Linear Image Filtering

Introduction

We have a fairly good idea of what happens at the initial stages of human visual processing

It is similar to some of the filtering we will discuss

We will also talk about

1. Extracting low-level features (tokens)
2. How to enhance image structures
3. Remove variability

Introduction

1. **first stage** in most computer vision algorithms, namely the use of **image processing** to preprocess the image and convert it into a form **suitable for further analysis**
    a. exposure correction and
    b. color balancing,
    c. reducing image noise,
    d. increasing sharpness, or
    e. straightening the image by rotating it

Signals and Images

1. "**A signal** is a measurement of some physical quantity (light, sound, height, temperature, etc.) as a function of another independent quantity (time, space, wavelength, etc.)."
2. "**A system** is a process/function that transforms a signal into another."
3. Continuous signals are sampled for processing by computers
4. The sampling equation: $\ell[n] = \ell[n\Delta T]$

Signals and Images

1. $\ell \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}}$, height x width

$$\ell = \begin{bmatrix}
160 & 175 & 171 & 168 & 168 & 172 & 164 & 158 & 167 & 173 & 167 & 163 & 162 & 164 & 160 & 159 & 163 & 162 \\
149 & 164 & 172 & 175 & 178 & 179 & 176 & 118 & 97 & 168 & 175 & 171 & 169 & 175 & 176 & 177 & 165 & 152 \\
161 & 166 & 182 & 171 & 170 & 177 & 175 & 116 & 109 & 169 & 177 & 173 & 168 & 175 & 175 & 159 & 153 & 123 \\
171 & 174 & 177 & 175 & 167 & 161 & 157 & 138 & 103 & 112 & 157 & 164 & 159 & 160 & 165 & 169 & 148 & 144 \\
163 & 163 & 162 & 165 & 167 & 164 & 178 & 167 & 77 & 55 & 134 & 170 & 167 & 162 & 164 & 175 & 168 & 160 \\
173 & 164 & 158 & 165 & 180 & 180 & 150 & 89 & 61 & 34 & 137 & 186 & 186 & 182 & 175 & 165 & 160 & 164 \\
152 & 155 & 146 & 147 & 169 & 180 & 163 & 51 & 24 & 32 & 119 & 163 & 175 & 182 & 181 & 162 & 148 & 153 \\
134 & 135 & 147 & 149 & 150 & 147 & 148 & 62 & 36 & 46 & 114 & 157 & 163 & 167 & 169 & 163 & 146 & 147 \\
135 & 132 & 131 & 125 & 115 & 129 & 132 & 74 & 54 & 41 & 104 & 156 & 152 & 156 & 164 & 156 & 141 & 144 \\
151 & 155 & 151 & 145 & 144 & 149 & 143 & 71 & 31 & 29 & 129 & 164 & 157 & 155 & 159 & 158 & 156 & 148 \\
172 & 174 & 178 & 177 & 177 & 181 & 174 & 54 & 21 & 29 & 136 & 190 & 180 & 179 & 176 & 184 & 187 & 182 \\
177 & 178 & 176 & 173 & 174 & 180 & 150 & 27 & 101 & 94 & 74 & 189 & 188 & 186 & 183 & 186 & 188 & 187 \\
160 & 160 & 163 & 163 & 161 & 167 & 100 & 45 & 169 & 166 & 59 & 136 & 184 & 176 & 175 & 177 & 185 & 186 \\
147 & 150 & 153 & 155 & 160 & 155 & 56 & 111 & 182 & 180 & 104 & 84 & 168 & 172 & 171 & 164 & 168 & 167 \\
184 & 182 & 178 & 175 & 179 & 133 & 86 & 191 & 201 & 204 & 191 & 79 & 172 & 220 & 217 & 205 & 209 & 200 \\
184 & 187 & 192 & 182 & 124 & 32 & 109 & 168 & 171 & 167 & 163 & 51 & 105 & 203 & 209 & 203 & 210 & 205 \\
191 & 198 & 203 & 197 & 175 & 149 & 169 & 189 & 190 & 173 & 160 & 145 & 156 & 202 & 199 & 201 & 205 & 202 \\
153 & 149 & 153 & 155 & 173 & 182 & 179 & 177 & 182 & 177 & 182 & 185 & 179 & 177 & 167 & 176 & 182 & 180
\end{bmatrix}$$

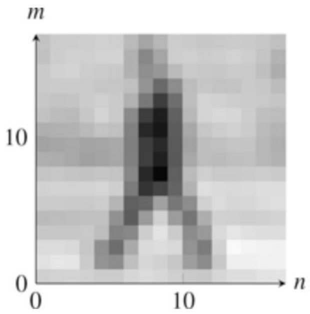
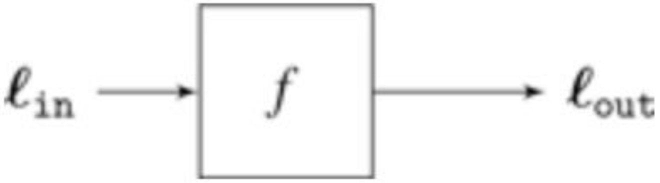


Figure 15.2: Grayscale image showing a person walking in the street. This tiny image has only 18 × 18 pixels.

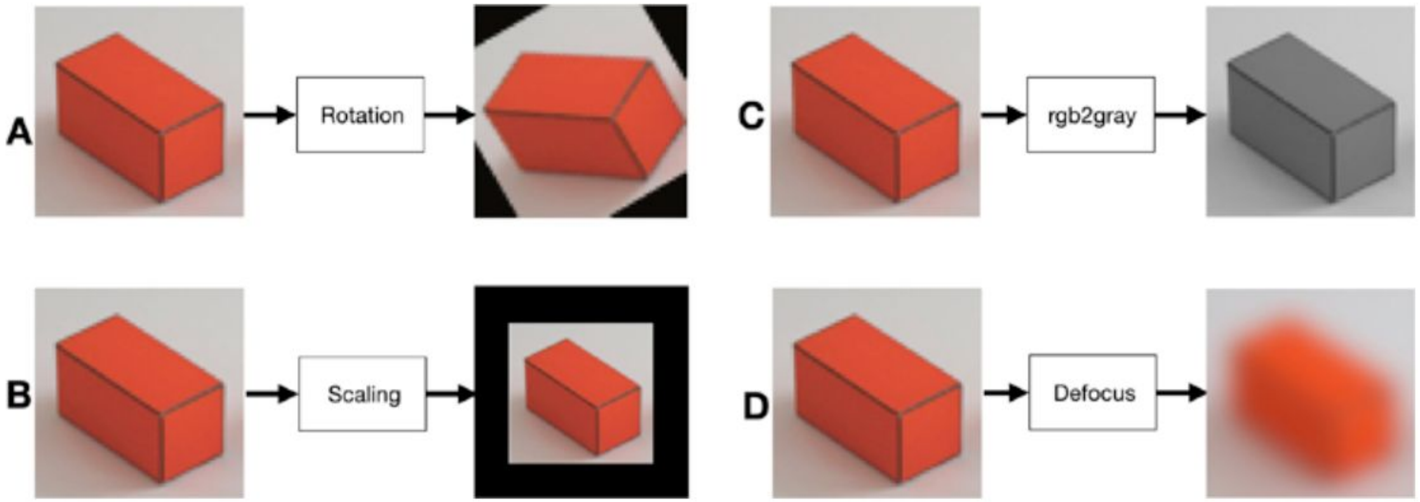
Linear Systems



1. Linear systems are capable of creating very interesting image transformations
2. "A function *f* is linear is it satisfies the following two properties:"
   a. $f(\ell_1+\ell_2) = f(\ell_1)+f(\ell_2)$
   b. $f(a\ell) = af(\ell)$ *for any scalar a*

