل 18 - فیلتر 9\*9 میانگین بدون وزن



شکل 19 - تصویر با لبه های تار شده



شکل 20- لبه ها

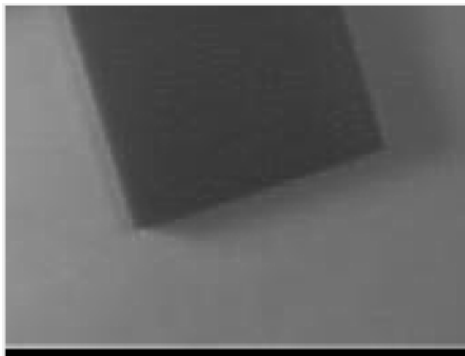


شکل 21 - تصویر با لبه های تیز شده

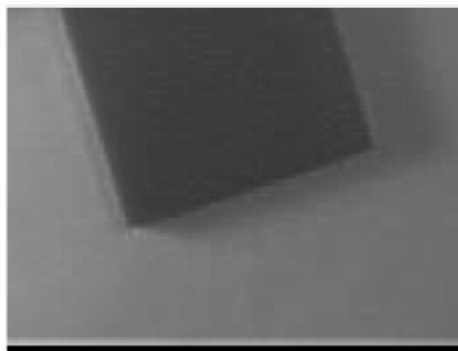
در این قسمت نتایج فیلتر های  $3 \times 3$  ،  $5 \times 5$  ،  $7 \times 7$  و  $9 \times 9$  را در کنار یکدیگر و با بزرگنمایی، مشاهده می نمایید. هرچه اندازه ماسک بزرگتر می شود، ابتدا لبه ها تار تر شده و سپس در نتیجه نهایی، لبه ها تیز تر می شوند.



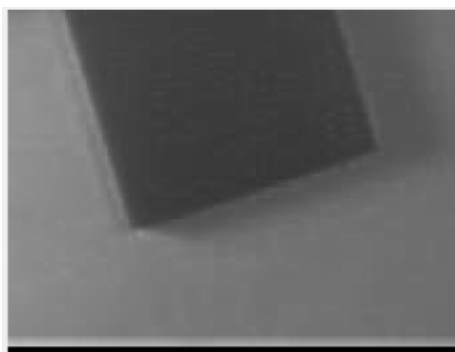
شکل 22 - ماسک 3\*3



شکل 23 - ماسک 5\*5



شکل 24 - ماسک 7\*7

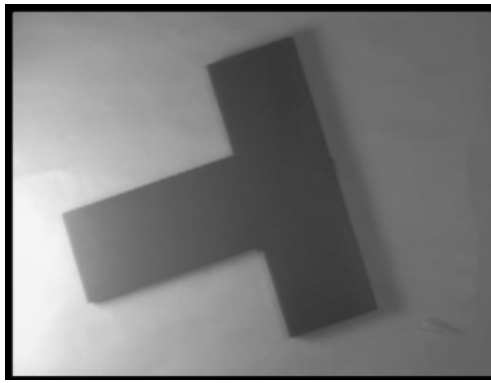


شکل 25 - ماسک 9\*9

#### مرحله چهارم:

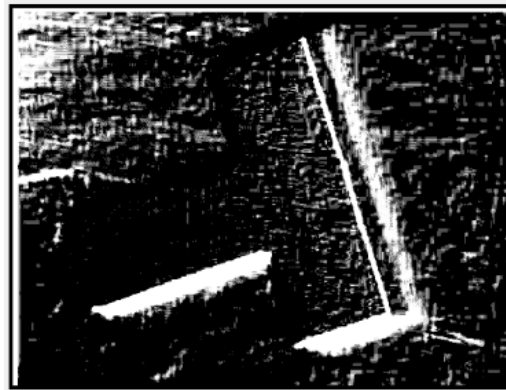
اگر در نقاب‌گذاری غیرتیز فیلتر میانه را به عنوان فیلتر هموارساز استفاده کنیم. چه نتیجه‌ای حاصل می‌شود؟ ابتدا فیلتر میانه را بر روی تصویر اصلی حرکت می‌دهیم. نتیجه به صورت شکل شماره (26) می‌باشد.





