Image Processing

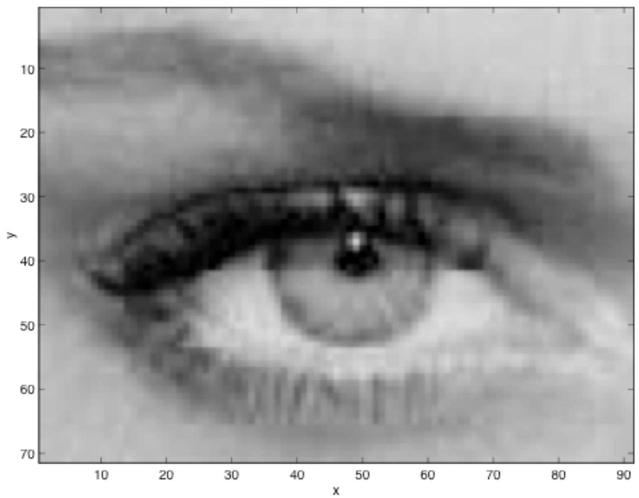
What is an image?

- We can think of an **image** as a function, f, from R^2 to R:
  - $f(x, y)$ gives the **intensity** at position (x, y)
  - Realistically, we expect the image only to be defined over a rectangle, with a finite range:
    - f: $[a,b] \times [c,d] \rightarrow [0,1]$

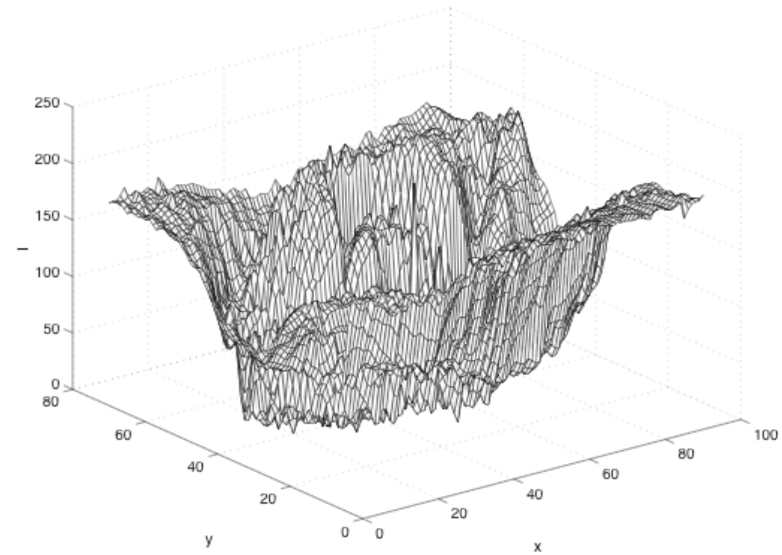
- A color image is just three functions pasted together. We can write this as a “vector-valued” function:

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

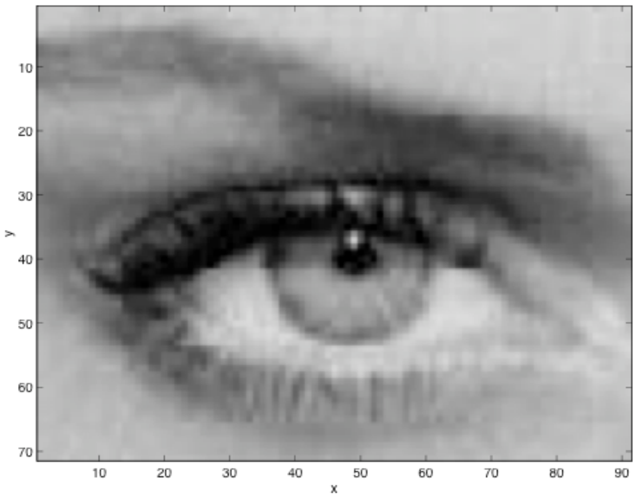
Image

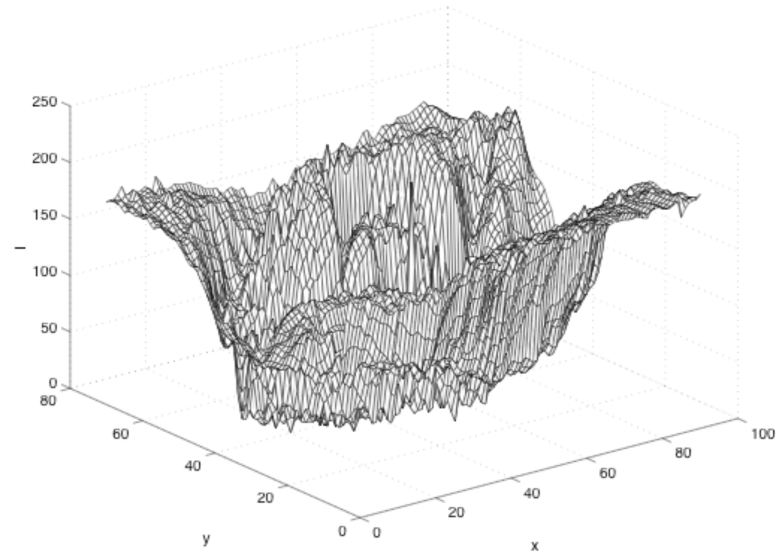


Brightness values



I(x,y)





62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

What is a digital image?

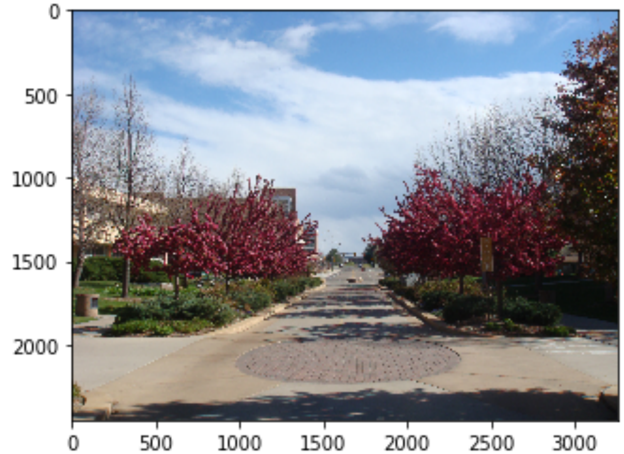
- In computer vision we usually operate on **digital** (**discrete**) images:
  - **Sample** the 2D space on a regular grid
  - **Quantize** each sample (round to nearest integer)
- If our samples are Δ apart, we can write this as:
- $f[i\ ,j] = \text{Quantize}\{\ f(i\ \Delta,\ j\ \Delta)\ \}$
- The image can now be represented as a matrix of integer values

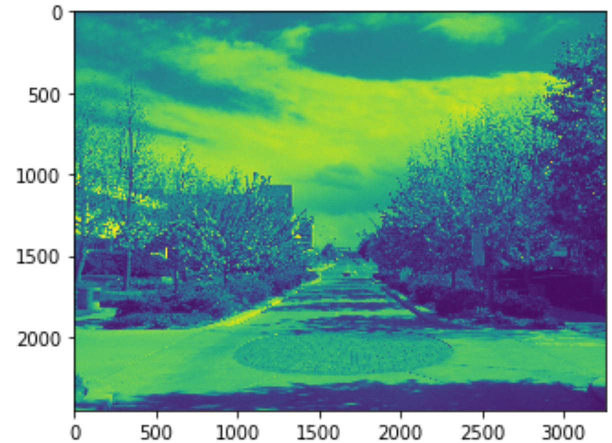
j →

i ↓

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

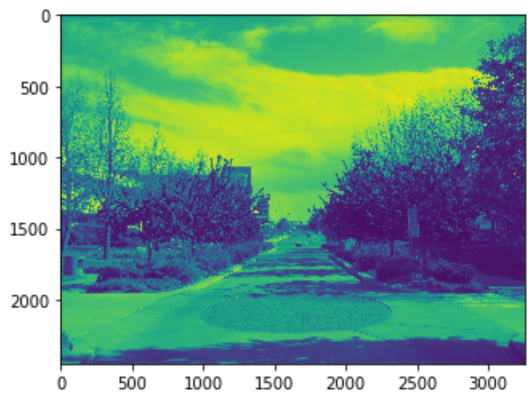
2448,3264,3





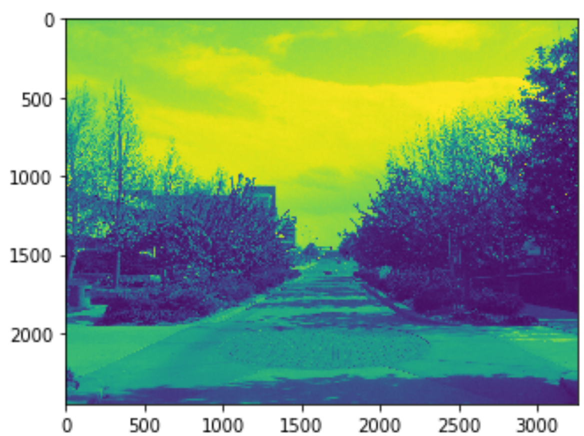
im[1:10,1:10,0]

```
array([[88, 89, 90, 90, 91, 91, 92, 92, 91],
       [89, 90, 91, 91, 91, 92, 92, 92, 92],
       [90, 92, 92, 92, 91, 92, 92, 93, 93],
       [91, 93, 93, 92, 91, 92, 92, 93, 93],
       [91, 92, 92, 92, 92, 92, 93, 93, 94],
       [91, 91, 92, 92, 93, 93, 94, 95, 94],
       [92, 92, 92, 93, 94, 94, 96, 96, 96],
       [93, 93, 93, 93, 92, 92, 91, 94, 94],
       [94, 94, 94, 94, 93, 92, 92, 94, 94]], dtype=uint8)
```



