Insper

Computer Vision

Class 7: Discrete Derivatives as Convolutional Filters

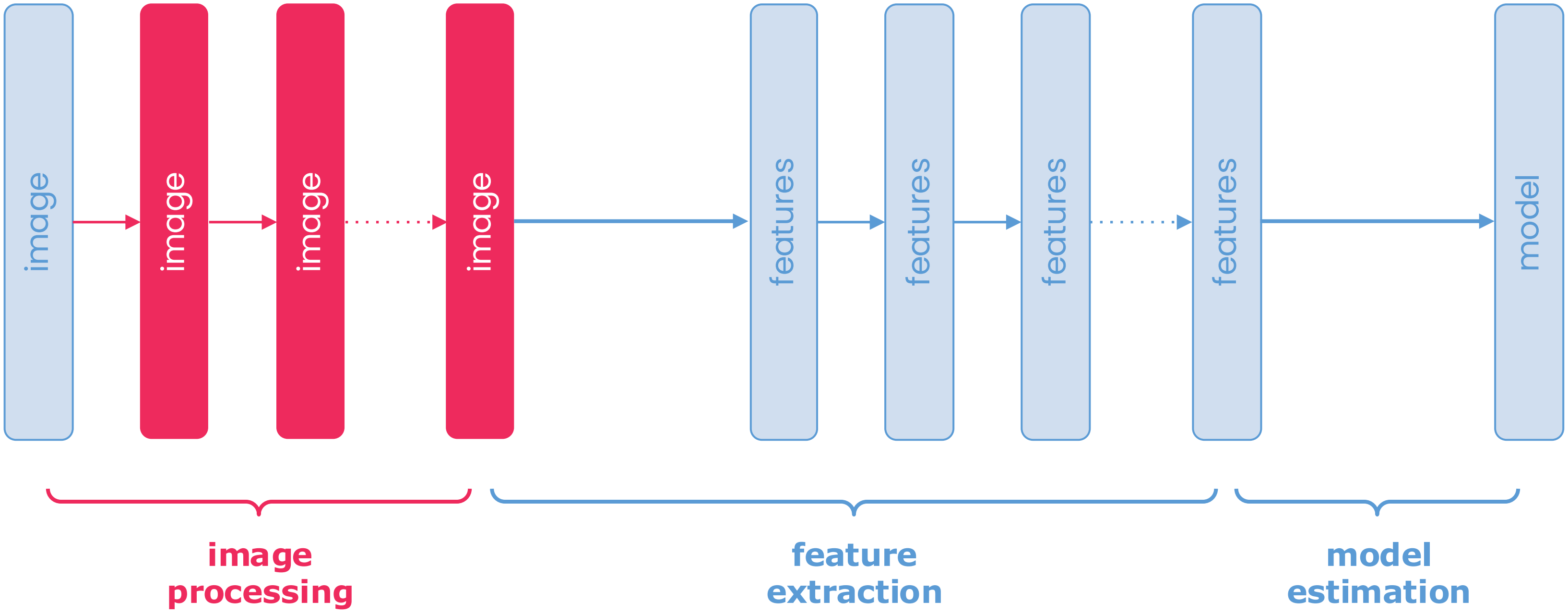


image
image
image
image
features
features
features
features
model
image processing
feature extraction
model estimation