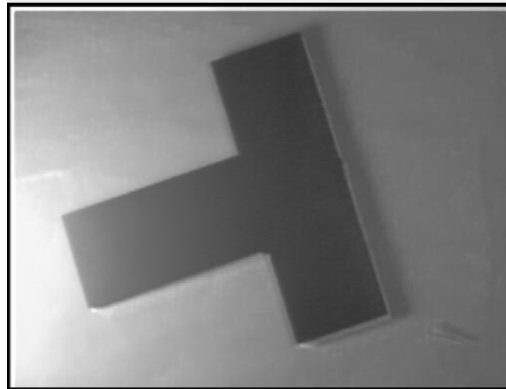
شکل 26- تصویر پس از اعمال فیلتر میانه

حال تصویر هموار شده را از تصویر اصلی کم می نماییم. تصویر حاصل شماره (27) است، که شامل لبه هاست. برای مشاهده بهتر تصویر خروجی، آن را در 40 ضرب کردیم.



شکل 27 حاصل تفریق تصویر اصلی و تصویر بالا

حال تصویر لبه را با تصویر اصلی جمع می کنیم. تصویر حاصل شماره (28) است، که لبه ها در آن تقویت شده است.



شکل 28 - تصویر با لبه های تیز شده

### مرحله پنجم:

در این مرحله می خواهیم عمل تیز کردن لبه های تصویر را به کمک الگوریتم لاپلاسیان انجام دهیم.

کلیشه زیر را به عنوان کلیشه لاپلاسیان در نظر می گیریم

-1/16	-2/16	-1/16
-2/16	12/16	-2/16

-1/16	-2/16	-1/16
-------	-------	-------

شکل 29 - کلیشه لاپلاسی

با استفاده از این کلیشه رابطه زیر می‌تواند موجب تیزشدن تصویر شود:

$$(1) \quad g(x, y) = f(x, y) + \nabla^2 f(x, y)$$

کلیشه فوق را می‌توان 1/16 مجموع کلیشه‌های زیر در نظر گرفت

0	0	0
-2	2	0
0	0	0

-1	0	0
0	1	0
0	0	0

0	-2	0
0	2	0
0	0	0

0	0	-1
0	1	0
0	0	0

0	0	0
0	2	-2
0	0	0

0	0	0
0	1	0
0	0	-1

0	0	0
0	2	0
0	-2	0

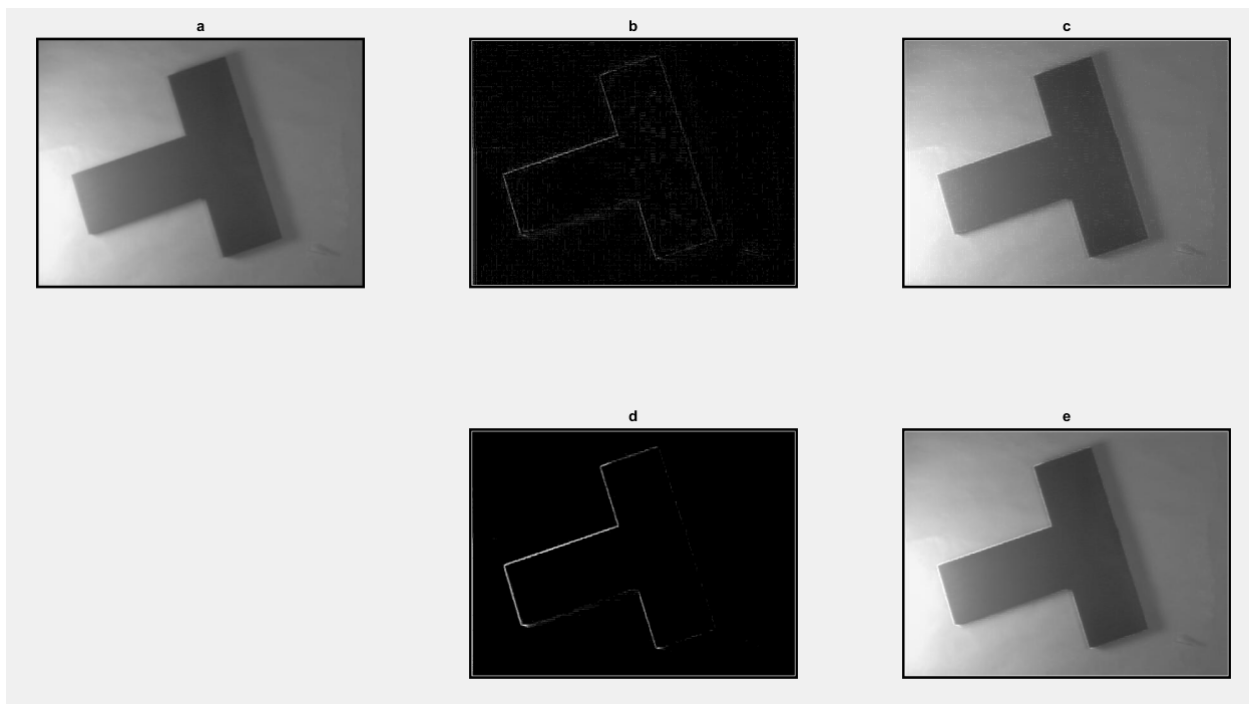
0	0	0
0	1	0
-1	0	0

با توجه به این کلیشه‌ها می‌توان یک تیز کردن و فقی را به این ترتیب پیاده‌سازی کرد که به هر نقطه از تصویر مبدا تمام این کلیشه‌ها را اعمال کرده و حاصل تمام فیلترهایی که از یک آستانه بزرگتر باشند را با هم جمع می‌کنیم. در آخر حاصل را به تصویر اصلی اضافه می‌کنیم.

در این پروژه، مقدار آستانه را برابر با 1 در نظر گرفته ایم.

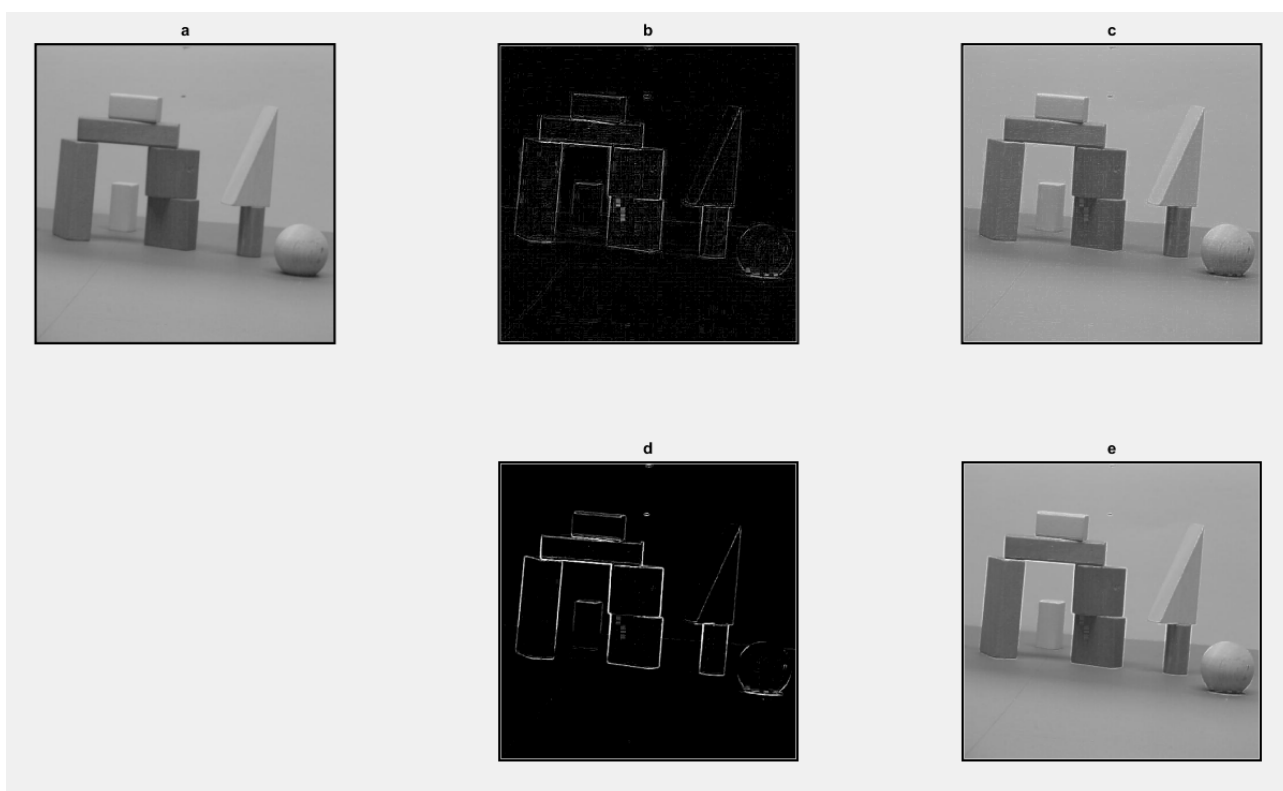
بر این اساس، تیزکردن تصویر به دو شکل غیروفتی و وفتی را بر روی تصاویر Im132 تا Im134 اعمال کرده و نتیجه را با هم مقایسه می‌کنیم.

