Informatique et systèmes de communication ISC
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

# *Signal and information (SigInf)*

## 204.1

Dr. Robin Delabays and Dre. Benedetta Franceschiello
benedetta.franceschiello@hevs.ch
Robin.delabays.hevs.ch

# *Lesson 8. Image processing*

Dr. Robin Delabays and Dre. Benedetta Franceschiello
benedetta.franceschiello@hevs.ch
Robin.delabays.hevs.ch

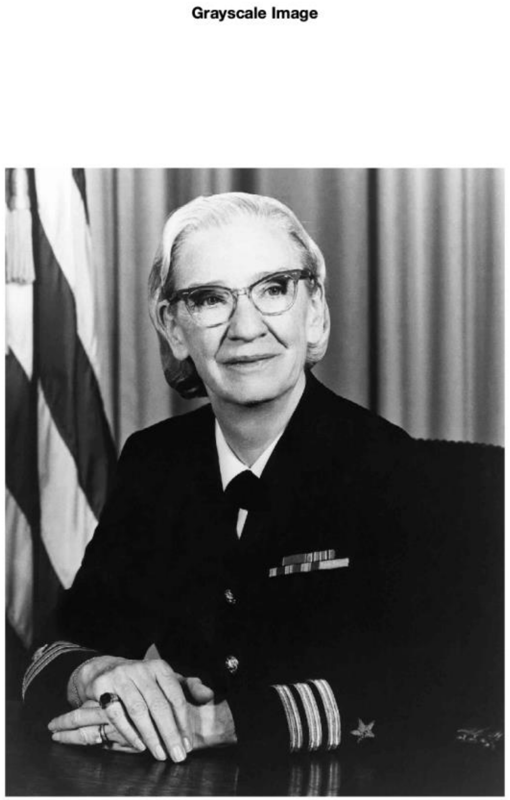
# 1. INTRODUCTION TO IMAGES

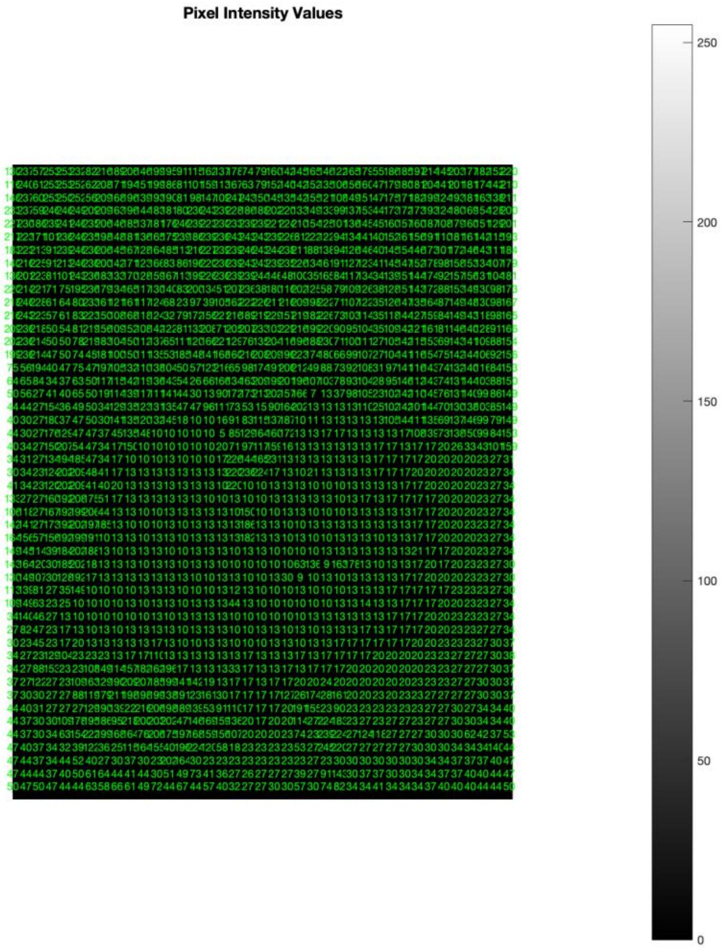
# An image is a 2D signal

**An image** is a 2D grid of pixel values, each representing light intensity. Greyscale images are single-channel arrays and color images are three-channel (RGB) arrays.

