Linear Filtering

Juan Park, Graduate Research Assistant
Chul Min Yeum, Assistant Professor
Civil and Environmental Engineering
University of Waterloo, Canada

CIVE 497 – CIVE 700: Smart Structure Technology

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

Last updated: 2020-01-24

Image as Functions

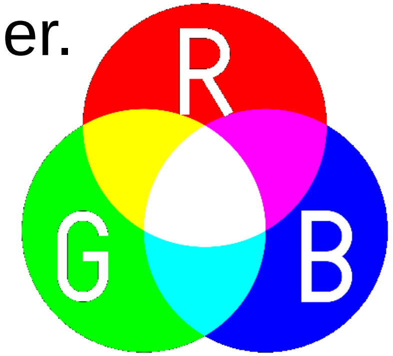
- We can think of an image as a function f, mapping $\mathbb{R}^2\ (space) \rightarrow \mathbb{R}\ (intensity)$:
  - $f(x, y)$ gives the intensity at point (x, y)
  - Realistically, images are rectangles, with a finite intensity range
  - (0-1 or 0-255). Thus:

$$f: [0, w] \times [0, h] \rightarrow [0,1],$$

$$w = width, h = height$$

- A color image is just three “grayscale” images pasted together.

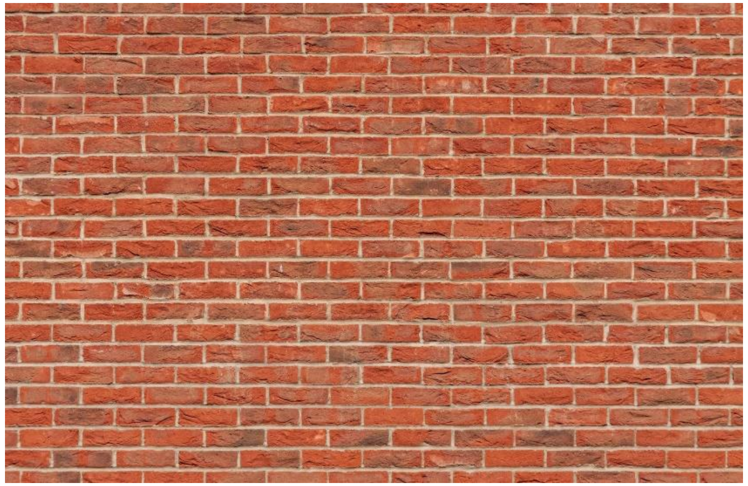
$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$, r=red, g=green, b=blue



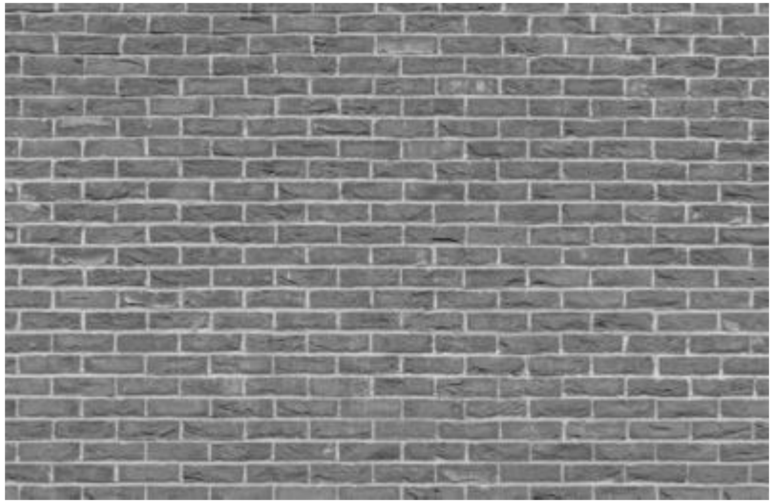
*RGB isn’t the only color space! There are others, like YCbCr, sRGB, CMY…

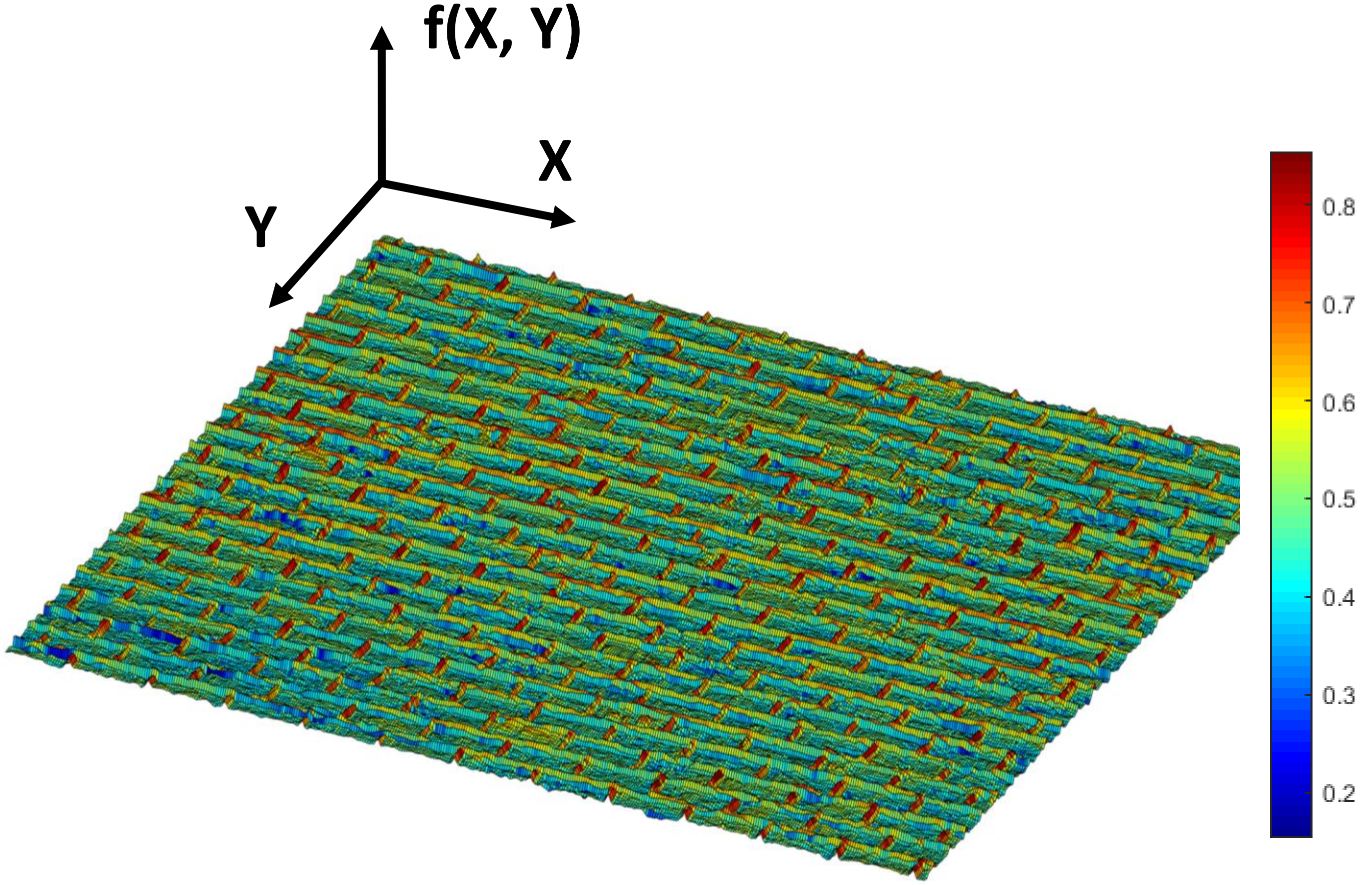
Image as Functions (Continue)

