Data Science

Lecture 10-1: Image Data Processing

(Image Filtering)

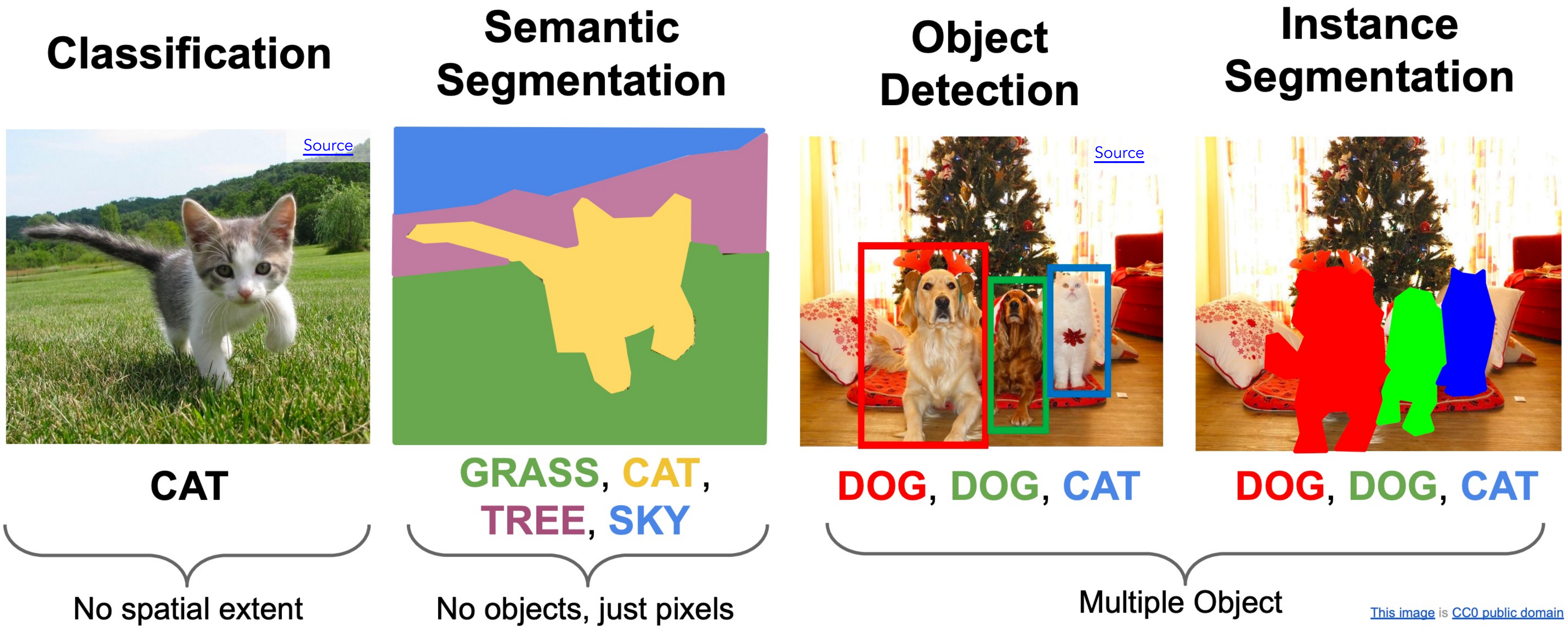
UNIVERSITY OF AMSTERDAM

Lecturer: Yen-Chia Hsu

Date: Mar 2023

This lecture covers basic concepts of image filtering and feature extraction.

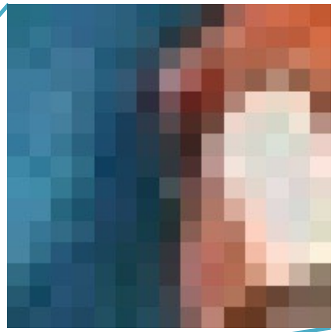
There are many different types of tasks in Computer Vision. This lecture will only cover the image/video classification, which gives only one label to an image/video.



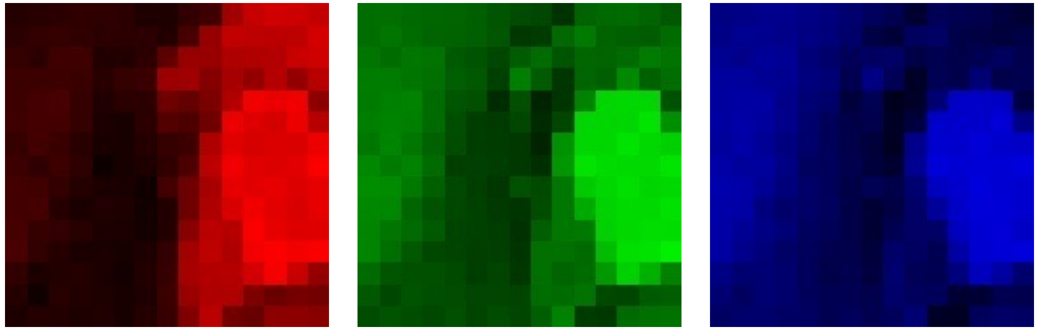
Source -- http://cs231n.stanford.edu/slides/2022/lecture_1_2_ruohan.pdf

People can see images directly, but computers read only numbers. Typically, computers store images as pixels with RGB channels with values ranging from 0 to 255.

