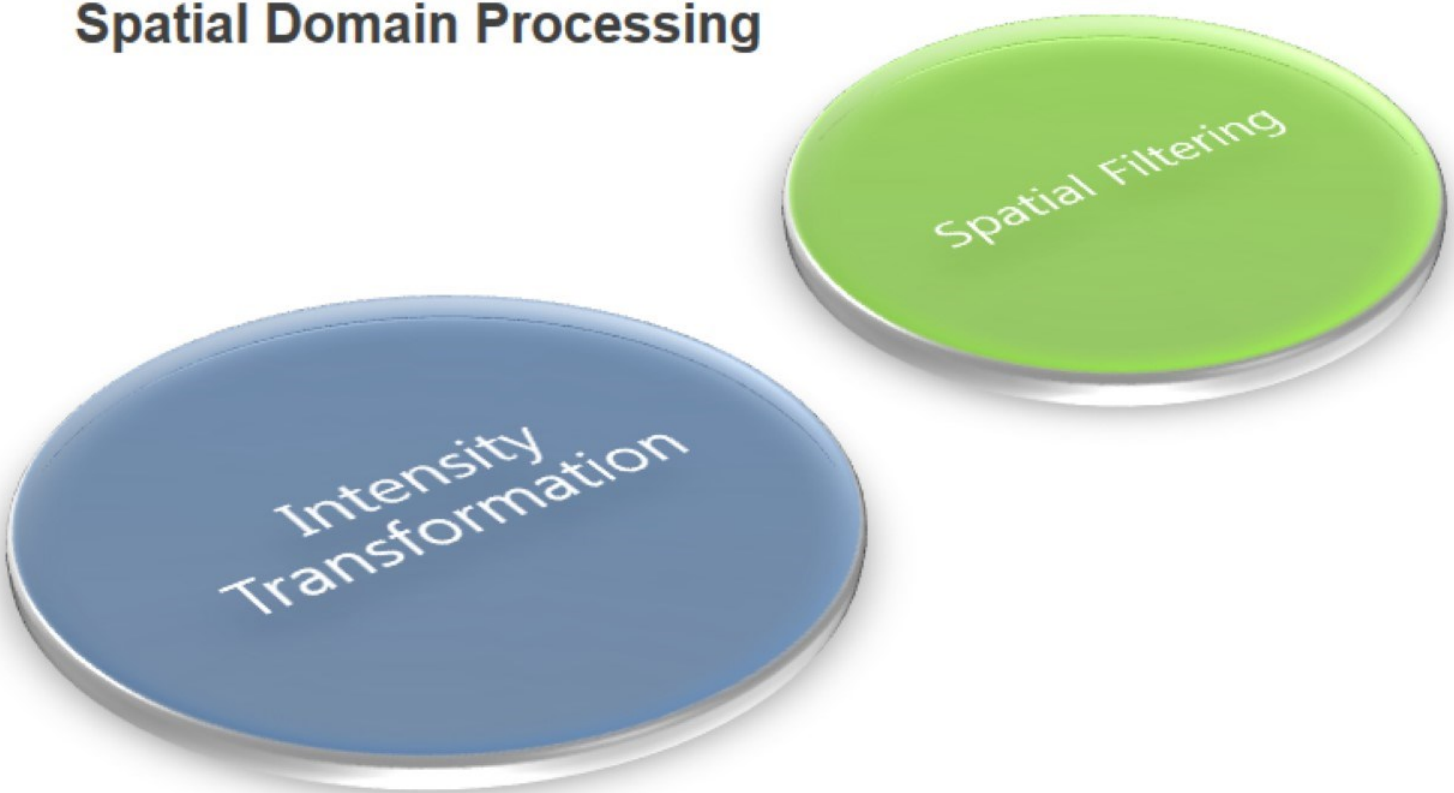
Image Processing

Intensity Transformation and Spatial Filtering (Part II)

Pattern Recognition and Image Processing Laboratory (Since 2012)

Introduction

Spatial Domain Processing



Spatial Filtering

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,-1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y-1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

Spatial Filtering

Spatial Filtering

Linear Spatial Filtering

-1	-1	-1
-1	8	-1
-1	-1	-1

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y-1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