Definition

# An image is a 2D signal: greyscale



# An image is a 2D signal: RGB image

RGB Image

**Red Channel**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 47 | 77 | 37 | 94 | 227 | 181 | 246 | 244 |
| 56 | 85 | 115 | 34 | 52 | 208 | 221 | 249 |
| 237 | 15 | 47 | 34 | 0 | 215 | 4 | 251 |
| 240 | 76 | 19 | 26 | 167 | 217 | 191 | 253 |
| 225 | 42 | 20 | 215 | 234 | 228 | 228 | 254 |
| 213 | 21 | 8 | 204 | 232 | 245 | 73 | 254 |
| 210 | 63 | 189 | 179 | 171 | 214 | 26 | 255 |
| 208 | 217 | 198 | 201 | 5 | 4 | 52 | 51 |

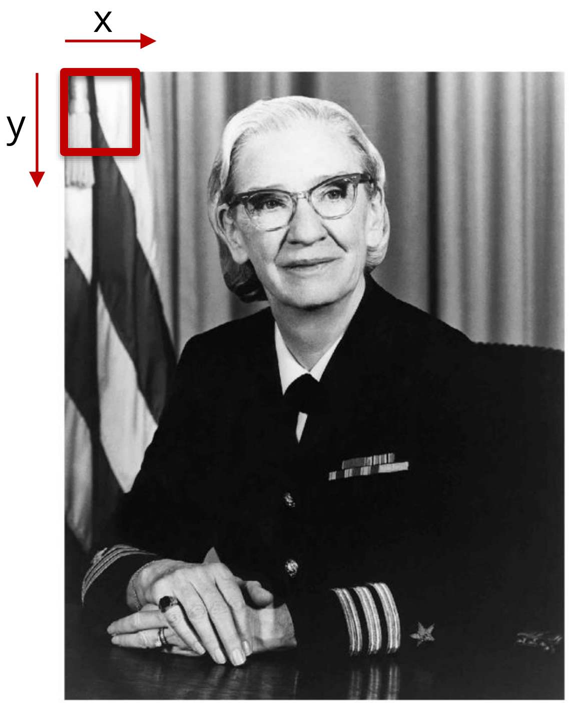
**Green Channel**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 57 | 72 | 47 | 85 | 221 | 172 | 241 | 240 |
| 61 | 77 | 108 | 42 | 61 | 181 | 208 | 245 |
| 214 | 16 | 57 | 40 | 0 | 141 | 3 | 249 |
| 204 | 57 | 25 | 29 | 119 | 133 | 140 | 253 |
| 16 | 41 | 26 | 163 | 4 | 159 | 165 | 254 |
| 7 | 24 | 10 | 137 | 8 | 188 | 65 | 254 |
| 139 | 47 | 4 | 75 | 10 | 138 | 26 | 255 |
| 135 | 171 | 152 | 113 | 3 | 3 | 51 | 50 |

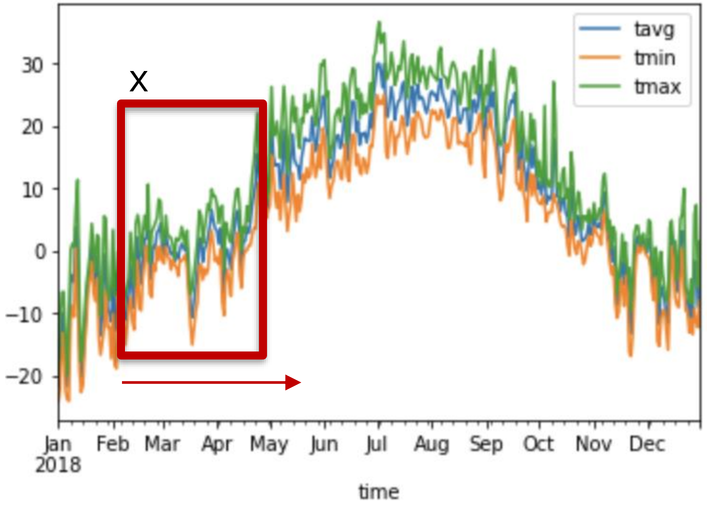
**Blue Channel**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 145 | 120 | 154 | 131 | 227 | 173 | 240 | 238 |
| 154 | 121 | 153 | 142 | 161 | 116 | 198 | 243 |
| 204 | 76 | 163 | 140 | 0 | 104 | 12 | 248 |
| 199 | 122 | 126 | 122 | 78 | 87 | 101 | 252 |
| 31 | 113 | 119 | 138 | 23 | 124 | 119 | 253 |
| 20 | 106 | 74 | 104 | 24 | 141 | 64 | 254 |
| 99 | 94 | 15 | 25 | 27 | 96 | 44 | 255 |
| 95 | 132 | 109 | 10 | 19 | 15 | 68 | 67 |

