Linear shift-invariant filtering.
Box filter.
Seperable filters
Conv and correlation
↪ ödeve bir daha bak !

image filtering

image transformation
→ filtering : changes pixel values
→ warping : changes pixel locations

image filtering
→ filtering
→ point processing
- invert $(255-x)$
- darken $(x-128)$
- lower contrast $(x/2)$
- non-linear raise contrast $\left(\left(\frac{x}{255}\right)^2 \cdot 255\right)$

Correlation

$$f \otimes h = \sum_k \sum_l \underbrace{f(k,l)}_{\text{image}} \cdot \underbrace{h(i+k,\, j+l)}_{\text{kernel}}$$

Convolution

$$f * h = \sum_k \sum_l \underbrace{f(k,l)}_{\text{image}} \cdot \underbrace{h(i-k,\, j-l)}_{\text{kernel}}$$

↪ we flip the kernel in x and y axes and then apply correlation.

Gaussian Filter → smoothing
→ low-pass filtering.

$$g(x) = e^{\frac{-x^2}{2\sigma^2}}$$

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

↪ mask values for gaussian filter

- most common natural model
- it has infinite number of derivatives
- Fourrier of a Gaussian is Gaussian.
  ↪ if we want to look at image in frequency level.
- Convolution of Gaussian with itself is Gaussian.
  ↪ if both image and the kernel is gaussian for ex.

- As σ increases more pixels are involved in avg.
- As σ increases image is more blurred
- As σ increases noise is more effectively suppressed.

kernel ⇒ $\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}_{g} = \underbrace{\begin{bmatrix}1\\2\\1\end{bmatrix} * \begin{bmatrix}1&2&1\end{bmatrix}}_{\text{seperable}}$ ♡

Laplace Filter → second derivative filter

1D derivative filter ⇒ $f'(x) = \lim_{h\to 0} \frac{f(x+0.5h) - f(x-0.5h)}{h}$

Laplace filter ⇒ $f''(x) = \lim_{h\to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$

↪ $\begin{bmatrix}1 & -2 & 1\end{bmatrix}$

Image Gradients

$$\text{magnitude} = \sqrt{(S_x^2 + S_y^2)}$$

$$\text{direction} = \theta = \tan^{-1} \frac{S_y}{S_x}$$

- gradient is perpendicular to the edge

Filtering

image ⊗ kernel = filter output

Gaussian filtering : weighted averaging.
↪ you wanna make center pixel more important than others.

Edge Detectors → we use derivative filters

Prewitt and Sobel

- Compute derivatives in x and y direction
- Find gradient and magnitude
- Threshold gradient magnitude

$\boxed{\text{image}} \xrightarrow[\text{in } x]{\text{avg. smoothing}} \boxed{\text{blurred}} \xrightarrow[\text{in } x]{\text{derivative filtering}} \boxed{\text{edges in } x}$

Sobel ⇒ $\underset{x}{\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix}}$ $\underset{y}{\begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix}}$

Laplacian of Gaussian

- Laplacian filters are derivative filters to find areas of edges. Since derivative filters are very sensitive to noise, it's common to smooth the image (eg. using a Gaussian) before applying the Laplacian.