Coefficient mask

Spatial Filtering

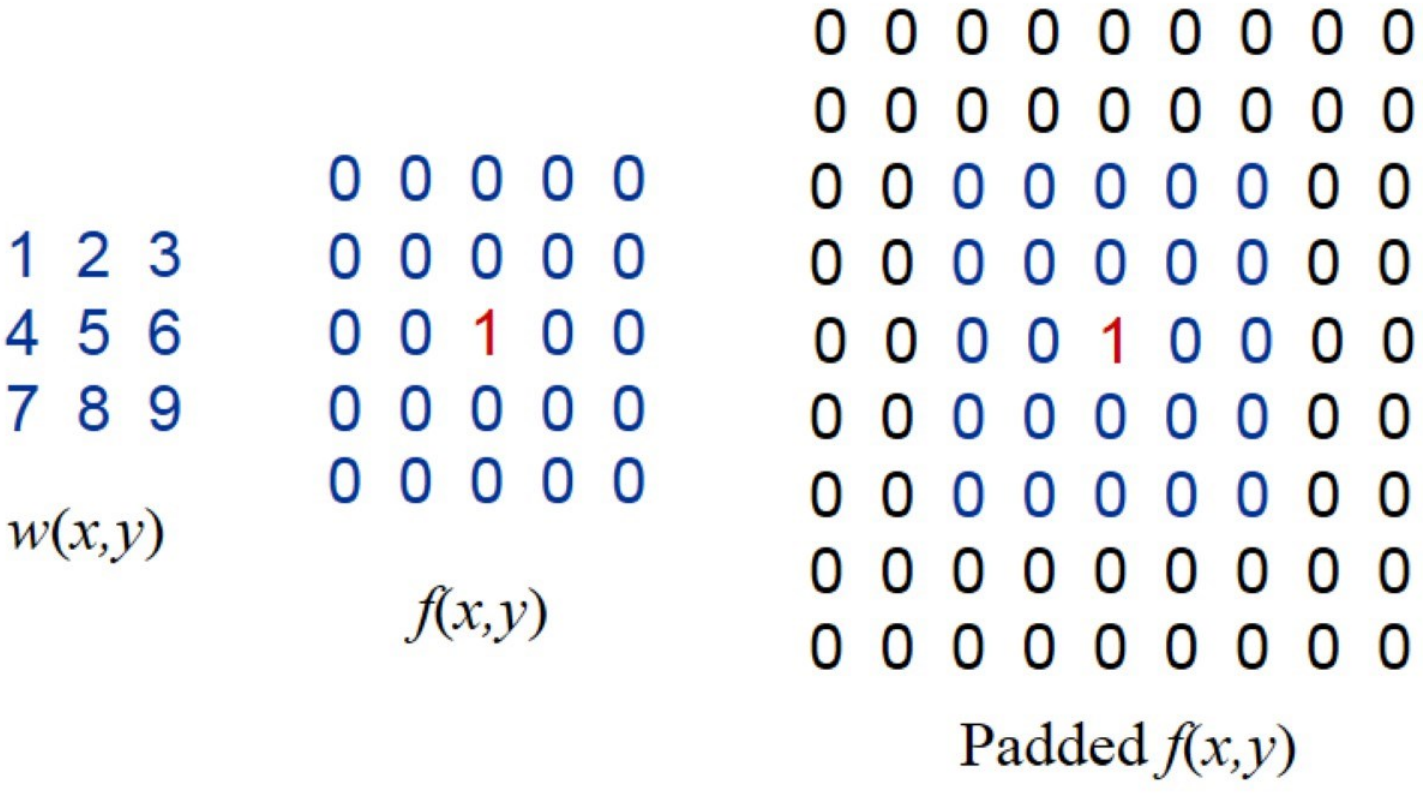
Linear Spatial Filtering

A 2-D linear spatial filter usually has the following properties:

- The mask size is symmetric, such as 3x3, 5x5, 7x7, ...
- The operation of a filter is based on convolution and correlation.