نتایج اعمال فیلتر لاپلاسی بر روی تصویر Im132 به صورت زیر می‌باشد:



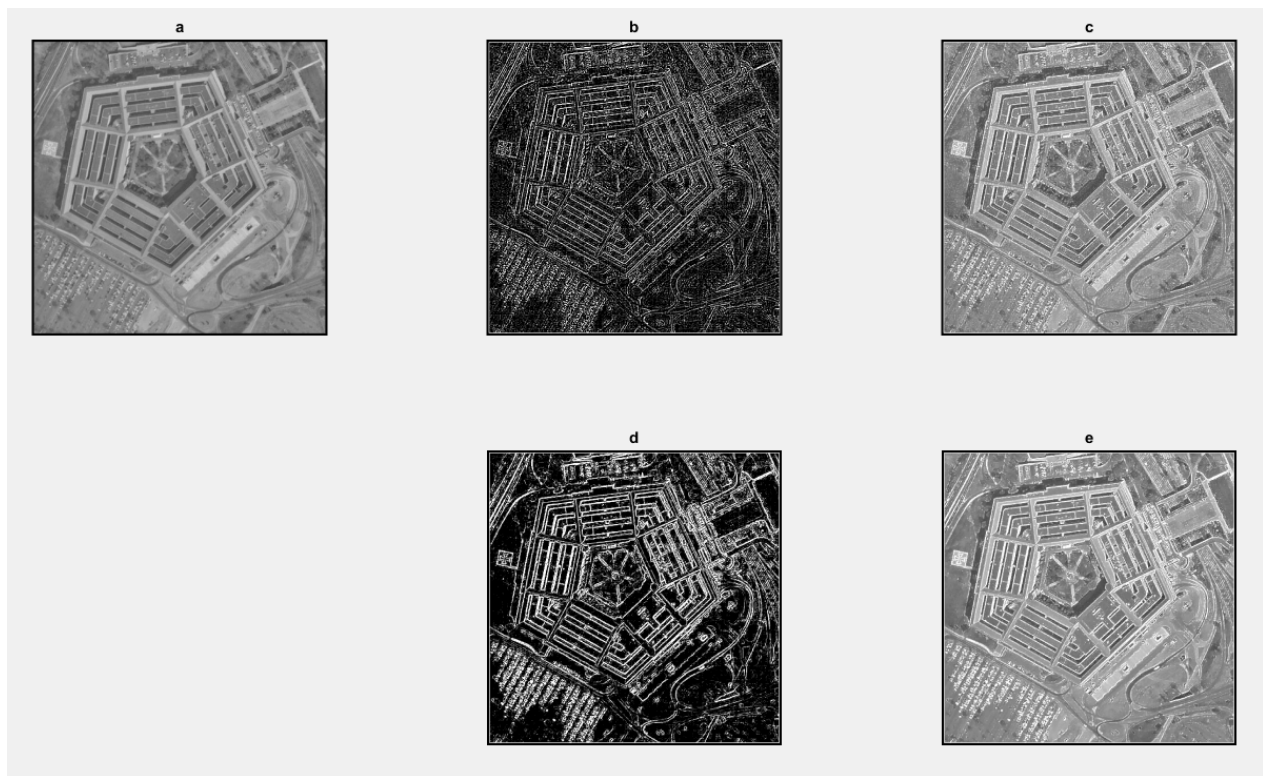
شکل 30- a - تصویر اصلی - b - لبه ها با فیلتر لاپلاسین غیر وفقی - c - تصویر با لبه های تیز شده با فیلتر لاپلاسین غیر وفقی - d - لبه ها با فیلتر لاپلاسین وفقی - e - تصویر با لبه های تیز شده با فیلتر لاپلاسین وفقی