Linear Systems

1. Which of the following are linear transformations?



Linear Systems

1. General form of a 1D linear system

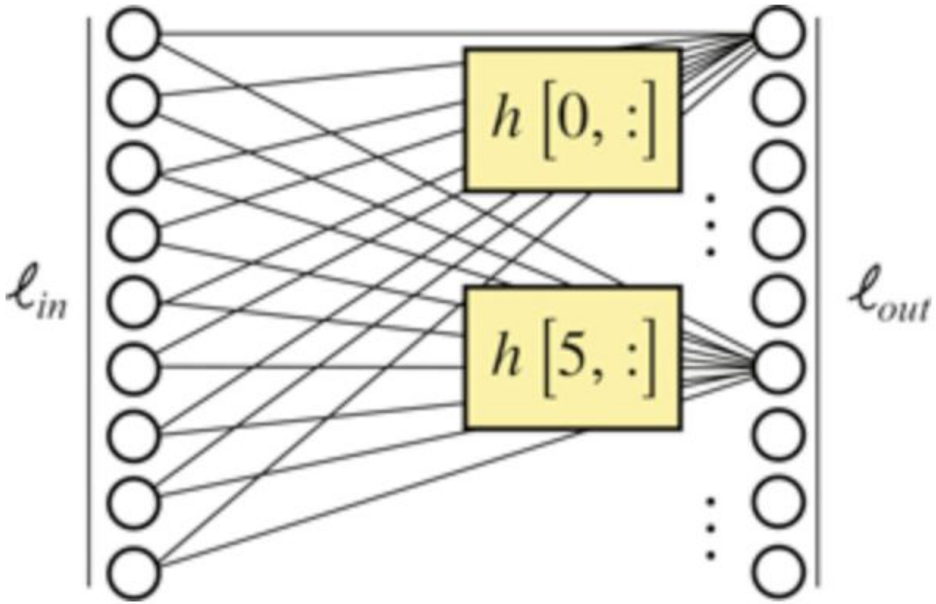
$$\ell_{\text{out}}[n] = \sum_{k=0}^{N-1} h[n,k]\,\ell_{\text{in}}[k] \quad \textit{for} \quad n \in [0, M-1]$$

$$\begin{bmatrix} \ell_{\text{out}}[0] \\ \ell_{\text{out}}[1] \\ \vdots \\ \ell_{\text{out}}[n] \\ \vdots \\ \ell_{\text{out}}[M-1] \end{bmatrix} = \begin{bmatrix} h[0,0] & h[0,1] & \ldots & h[0,N-1] \\ h[1,0] & h[1,1] & \ldots & h[1,N-1] \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & h[n,k] & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ h[M-1,0] & h[M-1,1] & \ldots & h[M-1,N-1] \end{bmatrix} \begin{bmatrix} \ell_{\text{in}}[0] \\ \ell_{\text{in}}[1] \\ \vdots \\ \ell_{\text{in}}[k] \\ \vdots \\ \ell_{\text{in}}[N-1] \end{bmatrix}$$

$$\boldsymbol{\ell}_{\text{out}} = \mathbf{H}\boldsymbol{\ell}_{\text{in}}$$

Linear Systems

1. General form of a 1D linear system



Linear Systems

- General form of a 2D linear system

$$\ell_{\text{out}}[n,m] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h[n,m,k,l]\, \ell_{\text{in}}[k,l]$$

- If we convert 2D images into a long 1D column vector, we can write

$$\boldsymbol{\ell}_{\text{out}} = \mathbf{H}\boldsymbol{\ell}_{\text{in}}$$

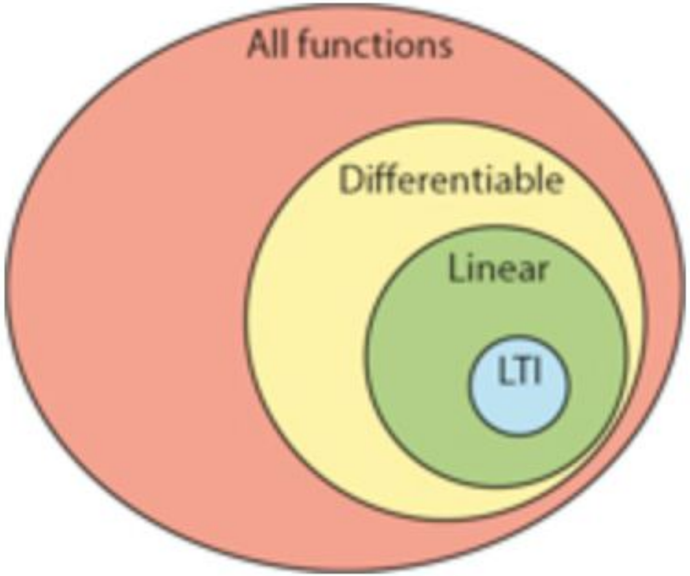
“Linear Translation Invariant Systems”

1. Why LTI?
   i. because we want to process the image in a spatially invariant manner
2. “A system is an LTI system if it is linear and when we translate the input signal by n_0, m_0, then output is also translated by n_0, m_0:”

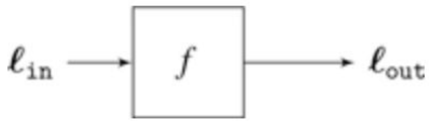
$$\ell_{\text{out}}[n-n_0, m-m_0] = f\left(\ell_{\text{in}}[n-n_0, m-m_0]\right)$$

“Linear Translation Invariant Systems”

1.



Analyzing LTI systems



1. LTI systems transforms input signal into output signal
2. We want to understand how the system is carrying out the transformation
3. We want to determine how a system reacts to different inputs, like in electrical circuits and mechanical systems.

Analyzing LTI systems

1. Every LTI system has a unique "impulse response," $h(t)$.
2. The impulse response help us analyze the system
3. Imagine a basic circuit with a resistor (R) and a capacitor (C). This is an LTI system.
    a. Impulse Response: The impulse response h(t) of this circuit is an exponential decay.

Convolution

1. Every LTI transformation can be represented by convolution.
2. All convolutions are LTI transformations
3. Convolution has widespread applications in engineering, physics, and computer science.

Applications of Convolution

1. Convolution is a fundamental tool in system analysis because **it helps us determine how an input signal transforms into an output**.
2. The system in question is doing convolution to the input signal and giving us output.
3. We can design our own system and convolve it with an input signal to modify it in a controlled way.
4. In signal processing, convolution is used to filter signals, remove noise, and enhance features.
5. The idea is to apply a convolution filter (also called an **impulse response**) to modify the signal in a controlled way.

Convolution

1. *Convolution between two continuous functions f and g*

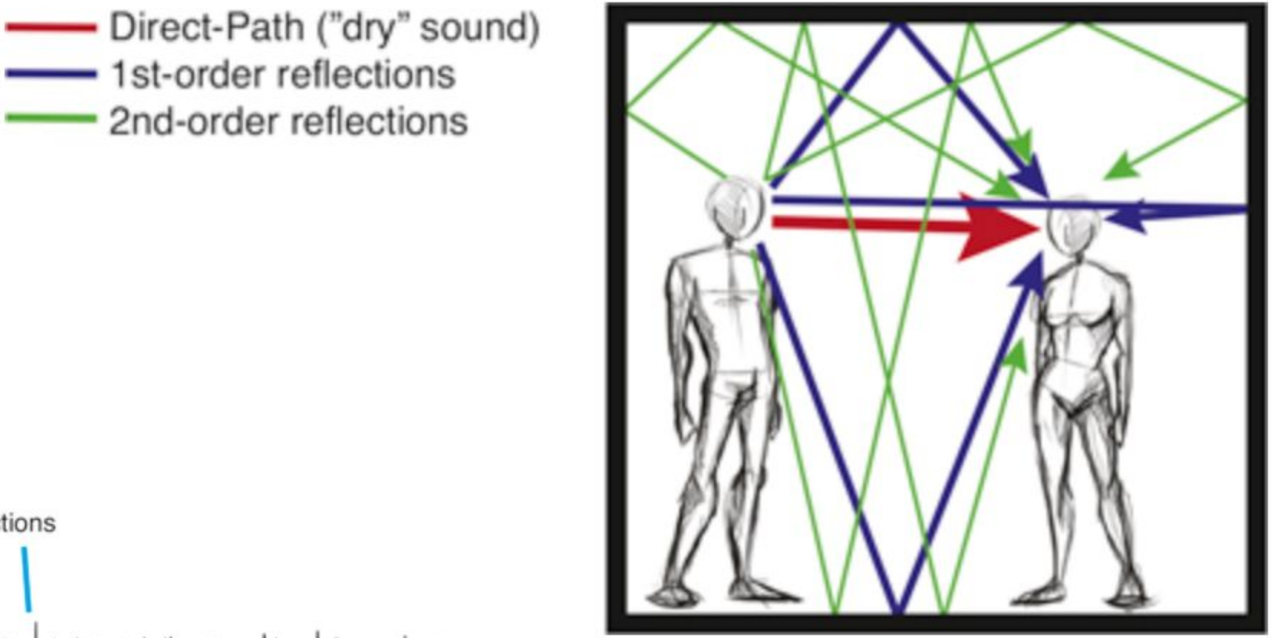
$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t - \tau)\, d\tau$$