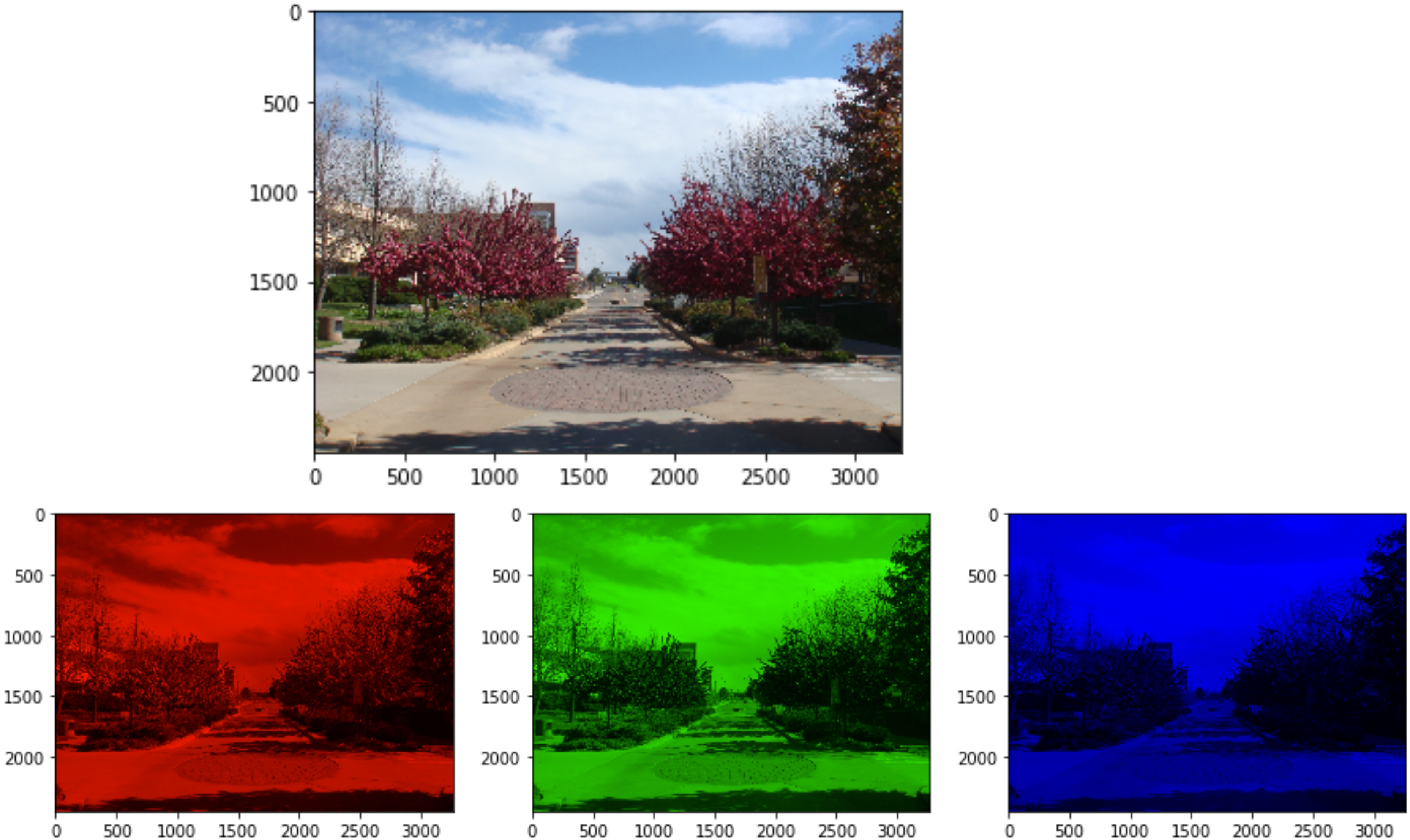
im[1:10,1:10,1]

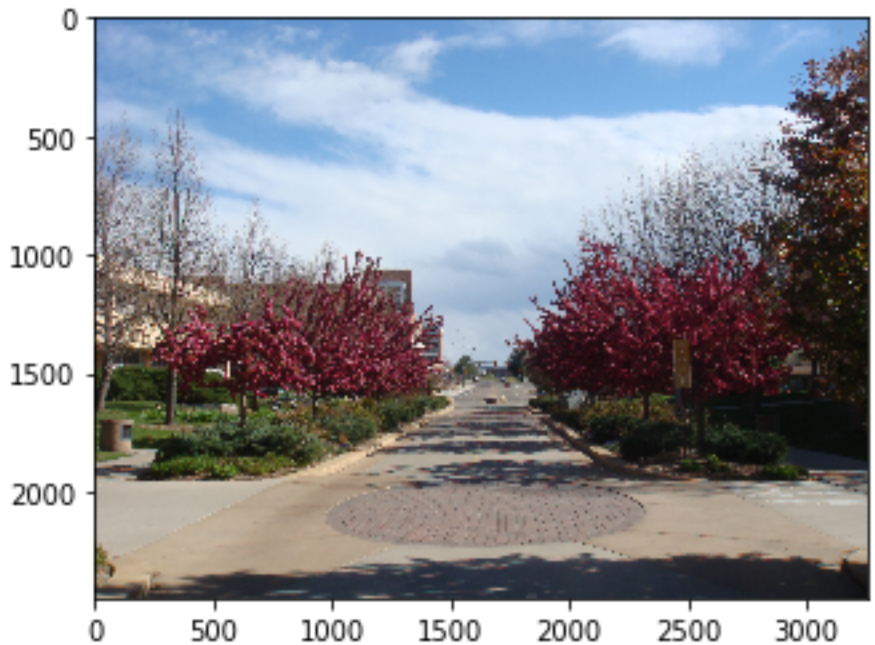
```
array([[134, 135, 136, 136, 137, 137, 138, 138, 137],
       [135, 136, 137, 137, 137, 138, 138, 138, 138],
       [136, 138, 138, 138, 137, 138, 138, 139, 139],
       [137, 139, 139, 138, 137, 138, 138, 139, 139],
       [137, 138, 138, 138, 138, 138, 139, 139, 139],
       [137, 137, 138, 138, 139, 139, 140, 140, 139],
       [138, 138, 138, 139, 140, 140, 141, 141, 141],
       [137, 137, 137, 137, 138, 138, 137, 140, 140],
       [138, 138, 138, 138, 139, 138, 138, 140, 140]], dtype=uint8)
```



im[1:10,1:10,2]

```
array([[194, 195, 196, 196, 197, 197, 198, 198, 196],
       [195, 196, 197, 197, 197, 197, 197, 197, 197],
       [195, 197, 197, 197, 196, 197, 197, 198, 198],
       [196, 198, 198, 197, 196, 197, 197, 198, 198],
       [196, 197, 197, 197, 197, 197, 198, 198, 198],
       [196, 196, 197, 197, 198, 198, 199, 199, 198],
       [197, 197, 197, 198, 199, 199, 200, 200, 200],
       [198, 198, 198, 198, 197, 197, 196, 199, 199],
       [199, 199, 199, 199, 198, 197, 197, 199, 199]], dtype=uint8)
```


0 500 1000 1500 2000
0 500 1000 1500 2000 2500 3000
0 500 1000 1500 2000
0 500 1000 1500 2000 2500 3000
0 500 1000 1500 2000
0 500 1000 1500 2000 2500 3000
0 500 1000 1500 2000
0 500 1000 1500 2000 2500 3000




```
array([[ 127.086,  128.086,  129.086,  129.086,  130.086,  130.086,
         131.086,  131.086,  129.972],
       [ 128.086,  129.086,  130.086,  130.086,  130.086,  130.972,
         130.972,  130.972,  130.972],
       [ 128.972,  130.972,  130.972,  130.972,  129.972,  130.972,
         130.972,  131.972,  131.972],
       [ 129.972,  131.972,  131.972,  130.972,  129.972,  130.972,
         130.972,  131.972,  131.972],
       [ 129.972,  130.972,  130.972,  130.972,  130.972,  130.972,
         131.972,  131.972,  132.271],
       [ 129.972,  129.972,  130.972,  130.972,  131.972,  131.972,
         132.972,  133.271,  132.271],
       [ 130.972,  130.972,  130.972,  131.972,  132.972,  132.972,
         134.271,  134.271,  134.271],
       [ 130.798,  130.798,  130.798,  130.798,  130.972,  130.972,
         129.972,  132.972,  132.972],
       [ 131.798,  131.798,  131.798,  131.798,  131.972,  130.972,
         130.972,  132.972,  132.972]])
```

RGB to Grayscale

- The relationship between grayscale reflectance of a surface and its RGB color equivalent is given by:

  $$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

Image Processing

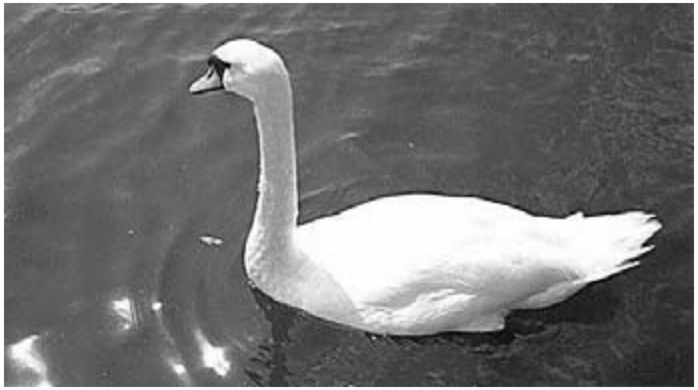
- An **image processing** operation typically defines a new image g in terms of an existing image f.

$$g(x, y) = t(f(x, y))$$

Image Processing at pixel level - Thresholding

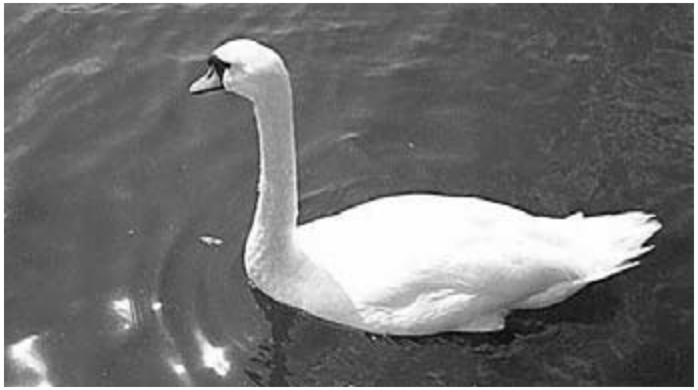
One of the simplest operations we can perform on an image is *thresholding*.

For example, take the swan image and threshold it with a value of T.



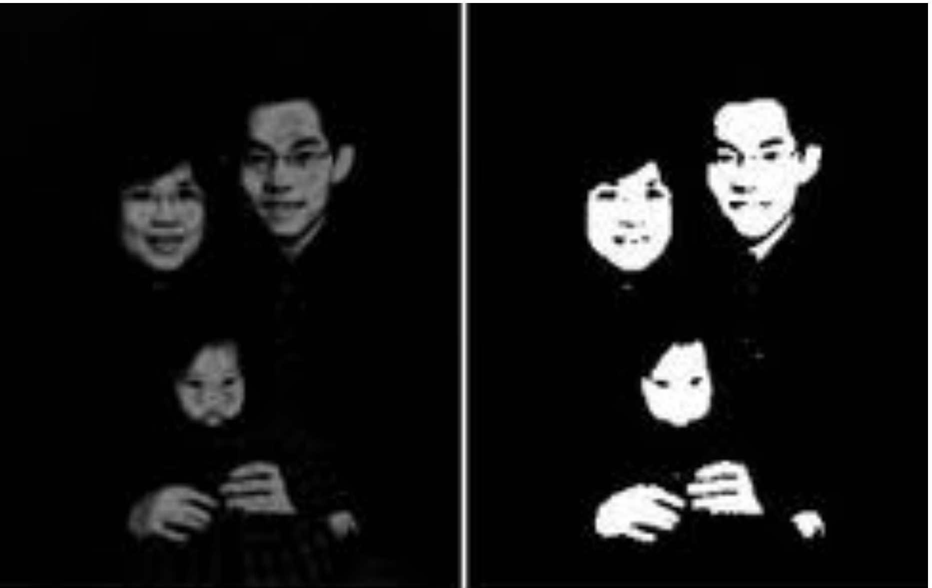
Make all pixels >= T into 1, and all pixels < T into 0.