نتایج اعمال فیلتر لاپلاسین بر روی تصویر Im133 به صورت زیر می باشد:



شکل 31 - a - تصویر اصلی - b - لبه ها با فیلتر لاپلاسین غیر وفقی - c - تصویر با لبه های تیز شده با فیلتر لاپلاسین غیر وفقی - d - لبه ها با فیلتر لاپلاسین وفقی - e - تصویر با لبه های تیز شده با فیلتر لاپلاسین وفقی

نتایج اعمال فیلتر لاپلاسین بر روی تصویر Im134 به صورت زیر می باشد:



شکل 32 - a - تصویر اصلی - b - لبه ها با فیلتر لاپلاسین غیر وقتی - c - تصویر با لبه های تیز شده با فیلتر لاپلاسین غیر وقتی - d - لبه ها با فیلتر لاپلاسین وقتی - e - تصویر با لبه های تیز شده با فیلتر لاپلاسین وقتی

### نتیجه گیری:

همانطور که ملاحظه می شود، هنگامی که از فیلتر های وقتی استفاده می کنیم، عمل لبه یابی با دقت بیشتری انجام شده و در نتیجه، تصویر نهایی دارای کیفیت لبه های بالاتری می باشد.

### پیوست:

سورس کد پروژه با نرم افزار متلب:

پیاده سازی تابع masking

```
function output = masking(img, x, y, mask_size, mask)

mask_offset = floor(mask_size/2);
sum = 0;
for row = 1 : mask_size
    for col = 1 : mask_size
        t = double(img(x - mask_offset - 1 + row, y - mask_offset - 1 + col));
        t = t * mask(row, col);
        sum = sum + t;
    end
end
output = sum;
end
```

پیاده سازی مرحله اول:

```
clear;
%read images
Im132 = imread('D:/Im132.png');

BoxFilterAverage = [1/9, 1/9, 1/9;...
                    1/9, 1/9, 1/9;...
                    1/9, 1/9, 1/9];

Im132_zp = padarray(Im132, [4 4]);
[Im132_zp_row_size, Im132_zp_col_size] = size(Im132_zp);
```

```

figure;
subplot(2,2,1), imshow(Im132_zp); title('Original Image');

Im132_zp_smooth = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size-8
    for col=1:Im132_zp_col_size-8
        Im132_zp_smooth(row+4, col+4) = masking(Im132_zp, row+4, col+4, 3,
BoxFilterAverage);
    end
end

subplot(2,2,2), imshow(Im132_zp_smooth); title('Blur Image');

Im132_zp_edges = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_edges(row, col) = Im132_zp(row, col) - Im132_zp_smooth(row, col);
    end
end

subplot(2,2,3), imshow(Im132_zp_edges .* 80); title('Edges');

Im132_zp_sharp = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_sharp(row, col) = Im132_zp(row, col) + Im132_zp_edges(row, col);
    end
end

subplot(2,2,4), imshow(Im132_zp_sharp); title('Sharp Image');

```

پیاده سازی مرحله دوم:

```

Im132 = imread('D:/Im132.png');

BoxFilterAverage = [1/16, 2/16, 1/16;...
                    2/16, 4/16, 2/16;...
                    1/16, 2/16, 1/16];

Im132_zp = padarray(Im132, [4 4]);
[Im132_zp_row_size, Im132_zp_col_size] = size(Im132_zp);

figure;
subplot(2,2,1), imshow(Im132_zp); title('Original Image');

Im132_zp_smooth = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size-8
    for col=1:Im132_zp_col_size-8
        Im132_zp_smooth(row+4, col+4) = masking(Im132_zp, row+4, col+4, 3,
BoxFilterAverage);
    end
end

subplot(2,2,2), imshow(Im132_zp_smooth); title('Blur Image');

Im132_zp_edges = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

```

```

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_edges(row, col) = Im132_zp(row, col) - Im132_zp_smooth(row, col);
    end
end

subplot(2,2,3), imshow(Im132_zp_edges .* 80); title('Edges');

Im132_zp_sharp = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_sharp(row, col) = Im132_zp(row, col) + Im132_zp_edges(row, col);
    end
end

subplot(2,2,4), imshow(Im132_zp_sharp); title('Sharp Image');