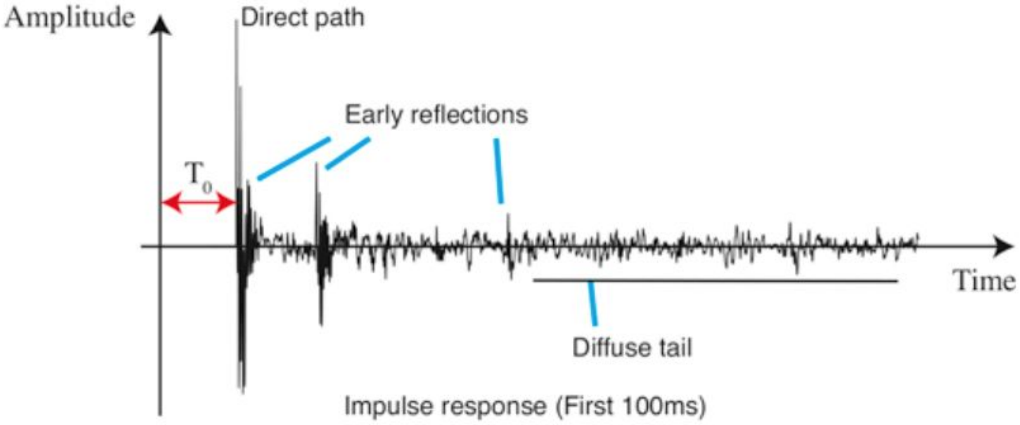
Convolution

1. Practice convolution of continuous functions
2. Convolution demo
    a. https://phiresky.github.io/convolution-demo/

System Identification - Acoustics

1.

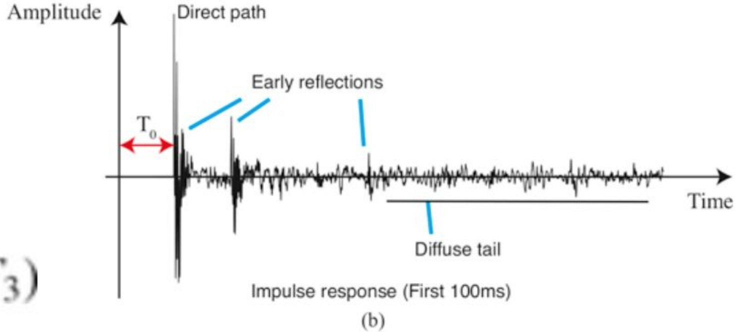




(b)

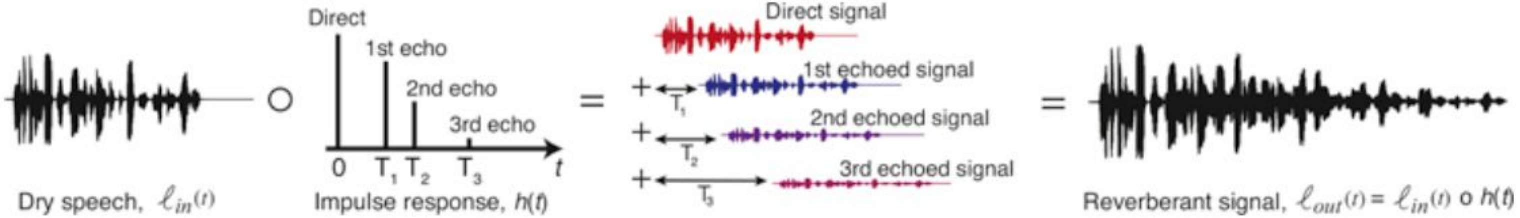
System Identification - Acoustics

1.



$$h(t) = a_0\delta(t) + a_1\delta(t - T_1) + a_2\delta(t - T_2) + a_3\delta(t - T_3)$$

$$\ell_{\mathrm{out}}(t) = \ell_{\mathrm{in}}(t) \circ h(t) = a_0\ell_{in}(t) + a_1\ell_{\mathrm{in}}(t - T_1) + a_2\ell_{\mathrm{in}}(t - T_2) + a_3\ell_{\mathrm{in}}(t - T_3)$$



Noise Cancellation

1. How can you cancel the noise when you have impulse response of a place?
2. Using Adaptive Filter
3. Taking a sample of the signal from the far end (before it goes to the speaker).
4. Passing this signal through its own internal filter. This filter has adjustable parameters (called coefficients or weights).
5. Comparing the output of its filter to the actual echo that's being picked up by the microphone.

Hack Audio: Convolution Function

1. Convolving audio with impulse response of a system that introduces echo
2. Convolving audio with impulse response of a system that introduces stereo effect
3. https://www.youtube.com/watch?v=NVSL9BcDKac

Convolution - Discrete

1. *“The convolution, denoted ○, between a signal ℓin [n] and the convolutional kernel h [n] is defined as follows”*

$$\ell_{\mathtt{out}}[n] = h[n] \circ \ell_{\mathtt{in}}[n] = \sum_{k=-\infty}^{\infty} h[n-k]\,\ell_{\mathtt{in}}[k]$$

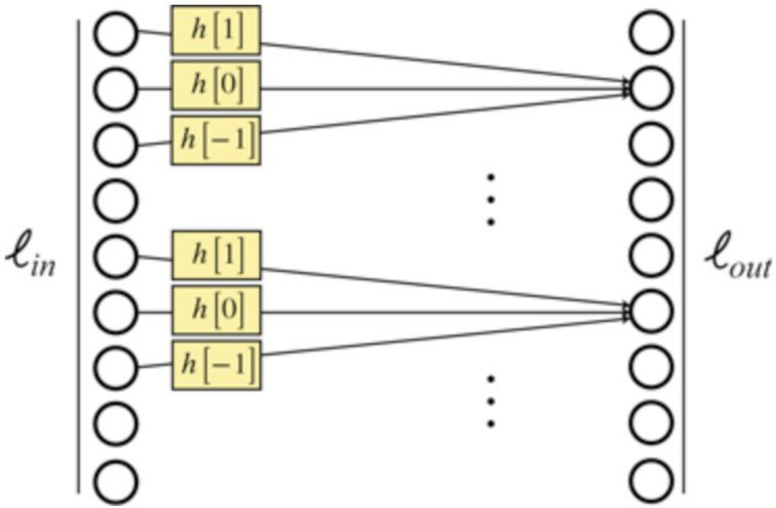
Convolution - Discrete

1. “The convolution operation can be described in words as:
   a. first take the kernel $h[k]$ and mirror it i.e. $h[-k]$,
   b. then shift the mirrored kernel so that the origin is at location n,
   c. then multiply the input values around location n by the mirrored kernel and sum the result.
   d. Store the result in the output vector $\ell_{out}[n]$.”

Convolution - Discrete

1.