RGB



Gray





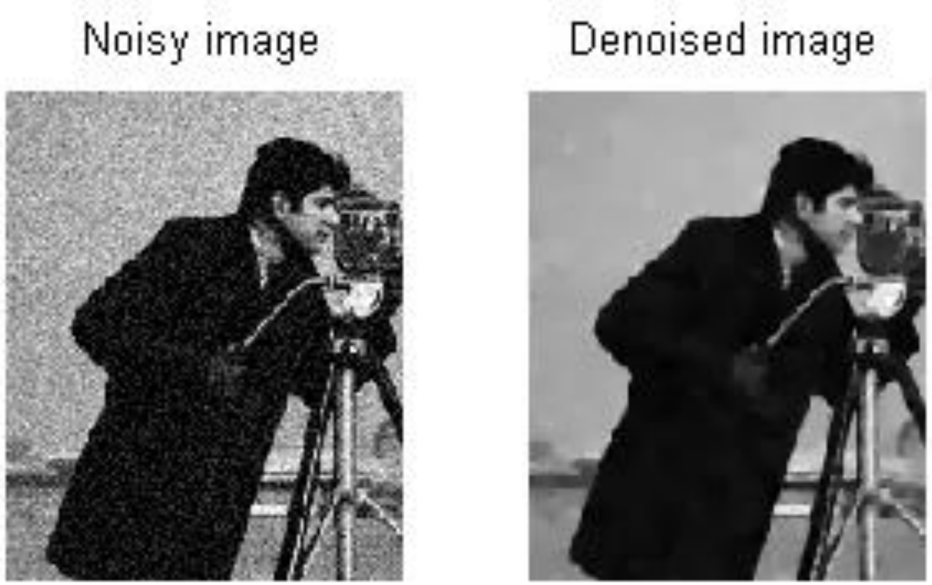
Linear Filtering?

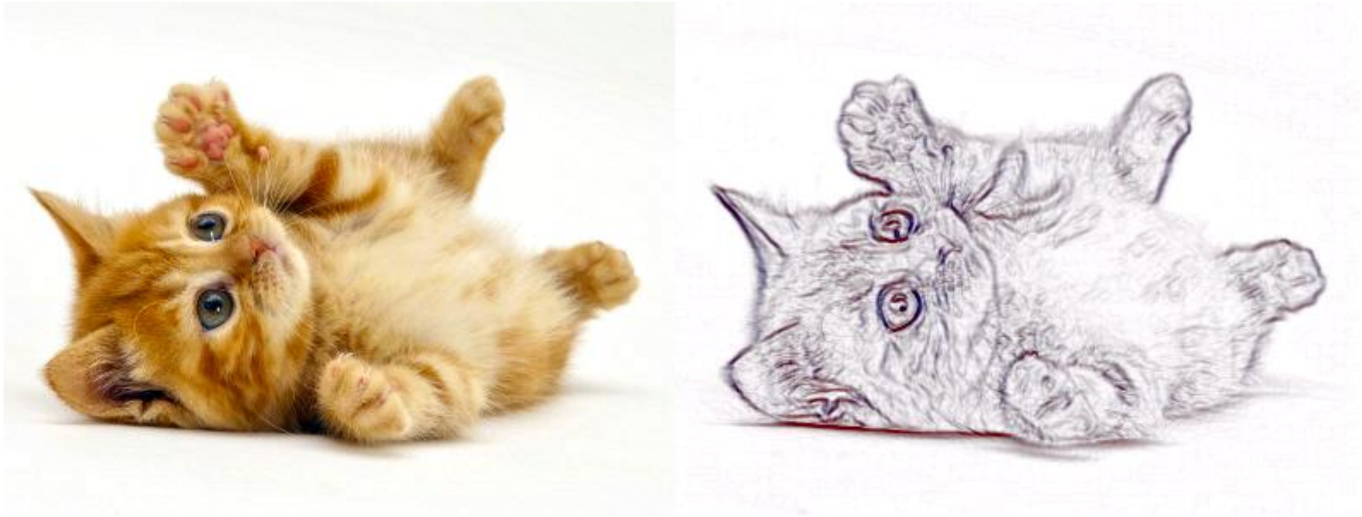
- **Filtering**: Modify or enhance data using a function (i.e. convolution kernel)
- **Linear**: The function follows linear properties of scaling and superposition.
  - Superposition: $\mathrm{h}(A+B)=h(A)+h(B)$
  - Scaling: $\mathrm{h}(\alpha A)=\alpha h(A)$
- Simply, it's linear combination: $\mathrm{h}(X,Y,Z)=aX+bY+cZ$

* h is a linear filter.

Linear Filtering?

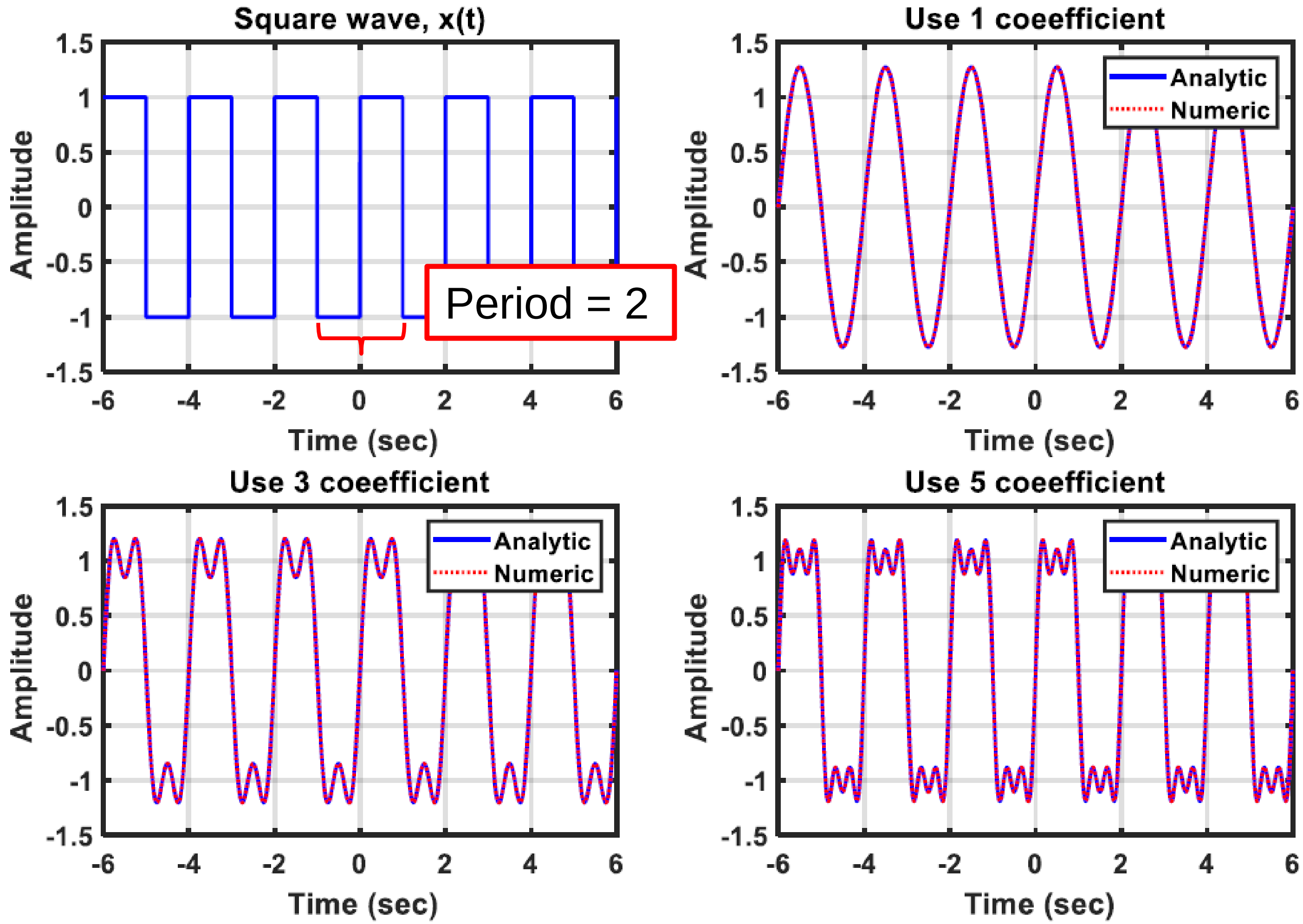
- Linear filtering is used to:
  - Reduce noise in data
  - Extract features from data