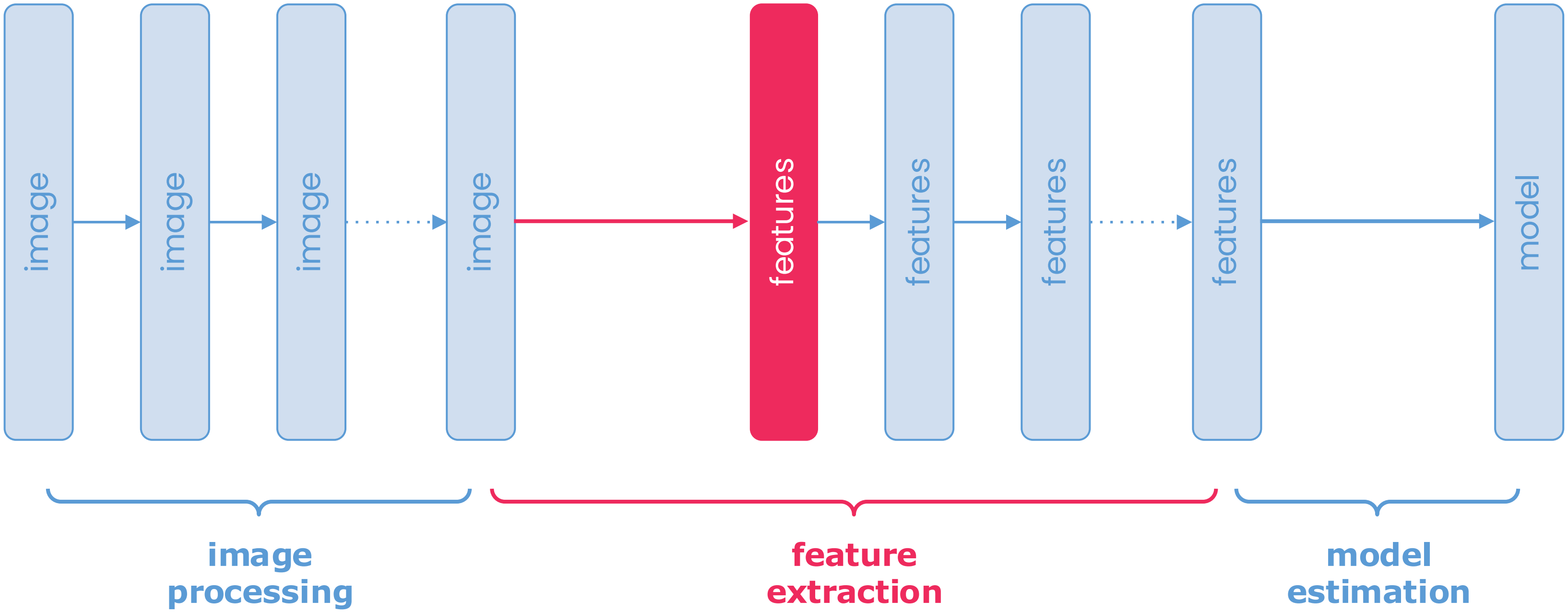


image
image
image
image
features
features
features
features
model
image
processing
feature
extraction
model
estimation

What do we want in a feature?

*Good features must be **robust**: detectable and recognizable under different conditions.*

Good features ideally are...

- ...robust to gray level reduction.
- ...robust to brightness changes.
- ...robust to contrast changes.
- ...robust to noise.
- ...robust to occlusion.
- ...robust to translation.
- ...robust to rotation.
- ...robust to scale.
- ...robust to perspective distortion.
- ...robust to deformation.

Edge detection:

a measure of how much a pixel seems to belong to a boundary

Edge detection is...

- ...robust to gray level reduction. *(mostly)*
- ...robust to brightness changes. *(within reason)*
- ...robust to contrast changes. *(within reason)*
- ...robust to occlusion. *(locally)*
- ...robust to translation.
- ...robust to rotation.
- ...robust to perspective distortion. *(locally)*
- ...robust to deformation. *(locally)*

Edges are ***discontinuities****:*
points of sudden variation.

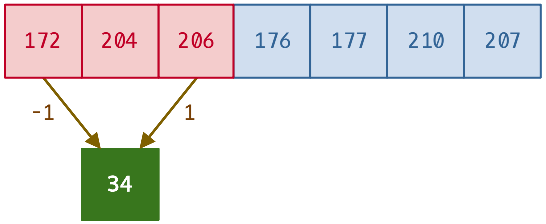
And what is the concept that measures variation?

Discrete derivative:

for each value, the difference of the pixels around it

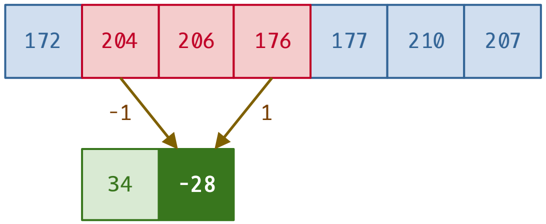
172	204	206	176	177	210	207

Discrete derivative:
for each value, the difference of the pixels around it

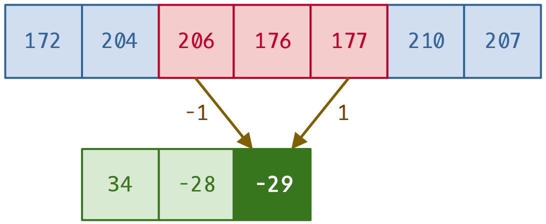


Discrete derivative:

for each value, the difference of the pixels around it

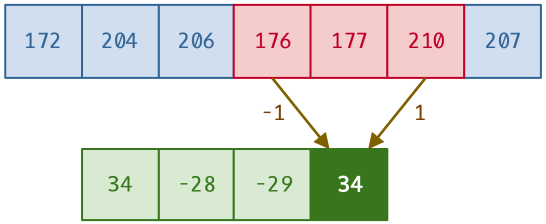


Discrete derivative:
for each value, the difference of the pixels around it



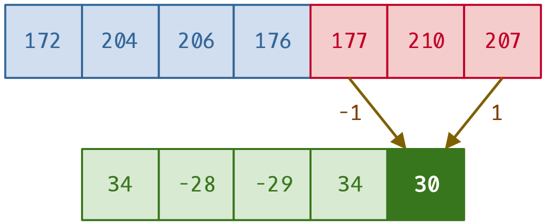
Discrete derivative:

for each value, the difference of the pixels around it



Discrete derivative:

for each value, the difference of the pixels around it



Discrete derivative:

for each value, the difference of the pixels around it

172	204	206	176	177	210	207

34	-28	-29	34	30

Discrete convolution:

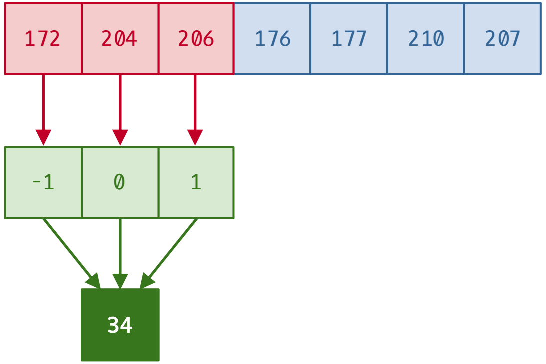
sum of the products with each possible shift of another function

$s(x)$	172	204	206	176	177	210	207

$d(x)$	-1	0	1

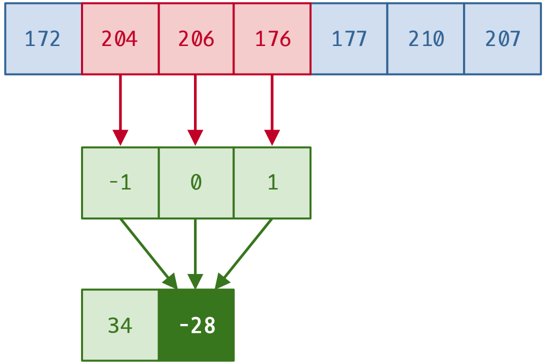
(technically, this is the definition of cross-correlation because does not flip the signal)

Discrete convolution:
sum of the products with each possible shift of another function



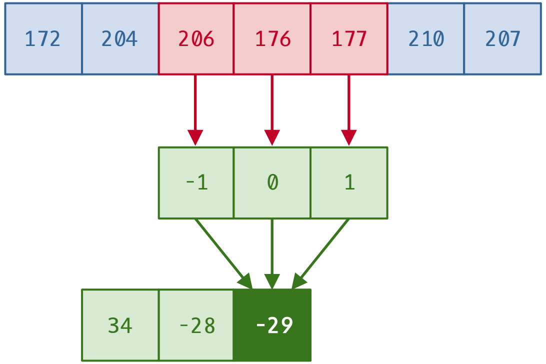
Discrete convolution:

sum of the products with each possible shift of another function

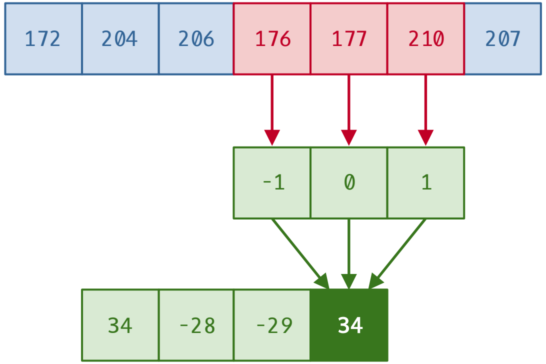


Discrete convolution:

sum of the products with each possible shift of another function

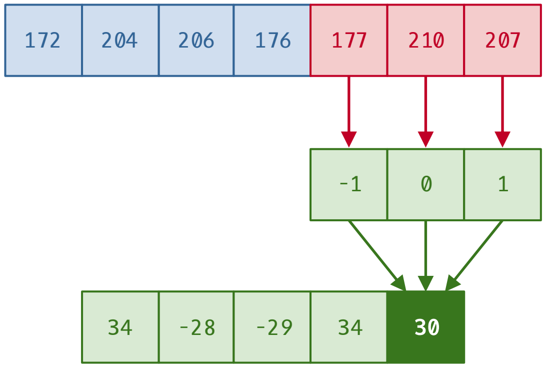


Discrete convolution:
sum of the products with each possible shift of another function