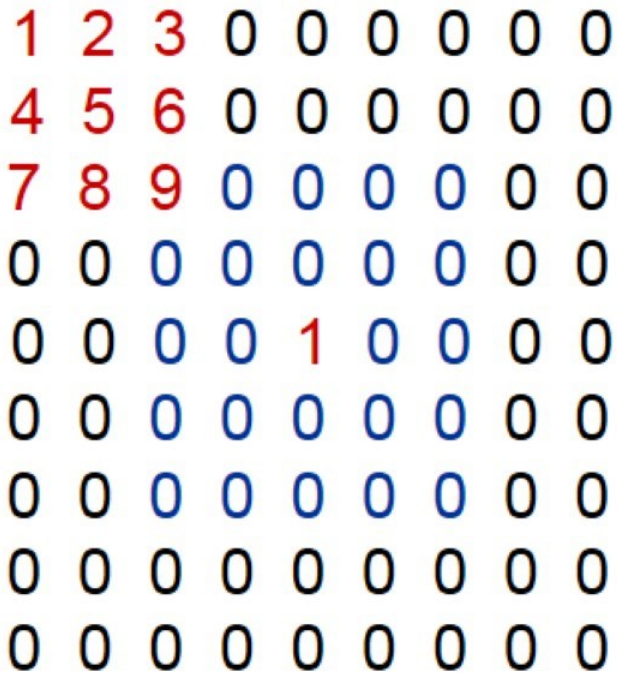
Spatial Filtering

Linear Spatial Filtering: Correlation

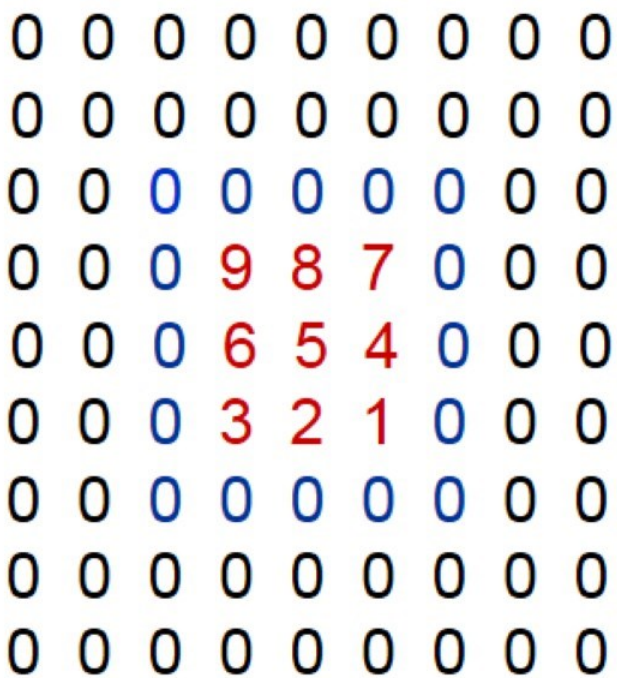


Spatial Filtering

Linear Spatial Filtering: Correlation



Initial operation



Correlation result

Spatial Filtering

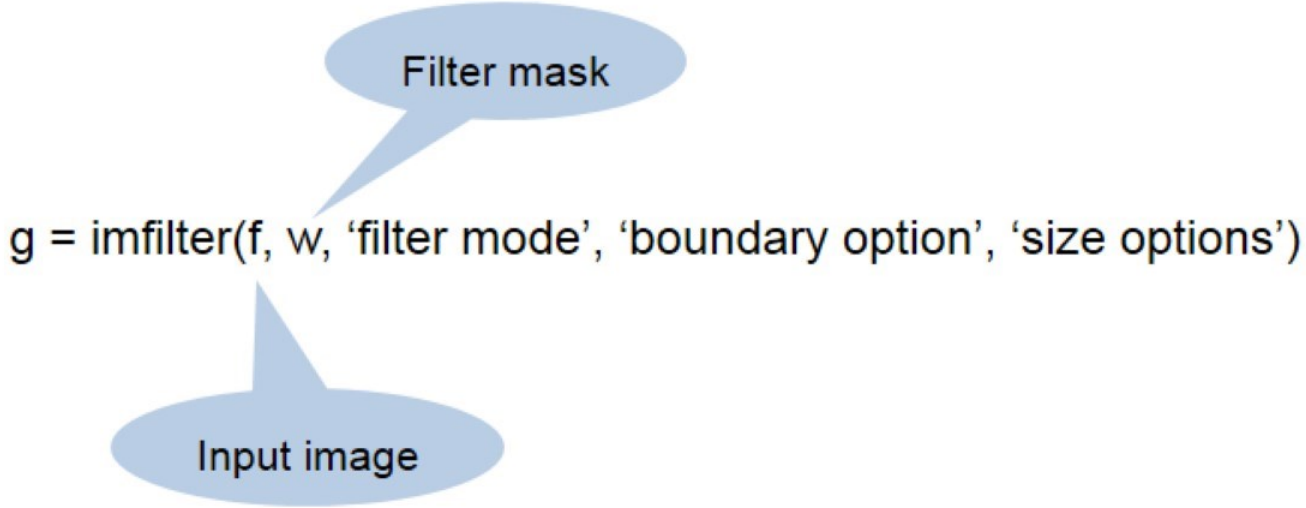
Linear Spatial Filtering: Convolution

Initial operation

Convolution result

Spatial Filtering

The following syntax is used when implementing IPT standard linear spatial filters.