```

پیاده سازی مرحله سوم برای فیلتر 5\*5:

```

Im132 = imread('D:/Im132.png');

BoxFilterAverage = [1/25, 1/25, 1/25, 1/25, 1/25;...
                    1/25, 1/25, 1/25, 1/25, 1/25;...
                    1/25, 1/25, 1/25, 1/25, 1/25;...
                    1/25, 1/25, 1/25, 1/25, 1/25;...
                    1/25, 1/25, 1/25, 1/25, 1/25];

Im132_zp = padarray(Im132, [4 4]);
[Im132_zp_row_size, Im132_zp_col_size] = size(Im132_zp);

figure;
subplot(2,2,1), imshow(Im132_zp); title('Original Image');

Im132_zp_smooth = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size-8
    for col=1:Im132_zp_col_size-8
        Im132_zp_smooth(row+4, col+4) = masking(Im132_zp, row+4, col+4, 5,
BoxFilterAverage);
    end
end

subplot(2,2,2), imshow(Im132_zp_smooth); title('Blur Image');

Im132_zp_edges = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_edges(row, col) = Im132_zp(row, col) - Im132_zp_smooth(row, col);
    end
end

subplot(2,2,3), imshow(Im132_zp_edges .* 80); title('Edges');

Im132_zp_sharp = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_sharp(row, col) = Im132_zp(row, col) + Im132_zp_edges(row, col);
    end
end

```

```
subplot(2,2,4), imshow(Im132_zp_sharp); title('Sharp Image');
```

پیاده سازی مرحله سوم برای فیلتر 7\*7:

```
Im132 = imread('D:/Im132.png');

BoxFilterAverage = [1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49;...
                    1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49;...
                    1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49;...
                    1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49;...
                    1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49;...
                    1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49;...
                    1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49];

Im132_zp = padarray(Im132, [4 4]);
[Im132_zp_row_size, Im132_zp_col_size] = size(Im132_zp);

figure;
subplot(2,2,1), imshow(Im132_zp); title('Original Image');

Im132_zp_smooth = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size-8
    for col=1:Im132_zp_col_size-8
        Im132_zp_smooth(row+4, col+4) = masking(Im132_zp, row+4, col+4, 7,
BoxFilterAverage);
    end
end

subplot(2,2,2), imshow(Im132_zp_smooth); title('Blur Image');

Im132_zp_edges = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_edges(row, col) = Im132_zp(row, col) - Im132_zp_smooth(row, col);
    end
end

subplot(2,2,3), imshow(Im132_zp_edges .* 80); title('Edges');

Im132_zp_sharp = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_sharp(row, col) = Im132_zp(row, col) + Im132_zp_edges(row, col);
    end
end

subplot(2,2,4), imshow(Im132_zp_sharp); title('Sharp Image');
```

پیاده سازی مرحله سوم برای فیلتر 9\*9:

%-----

```
Im132 = imread('D:/Im132.png');

BoxFilterAverage = [1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81;...
                    1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81;...
                    1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81;...
                    1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81;...
```

```

1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81;...
1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81;...
1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81;...
1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81;...
1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81, 1/81];

Im132_zp = padarray(Im132, [4 4]);
[Im132_zp_row_size, Im132_zp_col_size] = size(Im132_zp);

figure;
subplot(2,2,1), imshow(Im132_zp); title('Original Image');

Im132_zp_smooth = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size-8
    for col=1:Im132_zp_col_size-8
        Im132_zp_smooth(row+4, col+4) = masking(Im132_zp, row+4, col+4, 9,
BoxFilterAverage);
    end
end

subplot(2,2,2), imshow(Im132_zp_smooth); title('Blur Image');

Im132_zp_edges = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_edges(row, col) = Im132_zp(row, col) - Im132_zp_smooth(row, col);
    end
end

subplot(2,2,3), imshow(Im132_zp_edges .* 80); title('Edges');

Im132_zp_sharp = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_sharp(row, col) = Im132_zp(row, col) + Im132_zp_edges(row, col);
    end
end

subplot(2,2,4), imshow(Im132_zp_sharp); title('Sharp Image');

```

پیاده سازی مرحله چهارم:

```

Im132 = imread('D:/Im132.png');

Im132_zp = padarray(Im132, [4 4]);
[Im132_zp_row_size, Im132_zp_col_size] = size(Im132_zp);

figure;
subplot(2,2,1), imshow(Im132_zp); title('Original Image');

Im132_zp_smooth = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

mask = zeros(9);

for row=5:Im132_zp_row_size-8
    for col=5:Im132_zp_col_size-8
        i = 1;
        for mask_row = row-1:row+1
            for mask_col=col-1:col+1
                mask(i) = Im132_zp(mask_row, mask_col);
            end
        end
    end
end