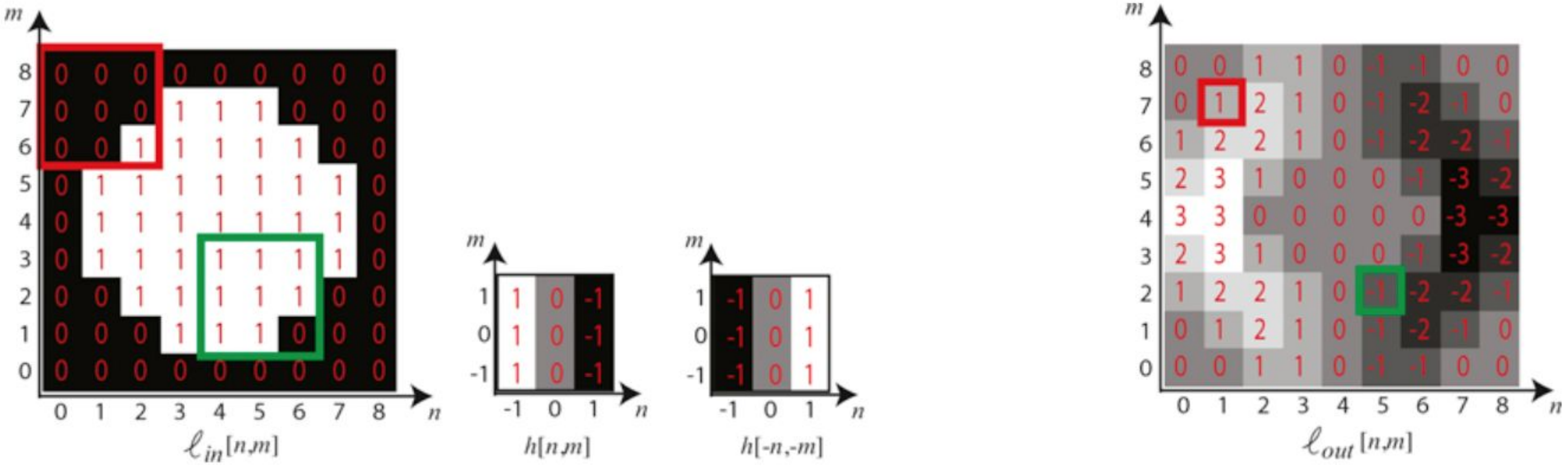
$$\begin{bmatrix} \ell_{\text{out}}[0] \\ \ell_{\text{out}}[1] \\ \ell_{\text{out}}[2] \\ \vdots \\ \ell_{\text{out}}[N-1] \end{bmatrix} = \begin{bmatrix} h[0] & h[-1] & 0 & \dots & 0 \\ h[1] & h[0] & h[-1] & \dots & 0 \\ 0 & h[1] & h[0] & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & \dots & h[0] \end{bmatrix} \begin{bmatrix} \ell_{\text{in}}[0] \\ \ell_{\text{in}}[1] \\ \ell_{\text{in}}[2] \\ \vdots \\ \ell_{\text{in}}[N-1] \end{bmatrix} \tag{15.13}$$



Convolution - Two dimensional

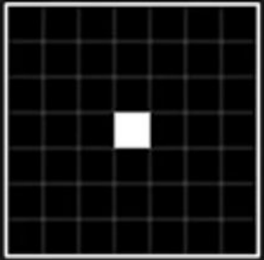
1.

$$\ell_{\text{out}}[n,m] = h[n,m] \circ \ell_{\text{in}}[n,m] = \sum_{k,l} h[n-k, m-l]\, \ell_{\text{in}}[k,l]$$



