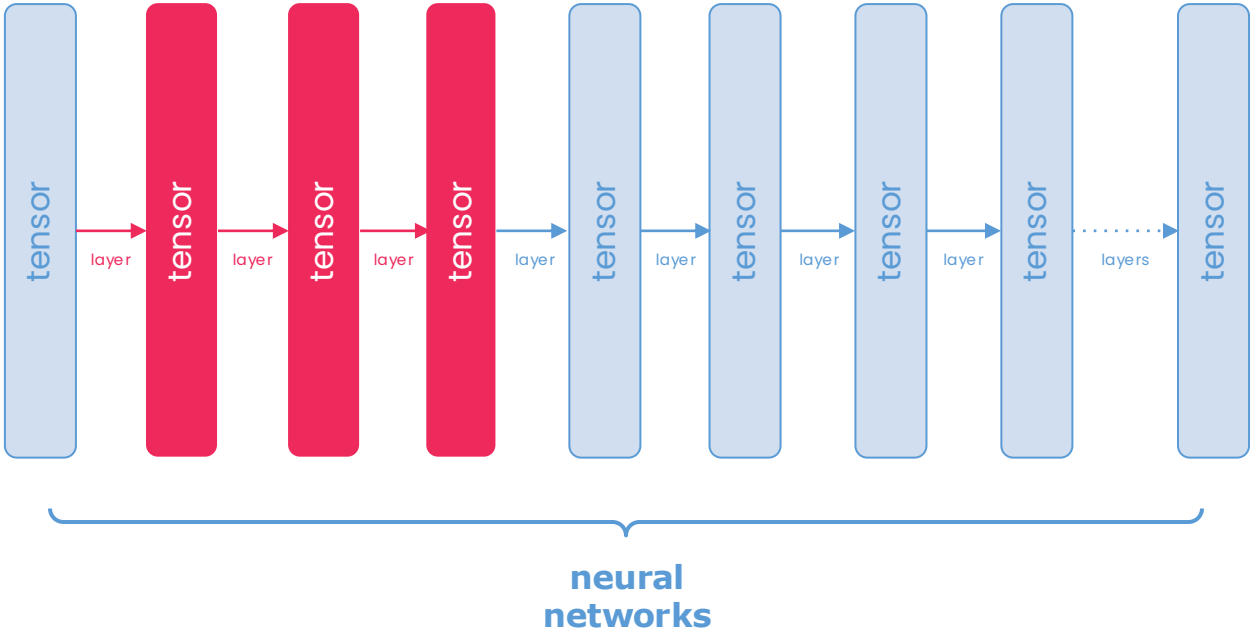
Insper

Computer Vision

# Class 6: Image Smoothing and Convolutional Filters


tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layers
tensor
neural
networks

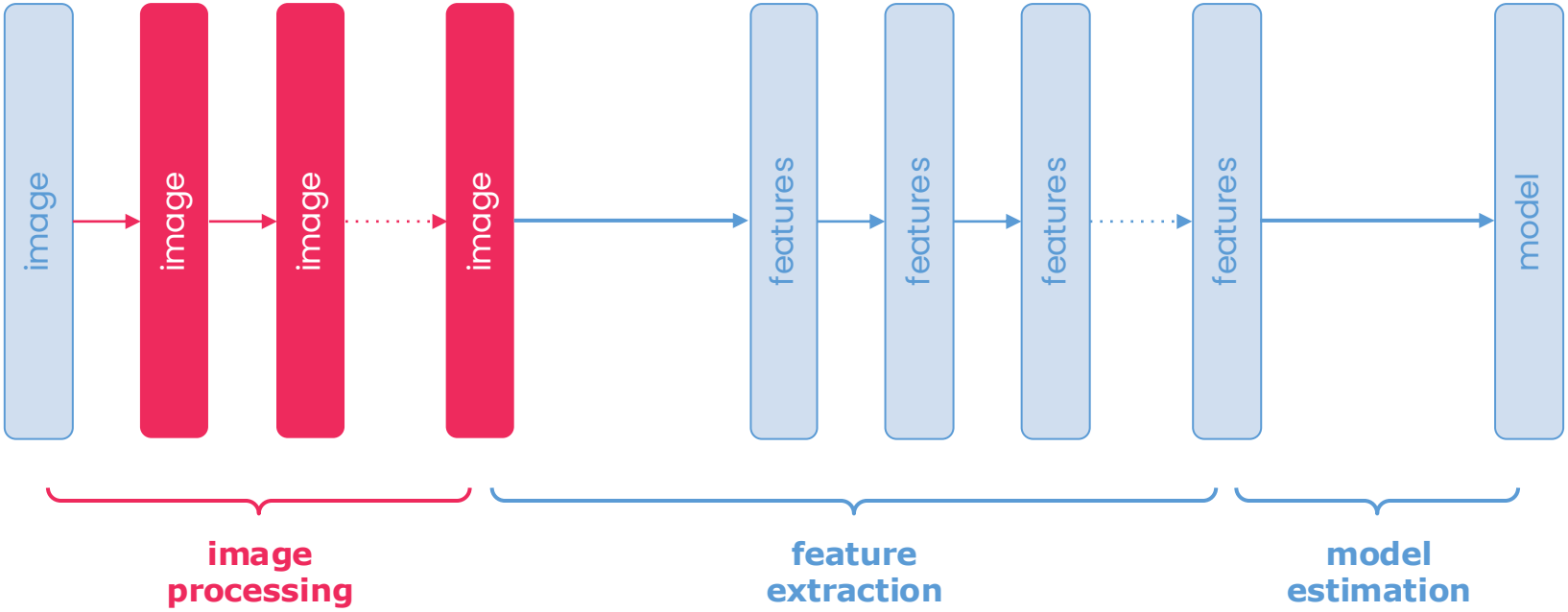


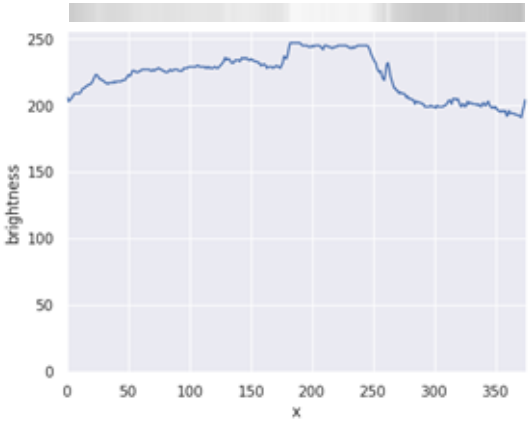
image
image
image
image
features
features
features
features
model
image processing
feature extraction
model estimation

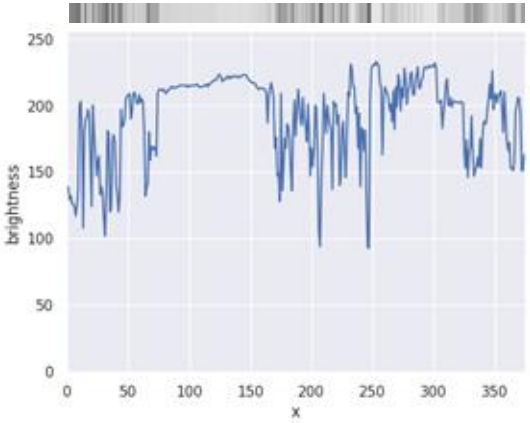
*We discussed how to enhance images to extract information...*