Threshold T=128

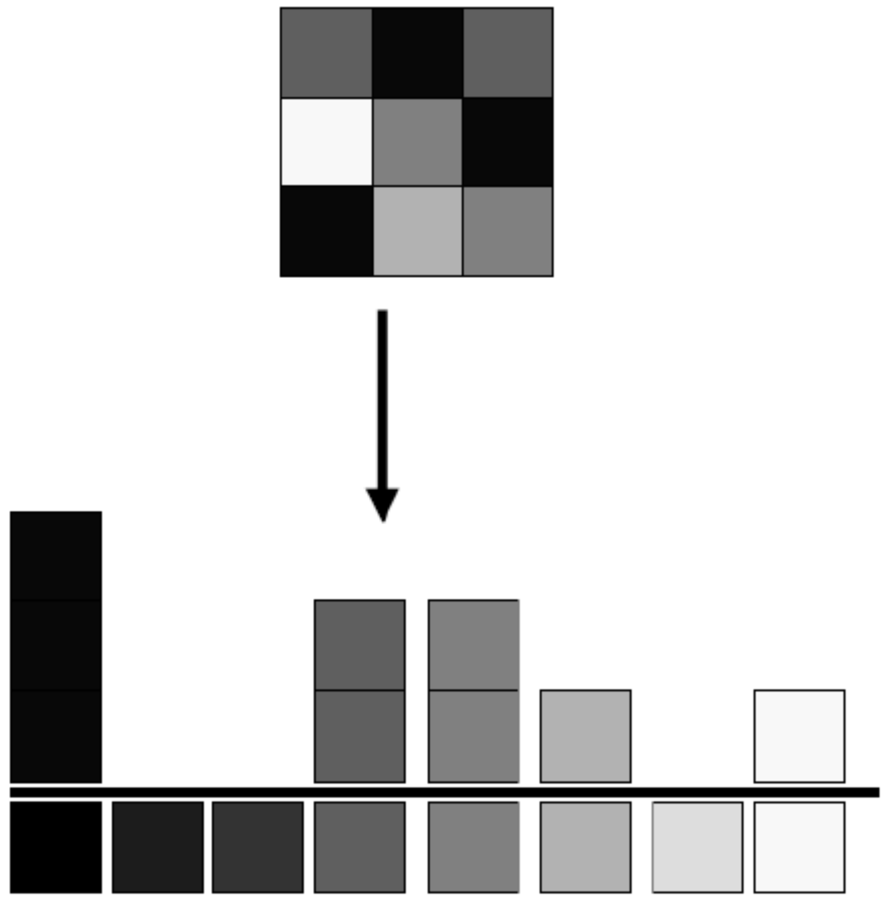


$$im[im > 128] = 255$$
$$im[im \leq 128] = 0$$

Examples



A simple image and its histogram

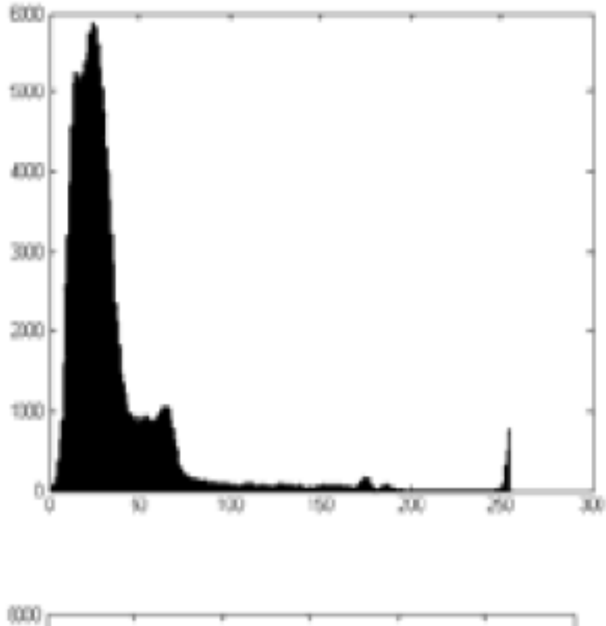


Definition of histogram

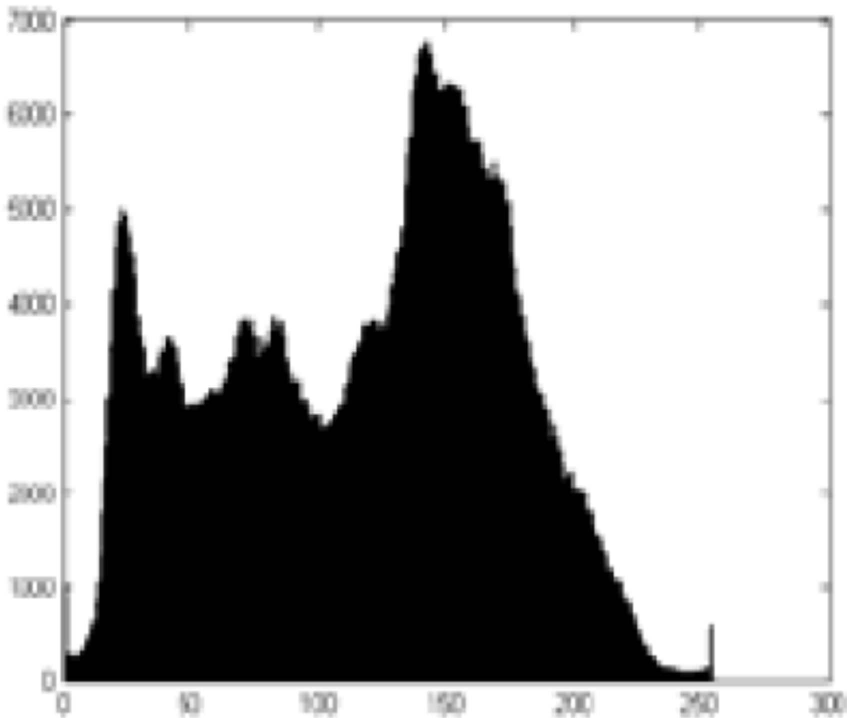
- To write this down, we might say that we have an image, I, in which the intensity at pixel with coordinates (x,y) is I(x,y).
- We would write the histogram h, as h(i) indicating that intensity i, appears h(i) times in the image.
- If we let the expression (a=b) have the value 1 when a=b, and 0 otherwise, we can write for histogram h(i):

$$h(i) = \sum_x \sum_y I(x, y)$$

Histogram example



Histogram example



NUMBER OF BINS AND SIZES

Pick a discrete set of bins, then put values into the bins

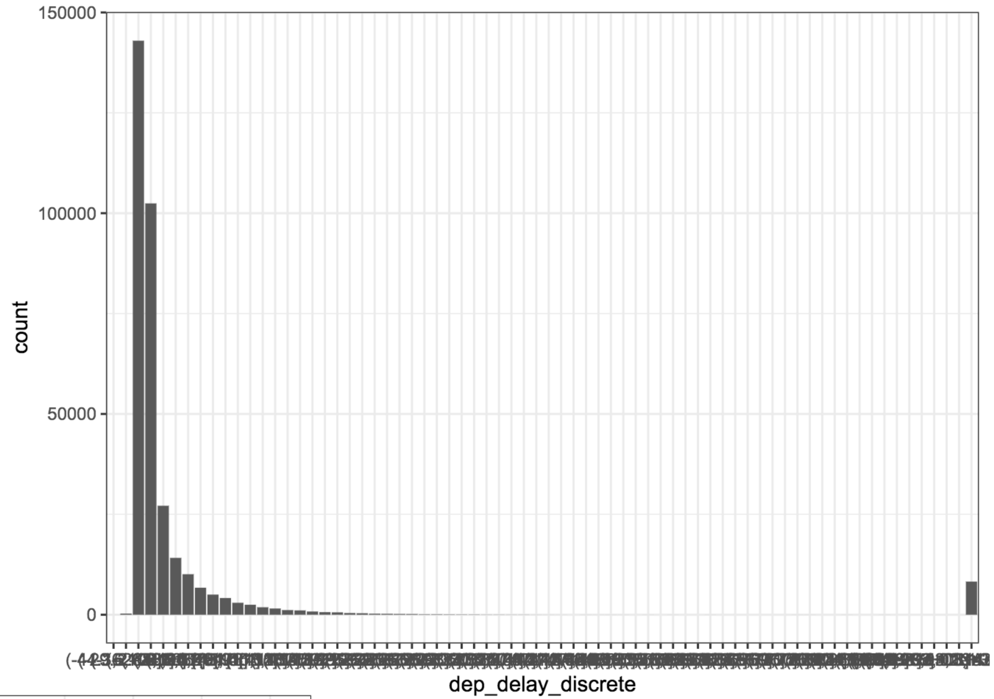
Equal-length bins:

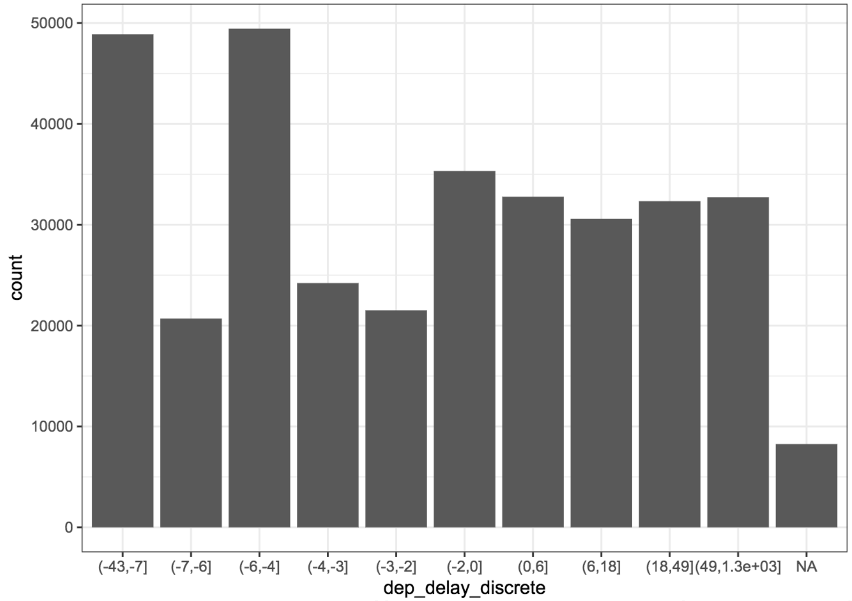
- Bins have an equal-length range and skewed membership
- Good/Bad ????????

Equal-sized bins:

- Bins have variable-length ranges but equal membership
- Good/Bad ????????