Input

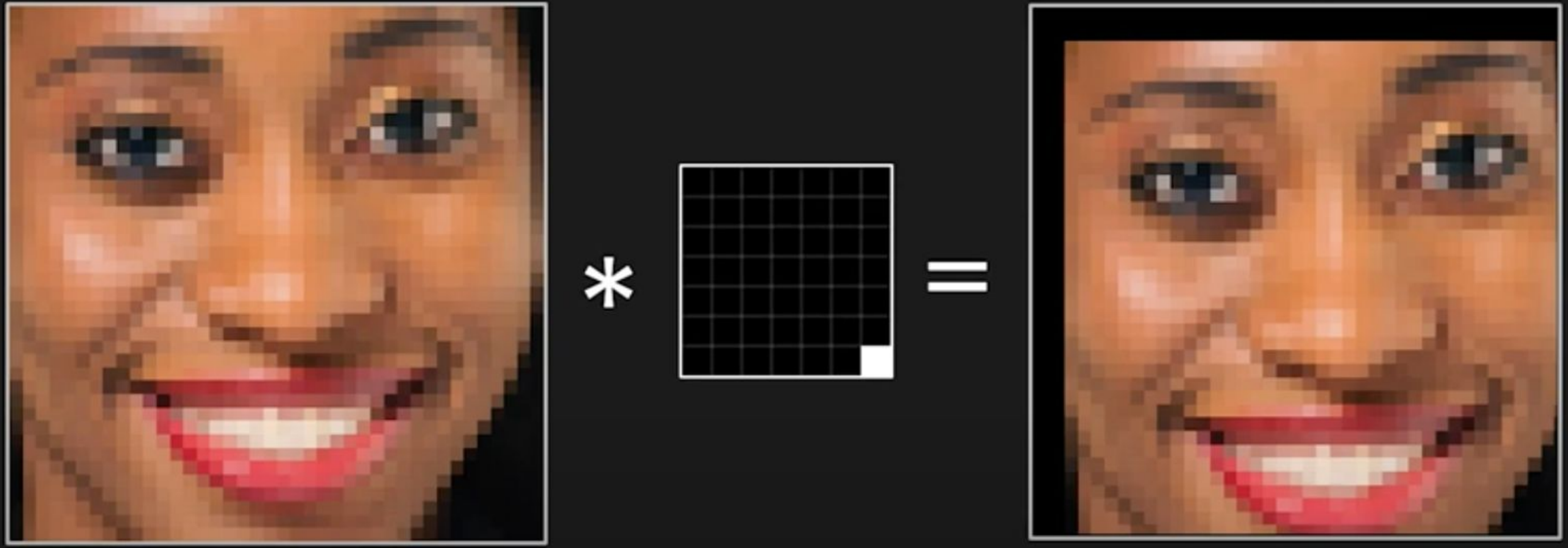
$$f(x,y) \quad * \quad \delta(x,y)$$



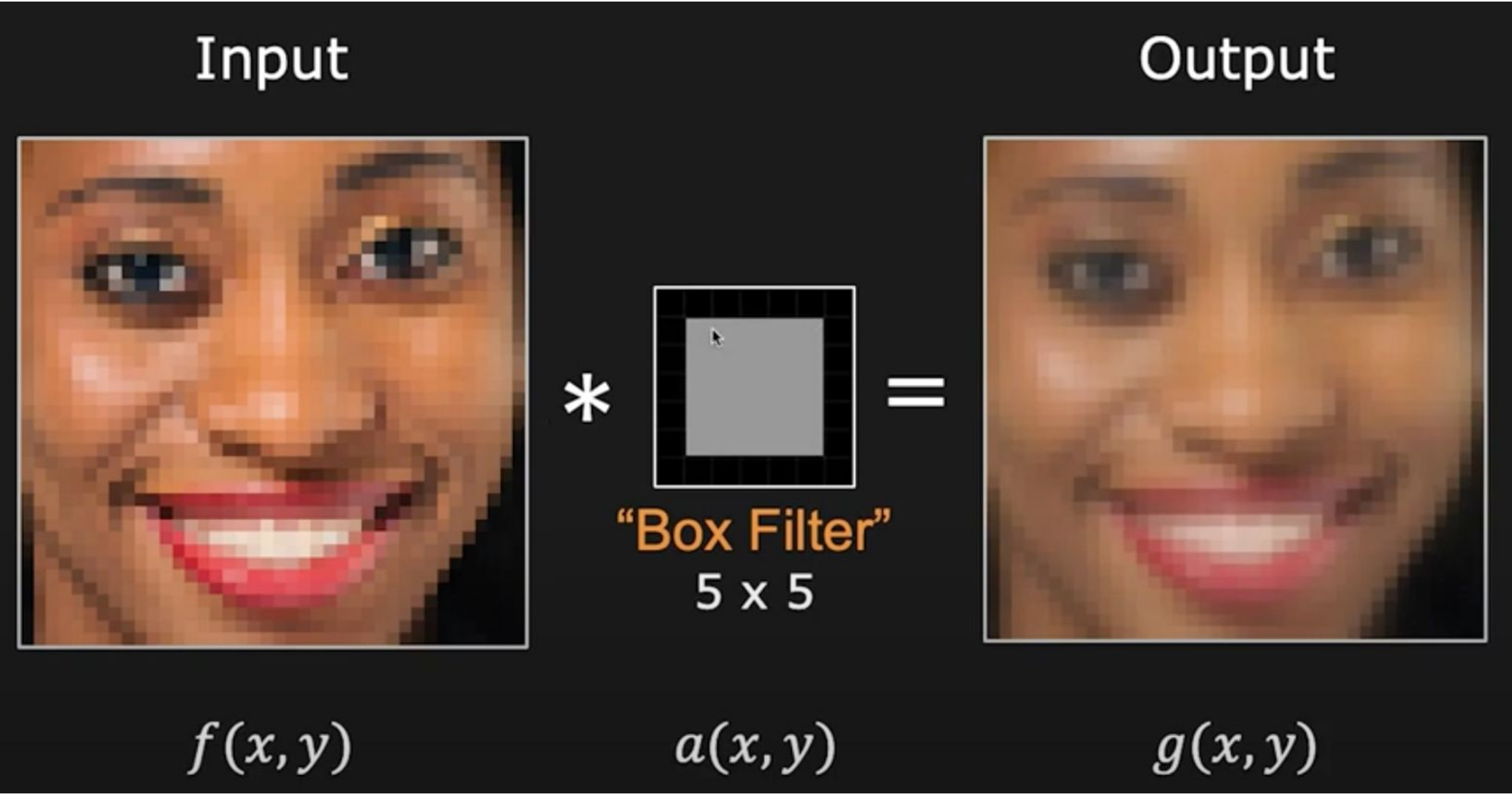
$$= \quad f(x,y)$$

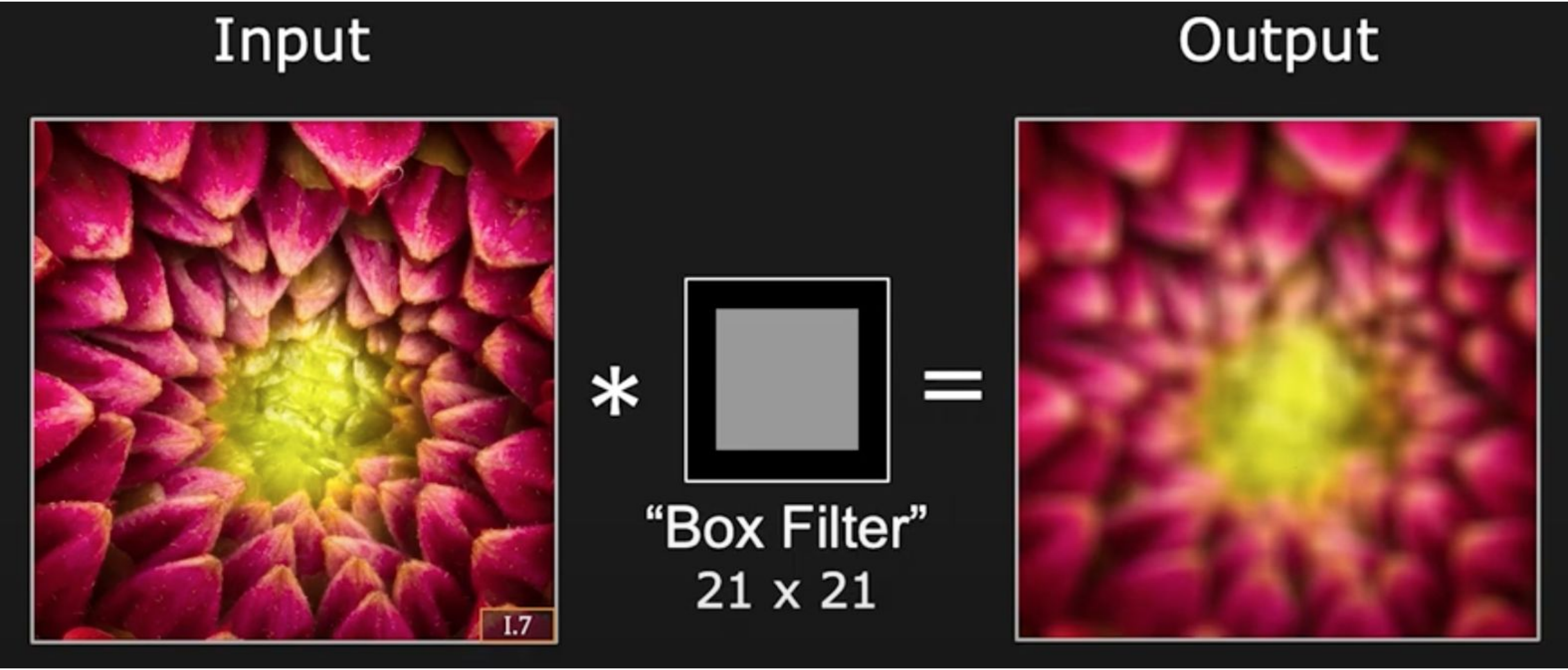
Output

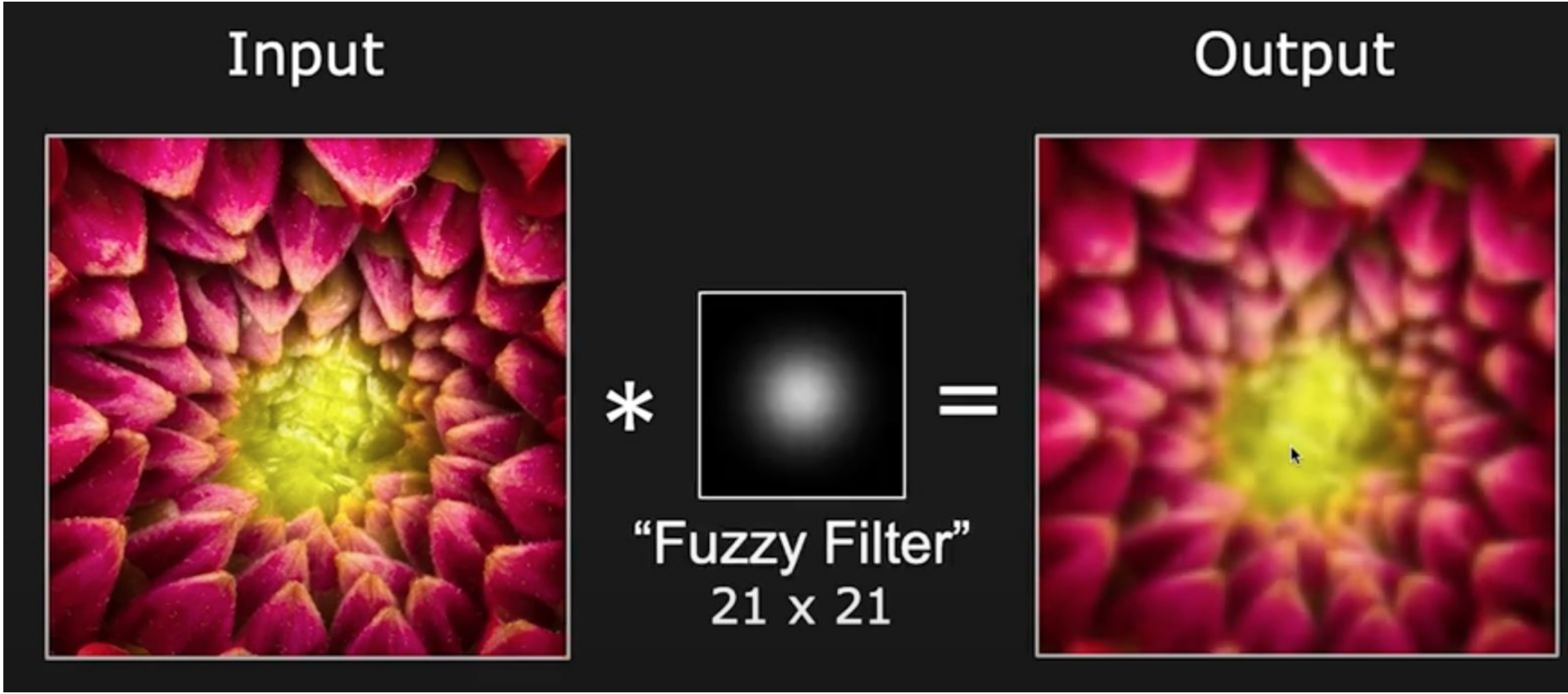
Input | Output



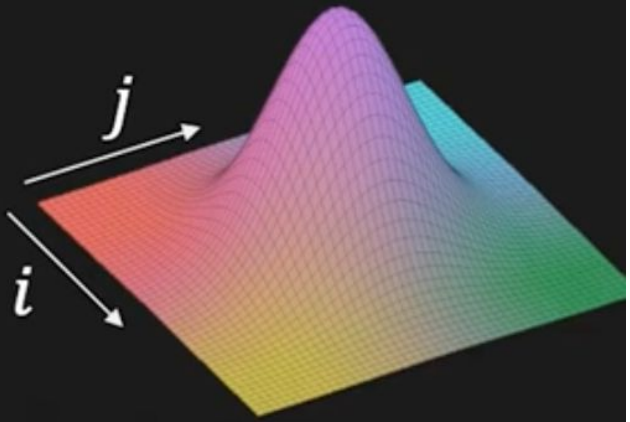
$f(x,y) * \delta(x-u, y-v) = f(x-u, y-v)$


Input
Output
*
=
"Box Filter"
5 x 5
$f(x,y)$
$a(x,y)$
$g(x,y)$


Input
Output
I.7
“Box Filter”
21 x 21

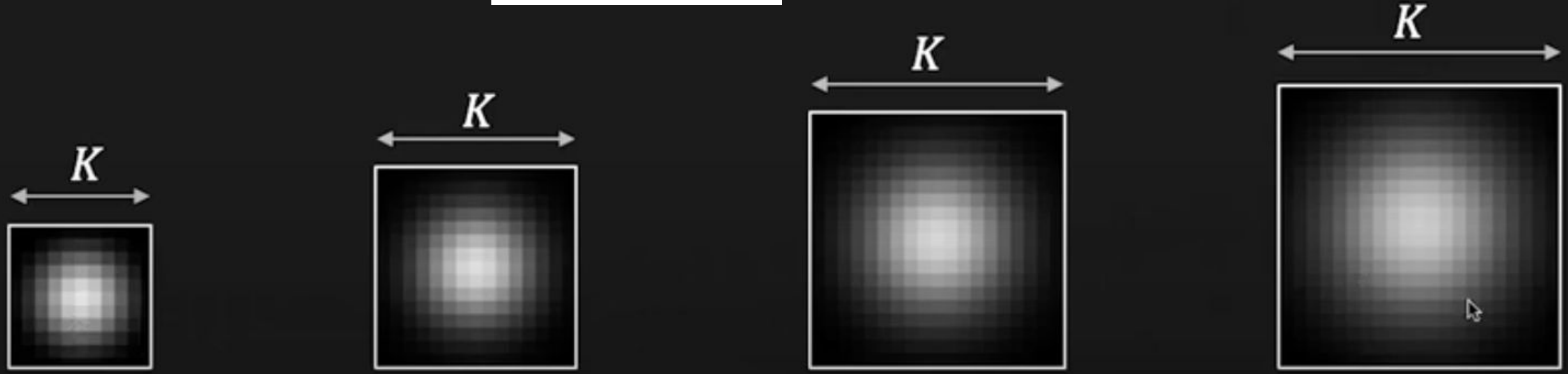

Input
Output
*
=
“Fuzzy Filter”
21 x 21