Discrete convolution:

sum of the products with each possible shift of another function



Discrete convolution:

sum of the products with each possible shift of another function

172	204	206	176	177	210	207

s

34	-28	-29	34	30

$s * d$

Sobel vertical edge filter:

convolution with a kernel that measures the gradient across the x dimension

normalized absolute values

P

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

D_x

$G_x = P * D_x$

Sobel vertical edge filter:

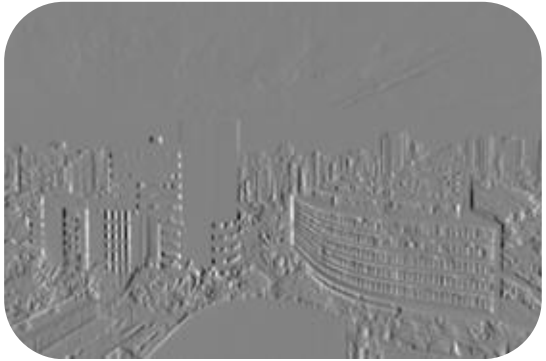
convolution with a kernel that measures the gradient across the x dimension

P

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

D_x

normalized signed values



$G_x = P * D_x$

Sobel horizontal edge filter:

convolution with a kernel that measures the gradient across the y dimension

normalized absolute values

P

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

D_y

$G_y = P * D_y$

Sobel horizontal edge filter:

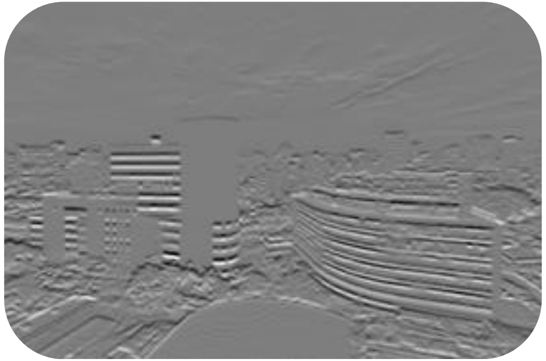
convolution with a kernel that measures the gradient across the y dimension

P

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

D_y

normalized signed values



$G_y = P * D_y$

Interpretation of the signs

- $g_x < 0$: brightness grows leftwards.
- $g_x > 0$: brightness grows rightwards.
- $g_y < 0$: brightness grows upwards.
- $g_y > 0$: brightness grows downwards.

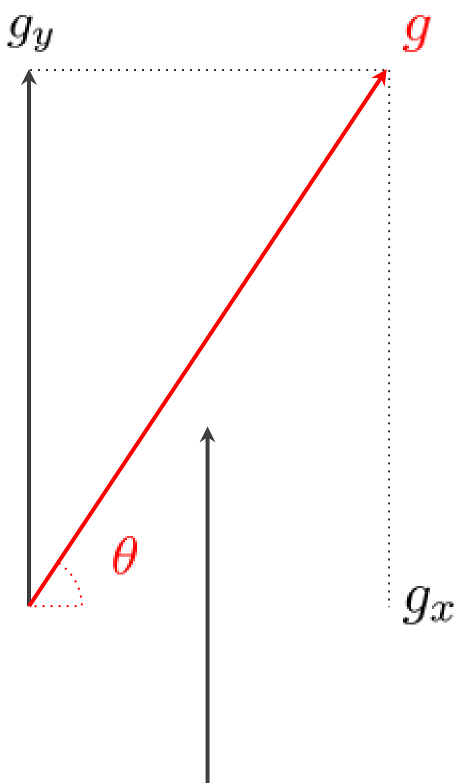
Interpretation as a vector

- Intensity:

$$|g| = \sqrt{|g_x|^2 + |g_y|^2}.$$

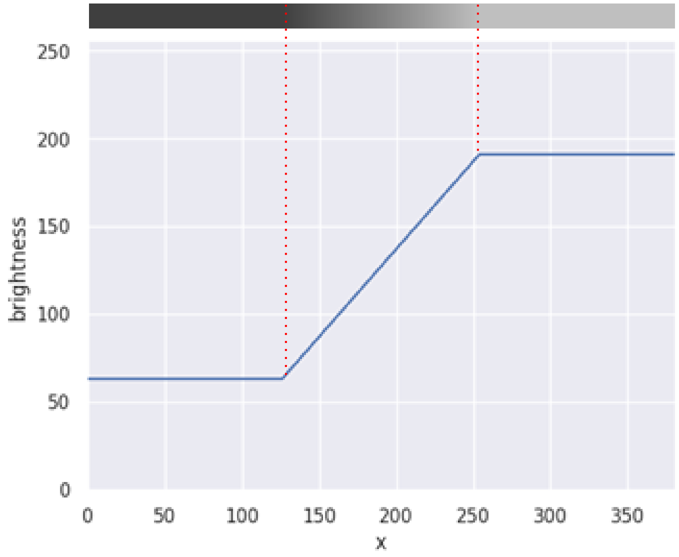
- Direction:

$$\theta = \arctan\left(\frac{|g_y|}{|g_x|}\right).$$

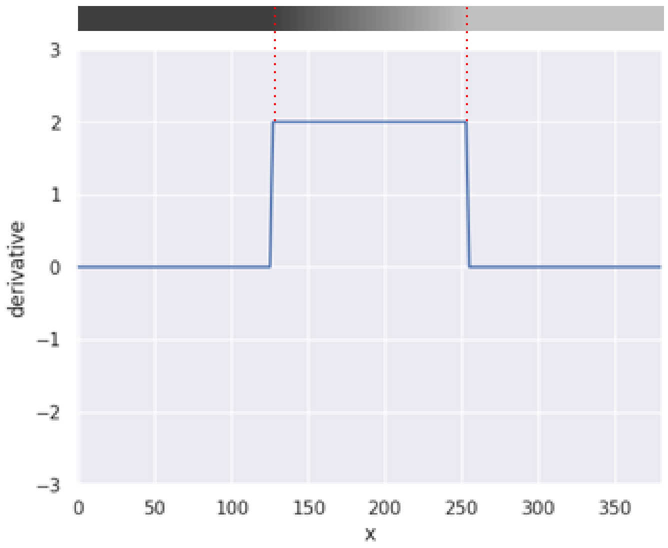


But, although the gradient works well in practice, it has a conceptual flaw.

Regions of constant variation:
not uniform, but not boundary either

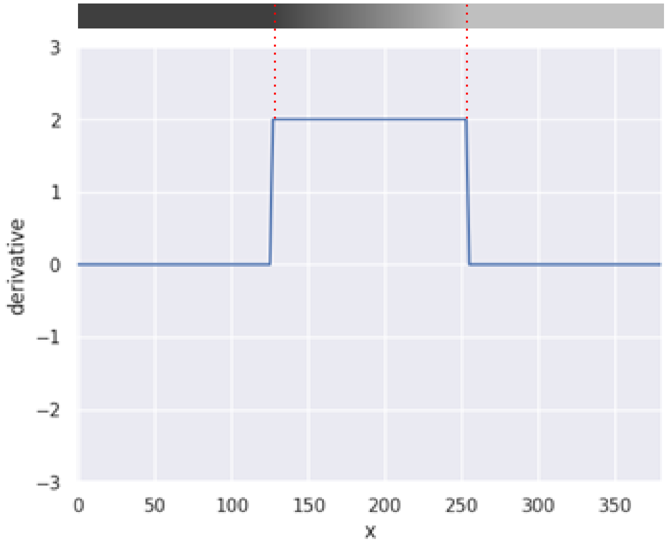


Derivative:
strong responses to the entire region

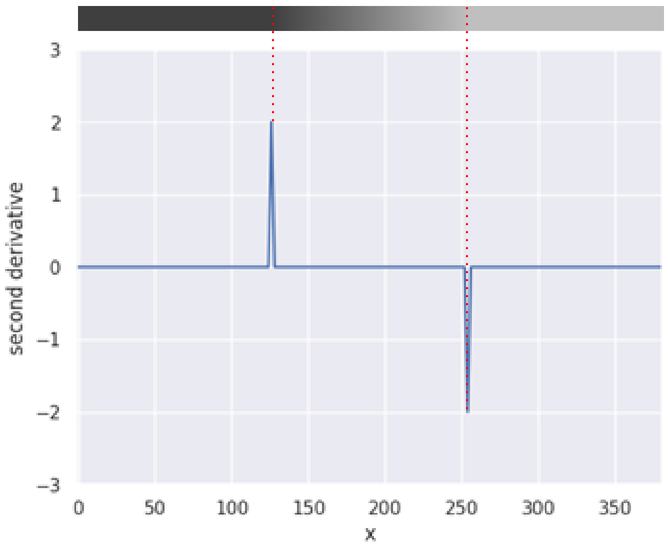


So let’s double our bet on derivatives.