Recall: 1D Signal Interpreted using Fourier Series

- Remember the square wave function and its Fourier series approximations?



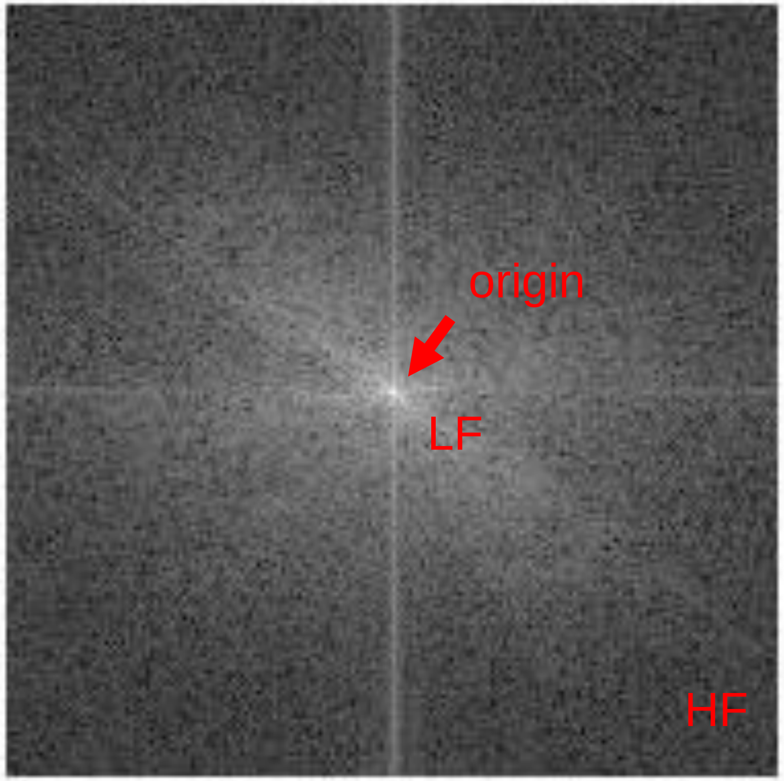
$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi n t}{T_p}\right) + b_n \sin\left(\frac{2\pi n t}{T_p}\right)$$

Back to images: Image Interpreted in a Frequency Domain

- Here is an image and its Fourier transform (remember Tp=infinity):

Image

Fourier transform



Frequency axis 2

Low frequency (LF) ≈ solid colors

High frequency (HF) ≈ noise/edges

Frequency axis 1

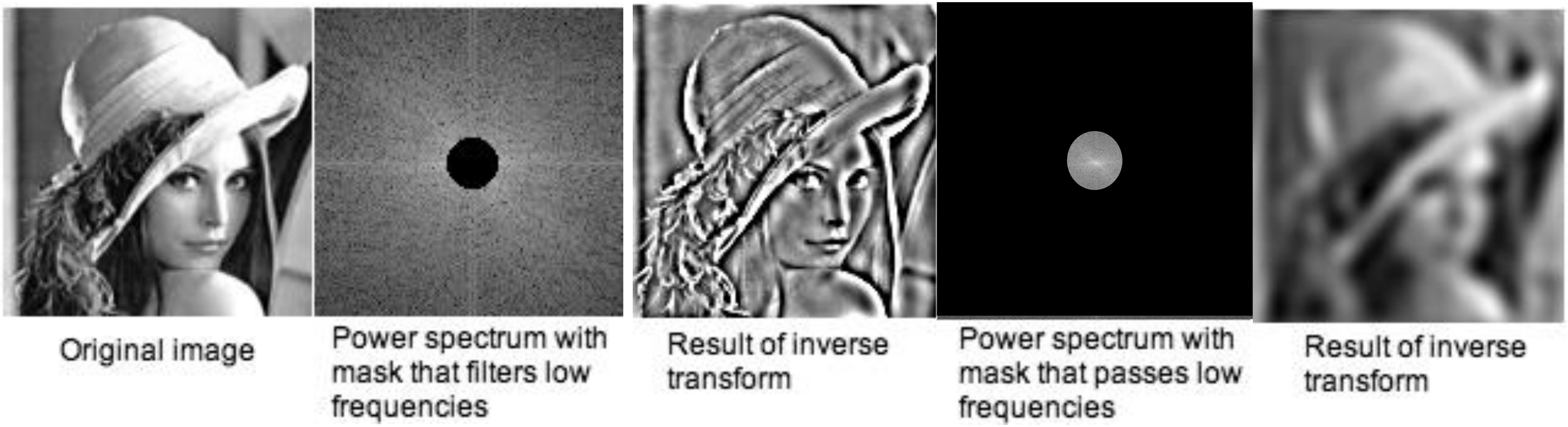
Question?

- What is blur?
- What is low resolution?

Back to images: Image Interpreted in a Frequency Domain

- If we remove low frequencies of an image, what do we get?
- How about high frequencies?

https://imagej.nih.gov/ij/docs/images/fft.jpg



Key point: Multiplication in the frequency domain is convolution in the spatial/time domain!

2D Cross-Correlation Example

Cross-correlation is a measure of similarity between two images. Cross correlation with a kernel can be viewed as comparing a litter “picture” or “region” of what you want to find across all sub-regions in the image.