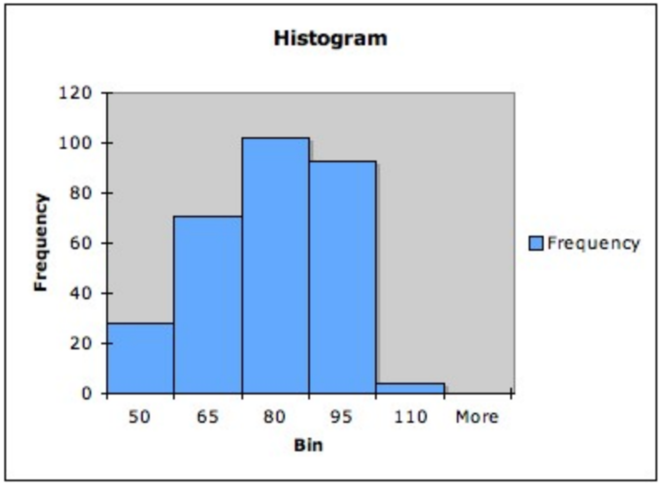
BIN SIZE

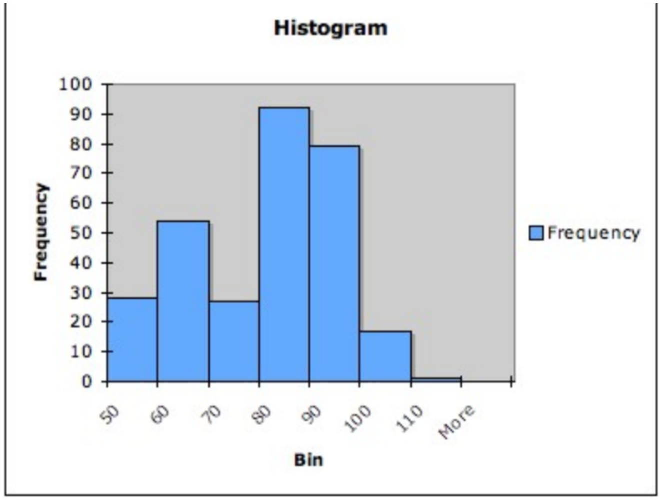


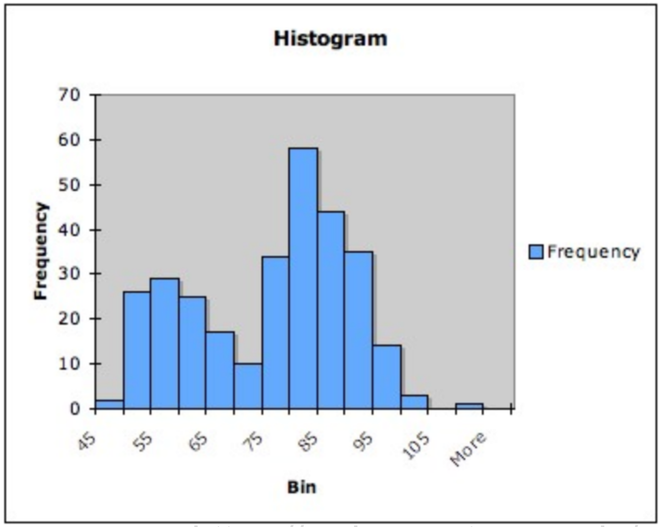


http://www.hcbravo.org/IntroDataSci/bookdown-notes/eda-data-transformations.html

DIFFERENT NUMBER OF BINS

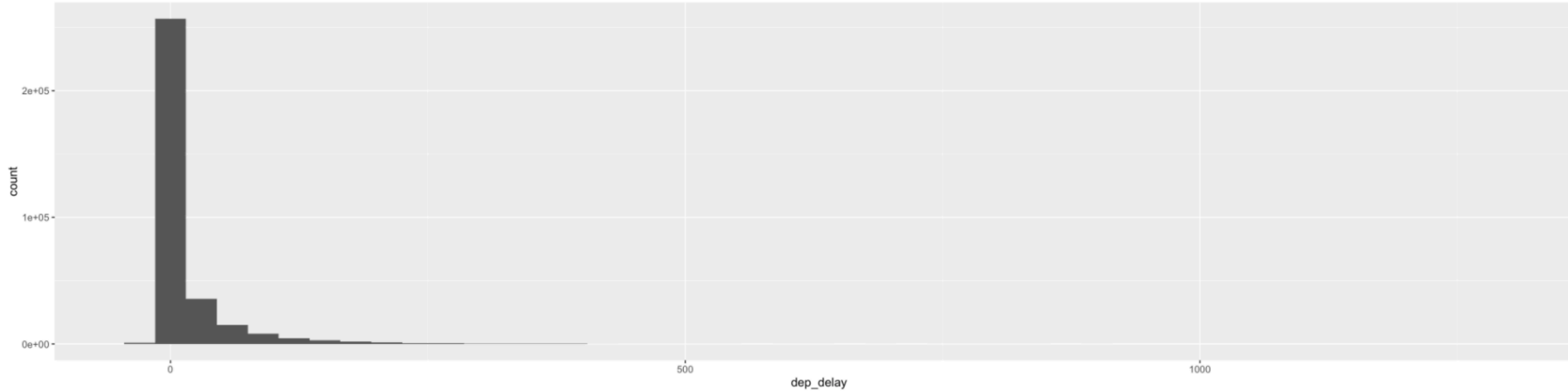




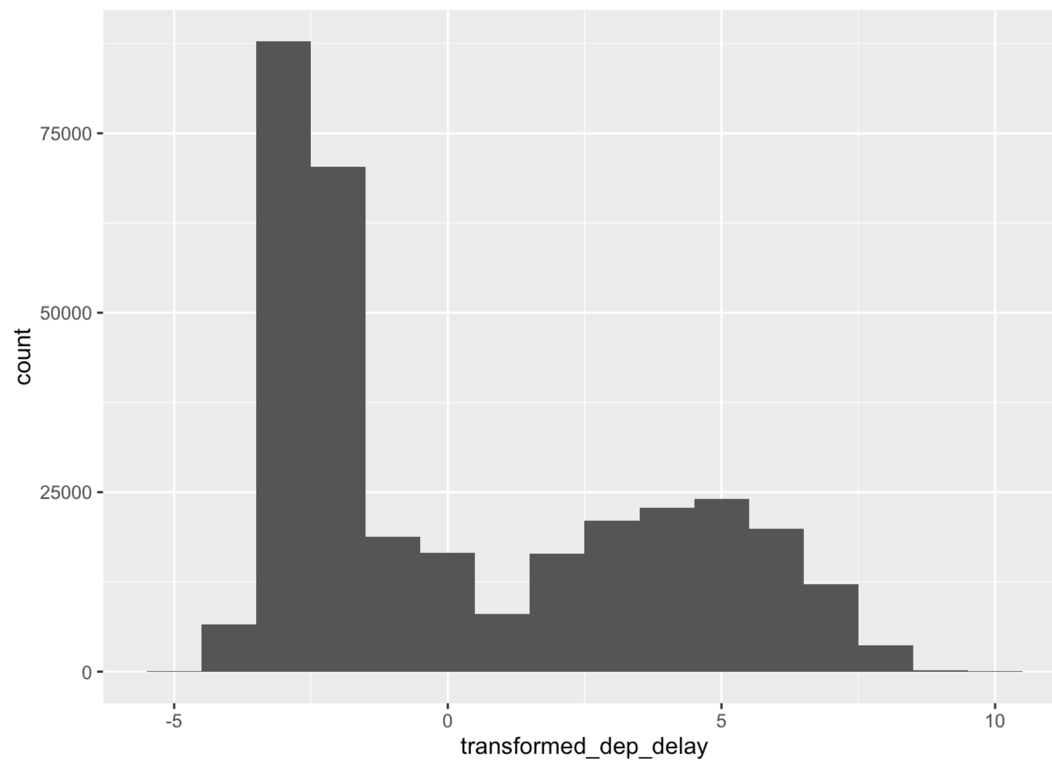


https://web.ma.utexas.edu/users/mks/statmistakes/dividingcontinuousintocategories.html

SKEWED DATA



log$_2$ transform



HISTOGRAM BINS AND WIDTHS

Square formula

$$bins = \sqrt{n} \qquad binwidth = \frac{max(valuees) - min(values)}{\sqrt{n}}$$

Sturges formula

$$bins = ceil(log_2 n) + 1 \qquad binwidth = \frac{max(valuees) - min(values)}{ceil(\log_2 n) + 1}$$

Rice formula

$$bins = 2 \times n^{1/3} \qquad binwidth = \frac{max(valuees) - min(values)}{bins}$$

Scott formula

$$bins = \frac{max(values) - min(values)}{3.5 \times \frac{stdev(values}{n^{1/3}}} \qquad binwidth = 3.5 \times \frac{stdev(values}{n^{1/3}}$$

Freedman-Diaconis formula

$$bins = \frac{max(values) - min(values)}{2 \times \frac{IQR(values)}{n^{1/3}}} \qquad binwidth = 2 \times \frac{IQR(values)}{n^{1/3}}$$

Histograms allow image manipulation

- One reason to compute a histogram is that it allows us to manipulate an image by changing its histogram. We do this by creating a new image, J, in which:

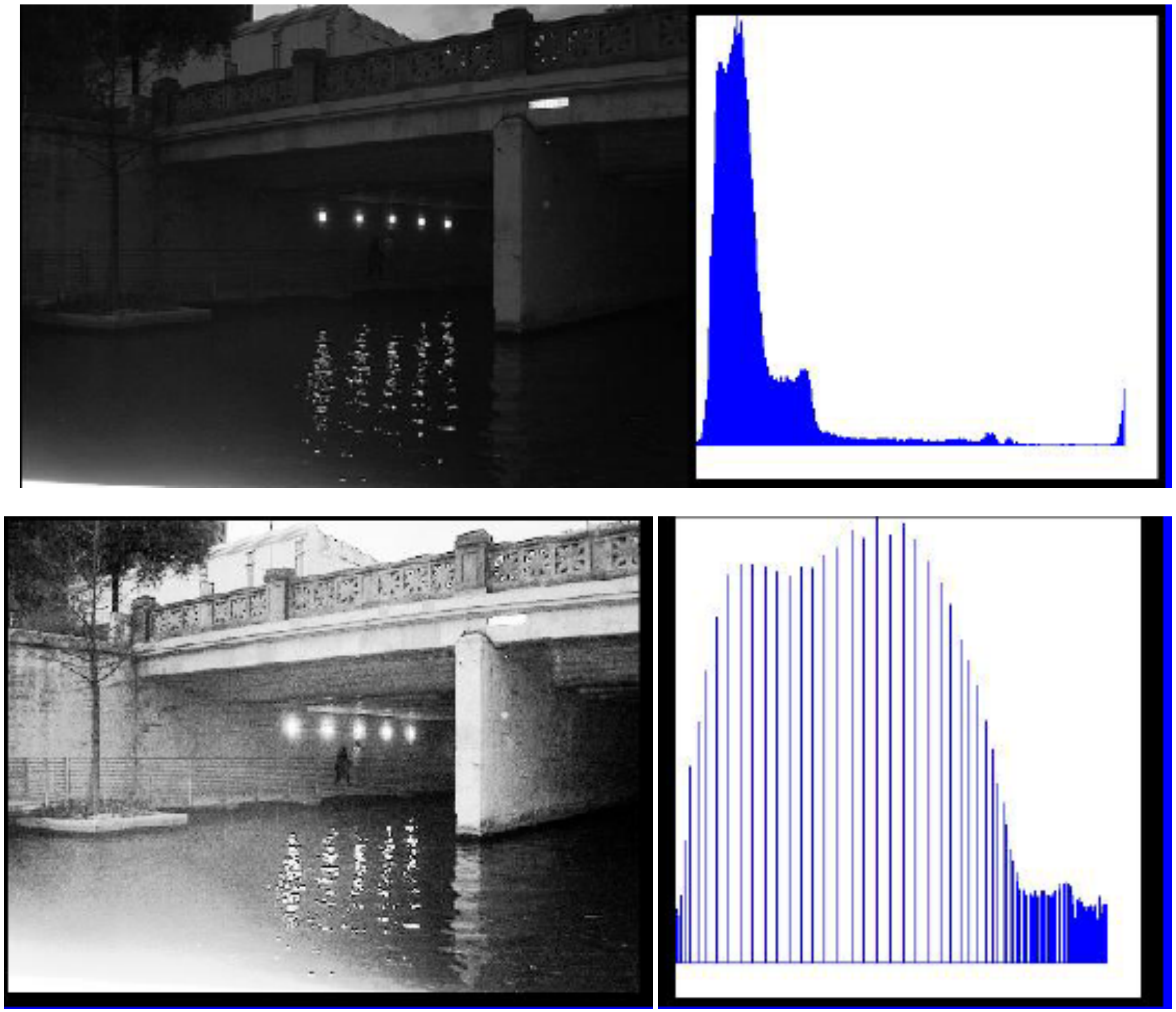
- The trick is to choose an f that will generate a nice or useful image. Typically, we choose f to be monotonic. This means that: if $u<v$ then $f(u) < f(v)$. Non-monotonic functions tend to make an image look truly different, while monotonic changes will be more subtle.

$$J(x, y) = f(I(x, y))$$

Histogram Equalization

- The idea is to spread out the histogram so that it makes full use of the dynamic range of the image.
- For example, if an image is very dark, most of the intensities might lie in the range 0-50. By choosing f to spread out the intensity values, we can make fuller use of the available intensities, and make darker parts of an image easier to understand.
- If we choose f to make the histogram of the new image, J, as uniform as possible, we call this **histogram equalization.**

Example of histogram equalization



8 x 8 Image

$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

8 x 8 Image

$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 \\ 69 & 85 & 64 \\ 70 & 87 & 69 \end{bmatrix}$$

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

8 x 8 Image

Value	Count	Value	Count	Value	Count
52	1	64	2	72	1
55	3	65	3	73	2
58	2	66	2	75	1
59	3	67	1	76	1
60	1	68	5	77	1
61	4	69	3	78	1
62	1	70	4	79	2
63	2	71	2	83	1

v, Pixel Intensity	cdf(v)	h(v), Equalized v
52	1	0
55	4	12
58	6	20
59	9	32
60	10	36
61	14	53
62	15	57
63	17	65
64	19	73
65	22	85
66	24	93

$$h(v) = round\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1)\right)$$

M – width M – height L – Number of gray levels

Comparing histograms

- **SSD** - let $h(i)$ and $g(i)$ be two histograms

$$||h - g|| = \sum_{i=1}^{N} (h(i) - g(i))^2$$

- **Cosine Distance**

$$cos(h, g) = \frac{h \cdot g}{||h||||g||}$$

- **Chi-square distance** - let $h(i)$ and $g(i)$ be two histograms

$$\chi^2(h(i), g(i)) = \frac{1}{2} \sum_{m=1}^{k} \frac{(h_i(m) - h_j(m))^2}{h_i(m) + h_j(m)}$$

Continuing operations at a pixel level

Cross - Correlation & Convolution

Correlation & Convolution

- Basic operation to extract information from an image.

- These operations have two key features:

    - shift invariant
    - linear

- Applicable to 1-D and multi dimensional images.

Correlation Example - 1D (Averaging)

Image I

2	3	6	5	5	1	8	9	7

$G = f(I)$

$I[2] = 3$

$$G[2] = \frac{2+3+6}{3} = \frac{11}{3}$$

2	$\frac{11}{3}$	6	5	5	1	8	9	7

$I[3] = 6$

$$G[3] = \frac{3+6+5}{3} = \frac{14}{3}$$

2	$\frac{11}{3}$	$\frac{14}{3}$	5	5	1	8	9	7

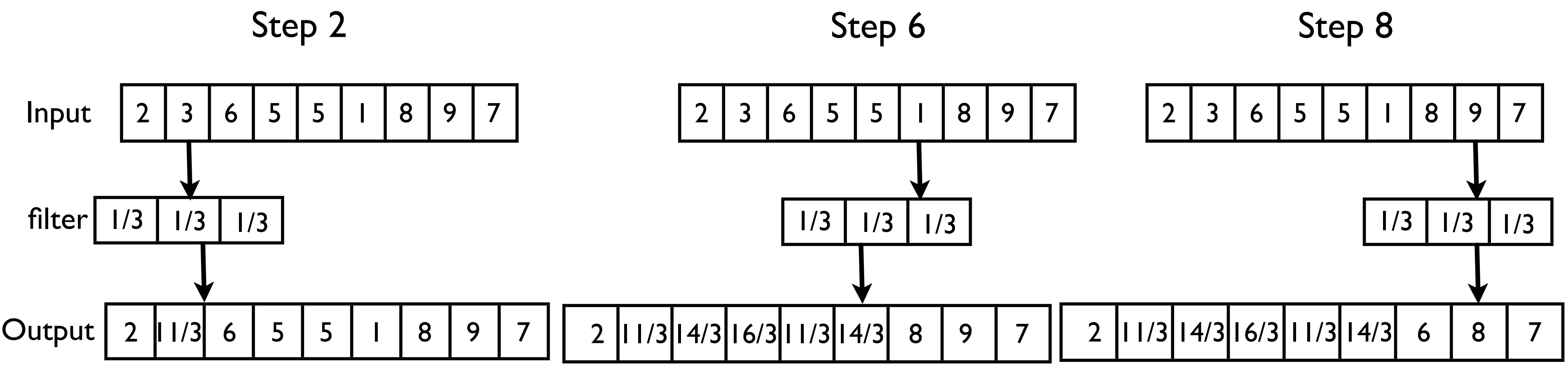
⋮

$I[8] = 9$

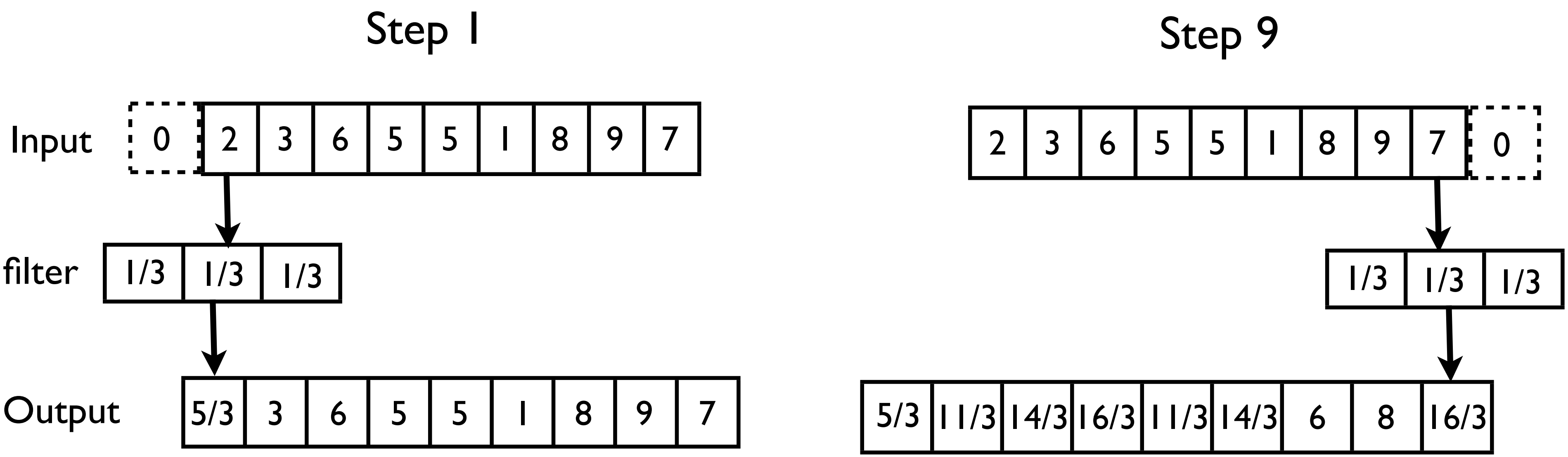
$$G[8] = \frac{8+9+7}{3} = 8$$

2	$\frac{11}{3}$	$\frac{14}{3}$	$\frac{16}{3}$	$\frac{11}{3}$	$\frac{14}{3}$	6	8	7