```



```

        i = i + 1;
    end
end
mask=sort(mask);
Im132_zp_smooth(row+4, col+4) = mask(5);
end
end

subplot(2,2,2), imshow(Im132_zp_smooth); title('Blur Image');

Im132_zp_edges = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_edges(row, col) = Im132_zp(row, col) - Im132_zp_smooth(row, col);
    end
end

subplot(2,2,3), imshow(Im132_zp_edges .* 80); title('Edges');

Im132_zp_sharp = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_sharp(row, col) = Im132_zp(row, col) + Im132_zp_edges(row, col);
    end
end

subplot(2,2,4), imshow(Im132_zp_sharp); title('Sharp Image');

```

پیاده سازی مرحله پنجم برای فیلتر لاپلاسیان:

```

Im132 = imread('D:/Im132.png');
Im133 = imread('D:/Im133.png');
Im134 = imread('D:/Im134.png');

BoxFilterLaplacian = [ -1/16, -2/16, -1/16;...
                      -2/16, 12/16, -2/16;...
                      -1/16, -2/16, -1/16];

BoxFilterLaplacianAdaptive = cat( 3, [0, 0, 0; ...
                                       -2/16, 2/16, 0; ...
                                       0, 0, 0 ...
                                       ], ...
                                   [-1/16, 0, 0; ...
                                    0, 1/16, 0; ...
                                    0, 0, 0 ...
                                    ], ...
                                   [0, -2/16, 0;...
                                    0, 2/16, 0;...
                                    0, 0, 0 ...
                                    ], ...
                                   [0, 0, -1/16;...
                                    0, 1/16, 0;...
                                    0, 0, 0 ...
                                    ], ...
                                   [0, 0, 0;...
                                    0, 2/16, -2/16;...
                                    0, 0, 0 ...
                                    ], ...
                                   [0, 0, 0;...
                                    0, 1/16, 0;...
                                    0, 0, -1/16 ...

```

```

], ...
[0, 0, 0;...
0, 2/16, 0;...
0, -2/16, 0 ...
], ...
[0, 0, 0;...
0, 1/16, 0;...
-1/16, 0, 0 ]);

```

```

Im132_zp = padarray(Im132, [4 4]);
Im133_zp = padarray(Im133, [4 4]);
Im134_zp = padarray(Im134, [4 4]);

```

```

[Im132_zp_row_size, Im132_zp_col_size] = size(Im132_zp);
[Im133_zp_row_size, Im133_zp_col_size] = size(Im133_zp);
[Im134_zp_row_size, Im134_zp_col_size] = size(Im134_zp);

```

```

Im132_zp_edges = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

```

```

for row=1:Im132_zp_row_size-8
    for col=1:Im132_zp_col_size-8
        Im132_zp_edges(row+4, col+4) = masking(Im132_zp, row+4, col+4, 3,
BoxFilterLaplacian);
    end
end

```

```

Im133_zp_edges = uint8(zeros(Im133_zp_row_size, Im133_zp_col_size));

```

```

for row=1:Im133_zp_row_size-8
    for col=1:Im133_zp_col_size-8
        Im133_zp_edges(row+4, col+4) = masking(Im133_zp, row+4, col+4, 3,
BoxFilterLaplacian);
    end
end

```

```

Im134_zp_edges = uint8(zeros(Im134_zp_row_size, Im134_zp_col_size));

```

```

for row=1:Im134_zp_row_size-8
    for col=1:Im134_zp_col_size-8
        Im134_zp_edges(row+4, col+4) = masking(Im134_zp, row+4, col+4, 3,
BoxFilterLaplacian);
    end
end

```

```

Im132_zp_sharp = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

```

```

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_sharp(row, col) = Im132_zp(row, col) + Im132_zp_edges(row, col)*10;
    end
end

```

```

Im133_zp_sharp = uint8(zeros(Im133_zp_row_size, Im133_zp_col_size));

```

```

for row=1:Im133_zp_row_size
    for col=1:Im133_zp_col_size
        Im133_zp_sharp(row, col) = Im133_zp(row, col) + Im133_zp_edges(row, col)*10;
    end
end