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u,j+v]$$

Point i, j

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

$\otimes$

1	1	1
1	1	1
1	1	1

$=$

4	7	5	4	2
6	9	7	5	3
4	6	6	5	4
3	4	4	3	2
1	1	2	3	4

Computation of the Cross-Correlation Between Two Matrices

F

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

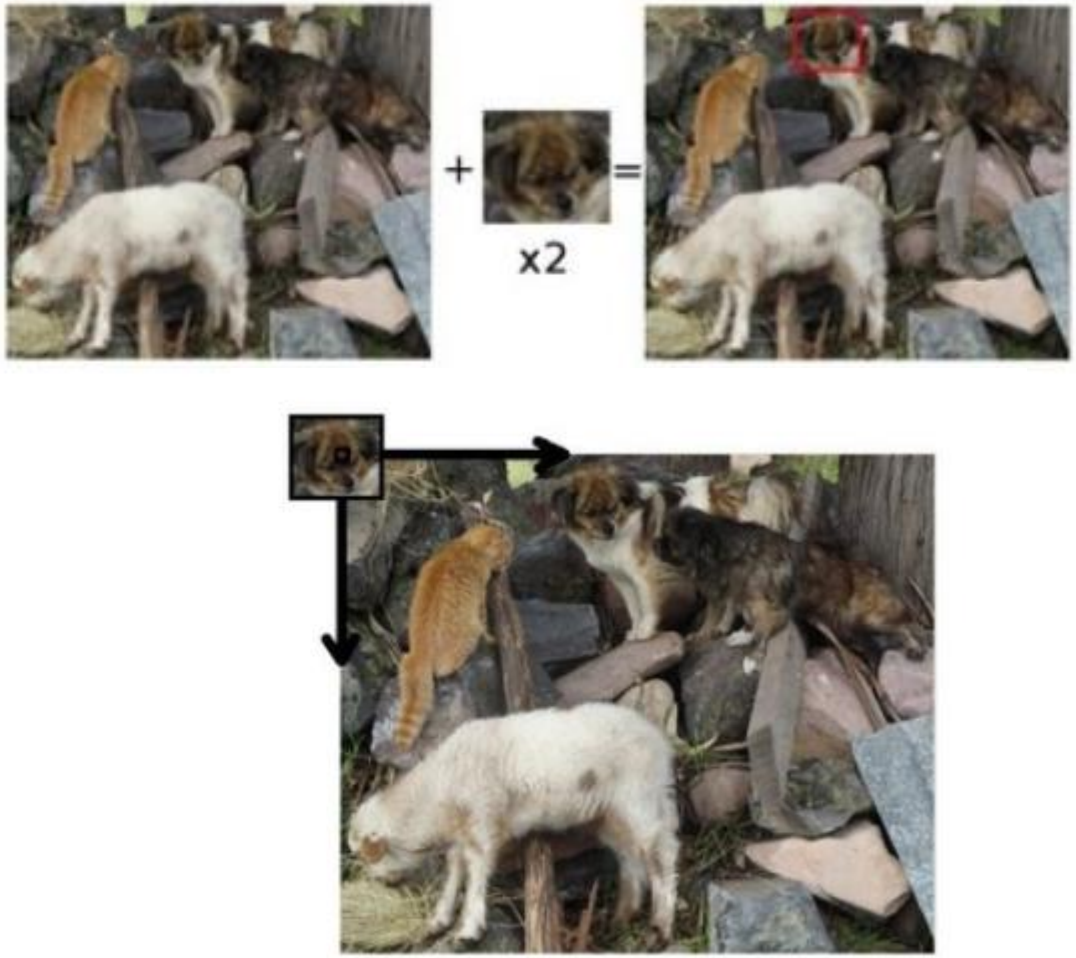
H

1	1	1
1	1	1
1	1	1

G

4	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Example: Finding Same Objects on Images



https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html

Q: Do we need CNN?

2D Convolution

Let F be the image, H be the kernel (of size 2k+1 x 2k+1),

G be the output image

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]\boxed{F[i-u,j-v]}$$

Only difference is that the kernel is "flipped" horizontally and vertically.

This is called a convolution operation:

$$G = H * F$$

Convolution Properties: Commutative, Associative, and Linearity

- Commutative: $F * H = H * F$
- Associative: $F * (H * L) = (H * F) * L$
- Linearity: $F * (H_1 + H_2) = F * H_1 + F * H_2$
- Relationship with differentiation: $(F * H)' = F' * H = F * H'$

Difference between Cross-Correlation and Convolution

X

A	B	C
D	E	F
G	H	I

Y

I	H	G
F	E	D
C	B	A

$$G = H * X = H \otimes Y$$

$\otimes$ cross-correlation

$*$ convolution

The basic difference between convolution and cross-correlation is that the convolution process flips the kernel horizontally and vertically.

Usage

Cross-Correlation: Process to measure a similarity between two signals.

Convolution: Process to transform a signal to another signal.

Guessing Game! Linear Filter: Generating an Identical Image

0	0	0
0	1	0
0	0	0

0	0	0
0	0	1
0	0	0

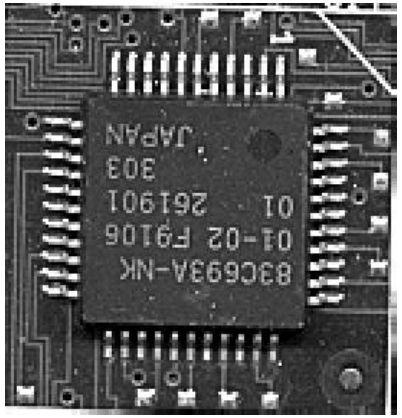
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

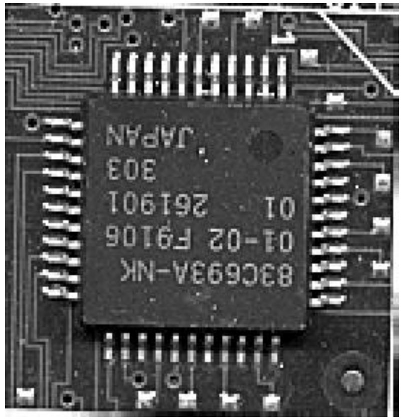
Guessing Game! Linear Filter: Generating an Identical Image