# An image is a 2D signal




We investigate spatial relationships between pixels




(where here your x is time)

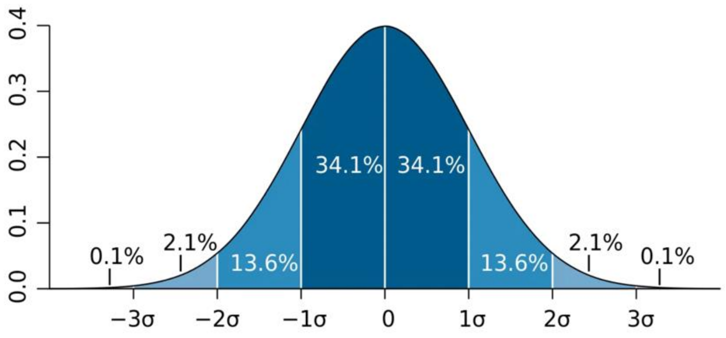
# 2. NOISE

# Noise in an image

**Noise:** as for 1D signal, noise is a variation that perturbs the signal clarity (or message that the image conveys), so it makes the interpretation harder.

**Filtering:** clean the noise.

# Gaussian Noise in an image

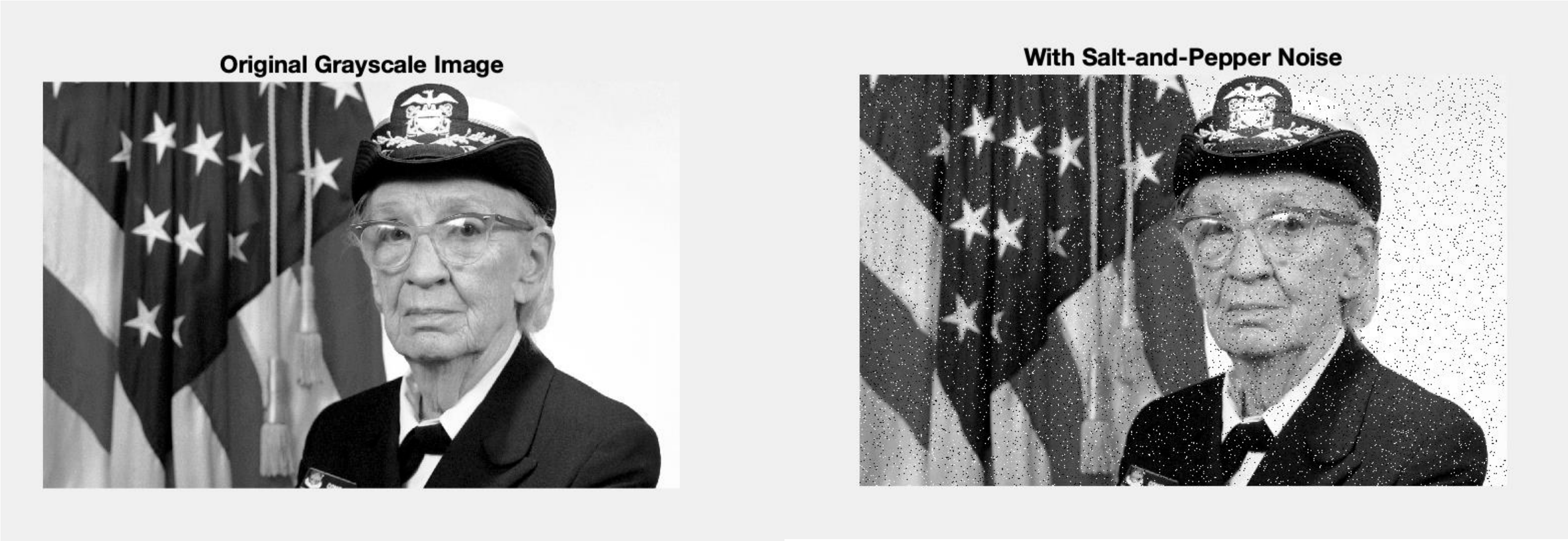



**Noise:** to all pixel values, random noise sample from a Gaussian distribution is added – so that it appears as random (but smooth) variations across the image.


https://wiki.cloudfactory.com/docs/mp-wiki/augmentations/gaussian-noise

# Salt and Pepper Noise in an image

**Noise:** randomly changes some pixels to either the maximum intensity (salt) or the minimum intensity (pepper) – this is caused by data transmission errors, for example.