$$n_\sigma[i,j] = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\left(\frac{i^2+j^2}{\sigma^2}\right)}$$



σ^2: Variance

$$\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$



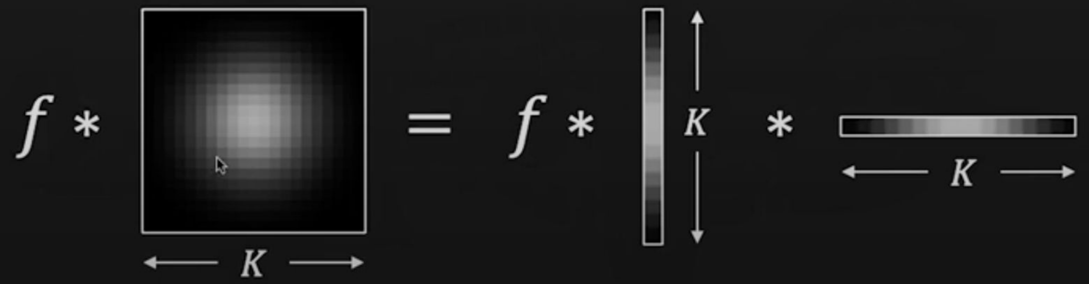
$\sigma = 2$ $\sigma = 3$ $\sigma = 4$ $\sigma = 5$

Gaussian Smoothing is Separable

$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)} f[i-m, j-n]$$

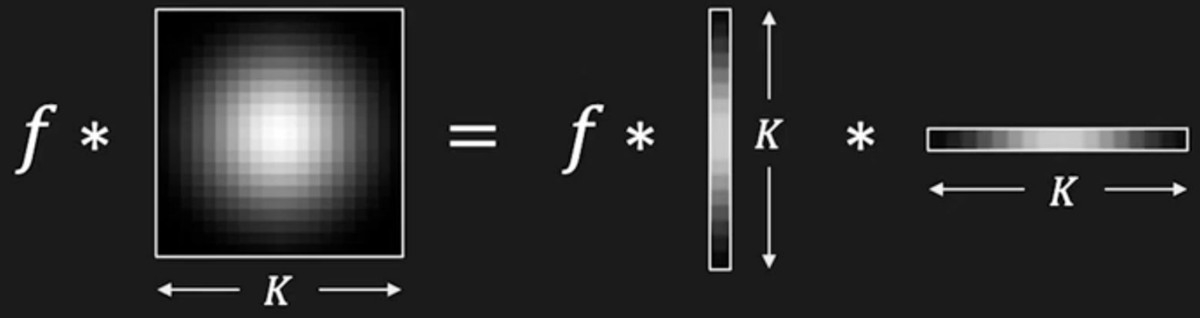
$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2}{\sigma^2}\right)} . \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{n^2}{\sigma^2}\right)} f[i-m, j-n]$$