Spatial Filtering

```
>> ex3_04  % See demonstration
```

Spatial Filtering

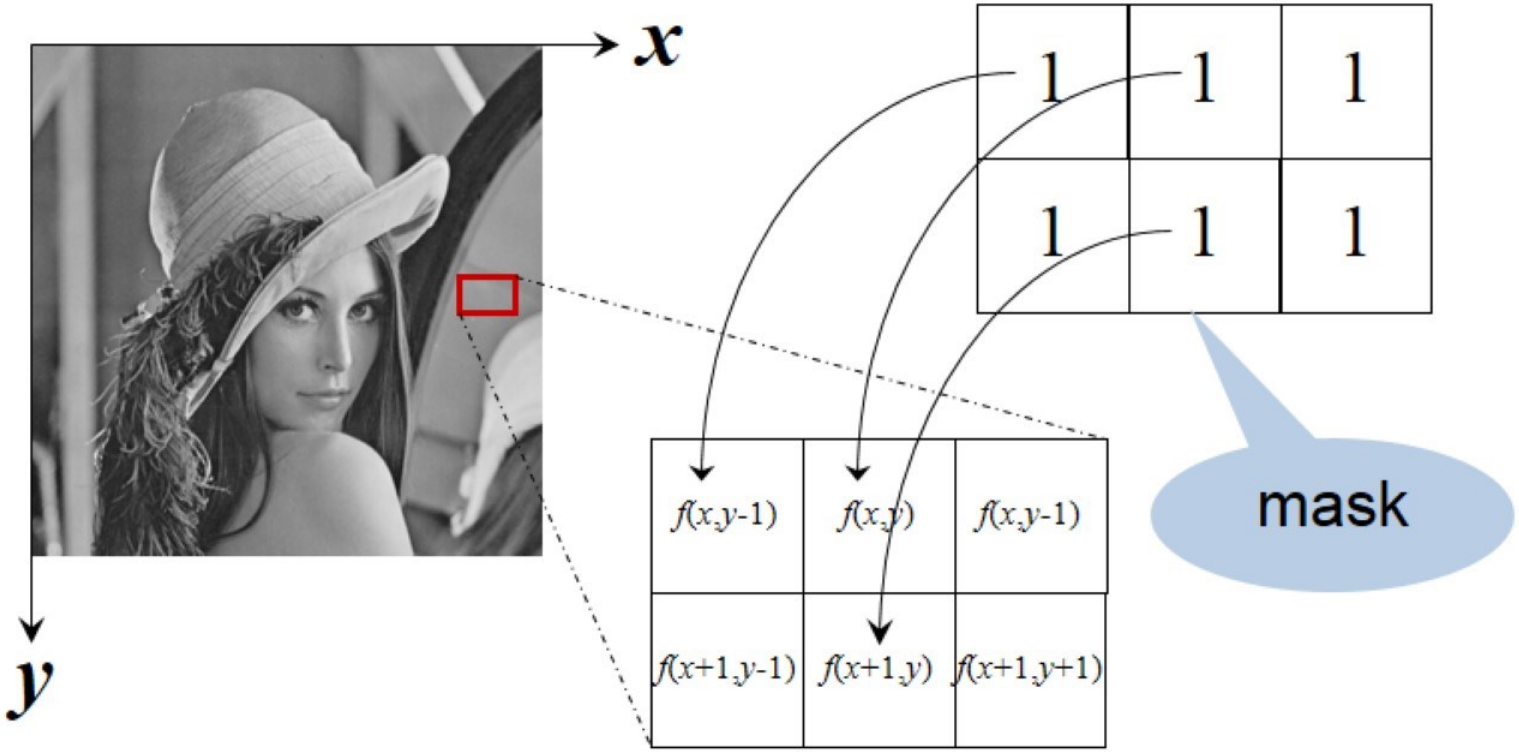
Non-linear Spatial Filtering

A 2-D non-linear spatial filter usually has the following properties:

- The mask size can be both symmetric and asymmetric forms, such as 2x2, 2x3, 3x3, 3x4, 5x7, ...
- The operation is directly performed on image pixels.