Image patch

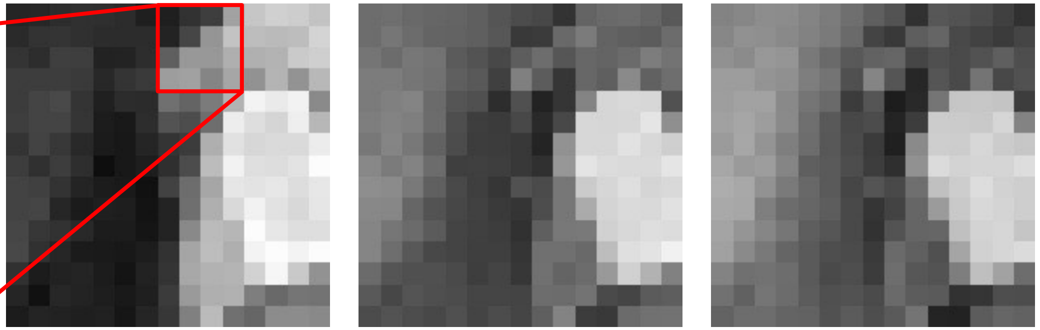


Red Green Blue



Colored visualization of the RGB channels

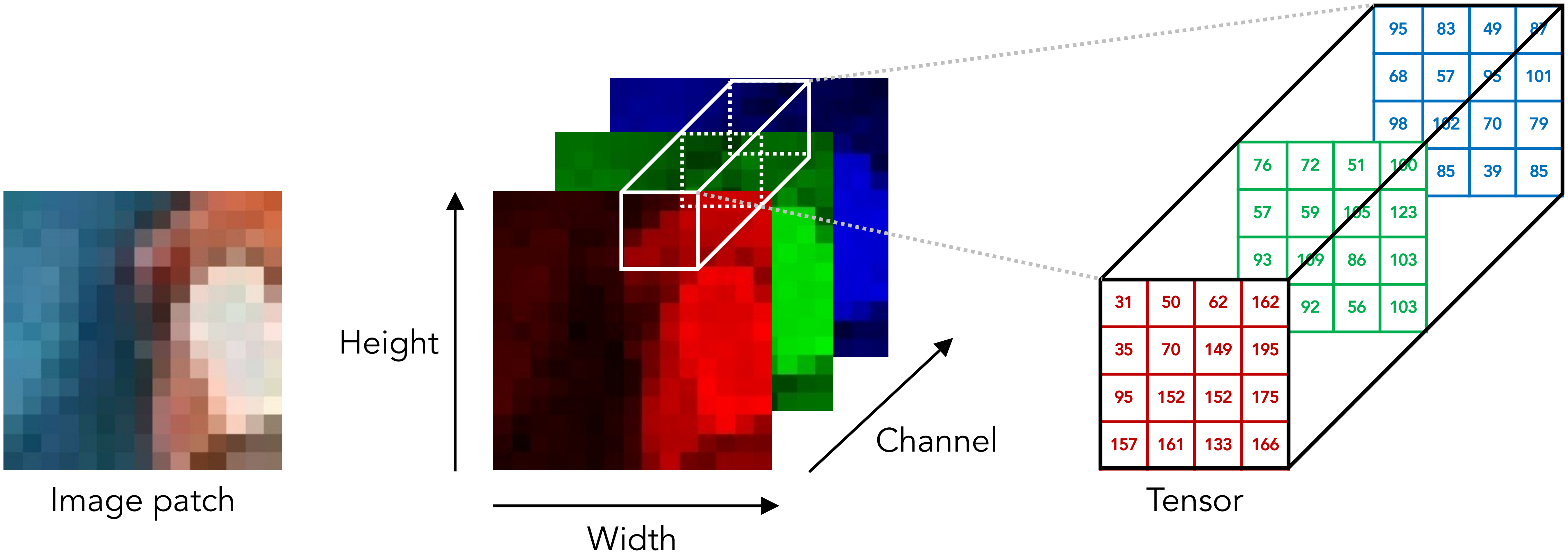
31	50	62	162
35	70	149	195
95	152	152	175
157	161	133	166