First derivative:
strong responses in the entire region

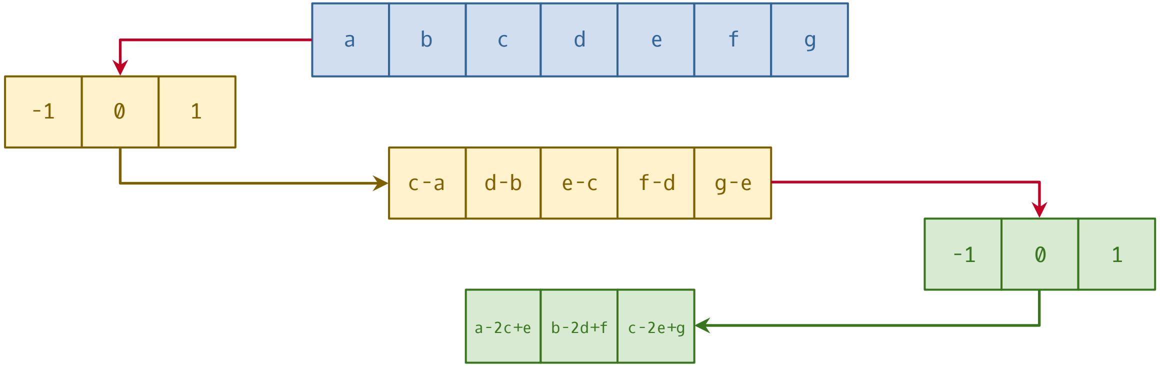


Second derivative:
strong responses only in boundaries



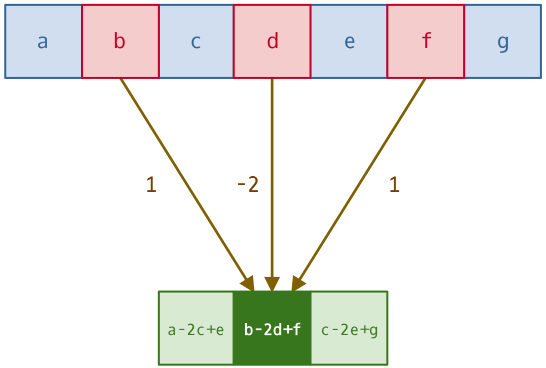
Second derivative:

two convolutions with the same kernel…



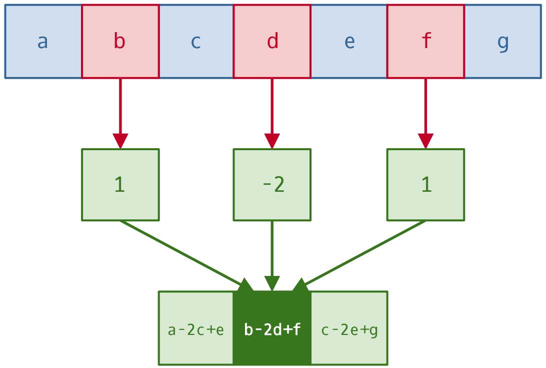
Discrete derivative:

...that...



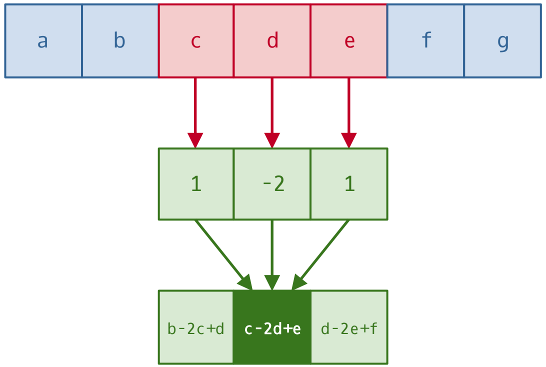
Discrete derivative:

...that...