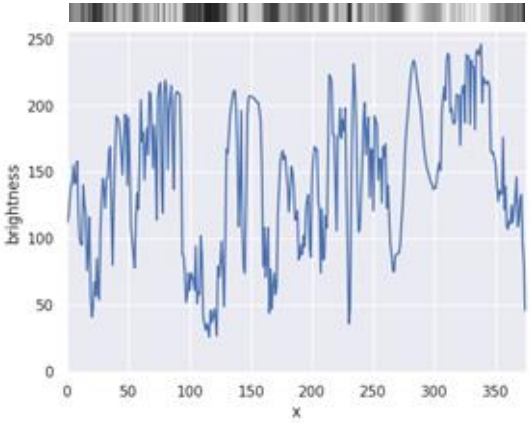
*...but assumed that data = information.*

**Gray level image:**
bivariate function from coordinates to levels

$$P(x, y)$$

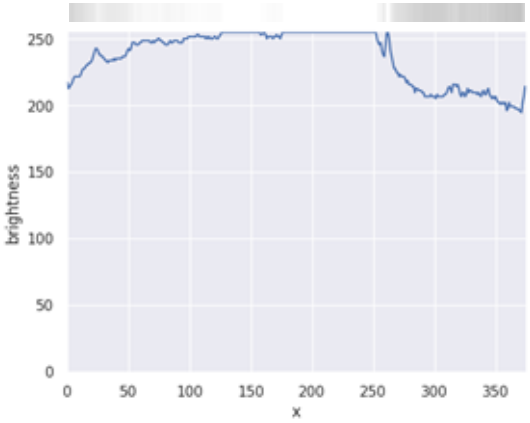


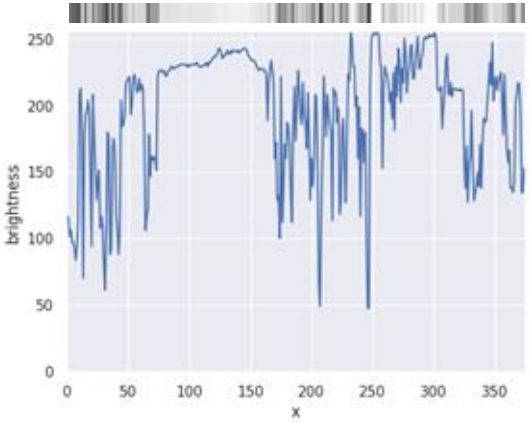



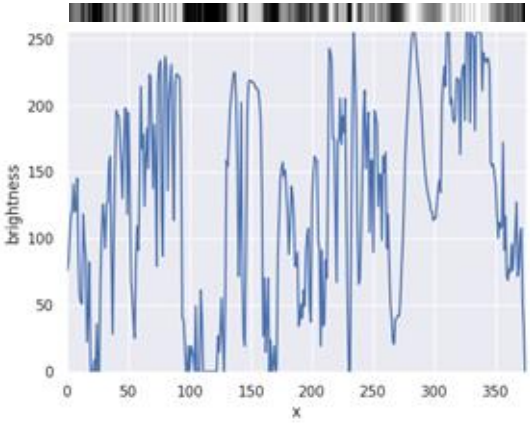



**Contrast:**
multiplicative constant,
but preserving the mean

$$1.5 \cdot (P(x,y) - \bar{p}) + \bar{p}$$



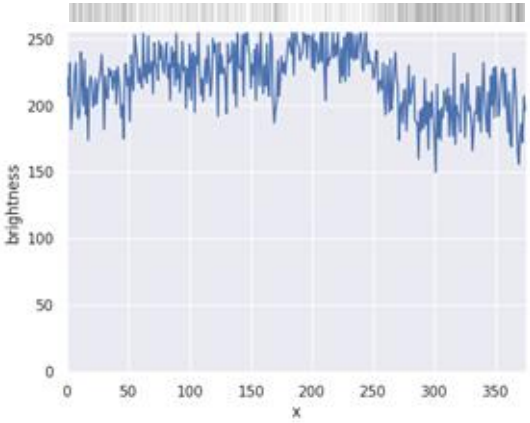



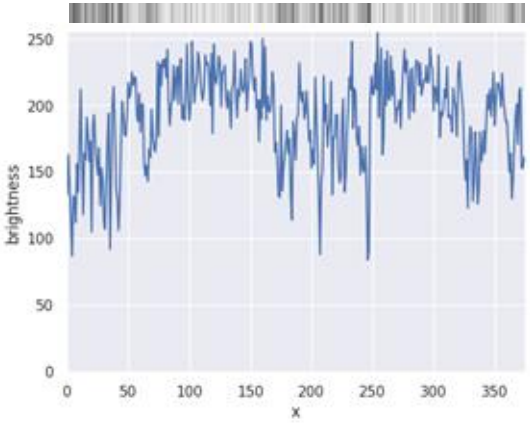
*Realistically, this is not true!*

*In most cases, we know that data = information + noise.*

**Image with noise:**
levels slightly below or above the correct value

$$P(x,y) + \epsilon(x,y)$$



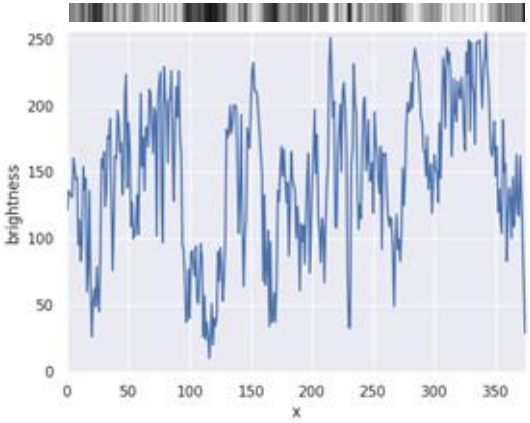
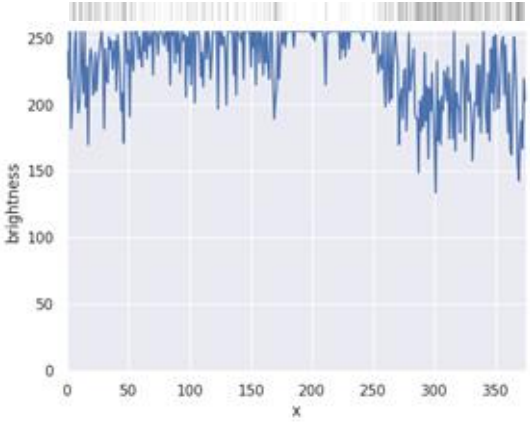
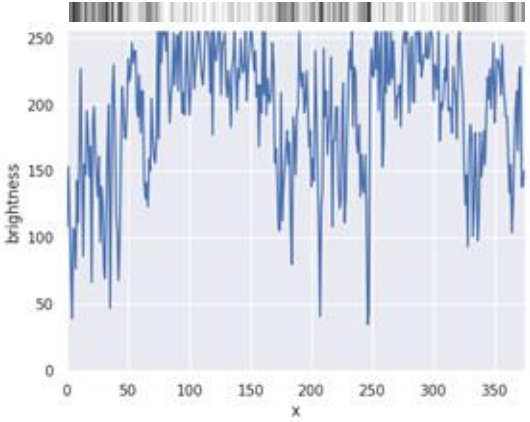
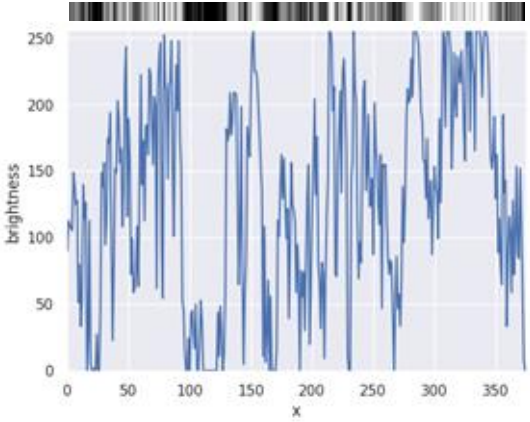