The intensity values (pixel values) per channel

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering

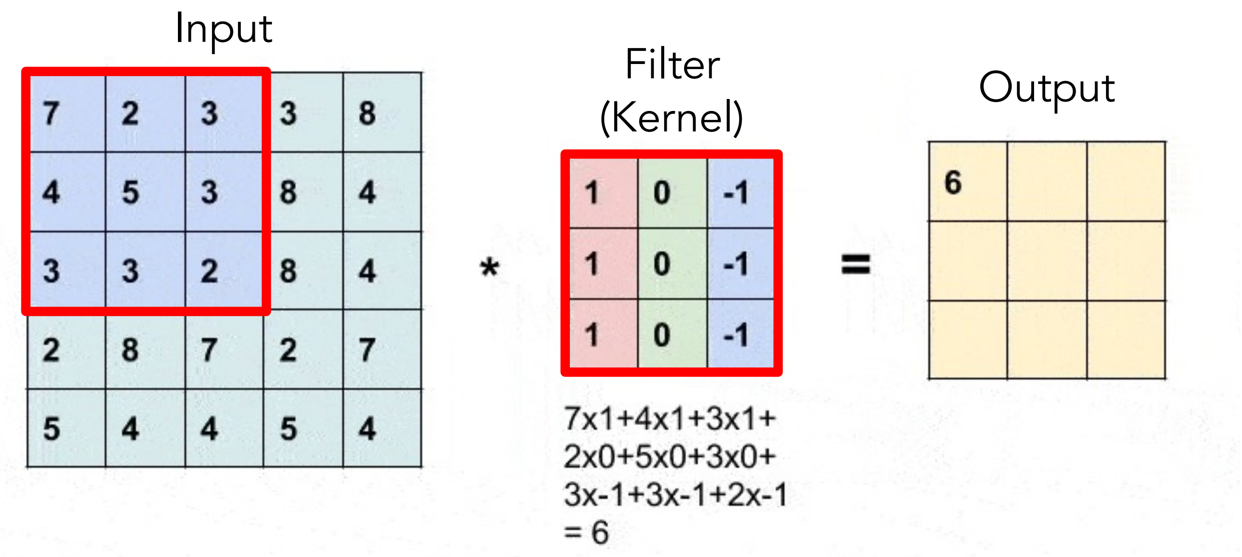
Images can be represented as a 3D tensor (width*height*channels). For RGB images, we have 3 channels corresponding to the pixel intensity of red, green, and blue colors.



Before the deep learning era, Computer Vision used hand-crafted features. Researchers developed different image filters/kernels to extract features using convolution.

Input image

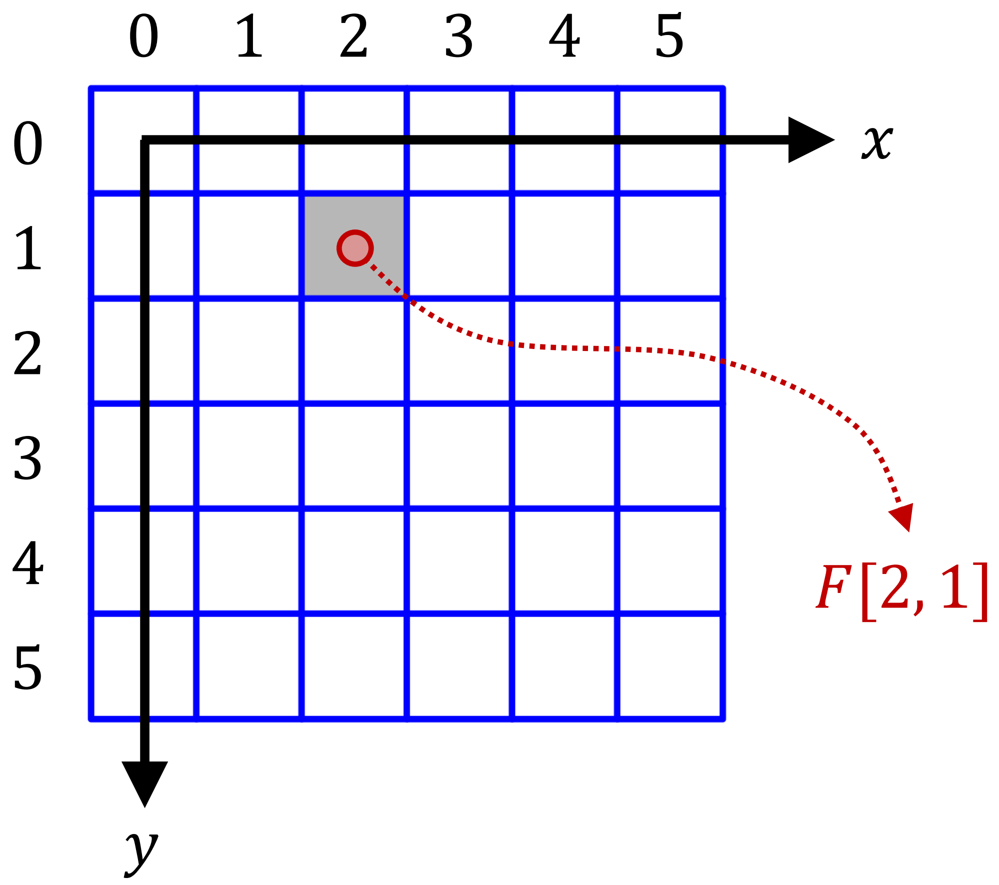
Discrete convolution: sum of the element-wise multiplication



Output image

Source -- https://docs.opencv.org/4.x/dc/ddf/tutorial_how_to_use_OpenCV_parallel_for_new.html

About image coordinates, the origin is at the top-left pixel. Notation $F[x, y]$ means the value of the center of the pixel for image F at location $[x, y]$ in the 2D array.



$$F[x, y]$$

F is an array of numbers (pixels)

x, y are integers (column/row indices)

Source -- https://courses.cs.washington.edu/courses/cse576/20sp/calendar/

Let's run the box filter

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

note that we assume that the kernel coordinates are centered

$$\underbrace{h[m,n]}_{\text{output}} = \sum_{k,l} \underbrace{g[k,l]}_{\text{filter}} \underbrace{f[m+k,n+l]}_{\text{image (signal)}}$$