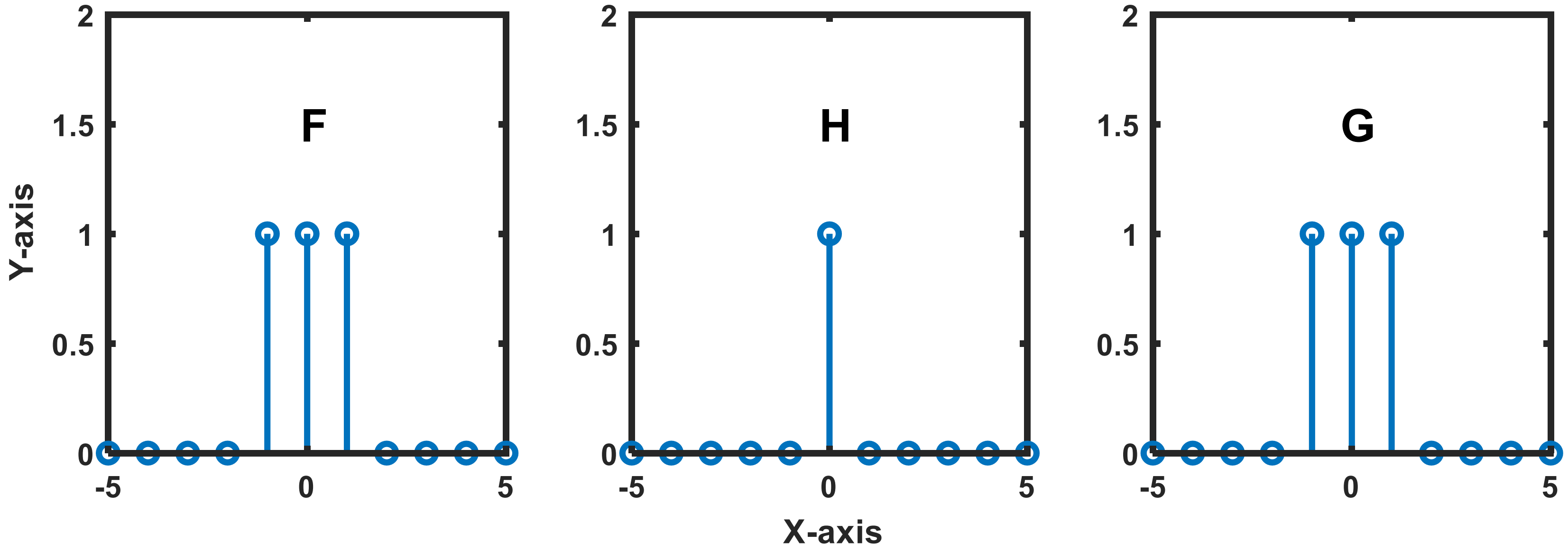


$*$

0	0	0
0	1	0
0	0	0

$=$





$$F * H = G$$

Guessing Game! Linear Filter: Shifting Pixels

0	0	0
0	1	0
0	0	0

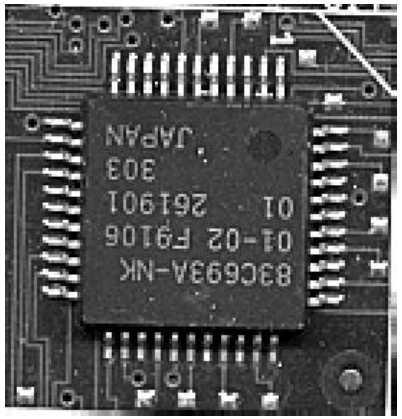
0	0	0
0	0	1
0	0	0

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

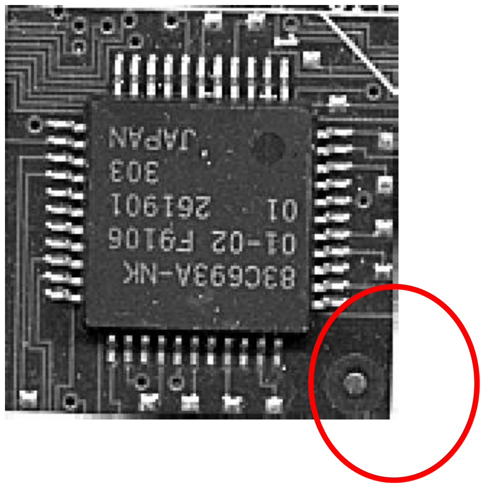
Guessing Game! Linear Filter: Shifting Pixels

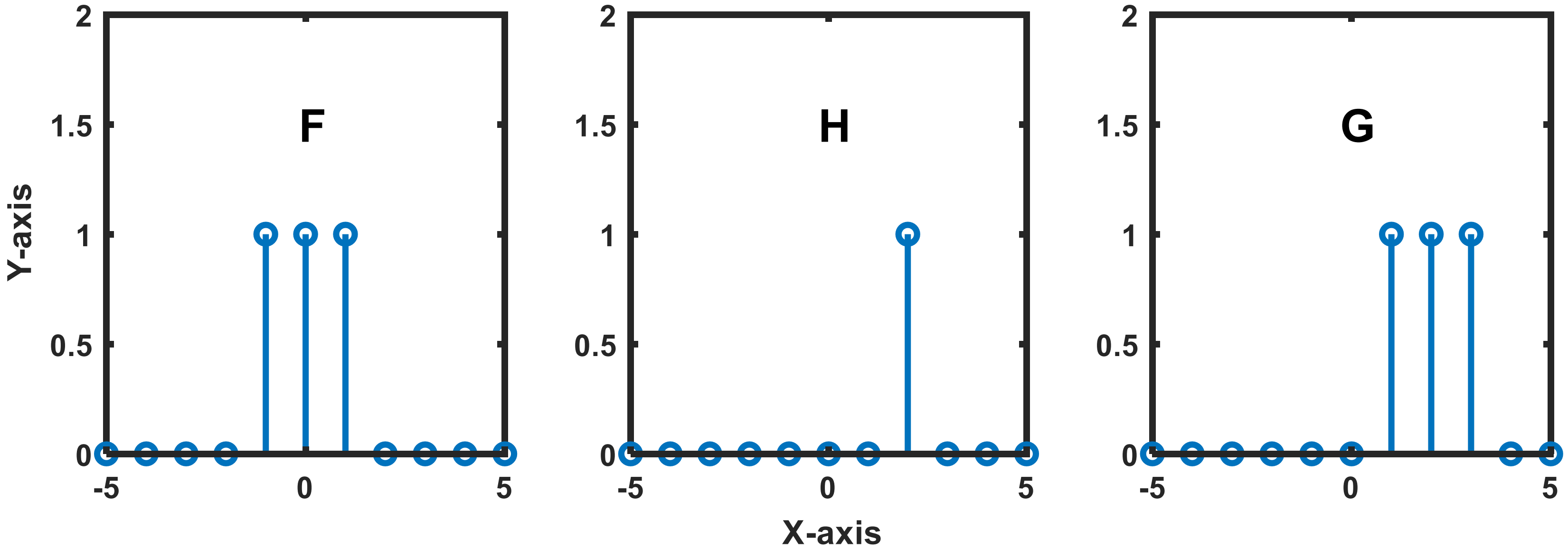


$*$

0	0	0
0	0	1
0	0	0

$=$





$$F * H = G$$

Guessing Game! Linear Filter: Blurring

0	0	0
0	1	0
0	0	0

0	0	0
0	0	1
0	0	0

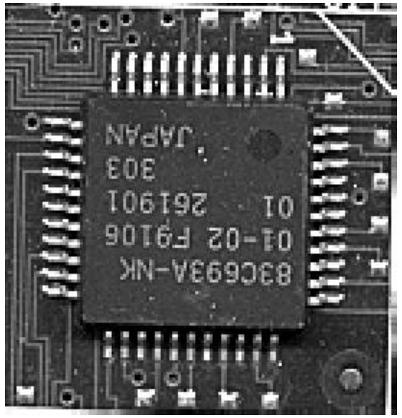
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

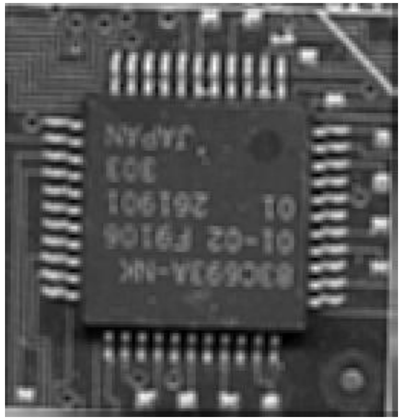
-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