Discrete derivative:

…can be approximated by a single kernel



Laplace operator:

sum of the second partial derivatives

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Laplace kernel:
sum of the second partial derivatives

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

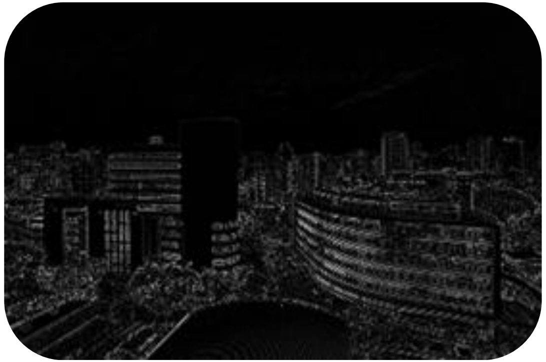
Laplace filter:
convolution with the Laplace kernel

normalized absolute values

P

0	1	0
1	-4	1
0	1	0

L



$\nabla^2 P = P * L$

handout

Toolkit

- **Language:** Python
- **Library:** OpenCV
- **Platform:** Google Colab

Instructions

1. Organize in groups of 2 or 3 members. No more, no less.

1. Make a copy of the notebook, read it, and do the activities.

1. Clean the notebook, save as `ipynb`, and submit via form.

Convolution associativity:

multiple convolutions with a single kernel

$$(P * G) * L$$

Convolution associativity:
multiple convolutions with a single kernel

$$P * (G * L)$$

Laplacian of Gaussian:
convolution of a Gaussian kernel with a Laplacian kernel

$$LoG = G * L$$

Laplacian of Gaussian:

convolution of a Gaussian kernel with a Laplacian kernel

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

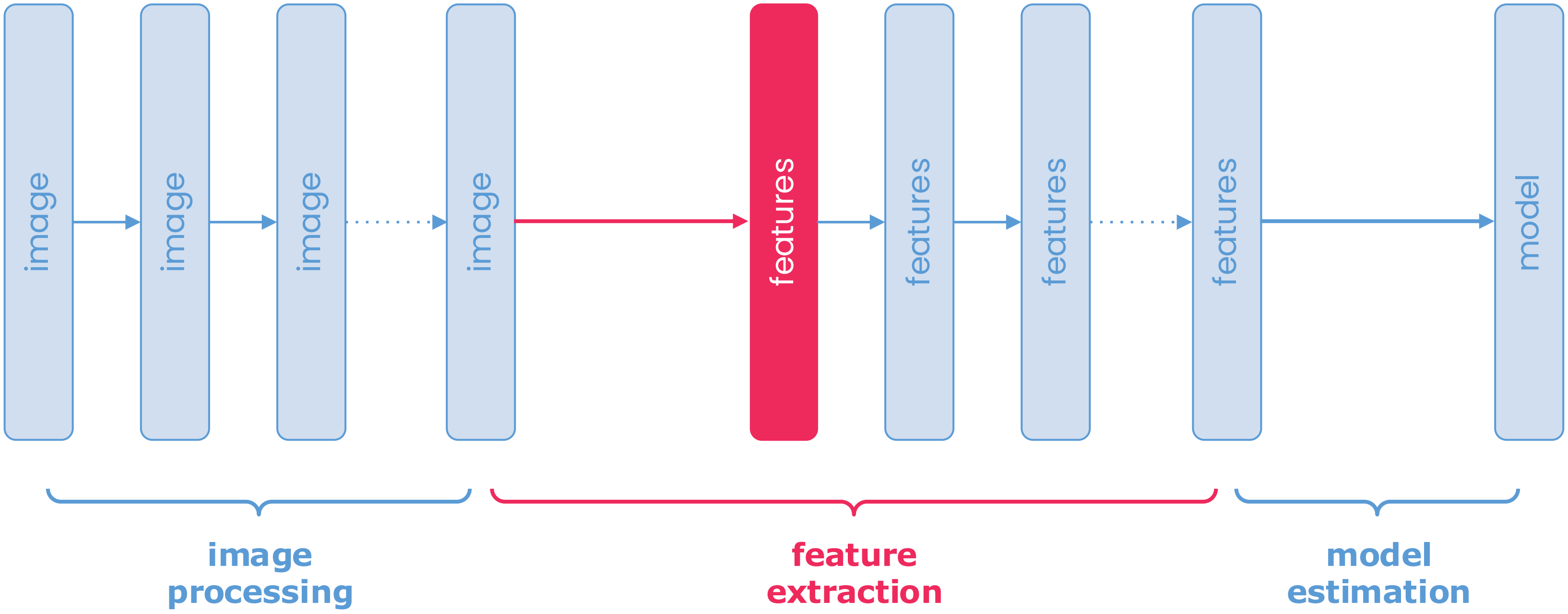


image
image
image
image
features
features
features
features
model
image
processing
feature
extraction
model
estimation