Let’s run the box filter

$g[\cdot,\cdot]$

kernel

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

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

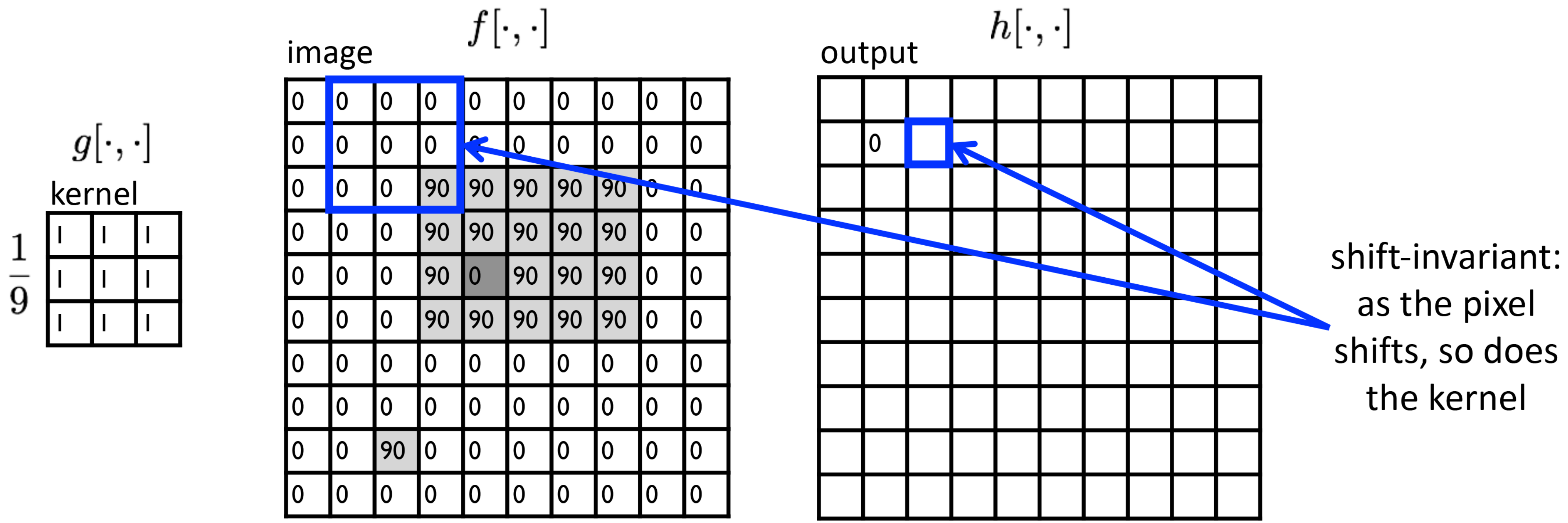
output $h[\cdot,\cdot]$

	0								

$$\underbrace{h[m,n]}_{\text{output}} = \sum_{k,l} \underbrace{g[k,l]}_{\text{filter}} \underbrace{f[m+k,n+l]}_{\text{image (signal)}}$$

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let’s run the box filter



$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output filter image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let’s run the box filter

$g[\cdot,\cdot]$

kernel

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

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10							

$$h[m,n]=\sum_{k,l}g[k,l]f[m+k,n+l]$$

output filter image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let’s run the box filter

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10							

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output filter image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let's run the box filter

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20						

$$h[m,n] = \sum_{k,l} g[k,l]\,f[m+k,n+l]$$

output — filter — image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let’s run the box filter

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20						

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output filter image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let's run the box filter

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20	30					

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output — filter — image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let's run the box filter

$g[\cdot,\cdot]$

kernel

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

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20	30	30				

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output — filter — image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let’s run the box filter

$g[\cdot,\cdot]$

kernel

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

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20	30	30	30			

$$h[m,n]=\sum_{k,l} g[k,l]\,f[m+k,n+l]$$

output — filter — image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let’s run the box filter

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20	30	30	30	20	10	
	0	20	40						

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output filter image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let’s run the box filter

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30							

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output filter image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let’s run the box filter

$g[\cdot,\cdot]$

kernel

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

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	0	10	20	30	30	30	20	10	
	10	10	10	10	0	0	0	0	
	10								

$$\underbrace{h[m,n]}_{\text{output}} = \sum_{k,l} \underbrace{g[k,l]}_{\text{filter}} \underbrace{f[m+k,n+l]}_{\text{image (signal)}}$$

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

Let's run the box filter

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	0	10	20	30	30	30	20	10	
	10	10	10	10	0	0	0	0	
	10	10	10	10	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output filter image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

… and the result is

$g[\cdot,\cdot]$

kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $f[\cdot,\cdot]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $h[\cdot,\cdot]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	0	10	20	30	30	30	20	10	
	10	10	10	10	0	0	0	0	
	10	10	10	10	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

output filter image (signal)

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering/

What will be the result of the following convolution operation?

Input image

$$*$$

0	0	0
0	1	0
0	0	0

Filter
(Kernel)

Source -- https://www.cs.cornell.edu/courses/cs5670/2022sp/lectures/lec01_filter_for_web.pdf

The following convolution operation produces an identical image.