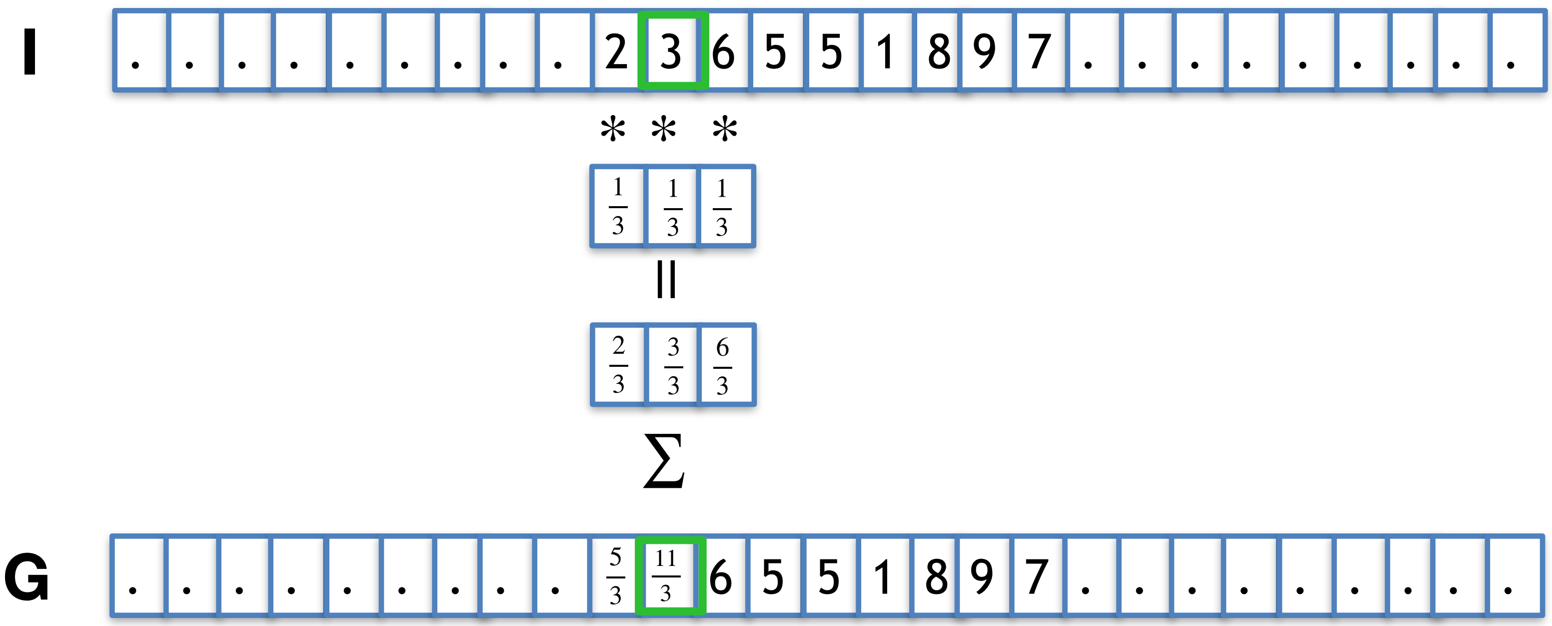
Correlation Example - 1D



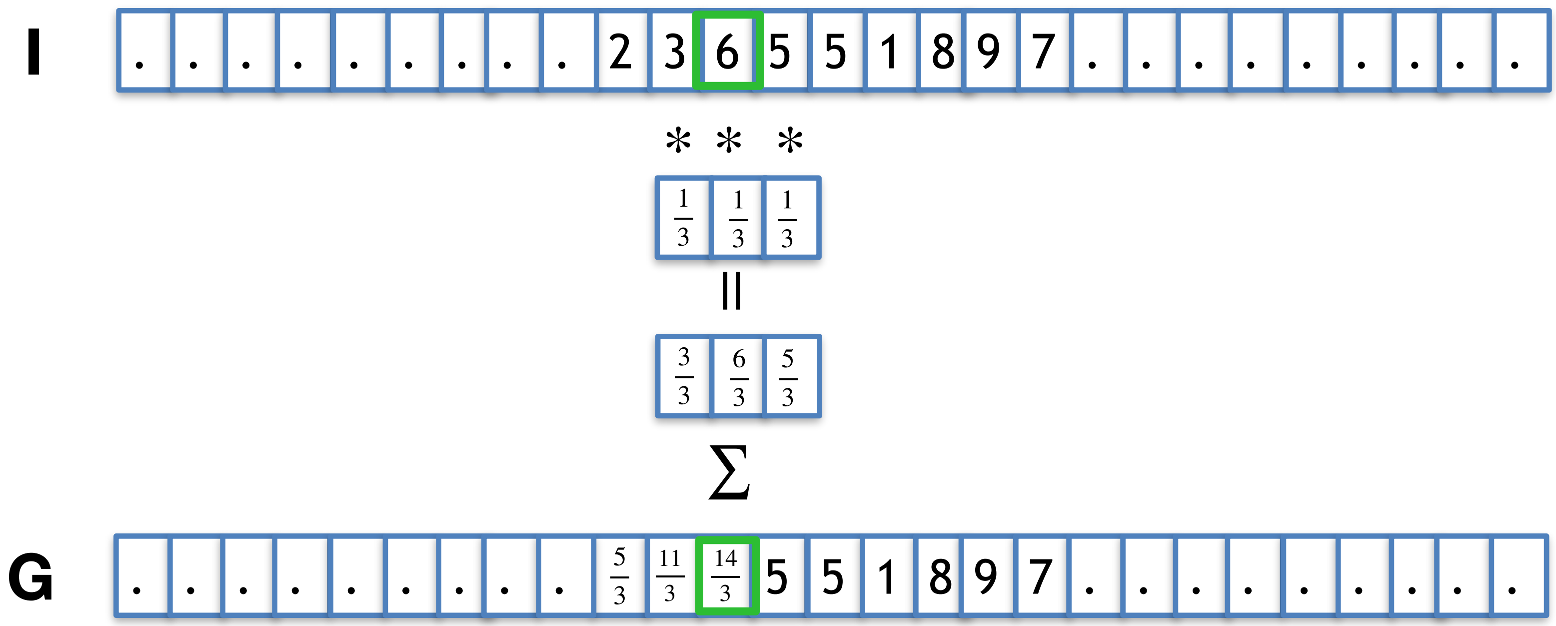
Correlation Example - 1D



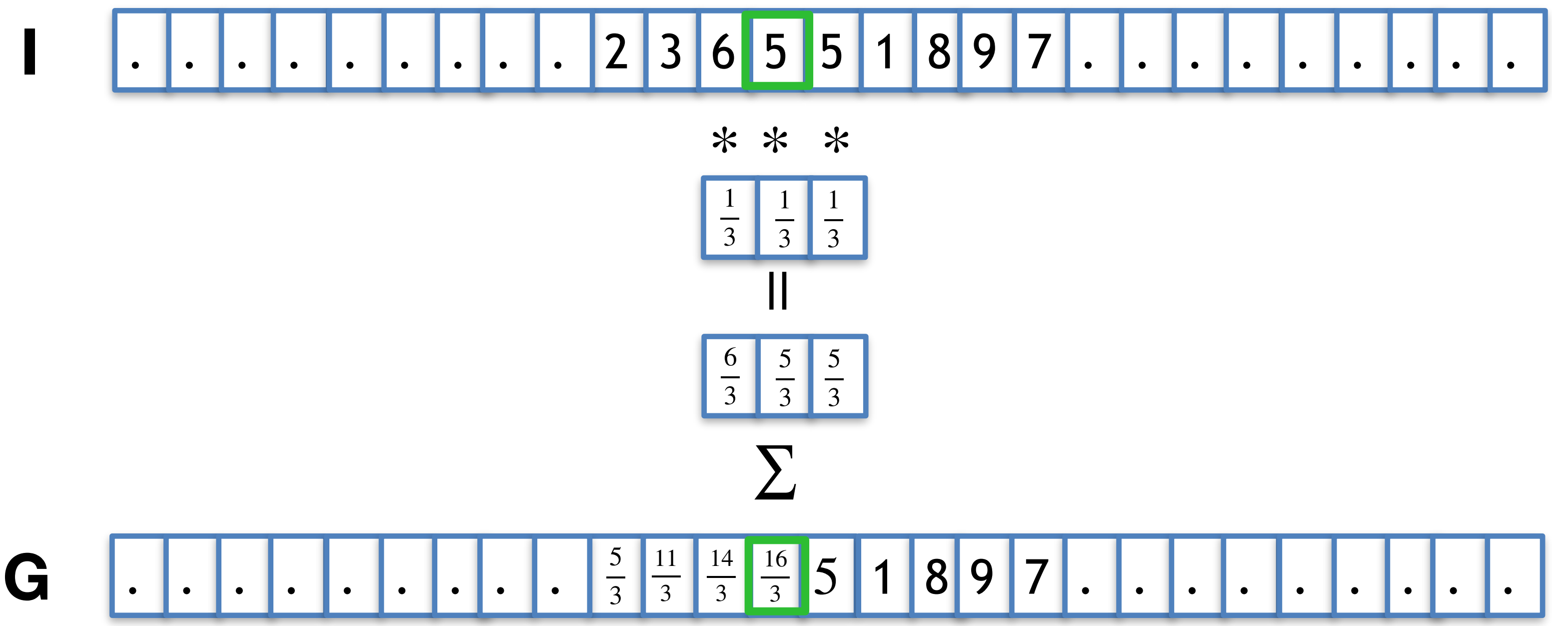
Correlation Example - 1D



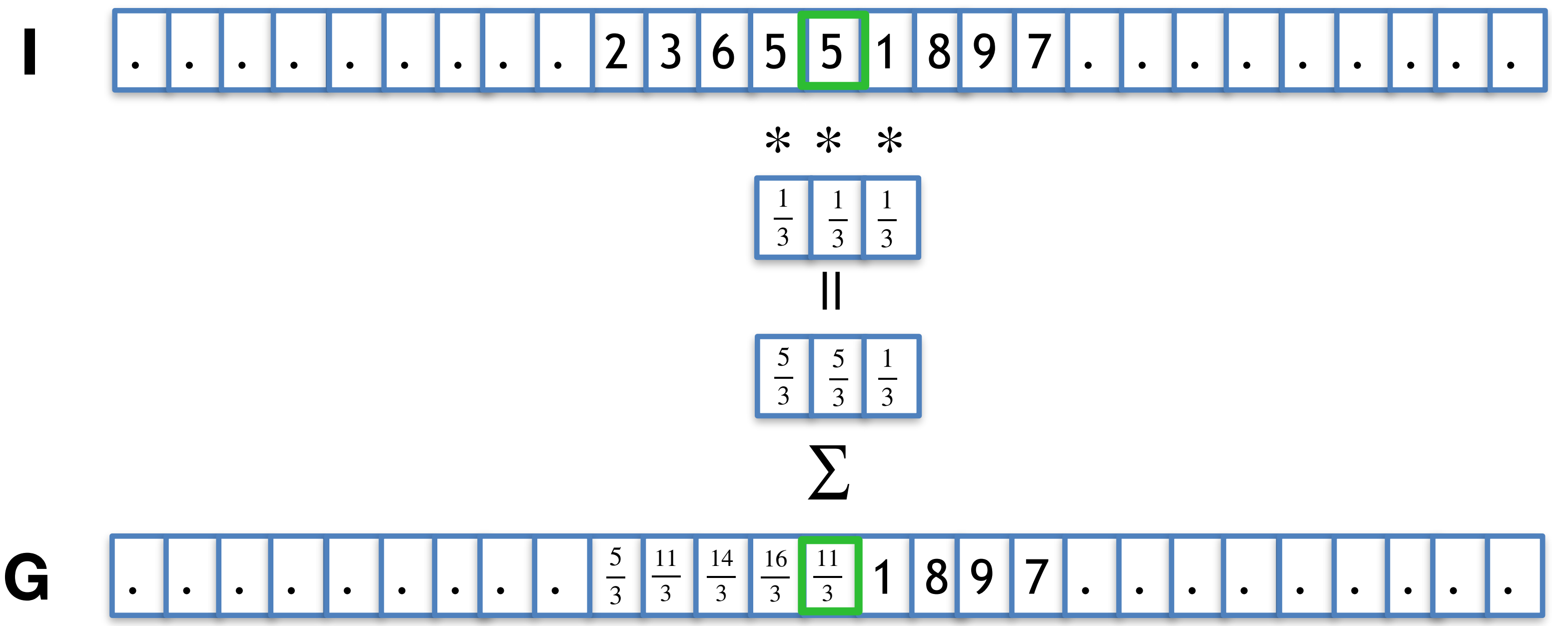
Correlation Example - 1D



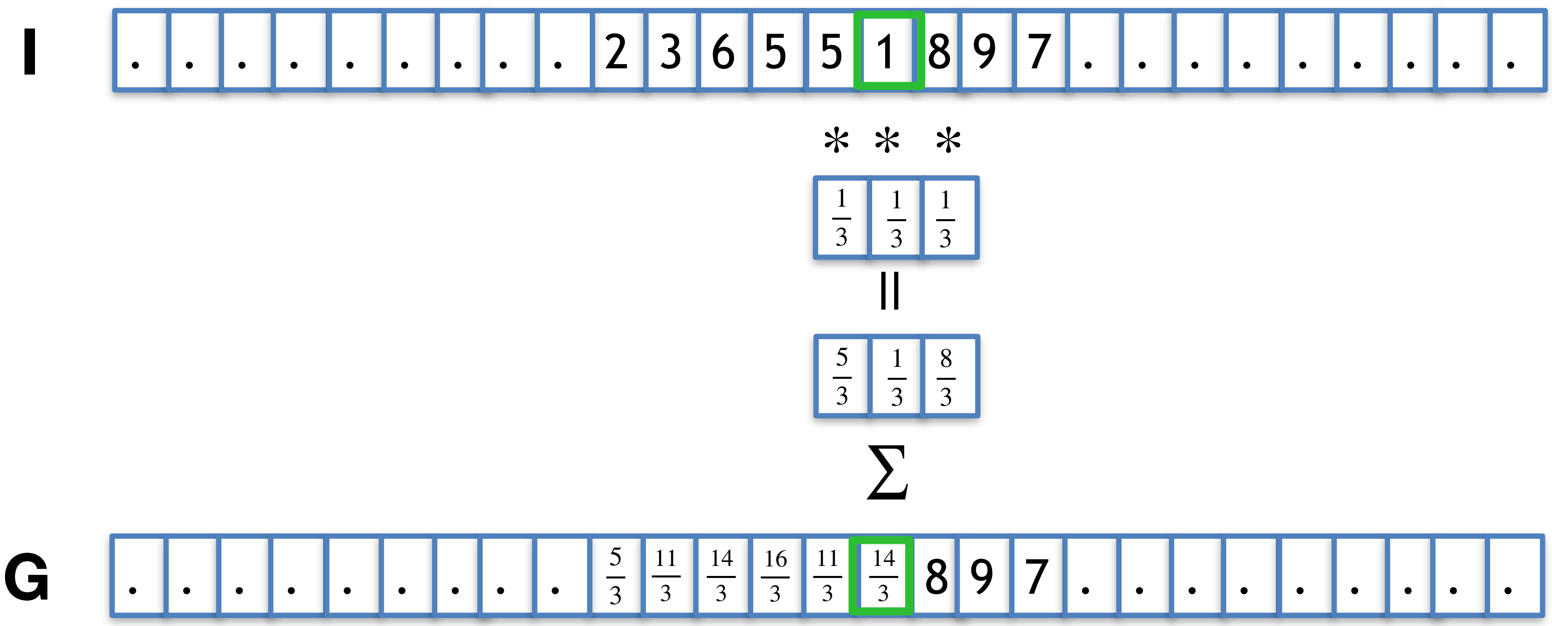
Correlation Example - 1D



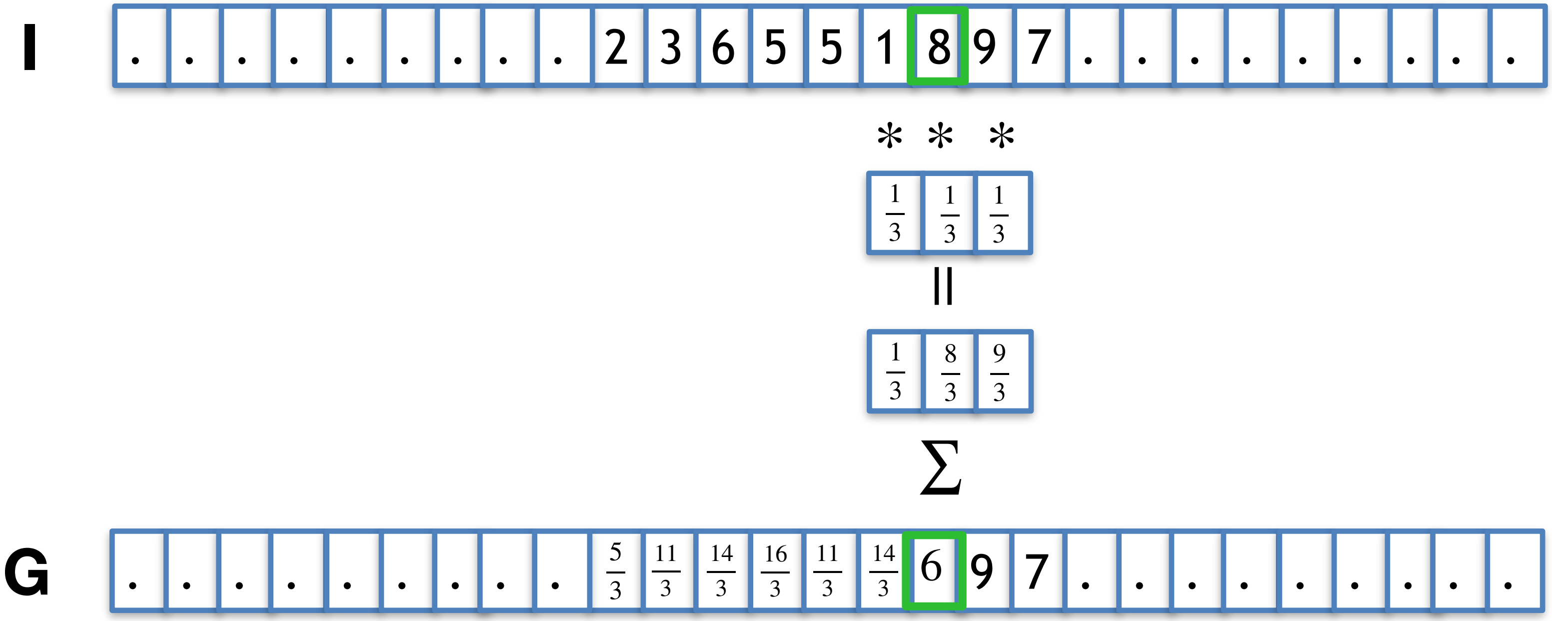
Correlation Example - 1D



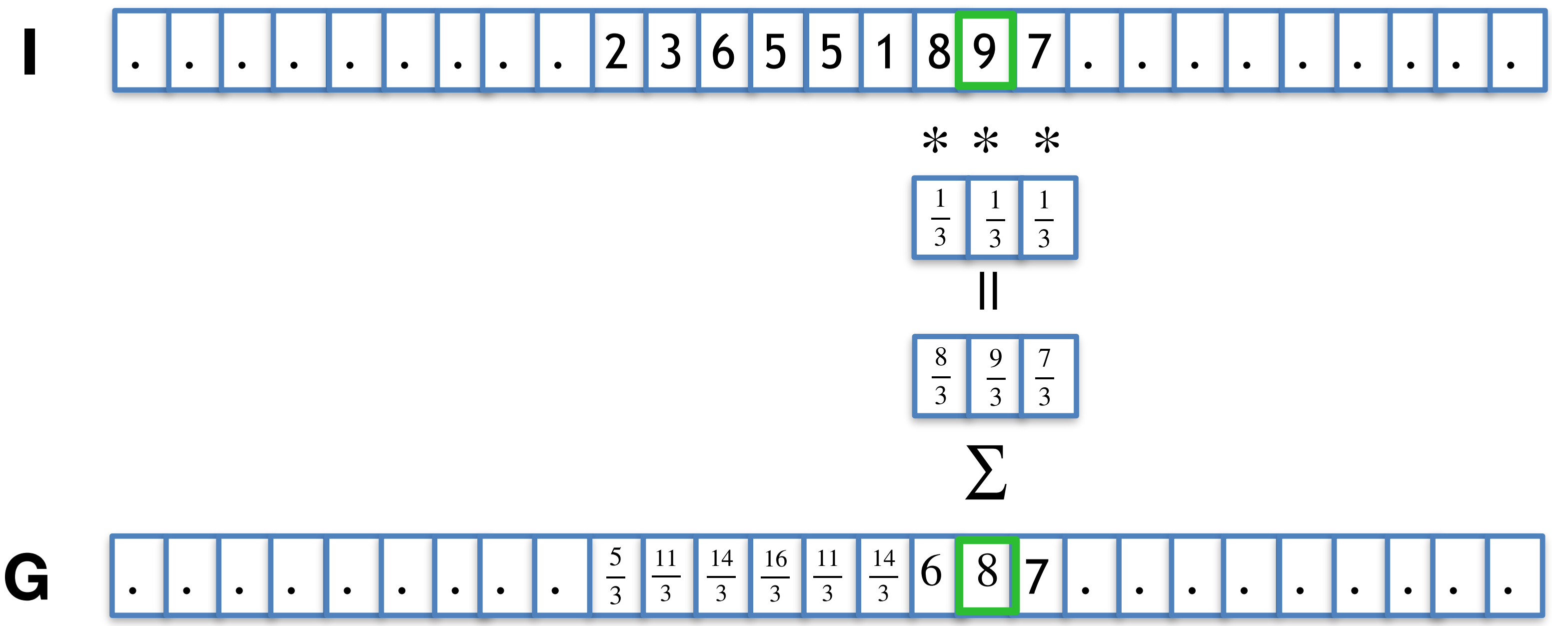
Correlation Example - 1D



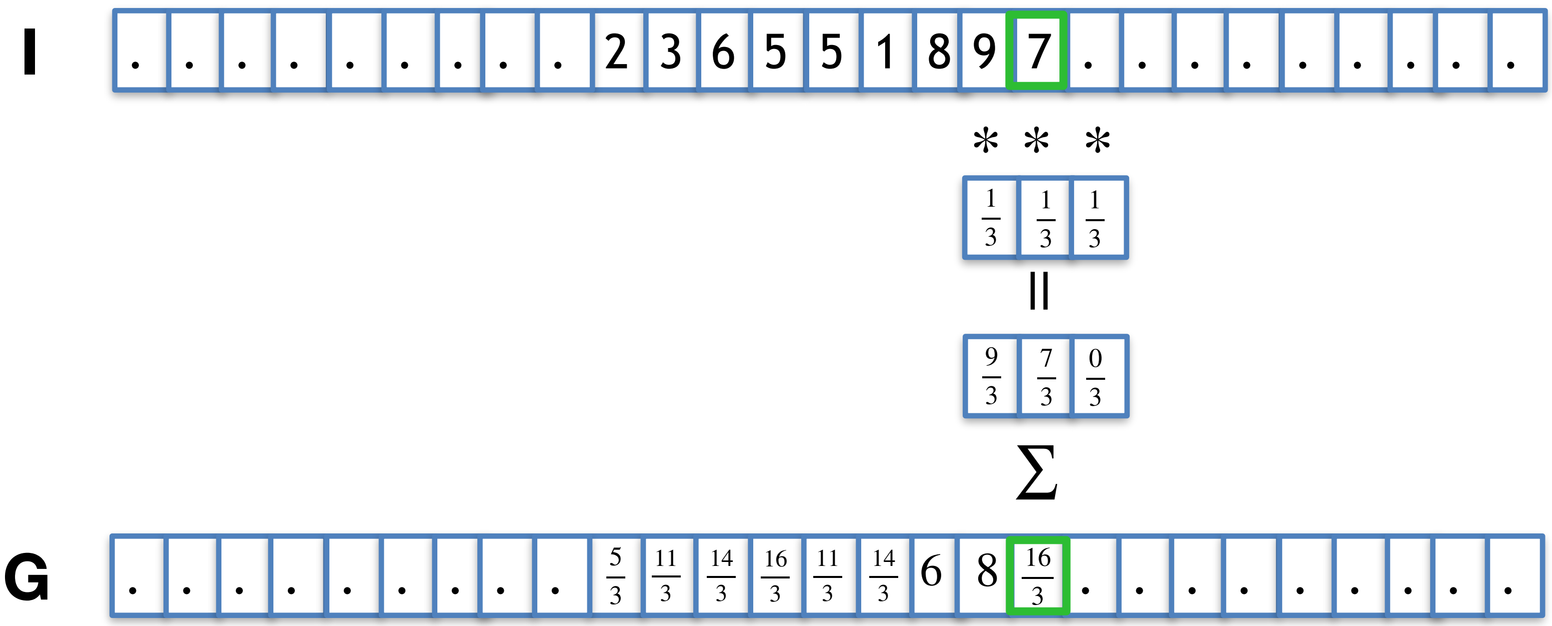
Correlation Example - 1D



Correlation Example - 1D



Correlation Example - 1D



Cross-Correlation and Convolution

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

Cross-Correlation and Convolution

$$2 \times 15 + 1 \times 1 = 31$$

$$\begin{bmatrix} 5 & 15 & 4 & 0 & -1 \\ 10 & 1 & 5 & 1 & 0 \\ 6 & 9 & 11 & 1 & -1 \\ 0 & -1 & 5 & 15 & 4 \\ 1 & 0 & 10 & 1 & 5 \end{bmatrix} \circ \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 31 & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$$

Cross-Correlation and Convolution

$$-2 \times 5 + 2 \times 4 - 1 \times 10 + 5 \times 1 = 7$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7			

Cross-Correlation and Convolution

$$-2 \times 15 - 1 \times 1 + 1 \times 1 = -30$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

∘

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

=

31	-7	-30		

Cross-Correlation and Convolution

$$-2 \times 4 - 1 \times 2 - 5 \times 1 = -15$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

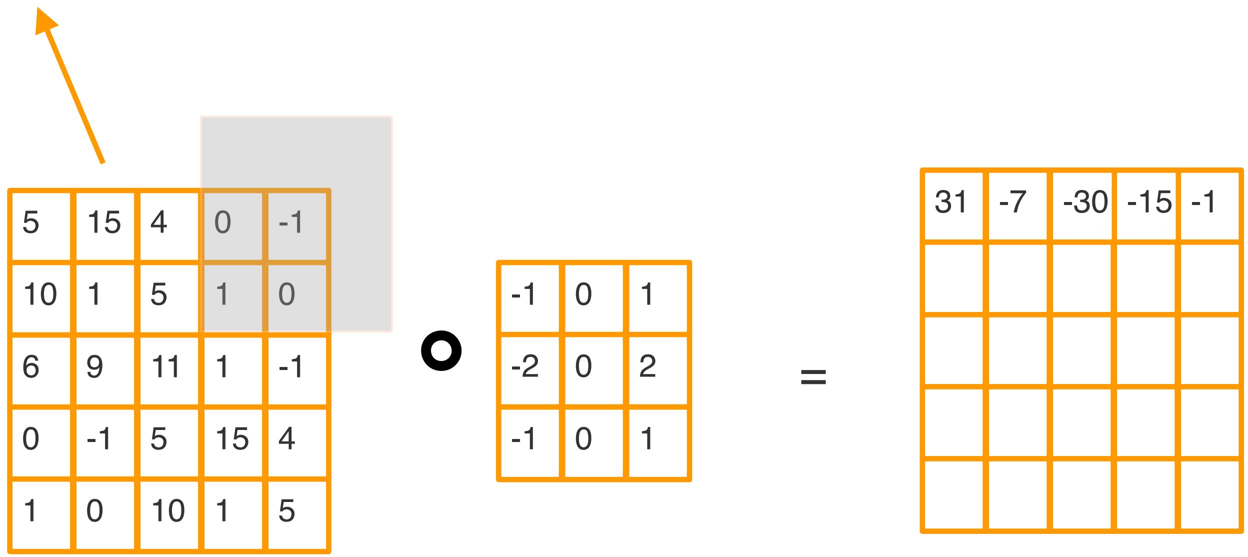
-1	0	1
-2	0	2
-1	0	1

=

31	-7	-30	-15	

Cross-Correlation and Convolution

$$-1 \times 1 = -1$$



Cross-Correlation and Convolution

$$15 \times 1 + 2 \times 1 + 9 \times 1 = 26$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26				

Cross-Correlation and Convolution

$$5 \times (-1) + 10 \times (-2) + 6 \times (-1) + 4 \times 1 + 5 \times 2 + 11 \times 1 = -6$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6			

Cross-Correlation and Convolution

$$15 \times (-1) + 1 \times (-2) + 9 \times (-1) + 0 \times 1 + 1 \times 2 + 1 \times 1 = -23$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23		

Cross-Correlation and Convolution

$$4 \times (-1) + 5 \times (-2) + 11 \times (-1) + (-1) \times 1 + 0 \times 2 + (-1) \times 1 = -27$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	

Cross-Correlation and Convolution