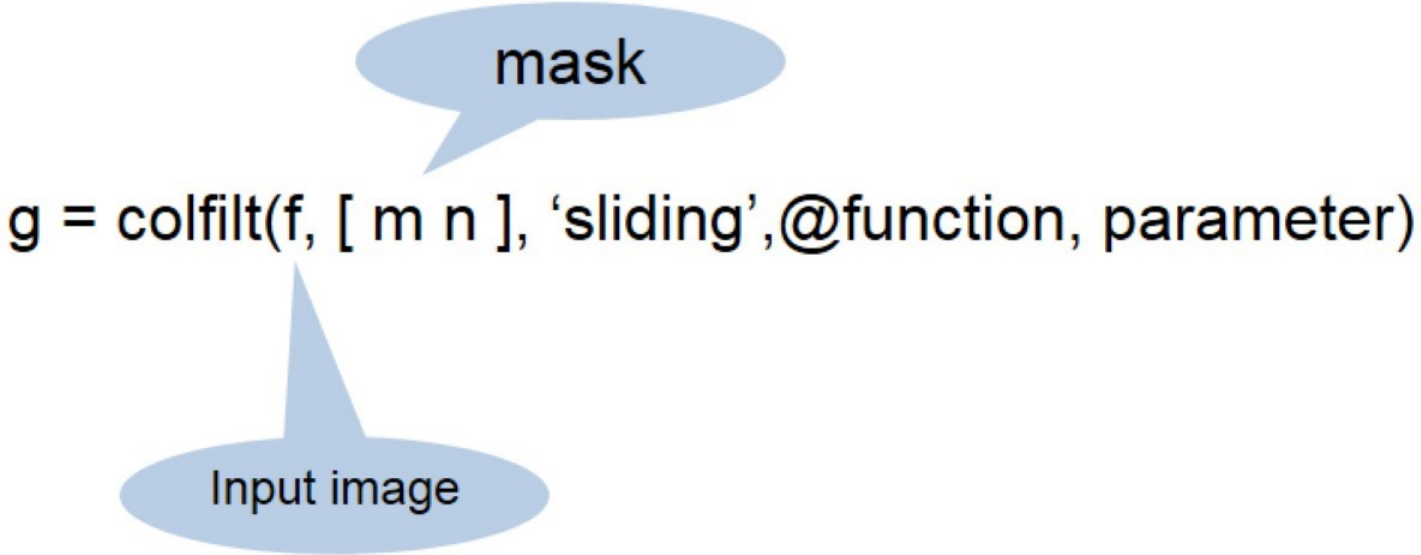
Spatial Filtering

Non-linear Spatial Filtering



Spatial Filtering

The following syntax is used for implementing generalized non-linear spatial filters.



Applications of Non-linear Spatial Filtering: Image Enhancement

```
>> ex3_04  % See demonstration
```

Applications of Non-linear Spatial Filtering: Noise Filtering

```
>> ex3_04  % See demonstration
```