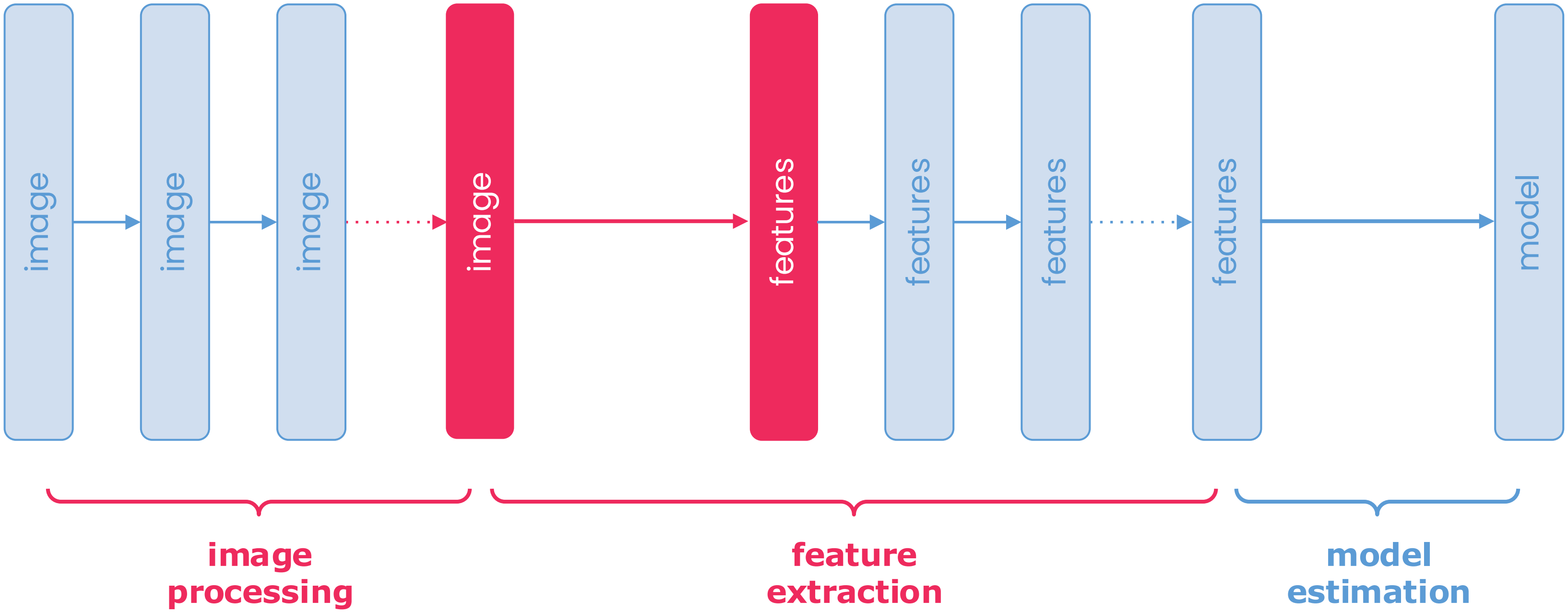


image
image
image
image
features
features
features
features
model
image processing
feature extraction
model estimation

Edge detection is not...

- ...robust to noise.
- ...robust to scale.

Edge detection is not…

- …robust to noise, but we can mitigate that with image smoothing.
- …robust to scale.

Edge detection is not...

- ...robust to noise, but we can mitigate that with image smoothing.
- ...robust to scale and... well... there's not much we can do about it.

Next class:

- robustness to scale and an important type of neural networks.

Credits

This material was based on the work of other professors, listed below.

- Fabio Miranda (fabiomiranda@insper.edu.br)
- Raul Ikeda (RaulIGS@insper.edu.br)
- Fabio Ayres (FabioJA@insper.edu.br)
- Igor Montagner (IgorSM1@insper.edu.br)
- Andrew Kurauchi (AndrewTNK@insper.edu.br)
- Luciano Silva (LucianoS4@insper.edu.br)
- Tiago Sanches (tiagoss4@insper.edu.br)

Well, except for the errors. Any errors you might find are probably my fault.

Images

https://www.insper.edu.br/graduacao/organizacoes-estudantis/
https://www.insper.edu.br/campus/
https://www.insper.edu.br/laboratorio-de-informatica/
https://terracoeconomico.com.br/insper-2/