$$0 \times (-1) + 1 \times (-2) + 1 \times (-1) = -3$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3

Cross-Correlation and Convolution

$$1 \times 1 + 9 \times 2 + -1 \times 1 = 18$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18				

Cross-Correlation and Convolution

$$10 \times (-1) + 6 \times (-2) + 5 \times 1 + 11 \times 2 + 5 \times 1 = 10$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

$=$

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10			

Cross-Correlation and Convolution

$$1 \times (-1) + 9 \times (-2) + (-1) \times (-1) + 1 \times 1 + 2 \times 1 + 15 \times 1 = 0$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0		

Cross-Correlation and Convolution

$$5 \times (-1) + 11 \times (-2) + 5 \times (-1) + 0 \times 1 + (-1) \times 2 + 4 \times 1 = -30$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	

Cross-Correlation and Convolution

$$1 \times (-1) + 1 \times (-2) + 15 \times (-1) + 0 \times 1 + 0 \times 2 + 0 \times 1 = -18$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18

Cross-Correlation and Convolution

$$0 \times (-1) + 0 \times (-2) + 0 \times (-1) + 9 \times 1 + (-1) \times 2 + 0 \times 1 = 7$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7				

Cross-Correlation and Convolution

$$6 \times (-1) + 0 \times (-2) + 1 \times (-1) + 11 \times 1 + 5 \times 2 + 10 \times 1 = 24$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24			

Cross-Correlation and Convolution

$$9 \times (-1) + -1 \times (-2) + 0 \times (-1) + 1 \times 1 + 15 \times 2 + 1 \times 1 = 25$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25		

Cross-Correlation and Convolution

$$11 \times (-1) + 5 \times (-2) + 10 \times (-1) + (-1) \times 1 + 4 \times 2 + 5 \times 1 = -19$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25	-19	

Cross-Correlation and Convolution

$$1 \times (-1) + 15 \times (-2) + 1 \times (-1) + 0 \times 1 + 0 \times 2 + 0 \times 1 = -32$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

∘

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25	-19	-32

Cross-Correlation and Convolution

$$0 \times (-1) + 0 \times (-2) + 0 \times (-1) + (-1) \times 1 + 0 \times 2 + 0 \times 1 = -1$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25	-19	-32
-1				

Cross-Correlation and Convolution

$$0 \times (-1) + 1 \times (-2) + 0 \times (-1) + 5 \times 1 + 10 \times 2 + 0 \times 1 = 23$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25	-19	-32