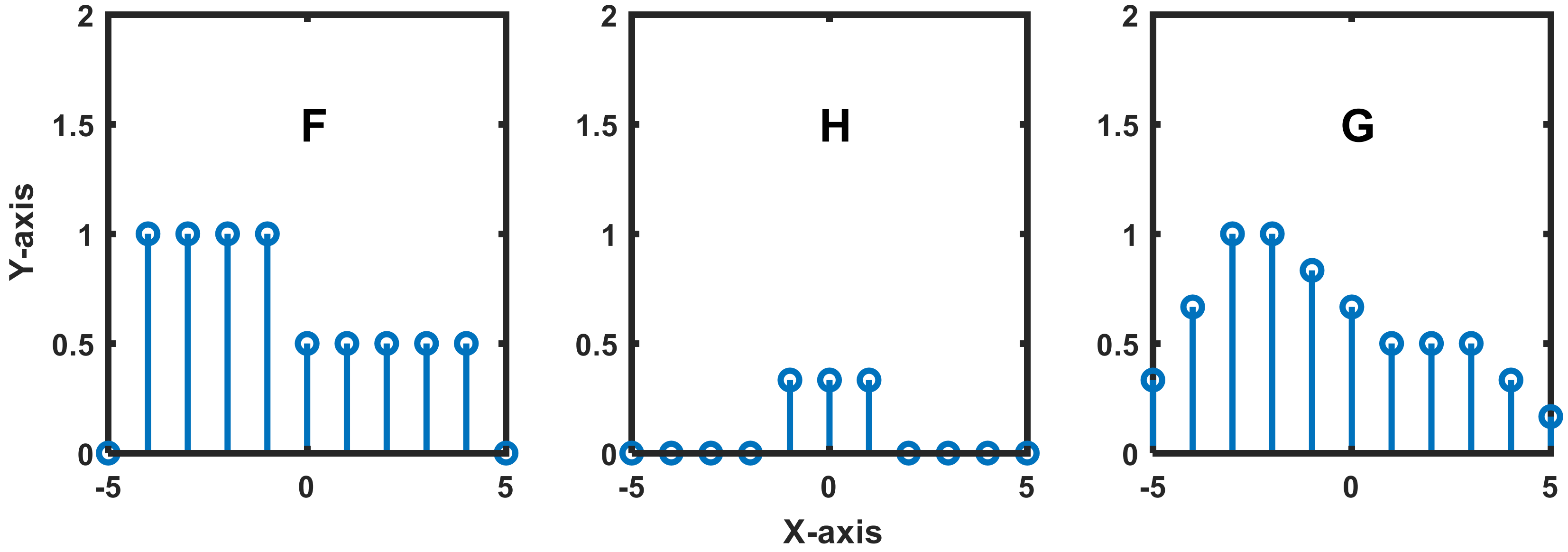
0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

Guessing Game! Linear Filter: Blurring



$$* \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$





$$F * H = G$$

Guessing Game! Linear Filter: Sharpening

0	0	0
0	1	0
0	0	0

0	0	0
0	0	1
0	0	0

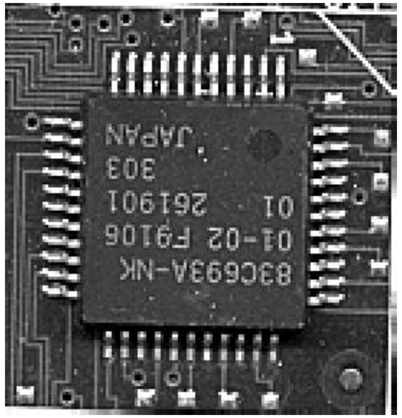
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

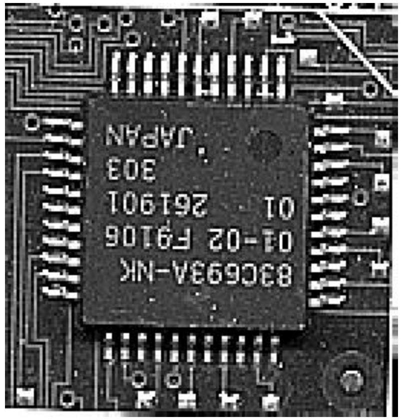
Guessing Game! Linear Filter: Sharpening