**Image with noise:**
increasing the contrast amplifies the problem

$$f_c(P(x,y) + \epsilon(x,y))$$

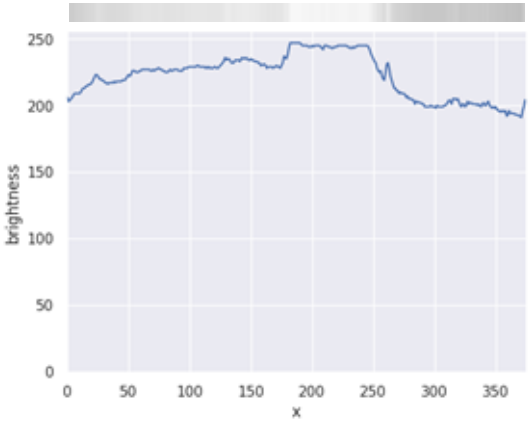


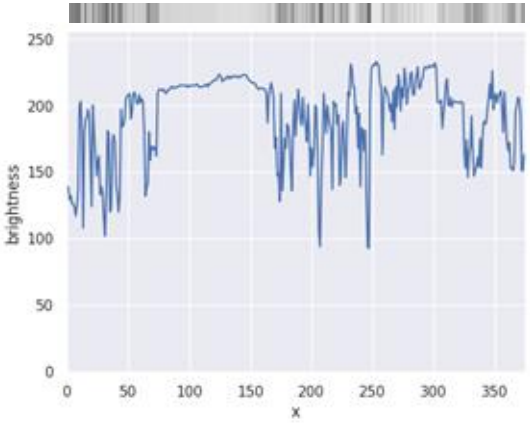



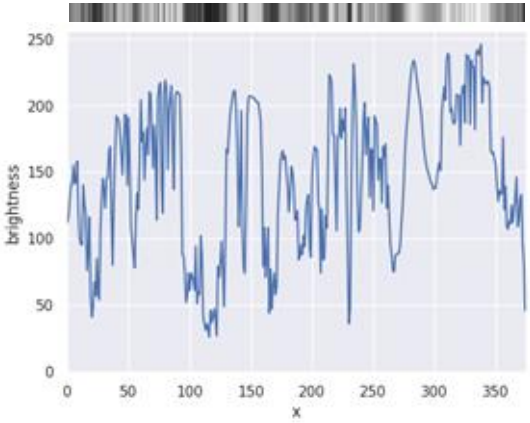



**Denoised image:**
usually not possible due to the loss of information

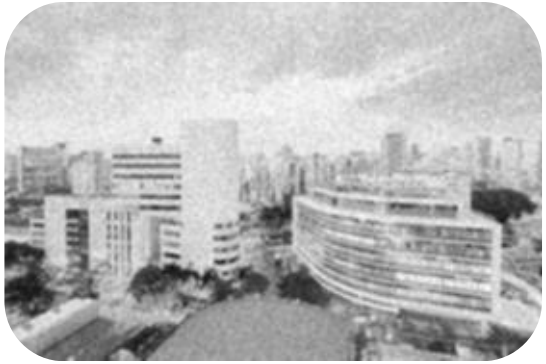
$$(P(x,y) + \epsilon(x,y)) - \epsilon(x,y)$$



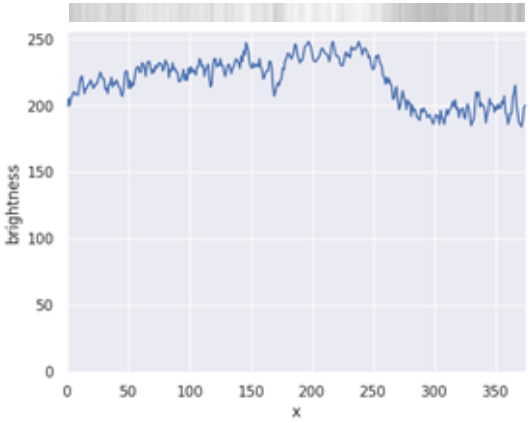



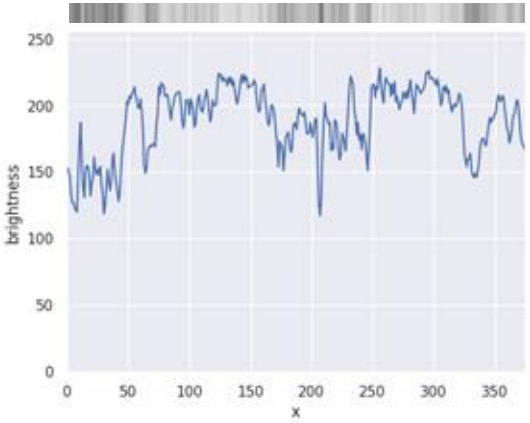



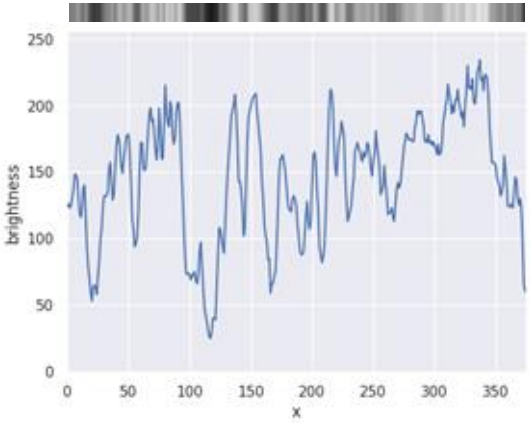
## Smoothed image?



$$f_s(P(x,y)+\epsilon(x,y))$$



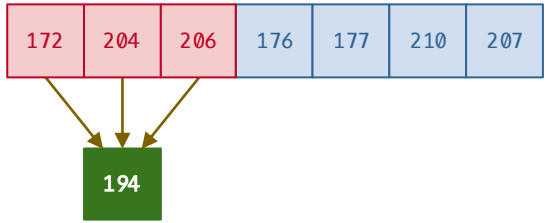



## Signal smoothing:

for each value, the mean of a window around it

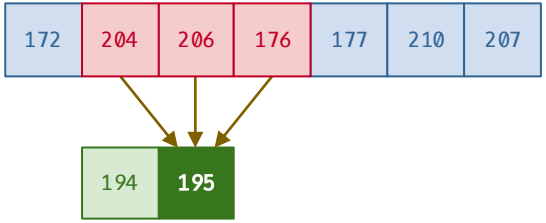
## Signal smoothing:

for each value, the mean of a window around it



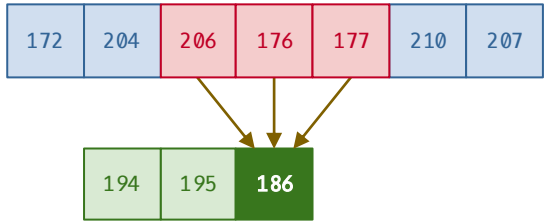
## Signal smoothing:

for each value, the mean of a window around it



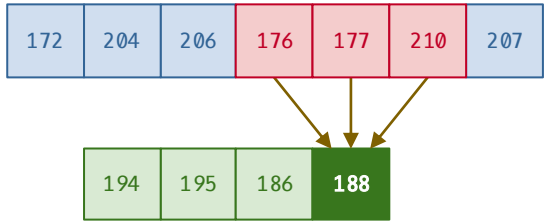
## Signal smoothing:

for each value, the mean of a window around it



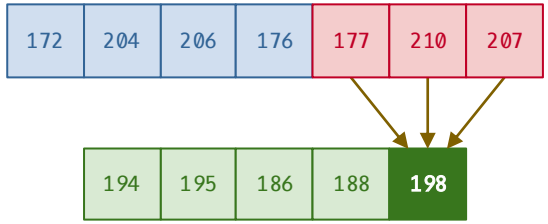
## Signal smoothing:

for each value, the mean of a window around it



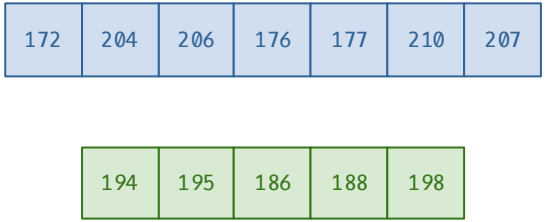
## Signal smoothing:

for each value, the mean of a window around it



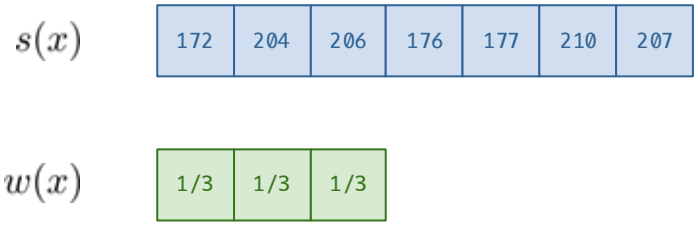
## Signal smoothing:

for each value, the mean of a window around it



## Discrete convolution:

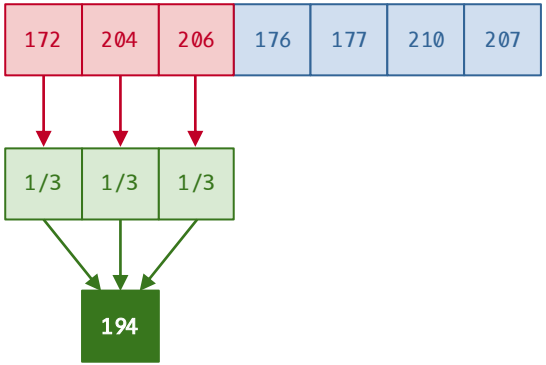
sum of the products with each possible shift of another function




*(technically, this is the definition of cross-correlation... we will explain more next class)*

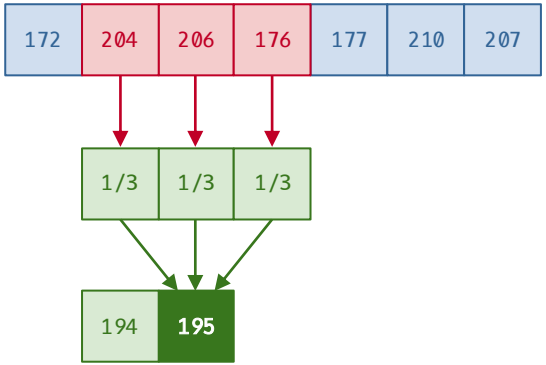
# Discrete convolution:

sum of the products with each possible shift of another function



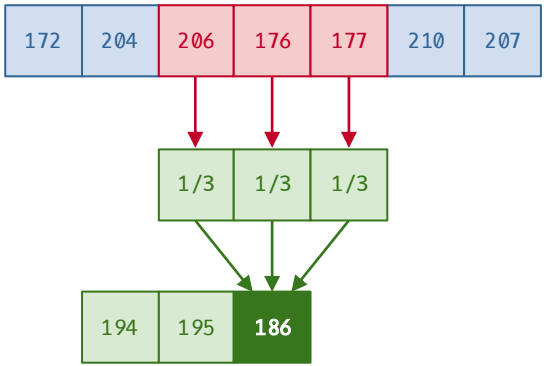
# Discrete convolution:

sum of the products with each possible shift of another function



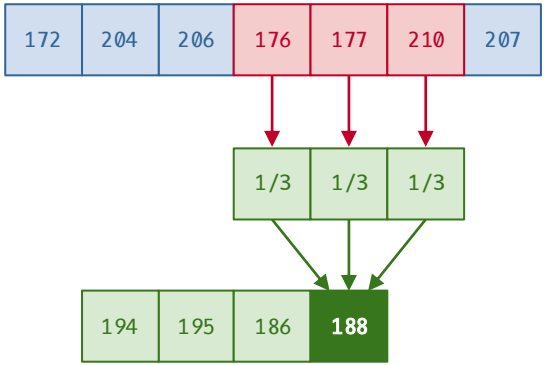
# Discrete convolution:

sum of the products with each possible shift of another function



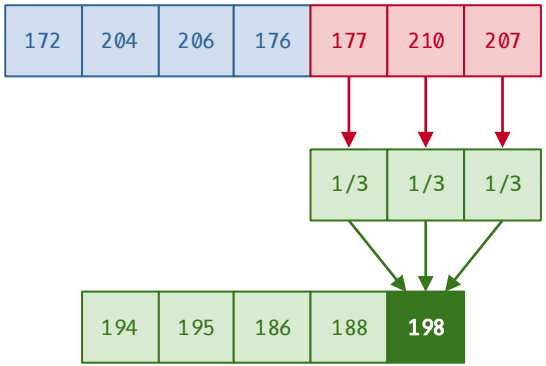
# Discrete convolution:

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


$s$


$s * w$

# Discrete convolution:

sum of the products with each possible shift of another function

| | | | | | | |
|---|---|---|---|---|---|---|
| 179 | 191 | 236 | 152 | 192 | 191 | 155 |
| 114 | 188 | 179 | 159 | 190 | 193 | 232 |
| 172 | 204 | 206 | 176 | 177 | 210 | 207 |
| 156 | 217 | 184 | 174 | 205 | 197 | 148 |
| 122 | 179 | 174 | 169 | 214 | 255 | 180 |

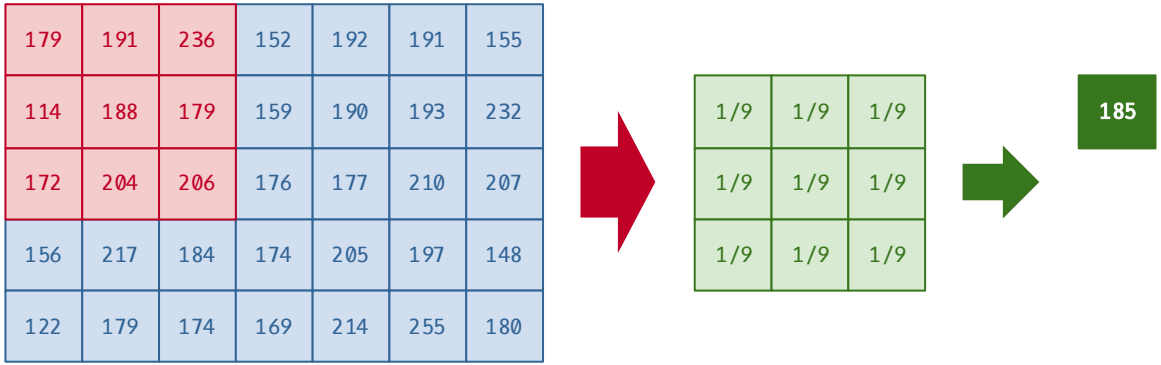
$P(x, y)$

| | | |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

$W(x, y)$

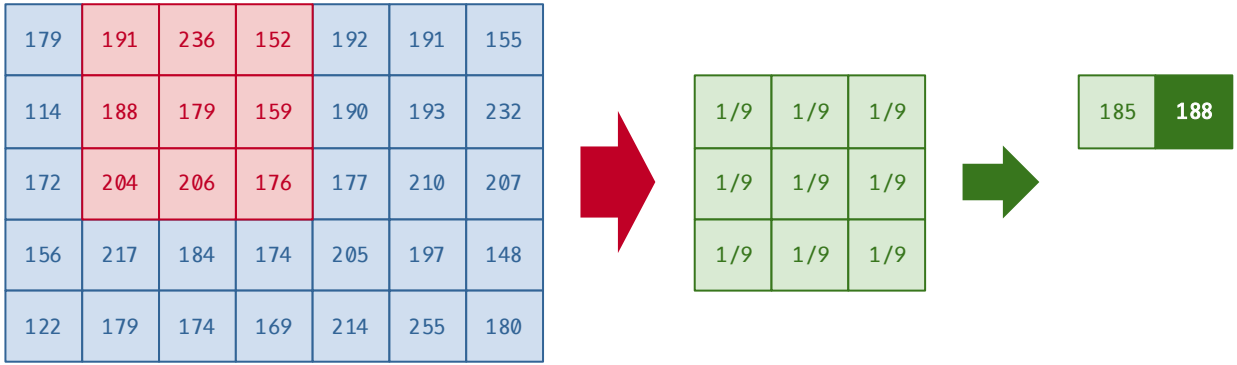
# Discrete convolution:

sum of the products with each possible shift of another function



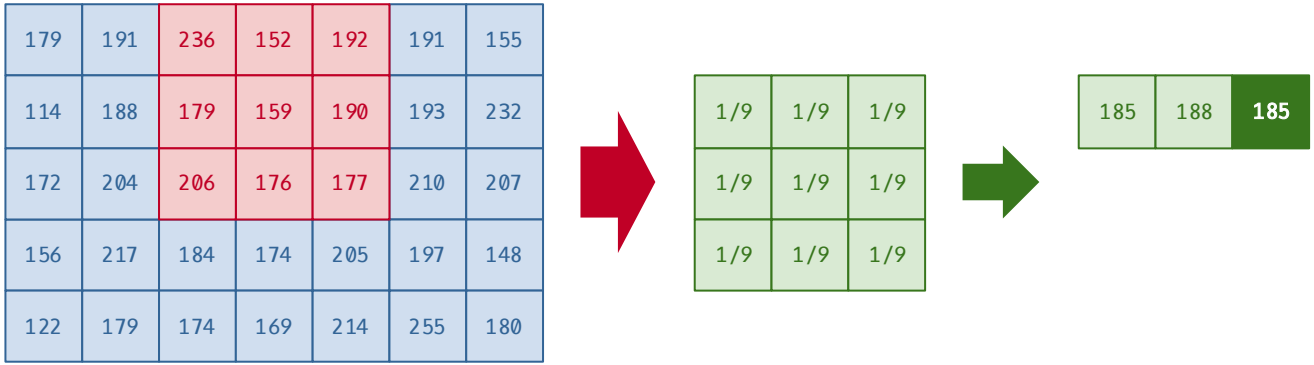
# Discrete convolution:

sum of the products with each possible shift of another function



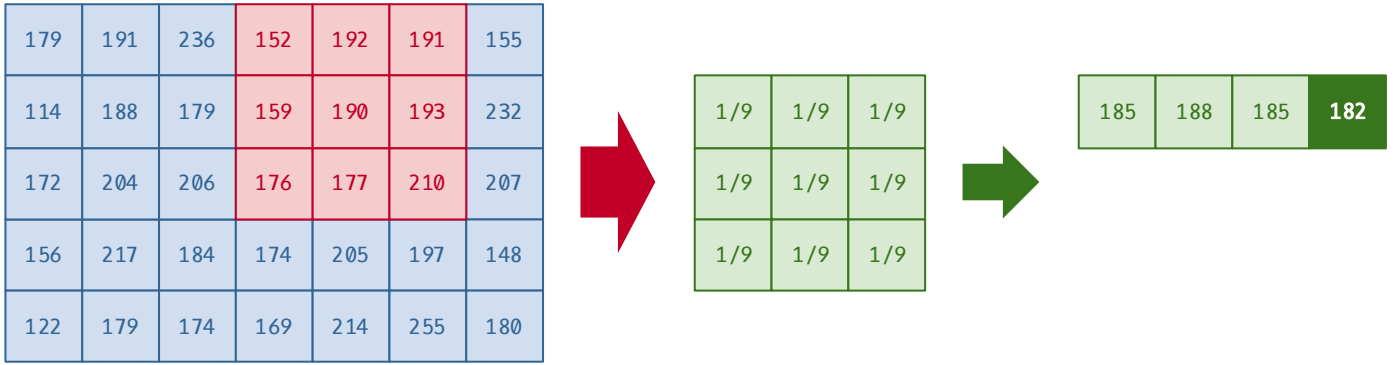
# Discrete convolution:

sum of the products with each possible shift of another function



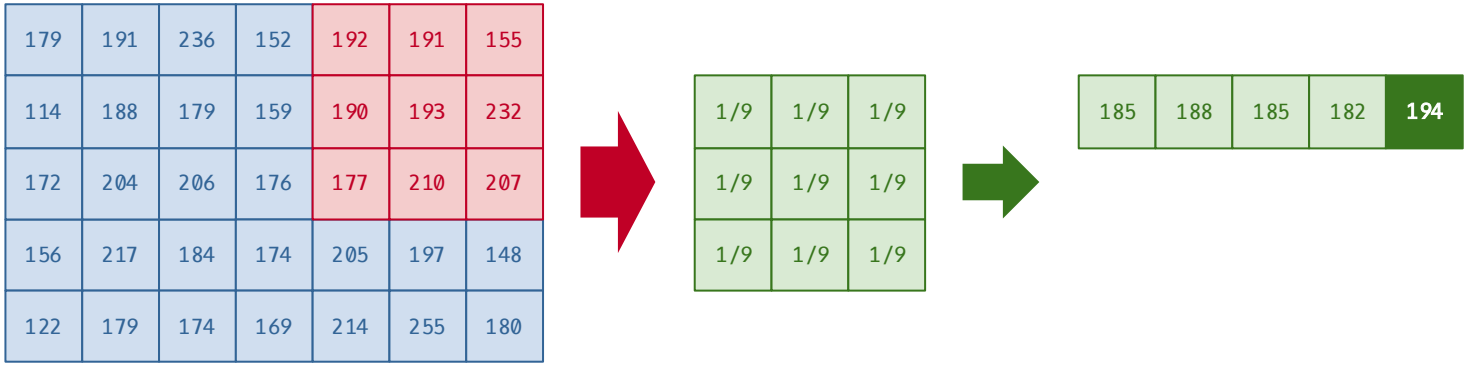
# Discrete convolution:

sum of the products with each possible shift of another function



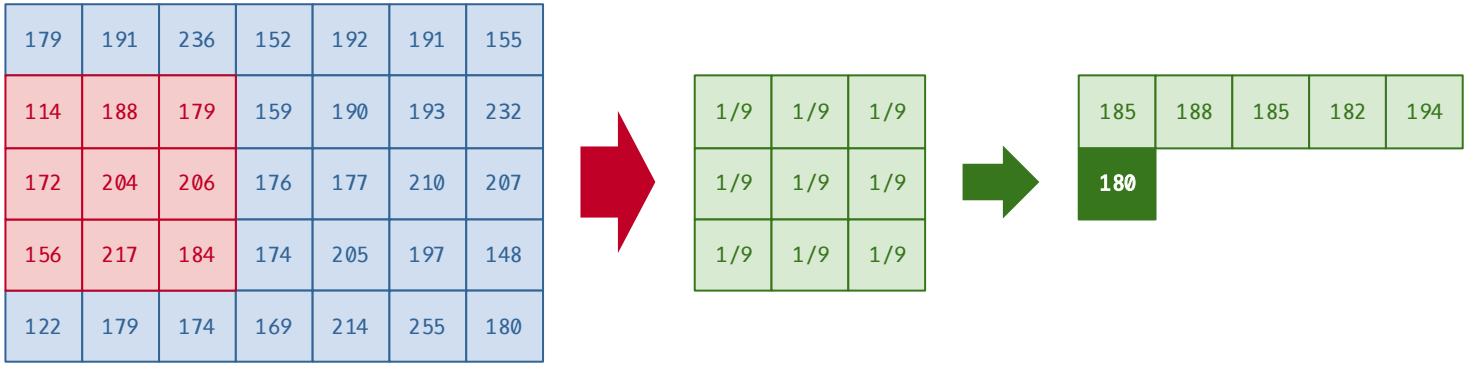
# Discrete convolution:

sum of the products with each possible shift of another function



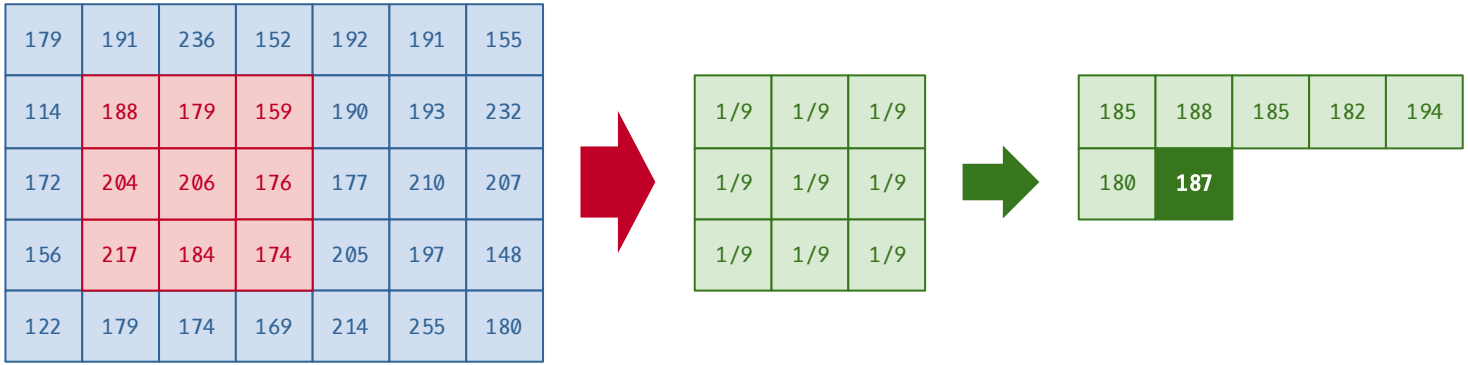
# Discrete convolution:

sum of the products with each possible shift of another function



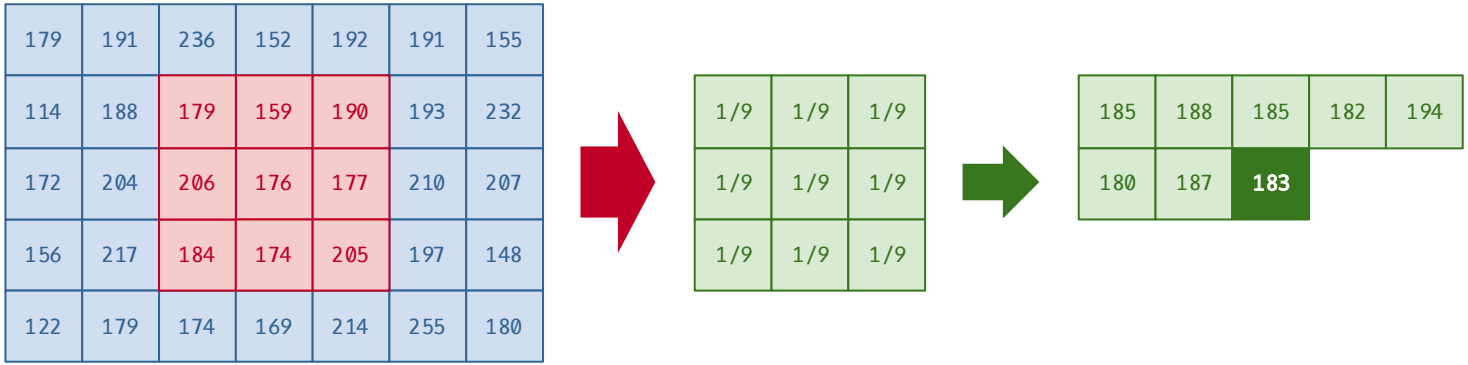
# Discrete convolution:

sum of the products with each possible shift of another function



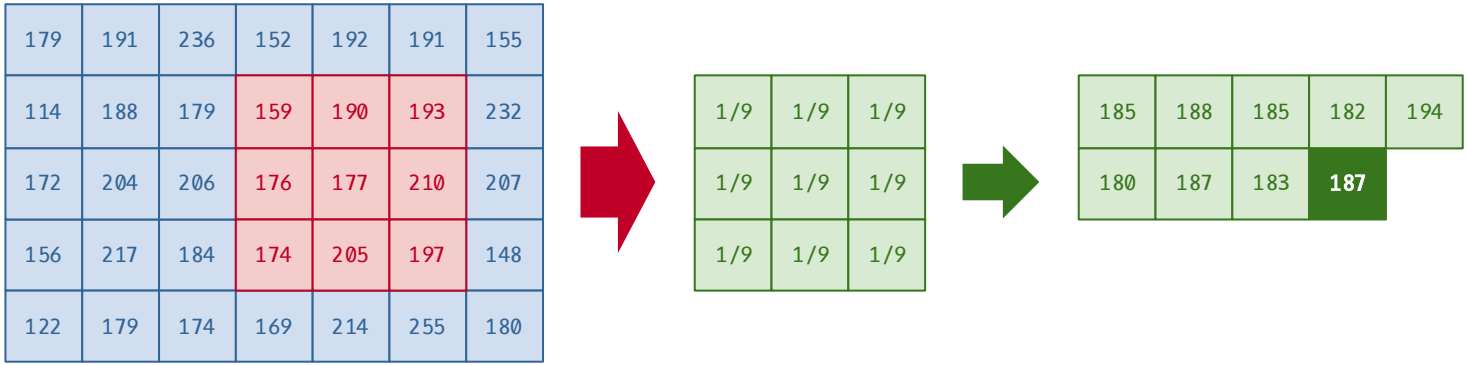
# Discrete convolution:

sum of the products with each possible shift of another function



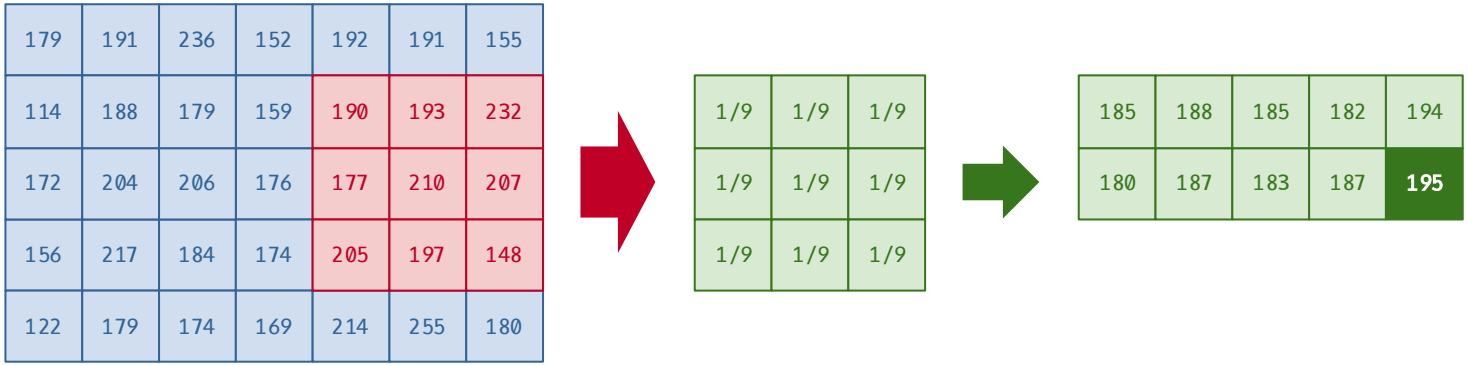
# Discrete convolution:

sum of the products with each possible shift of another function



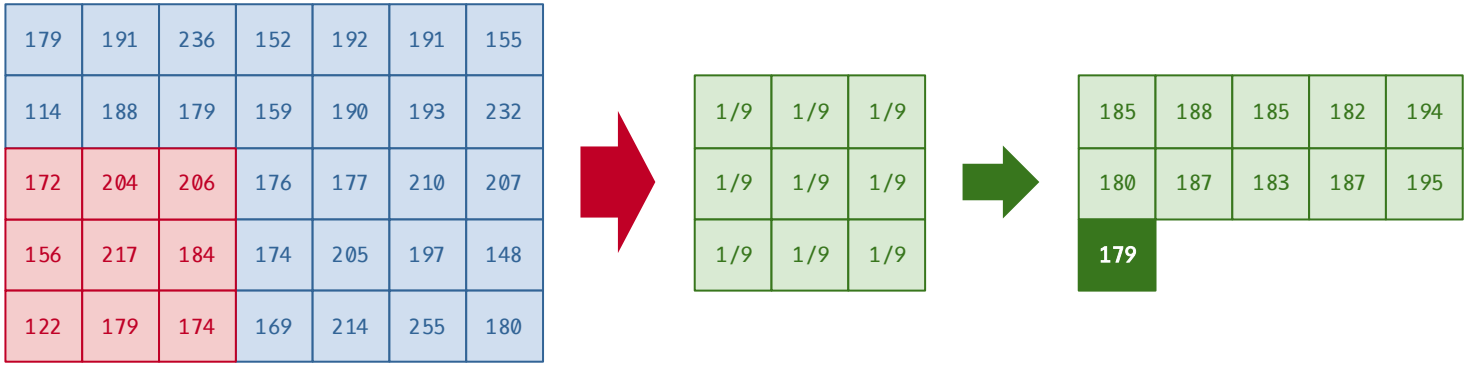
# Discrete convolution:

sum of the products with each possible shift of another function



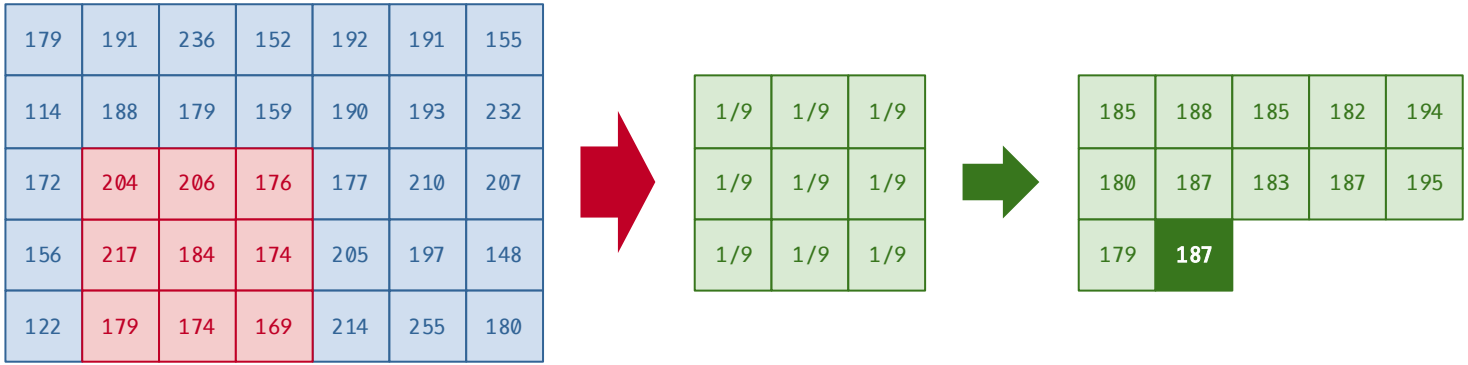
# Discrete convolution:

sum of the products with each possible shift of another function



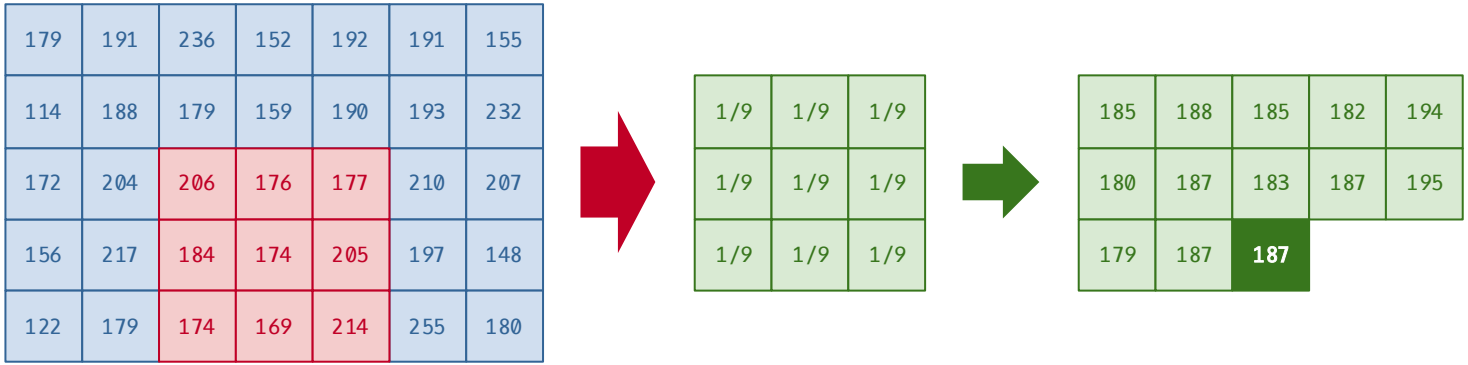
# Discrete convolution:

sum of the products with each possible shift of another function



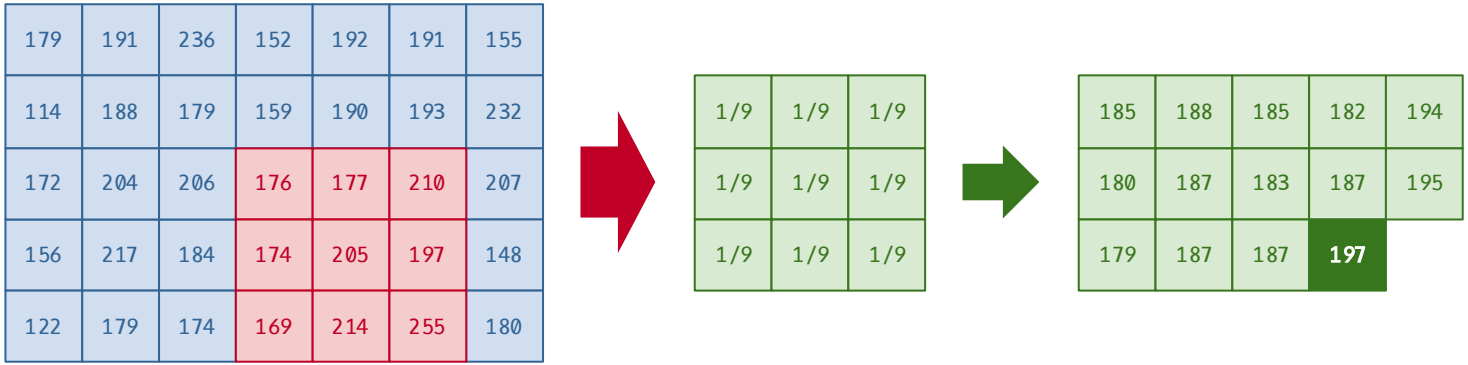
# Discrete convolution:

sum of the products with each possible shift of another function



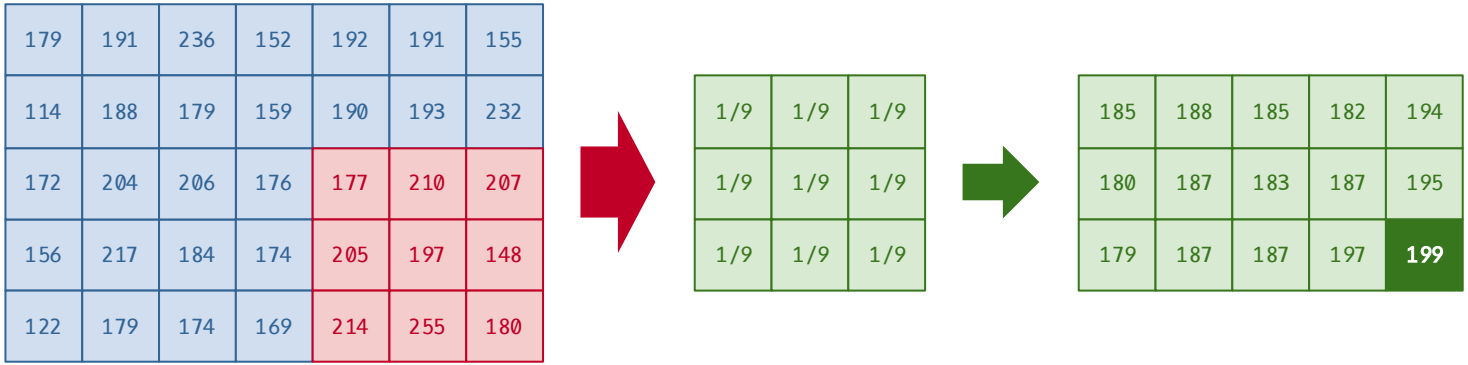
# Discrete convolution:

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

| | | | | | | |
|---|---|---|---|---|---|---|
| 179 | 191 | 236 | 152 | 192 | 191 | 155 |
| 114 | 188 | 179 | 159 | 190 | 193 | 232 |
| 172 | 204 | 206 | 176 | 177 | 210 | 207 |
| 156 | 217 | 184 | 174 | 205 | 197 | 148 |
| 122 | 179 | 174 | 169 | 214 | 255 | 180 |

$P$

| | | | | |
|---|---|---|---|---|
| 185 | 188 | 185 | 182 | 194 |
| 180 | 187 | 183 | 187 | 195 |
| 179 | 187 | 187 | 197 | 199 |

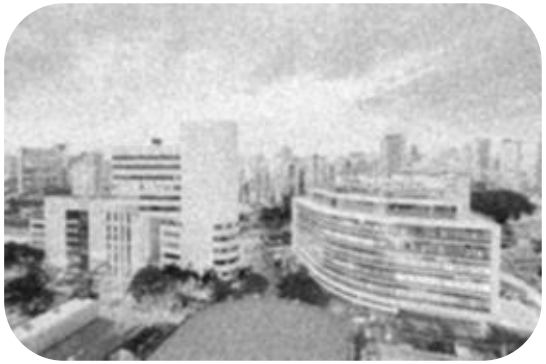
$P * W$

# Image smoothing:
convolution with a kernel of weights

$P$

| | | |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

$W$



$P * W$

# Image smoothing:
convolution with a kernel of weights

$P$

| | | | | |
|---|---|---|---|---|
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |

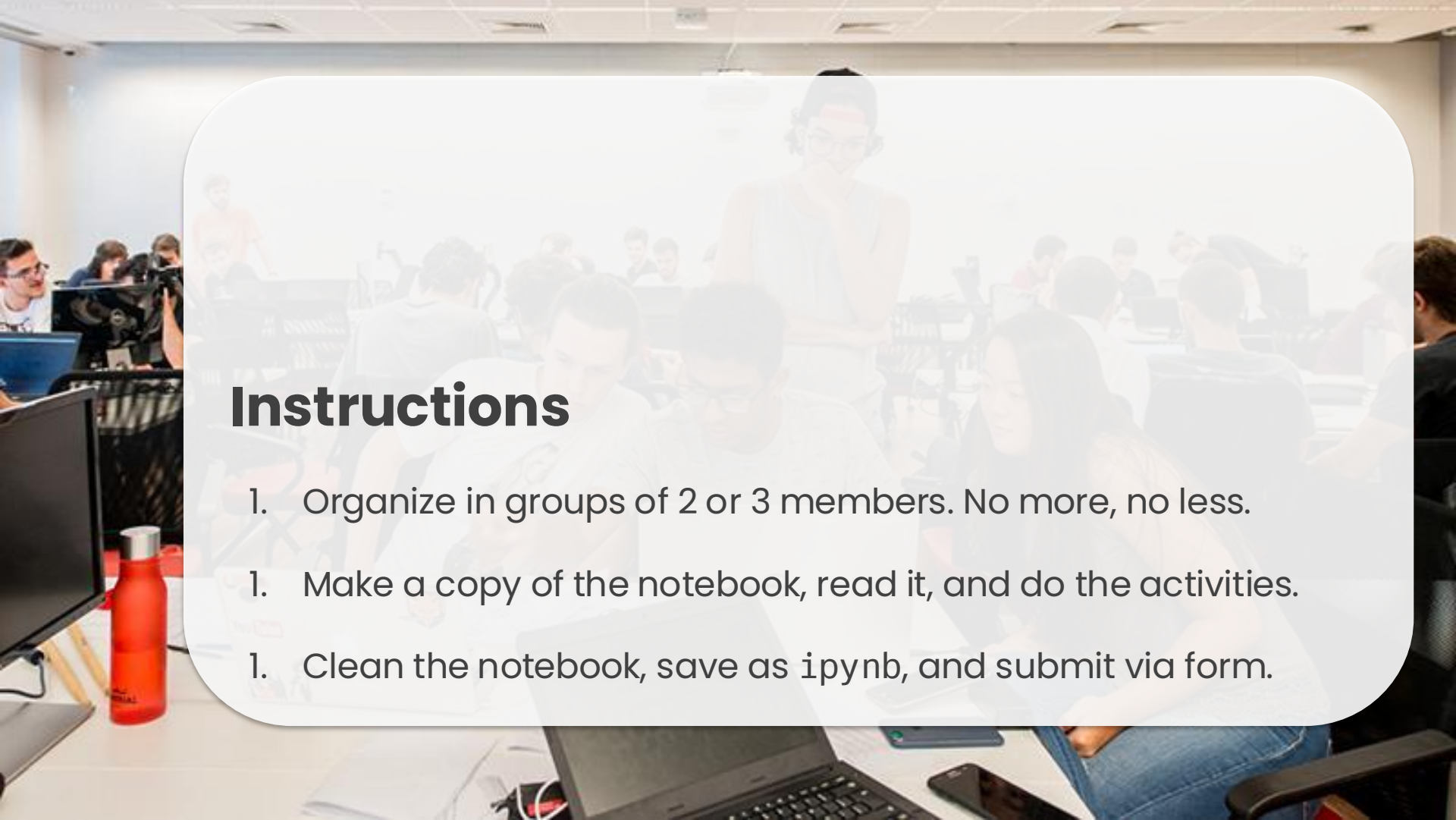
$W$

$P * W$

handout

# Toolkit

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

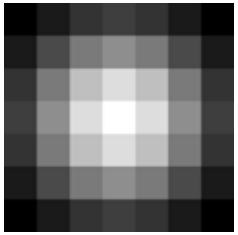


## Instructions

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.

# Image smoothing:
convolution with a kernel of weights

$P$



$G$

$S = P * G$

# Next class:

- edge detection.

# 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/