Input image

$*$

0	0	0
0	1	0
0	0	0

Filter
(Kernel)

$=$

Identical image

Source -- https://www.cs.cornell.edu/courses/cs5670/2022sp/lectures/lec01_filter_for_web.pdf

What will be the result of the following convolution operation?

Input image

*

0	0	0
1	0	0
0	0	0

Filter
(Kernel)

Source -- https://www.cs.cornell.edu/courses/cs5670/2022sp/lectures/lec01_filter_for_web.pdf

The following convolution operation shifts the image left by one pixel.

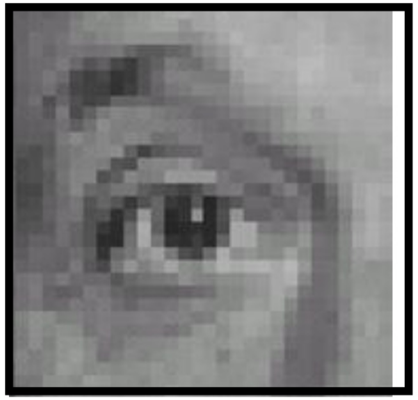
Input image

$*$

0	0	0
1	0	0
0	0	0

Filter
(Kernel)

$=$



Shifted left by 1 pixel

Source -- https://www.cs.cornell.edu/courses/cs5670/2022sp/lectures/lec01_filter_for_web.pdf

What will be the result of the following convolution operation?

Input image

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

Filter
(Kernel)

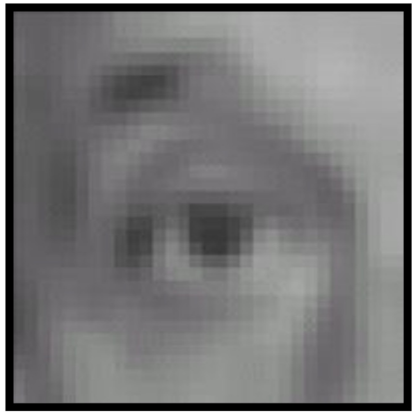
Source -- https://www.cs.cornell.edu/courses/cs5670/2022sp/lectures/lec01_filter_for_web.pdf

The following convolution operation blurs the image.

Input image

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

Filter
(Kernel)



Blurred image

Source -- https://www.cs.cornell.edu/courses/cs5670/2022sp/lectures/lec01_filter_for_web.pdf

What will be the result of the following convolution operation?

Input image

$$* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right)$$

Filter
(Kernel)

Source -- https://www.cs.cornell.edu/courses/cs5670/2022sp/lectures/lec01_filter_for_web.pdf

The following convolution operation sharpens the image.