The end of
part II

File_name: Ex3_04.m

```
clear all;
close all;
%% -------------- Linear Filtering ------------------
f = imread('circbw.tif');
w = [0.1 0.1 0.1;
     0.1 0.2 0.1;
     0.1 0.1 0.1];

afterfilter = conv2(double(f), w);
figure(1);
subplot(1,2,1); imshow(f);
subplot(1,2,2); imshow(uint8(afterfilter));

%% --- MATLAB toolbox implementing linear spatial filtering --
%%
%% its syntex is g = imfilter(f, w, filtering_mode,
%% boundary_options, size_options), where f is the input
%% image, w is the filter mask, g is the filtered result.

% w = [0.1 0.1 0.1;
%      0.1 0.2 0.1;
%      0.1 0.1 0.1];

lowresult1 = imfilter(double(f), w, 'conv', 'replicate');
lowresult2 = imfilter(double(f), w, 'corr', 'replicate');

figure(2);
subplot(2,2,1); imshow(f);
subplot(2,2,2); imshow(lowresult1);
subplot(2,2,3); imshow(lowresult2);
```

```
hw = [1   1   1;
      1  -8   1;
      1   1   1];

hiresult = imfilter(double(f), hw, 'conv', 'replicate');

figure;
subplot(2,2,1); imshow(f);
subplot(2,2,2); imshow(uint8(lowresult1));
subplot(2,2,3); imshow(uint8(hiresult));

% --------- generating a black-white image ----------
white(1:50,1:50) = 1;
black(1:50,1:50) = 0;
blkwht = [black white; white black];

w5 = ones(5);
result1 = imfilter(double(blkwht), w5, 'conv');
result2 = imfilter(double(blkwht), w5, 'conv',
'replicate');
result3 = imfilter(double(blkwht), w5, 'conv',
'symmetric');
result4 = imfilter(double(blkwht), w5, 'conv',
'circular');

figure(3);
subplot(2,3,1); imshow(blkwht);
subplot(2,3,2); imshow(im2uint8(mat2gray(result1)));
subplot(2,3,3); imshow(im2uint8(mat2gray(result2)));
subplot(2,3,4); imshow(im2uint8(mat2gray(result3)));
subplot(2,3,5); imshow(im2uint8(mat2gray(result4)));
```

```
%% -- MATLAB toolbox implementing non-linear spatial
filtering ---
%%
%% its syntex is g = ordfilt2(f, order, domain), where
%% f is the input image, w is the filter mask, g is
%% the filtered result.

f = imread('circuit.tif');
nonsf1 = ordfilt2(f, 1, ones(3));
nonsf2 = ordfilt2(f, 5, ones(3));
nonsf3 = ordfilt2(f, 9, ones(3));

figure(4);
subplot(2,2,1); imshow(f);
subplot(2,2,2); imshow(nonsf1);
subplot(2,2,3); imshow(nonsf2);
subplot(2,2,4); imshow(nonsf3);

% ---- Add noise ----------------------
fn = imnoise(f, 'salt & pepper', 0.2);
mf3 = ordfilt2(fn, 5, ones(3));
mf5 = ordfilt2(fn, 13, ones(5));

mf6 = medfilt2(fn, [5 5], 'symmetric');

figure(5);
subplot(2,2,1); imshow(f);
subplot(2,2,2); imshow(fn);
subplot(2,2,3); imshow(mf3);
subplot(2,2,4); imshow(mf5);

figure(6); imshow(mf6);
```

```
%% the toolbax supports a number of predefined 2-D
linear spatial filters

f = imread('lena.bmp');

w1 = fspecial('laplacian', 0);
w2 = fspecial('sobel');
w3 = fspecial('prewitt');

sf1 = imfilter(f, w1, 'conv');
sf2 = imfilter(f, w2, 'conv');
sf3 = imfilter(f, w3, 'conv');

figure(7);
subplot(2,2,1); imshow(f);
subplot(2,2,2); imshow(sf1);
subplot(2,2,3); imshow(sf2);
subplot(2,2,4); imshow(sf3);
```

Image Processing

Workshop on Intensity Transformation and Spatial Filtering (Part II)

Pattern Recognition and Image Processing Laboratory (Since 2012)

Workshop on Intensity Transformation (Part II)

1. จงคำนวณหาผลลัพธ์ของการ convolution "ด้วยมือ" เมื่อกำหนดให้ mask คือ w(x,y) และรูปภาพ คือ f(x,y)

-1	-1	-1
-1	8	-1
-1	-1	-1

w(x,y)

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

f(x,y)

Workshop on Intensity Transformation (Part II)

2. จงเขียน MATLAB Script เพื่อคำนวณหาผลลัพธ์ของการ convolution ระหว่าง w(x,y) และ f(x,y)

-1	-1	-1
-1	8	-1
-1	-1	-1

w(x,y)

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

f(x,y)

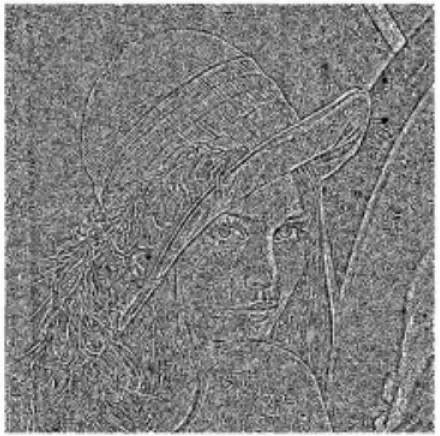
Workshop on Intensity Transformation (Part II)

3. จงเขียนฟังก์ชัน myFilter เพื่อทำการกรอง (filtering) รูปภาพ โดยมี mask และ รูปภาพ f เป็นข้อมูลนำเข้า

-1	-1	-1
-1	8	-1
-1	-1	-1

Filter mask

Original image (f)



Filtered image