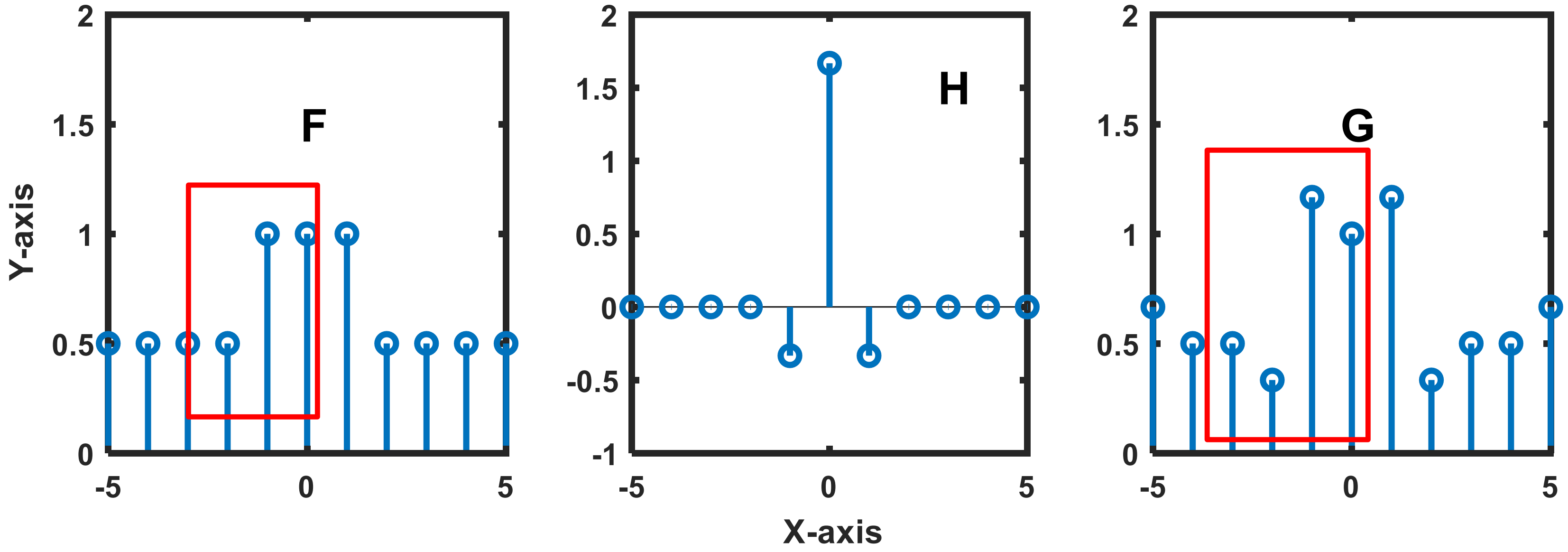


*

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

=





$$F * H = G$$

How the Sharpening Filter Works

0	0	0
0	2	0
0	0	0

$-$

0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

$=$

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

$$F + (F - F * H) = G$$

0	0	0
0	α+1	0
0	0	0

$- \alpha$

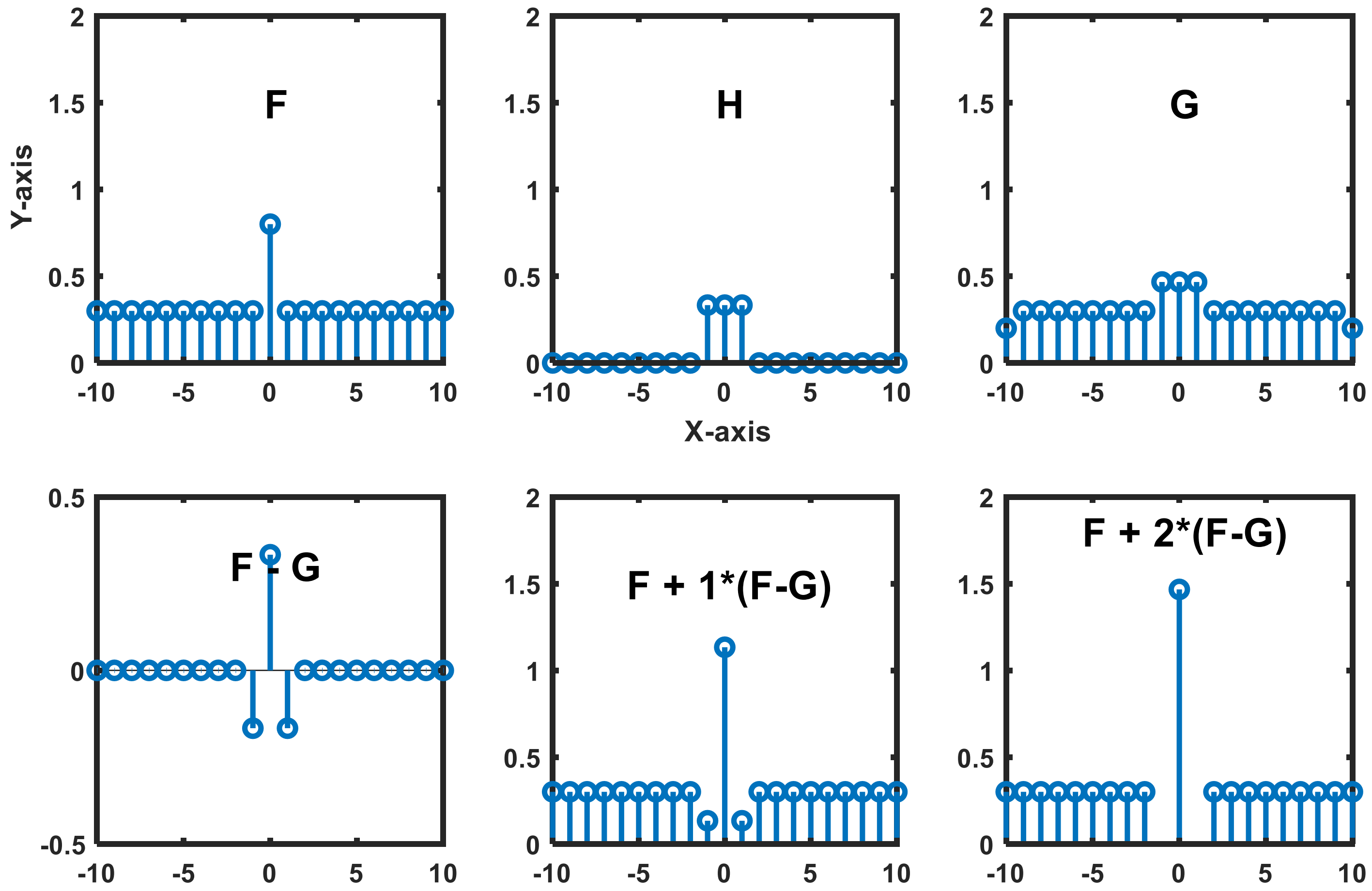
0.11	0.11	0.11
0.11	0.11	0.11
0.11	0.11	0.11

$=$

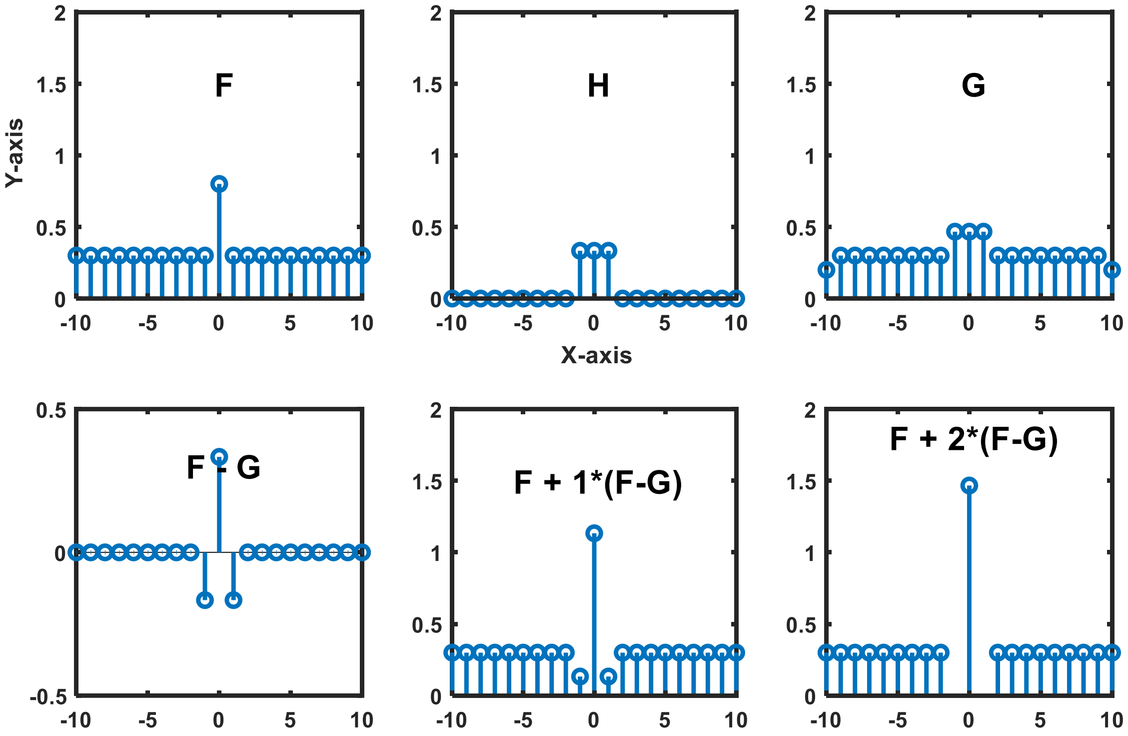
-0.11α	-0.11α	-0.11α
-0.11α	1+0.89α	-0.11α
-0.11α	-0.11α	-0.11α

$$F + \alpha(F - F * H) = G$$

How the Sharpening Filter Works (Signal Example) (Continue)