-1	23			

Cross-Correlation and Convolution

$$-1 \times (-1) + 0 \times (-2) + 0 \times (-1) + 15 \times 1 + 01 \times 2 + 0 \times 1 = 16$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25	-19	-32
-1	23	16		

Cross-Correlation and Convolution

$$5 \times (-1) + 10 \times (-2) + 0 \times (-1) + 4 \times 1 + 5 \times 2 + 0 \times 1 = -11$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25	-19	-32
-1	23	16	-11	

Cross-Correlation and Convolution

$$15 \times (-1) + 1 \times (-2) + 0 \times (-1) + 0 \times 1 + 0 \times 2 + 0 \times 1 = -17$$

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$\circ$

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

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25	-19	-32
-1	23	16	-11	-17

Cross-Correlation and Convolution

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

Image, I

$\circ$

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

Filter/template

=

31	-7	-30	-15	-1
26	-6	-23	-27	-3
18	10	0	-30	-18
7	24	25	-19	-32
-1	23	16	-11	-17

Output image

Cross-Correlation - Mathematically

$1D$

$$G = F \circ I[i] = \sum_{u=-k}^{k} F[u] I[i+u]$$ F has $2k+1$ elements

Box filter $F[u] = \frac{1}{3}$ for $u = -1,0,1$ and 0 otherwise

Cross-correlation filtering - 2D

Let's write this down as an equation. Assume the averaging window is (2k+1)x(2k+1):

$$G[i,j] = \frac{1}{(2k+1)^2} \sum_{u=-k}^{k} \sum_{v=-k}^{k} F[i+u, j+v]$$

We can generalize this idea by allowing different weights for different neighboring pixels:

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

This is called a **cross-correlation** operation and written:

$$G = F \circ I$$

F is called the "filter," "kernel," or "mask."

Convolution

Filter is flipped before correlating

$1D$ $\quad F$ has $2k+1$ elements

$$G = F * I[i] = \sum_{u=-k}^{k} F[u] I[i-u]$$

Box filter $F[u] = \dfrac{1}{3}$ for $u = -1,0,1$ and 0 otherwise

for example, convolution of 1D image with the filter [3,5,2]

is exactly the same as correlation with the filter [2,5,3]

Convolution filtering - 2D

For 2D the filter is flipped and rotated

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

Correlation and convolution are identical for symmetrical filters

Convolution with the filter

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

is the same as Correlation with the filter

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