Using One 2D Gaussian Filter ≡ Using Two 1D Gaussian Filters



Gaussian Smoothing is Separable

Using One 2D Gaussian Filter ≡ Using Two 1D Gaussian Filters

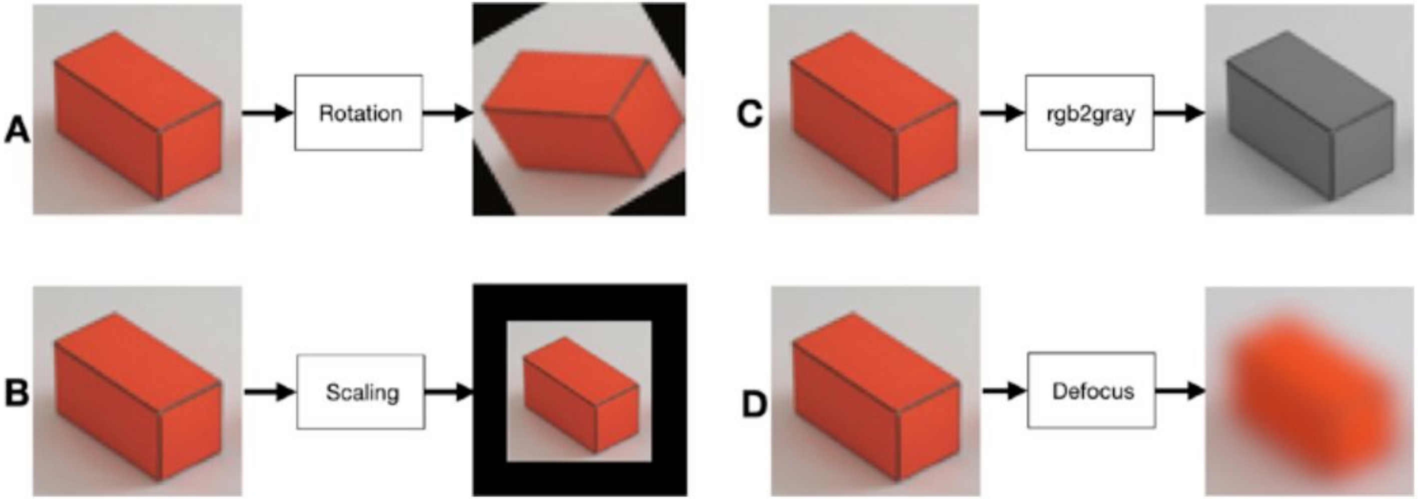


Which one is faster? Why?

K^2 Multiplications	$2K$ Multiplications
$K^2 - 1$ Additions	$2(K-1)$ Additions

Linear Systems

1. Which of the following are LTI transformations?

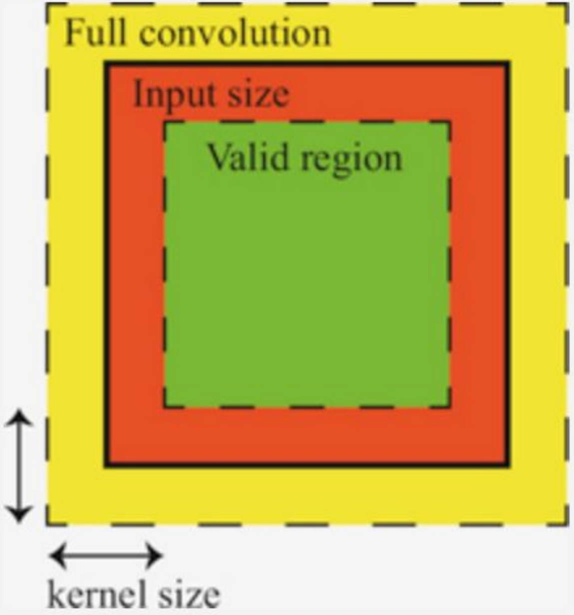


Properties of Convolution

1. Commutative $$h[n] \circ \ell[n] = \ell[n] \circ h[n]$$
2. Associative $$\ell_1[n] \circ \ell_2[n] \circ \ell_3[n] = \ell_1[n] \circ (\ell_2[n] \circ \ell_3[n]) = (\ell_1[n] \circ \ell_2[n]) \circ \ell_3[n]$$
3. Distributive $$\ell_1[n] \circ (\ell_2[n] + \ell_3[n]) = \ell_1[n] \circ \ell_2[n] + \ell_1[n] \circ \ell_3[n]$$
4. Shift $$\ell_{\text{out}}[n - n_0] = h[n] \circ \ell_{\text{in}}[n - n_0]$$
5. Identity $$\delta[n] \circ \ell[n] = \ell[n]$$ $$\delta[n] = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0 \end{cases}$$

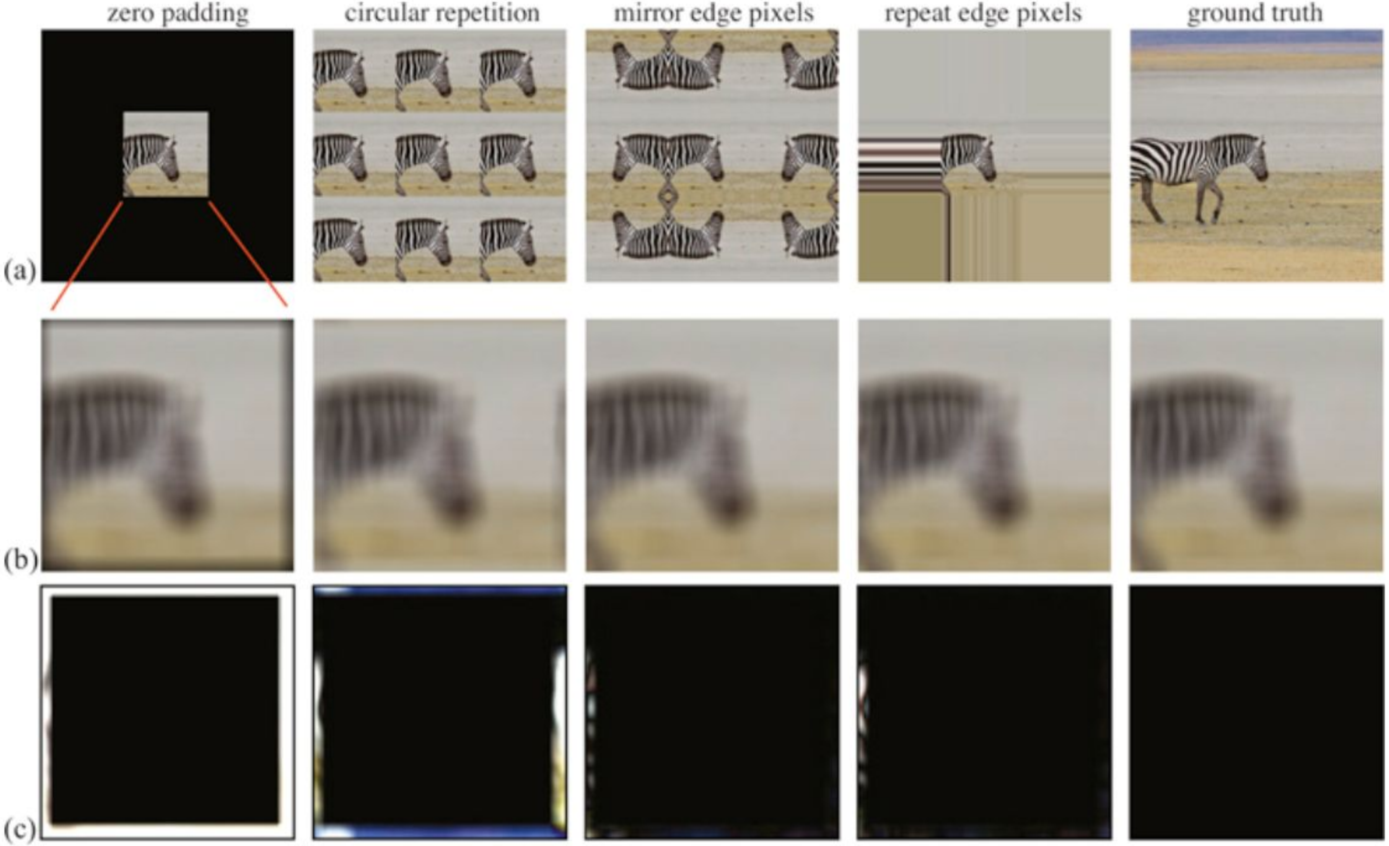
Handling Boundary Conditions

1. Only the green region contains values that can be computed, the rest will be affected by how we decide to handle the boundaries:



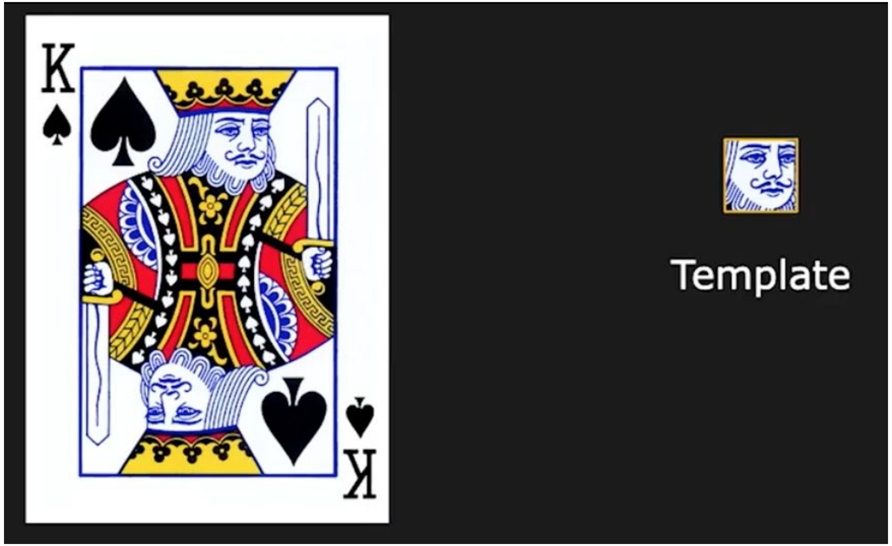
Handling Boundary Conditions

1.



Template Matching

1. How do we match a template with image?



2. Minimize

$$E[n,m] = \sum_{k,l} (\ell_{in}[n+k, m+l] - T[k,l])^2$$

Cross-Correlation vs Convolution

1. Cross-Correlation

$$\ell_{\text{out}}[n,m] = \ell_{\text{in}} \star h = \sum_{k,l=-N}^{N} \ell_{\text{in}}[n+k, m+l]\, h[k,l]$$

   a. Not Commutative

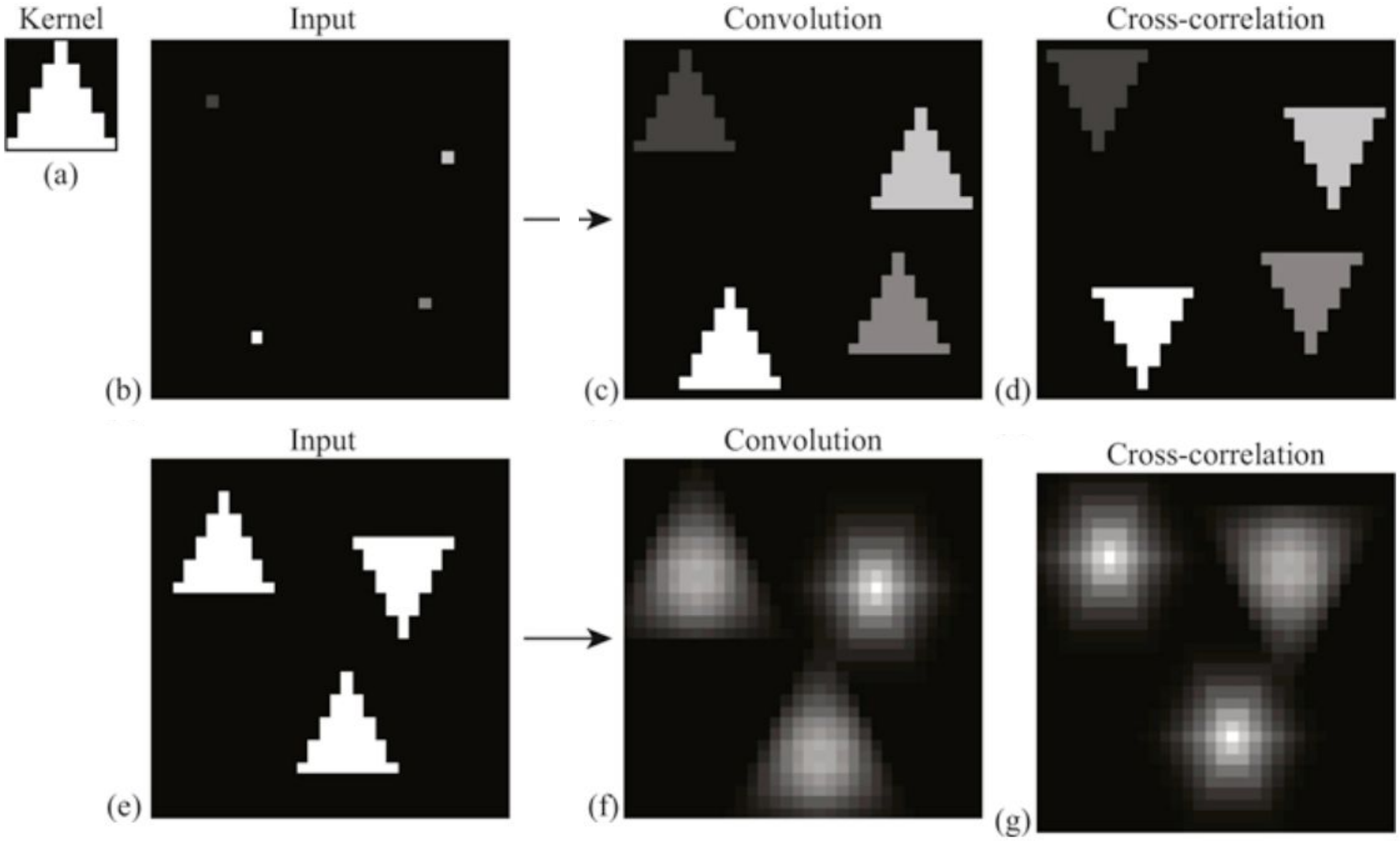
   b. Not Associative

2. Convolution

$$\ell_{\text{out}}[n,m] = \ell_{\text{in}} \circ h = \sum_{k,l=-N}^{N} \ell_{\text{in}}[n-k, m-l]\, h[k,l]$$

“cross-correlation and convolution outputs are identical when the kernel h has central symmetry.”

Cross-Correlation vs Convolution

1. 

Template Matching and Normalized Correlation

1. "Normalized correlation will detect the object independently of location, but it will not be robust to any other changes such as rotation, scaling, and changes in appearance."

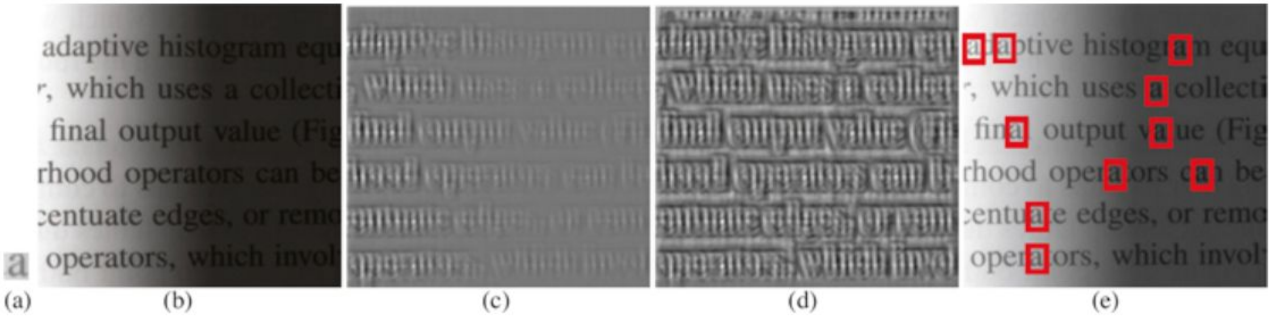


Figure 15.13: (a) Template. (b) Input image. (c) Correlation between input (b) and template (a). (d) Normalized correlation. e) Locations with cross-correlation above 75 percent of its maximum value.

References

1. Foundations of Computer Vision - Chapter 15
2. https://www.iaincollings.com/signals-and-systems#h.2uortdygbgqy
3. https://phiresky.github.io/convolution-demo/
4. https://www.youtube.com/watch?v=NVSL9BcDKac
5. Columbia University https://fpcv.cs.columbia.edu