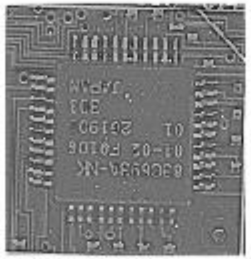


How the Sharpening Filter Works (Image Example)

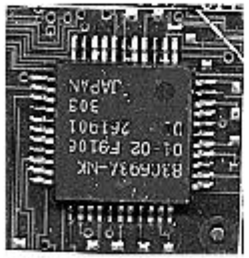


F

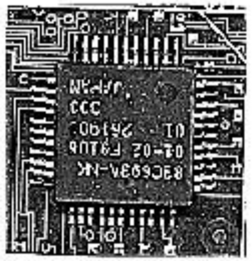
G



$F - G$



$F + (F - G)$



$F + 5(F - G)$

Guessing Game! Gaussian Kernel

0	0	0
0	1	0
0	0	0

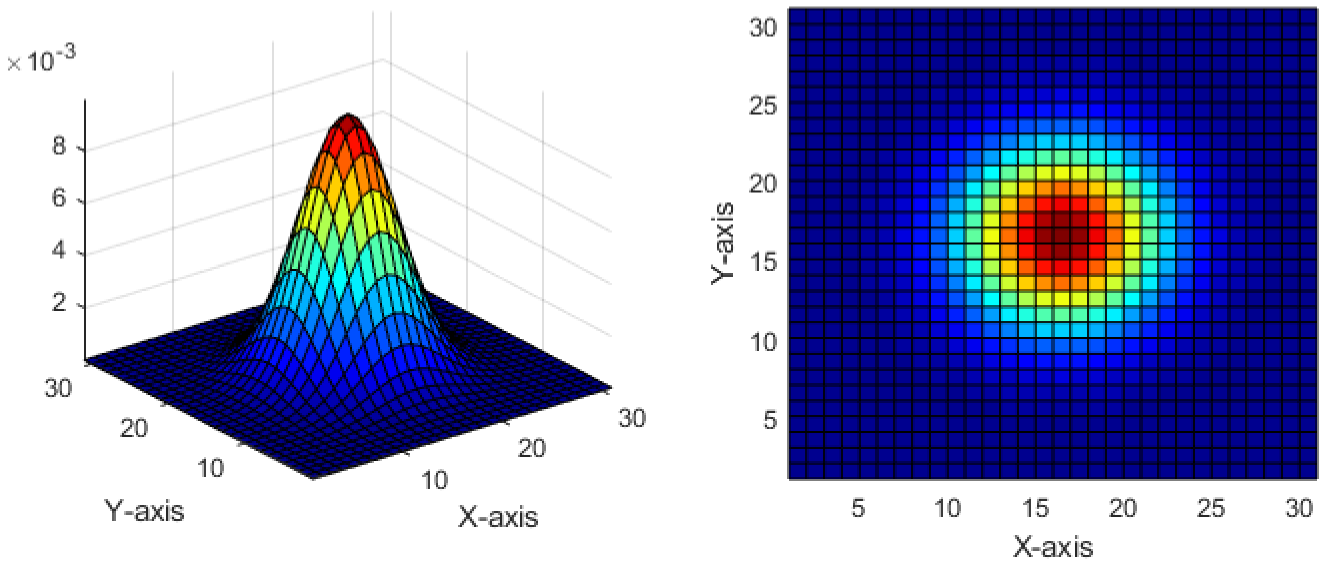
0	0	0
0	0	1
0	0	0

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

-0.11	-0.11	-0.11
-0.11	1.89	-0.11
-0.11	-0.11	-0.11

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

Gaussian Kernel



$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Gaussian Filter

```
clear; close all; clc;

h = fspecial('gaussian',3,1);

numSize = 3;
[x, y] = meshgrid(1:numSize);
x = x-(round(numSize/2));
y = y-(round(numSize/2));
sigma = 1;

G_sigma = 1/(2*pi*sigma^2)*exp(-(x.^2 + y.^2)/(2*sigma^2));
G_sigma = G_sigma/sum(G_sigma,'all');

figure(1);
subplot(121); PlotMat(h,gca,'float');
subplot(122); PlotMat(G_sigma,gca,'float');

fig2 = figure(2);
subplot(121); h = fspecial('gaussian', 31,4);
surf(h); axis tight; colormap(jet)
set(fig2,'Position', [100 100 800 300]);
xlabel('X-axis'); ylabel('Y-axis');
subplot(122); surf(h); view(0,90);
xlabel('X-axis'); ylabel('Y-axis'); axis tight
```

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

0.08	0.12	0.08
0.12	0.20	0.12
0.08	0.12	0.08

Effect of Gaussian Window Sizes

$f1$

Std=1

$f2$

Std=5

$f3$

Std=10

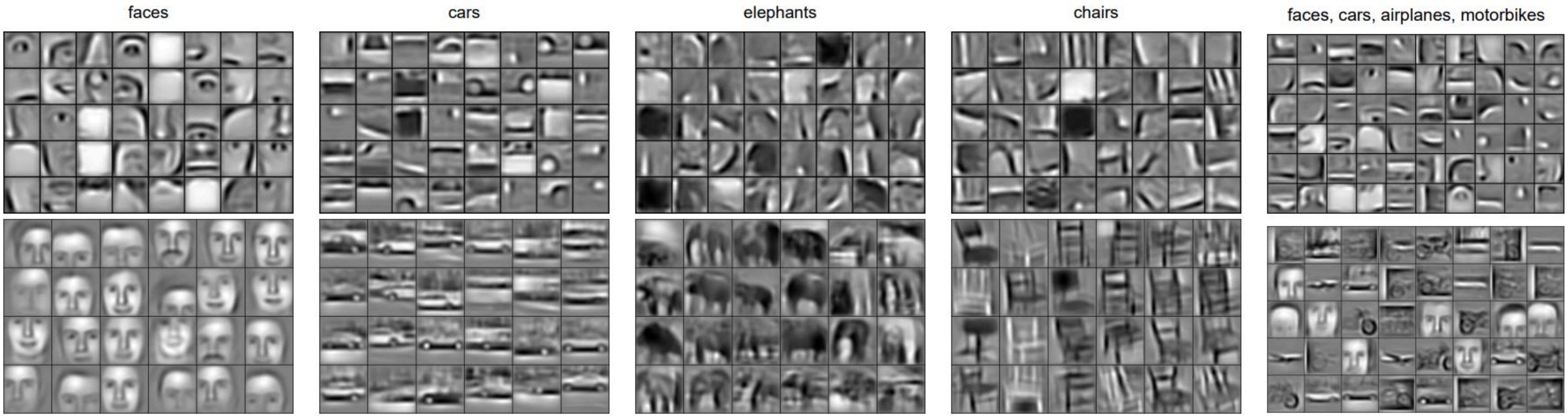
$f4$

Std=30

```
f1 =  fspecial('gaussian', 101,1);
f2 =  fspecial('gaussian', 101,5);
f3 =  fspecial('gaussian', 101,10);
f4 =  fspecial('gaussian', 101,30);
```

Challenges (template matching) !!!!!!

Applications: Convolution Neural Network



http://web.eecs.umich.edu/~honglak/icml09-ConvolutionalDeepBeliefNetworks.pdf