```

```

Im134_zp_sharp = uint8(zeros(Im134_zp_row_size, Im134_zp_col_size));

```

```

for row=1:Im134_zp_row_size

```

```

    for col=1:Im134_zp_col_size
        Im134_zp_sharp(row, col) = Im134_zp(row, col) + Im134_zp_edges(row, col)*10;
    end
end

threshold = 1;

Im132_zp_edges_Adaptive = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size-8
    for col=1:Im132_zp_col_size-8

        sum = 0;
        for mask_number = 1:8
            mask = BoxFilterLaplacianAdaptive(:, :, mask_number);
            temp = masking(Im132_zp, row+4, col+4, 3, mask);

            if temp >= threshold
                sum = sum + temp;
            end
        end
        Im132_zp_edges_Adaptive(row+4, col+4) = sum;
    end
end

Im132_zp_sharp_Adaptive = uint8(zeros(Im132_zp_row_size, Im132_zp_col_size));

for row=1:Im132_zp_row_size
    for col=1:Im132_zp_col_size
        Im132_zp_sharp_Adaptive(row, col) = Im132_zp(row, col) +
        Im132_zp_edges_Adaptive(row, col)*10;
    end
end

Im133_zp_edges_Adaptive = uint8(zeros(Im133_zp_row_size, Im133_zp_col_size));

for row=1:Im133_zp_row_size-8
    for col=1:Im133_zp_col_size-8

        sum = 0;
        for mask_number = 1:8
            mask = BoxFilterLaplacianAdaptive(:, :, mask_number);
            temp = masking(Im133_zp, row+4, col+4, 3, mask);

            if temp >= threshold
                sum = sum + temp;
            end
        end
        Im133_zp_edges_Adaptive(row+4, col+4) = sum;
    end
end

Im133_zp_sharp_Adaptive = uint8(zeros(Im133_zp_row_size, Im133_zp_col_size));

for row=1:Im133_zp_row_size
    for col=1:Im133_zp_col_size
        Im133_zp_sharp_Adaptive(row, col) = Im133_zp(row, col) +
        Im133_zp_edges_Adaptive(row, col)*10;
    end
end

Im134_zp_edges_Adaptive = uint8(zeros(Im134_zp_row_size, Im134_zp_col_size));

```

```

for row=1:Im134_zp_row_size-8
    for col=1:Im134_zp_col_size-8

        sum = 0;
        for mask_number = 1:8
            mask = BoxFilterLaplacianAdaptive(:, :, mask_number);
            temp = masking(Im134_zp, row+4, col+4, 3, mask);

            if temp >= threshold
                sum = sum + temp;
            end
        end
        Im134_zp_edges_Adaptive(row+4, col+4) = sum;
    end
end

Im134_zp_sharp_Adaptive = uint8(zeros(Im134_zp_row_size, Im134_zp_col_size));

for row=1:Im134_zp_row_size
    for col=1:Im134_zp_col_size
        Im134_zp_sharp_Adaptive(row, col) = Im134_zp(row, col) +
        Im134_zp_edges_Adaptive(row, col)*10;
    end
end

figure
subplot(2,3,1), imshow(Im132_zp); title('a');
subplot(2,3,2), imshow(Im132_zp_edges*40); title('b');
subplot(2,3,3), imshow(Im132_zp_sharp); title('c');
subplot(2,3,5), imshow(Im132_zp_edges_Adaptive*40); title('d');
subplot(2,3,6), imshow(Im132_zp_sharp_Adaptive); title('e');

figure
subplot(2,3,1), imshow(Im133_zp); title('a');
subplot(2,3,2), imshow(Im133_zp_edges*40); title('b');
subplot(2,3,3), imshow(Im133_zp_sharp); title('c');
subplot(2,3,5), imshow(Im133_zp_edges_Adaptive*40); title('d');
subplot(2,3,6), imshow(Im133_zp_sharp_Adaptive); title('e');

figure
subplot(2,3,1), imshow(Im134_zp); title('a');
subplot(2,3,2), imshow(Im134_zp_edges*40); title('b');
subplot(2,3,3), imshow(Im134_zp_sharp); title('c');
subplot(2,3,5), imshow(Im134_zp_edges_Adaptive*40); title('d');
subplot(2,3,6), imshow(Im134_zp_sharp_Adaptive); title('e');

```