Input image

$$* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) =$$

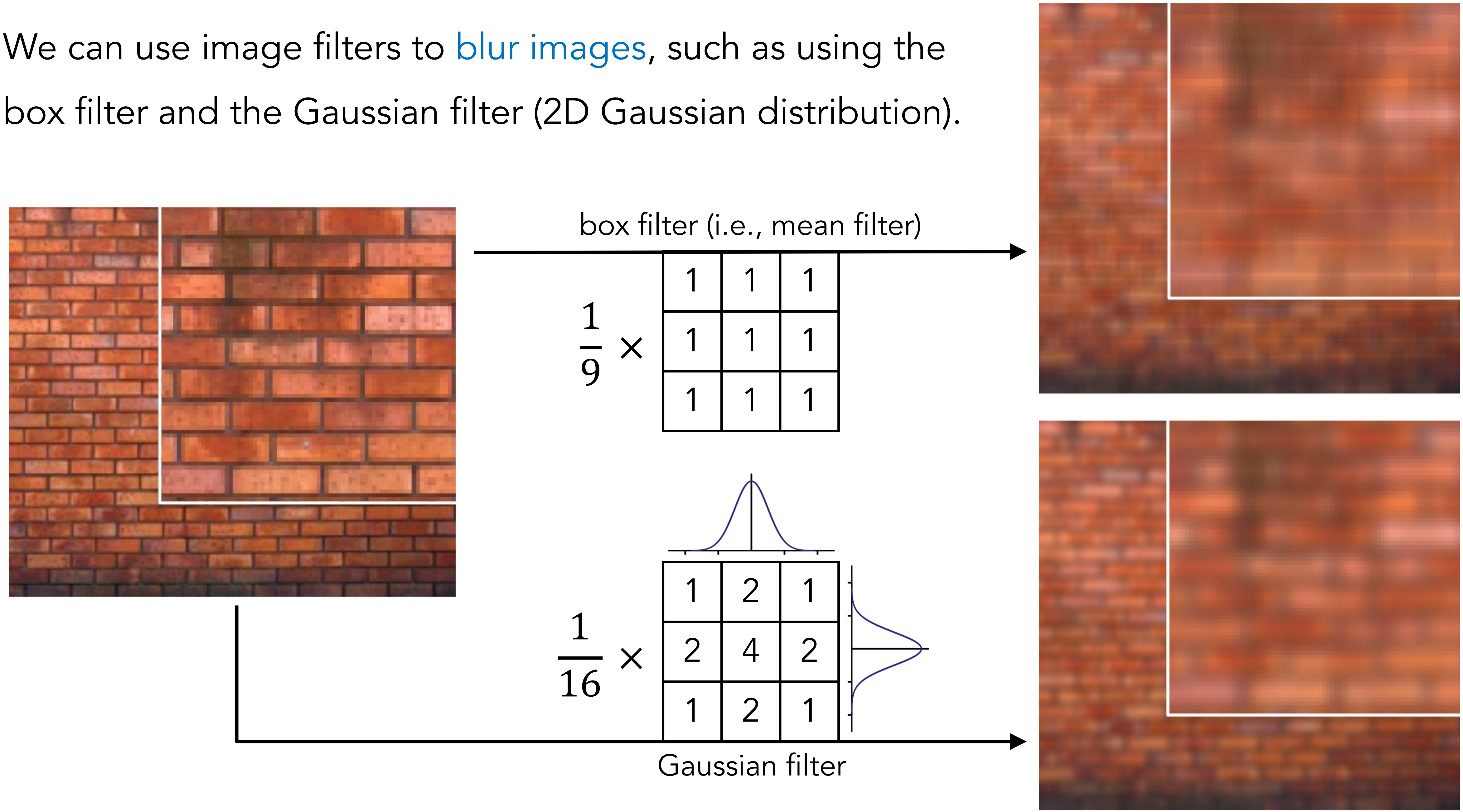
Scale up the intensity

Remove blurred signals

Sharpened image

Source -- https://www.cs.cornell.edu/courses/cs5670/2022sp/lectures/lec01_filter_for_web.pdf

We can use image filters to blur images, such as using the box filter and the Gaussian filter (2D Gaussian distribution).

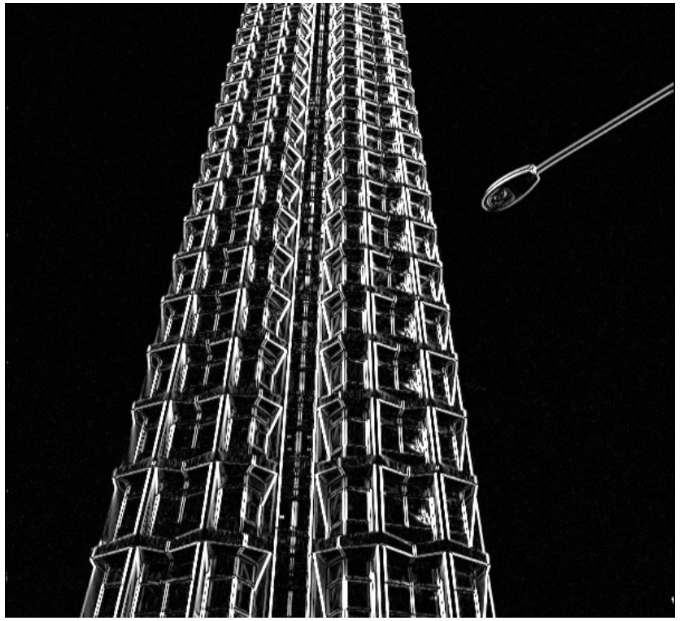


Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering

We can use image filters to detect edges, such as using the Sobel filter to compute the derivative of signals.

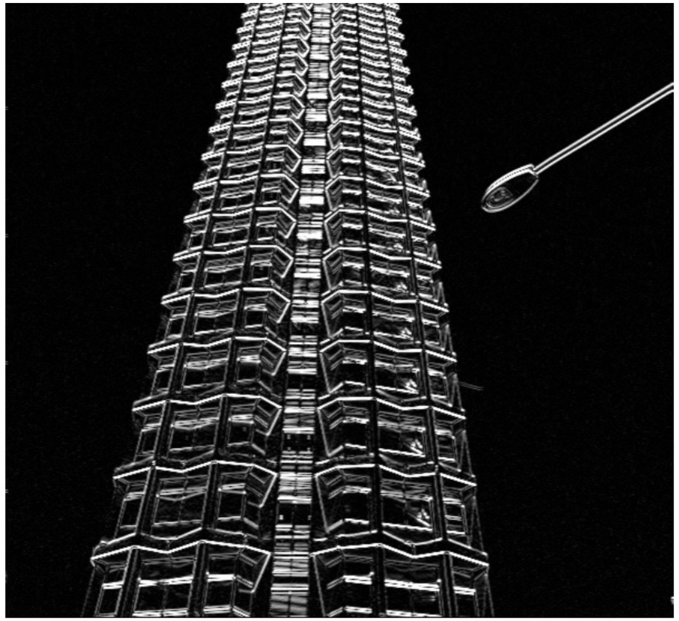
Horizontal Sobel filter

1	0	-1
2	0	-2
1	0	-1



1	2	1
0	0	0
-1	-2	-1

Vertical Sobel filter

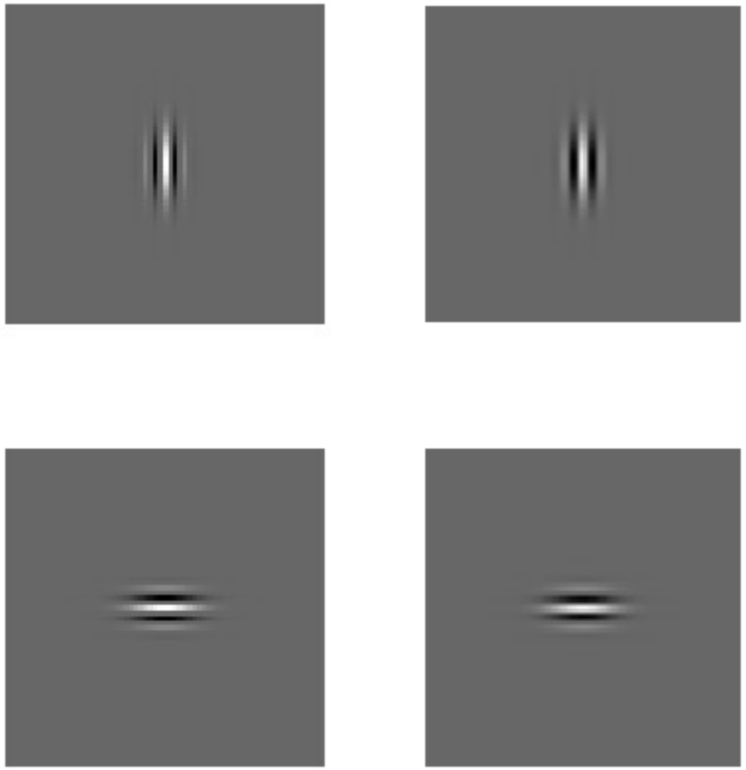


Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/filtering

We can use various hand-crafted convolutional kernels (i.e., filter banks) to extract different kinds of information in the image, such as using the Gabor filters.

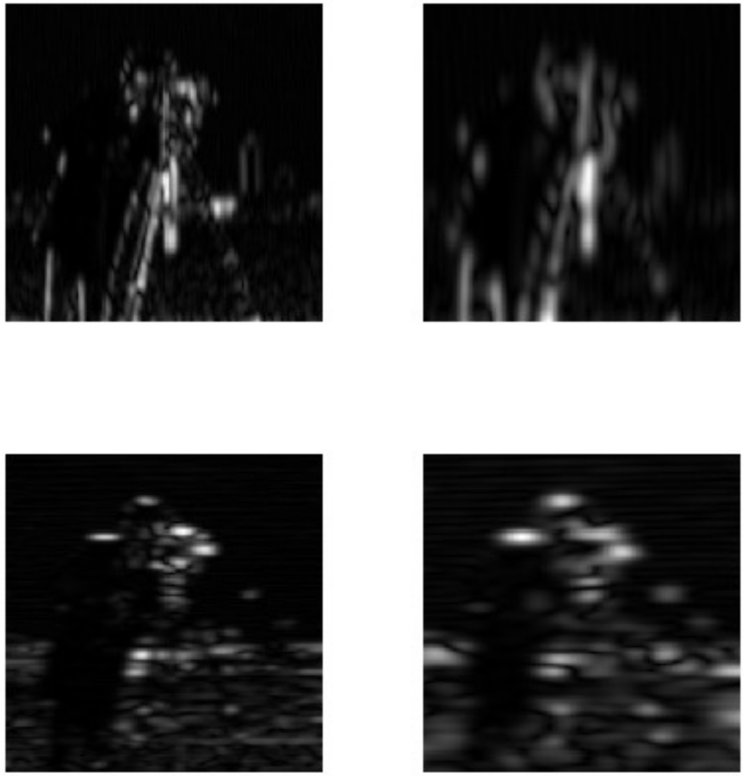
Input image

*



A partial set of Gabor filters

=

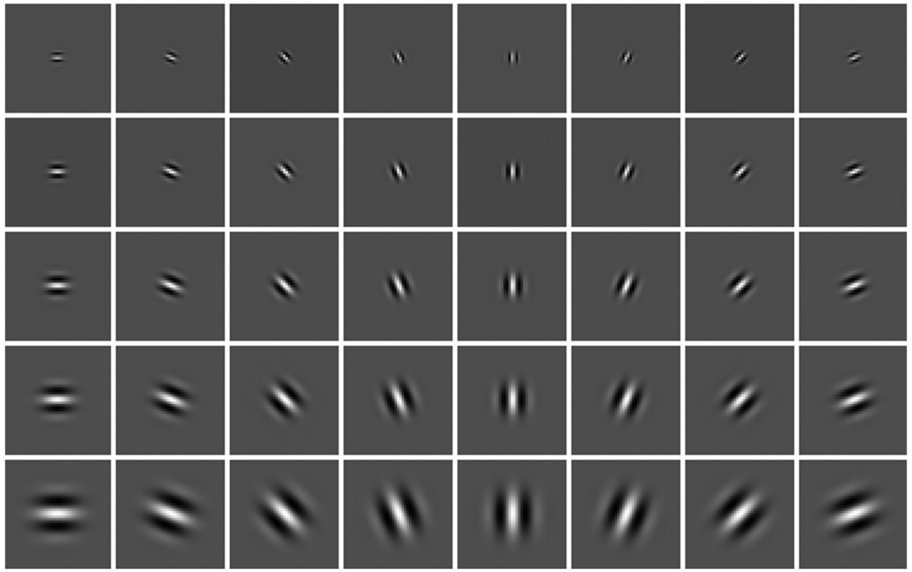


Output images

Source -- https://www.baeldung.com/cs/ml-gabor-filters

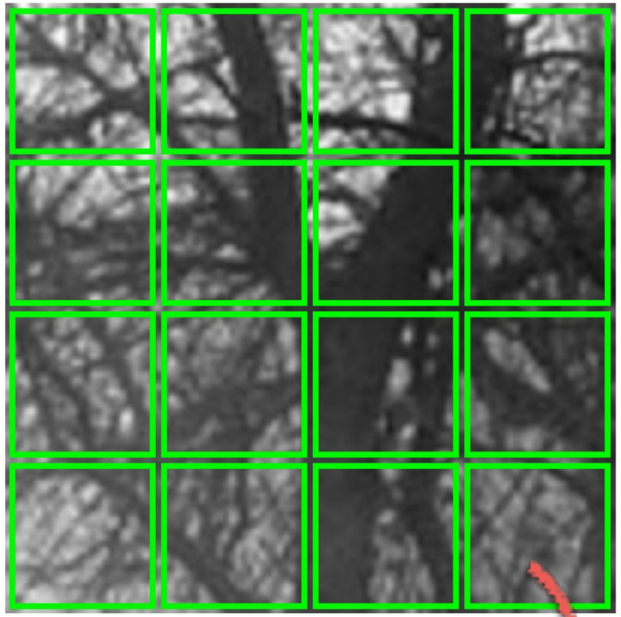
To construct features, we can perform convolution on image patches using a set of kernels (i.e., filter banks) and then aggregate them into a feature vector.

A set of N convolutional kernels



For each cell, compute filter responses using convolution, and then average them

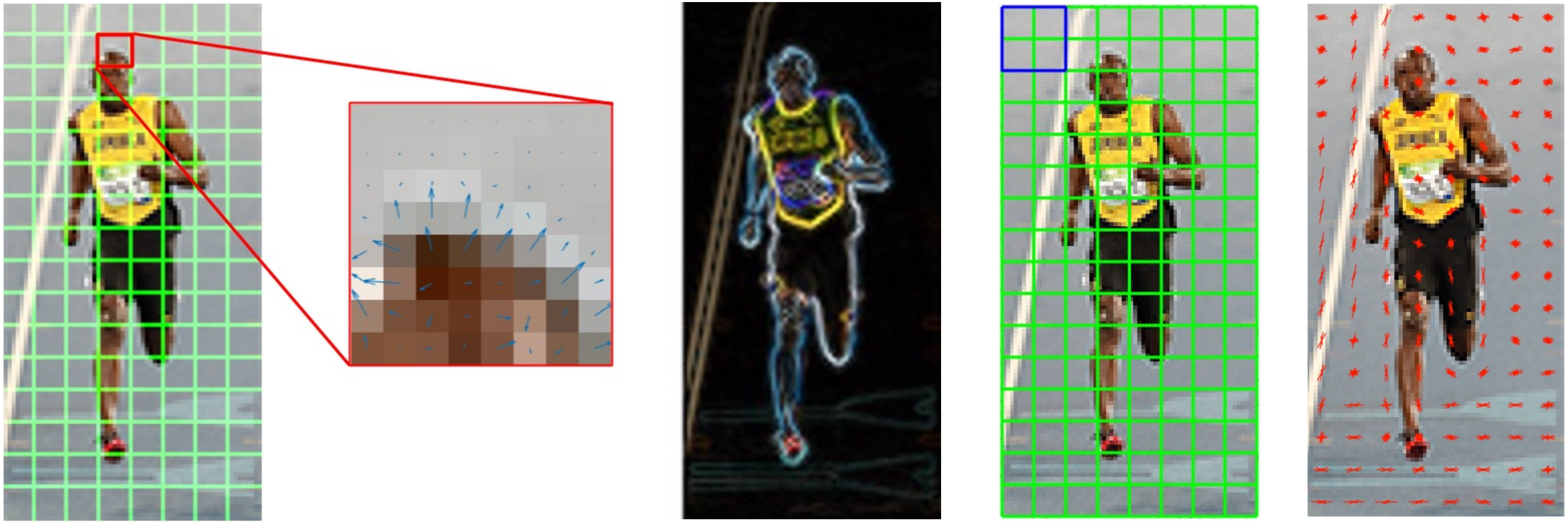
4 x 4 cell



Get a $4 \times 4 \times N$ feature vector

Source -- http://16385.courses.cs.cmu.edu/fall2022/lecture/features

There are many ways of constructing feature vectors, such as Histogram of Gradients.



Compute gradients using Sobel filter → Aggregate gradients into histograms

Source -- https://learnopencv.com/histogram-of-oriented-gradients/

Take-Away Messages

- Computers represent images as tensors with numbers in several channels (e.g., red, green, and blue).
- The origin (i.e., index $[0,0]$) of the image coordinate is at the top-left corner of the image.
- Convolution (the discrete version) means sliding a kernel/filter over the image to compute the sum of element-wise multiplication, which can also be seen as a dot product.
- We can use convolution to perform many different image-filtering operations, such as shifting pixels, blurring/sharpening images, and detecting edges.
- We can use a filter bank (i.e., a collection of kernels) to perform convolution on image patches, and the result (i.e., a set of filter responses) can be used as features.

Questions?