# 3. FILTERING IN THE SPATIAL DOMAIN

# Image preprocessing techniques in the spatial domain

**Filtering:** we apply filtering to manipulate pixel values in an image, in order to make certain features appearing, such as edges, or to erase noise.

| Category | Technique | Purpose | Examples |
|---|---|---|---|
| **Smoothing** | Mean Filter | Reduces Gaussian noise, smooths image. | Applies uniform averaging to pixel neighborhoods, but blurs edges. |
| | Gaussian Filter | Reduces noise while preserving edges. | Uses a Gaussian kernel for weighted averaging (smooths without excessive blurring). |
| | Median Filter | Removes salt-and-pepper noise, preserves edges. | Replaces pixel with the median value of its neighbors. |
| | Bilateral Filter | Smooths image while keeping edges sharp. | Combines spatial and intensity weighting to prevent edge blurring. |
| **Sharpening** | Laplacian Filter | Highlights edges and fine details by enhancing high frequencies. | Computes second-order derivatives to detect intensity changes. |
| | Unsharp Masking | Enhances details while keeping the image natural. | Subtracts a blurred image from the original, amplifies differences, and adds them back. |
| | Gradient Filters | Detects edges (intensity gradients). | Sobel, Prewitt operators calculate gradients in horizontal or vertical directions. |
| **Noise Reduction** | Gaussian Filter | Reduces random noise. | Blurs high-frequency noise using a Gaussian kernel. |
| | Median Filter | Removes noise while preserving edges. | Excellent for removing salt-and-pepper noise. |
| | Bilateral Filter | Reduces noise, preserves edges. | Suitable for applications like medical imaging or facial feature smoothing. |
| **Contrast Enhancement** | Histogram Equalization | Enhances global contrast. | Redistributes pixel intensities to span the full intensity range. |
| | Adaptive Histogram Equalization (AHE) | Improves local contrast, enhances details in varying lighting. | Divides the image into smaller regions (tiles) for local intensity redistribution (e.g., CLAHE). |

# Smoothing filtering (low-pass filtering): mean filtering

**Smoothing:** Smoothing reduces noise and removes sharp intensity variations by averaging pixel intensities.

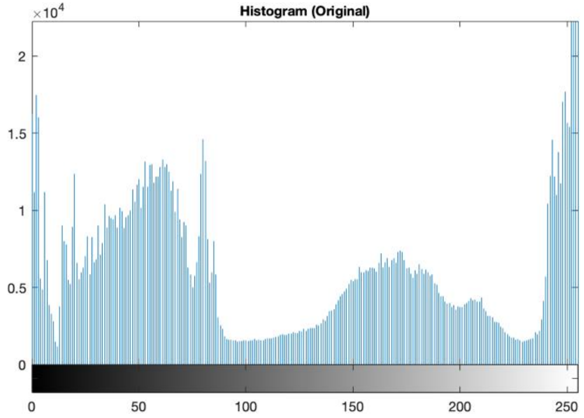
**Mean filtering:**

- Replaces each pixel with the average of its neighbourhood.
- **Kernel Example**:

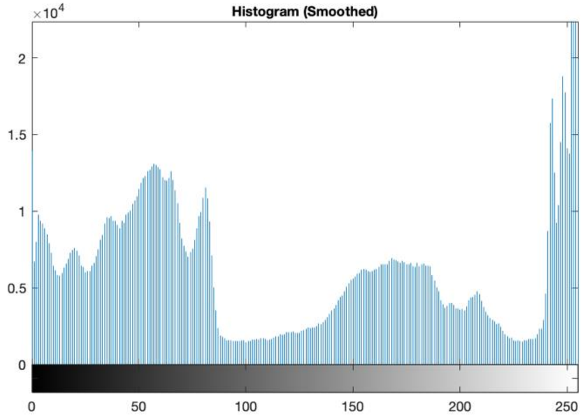
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

- **Use Case**: Reduces Gaussian noise but blurs edges.

Original Grayscale Image




Smoothed Image (Mean Filter)



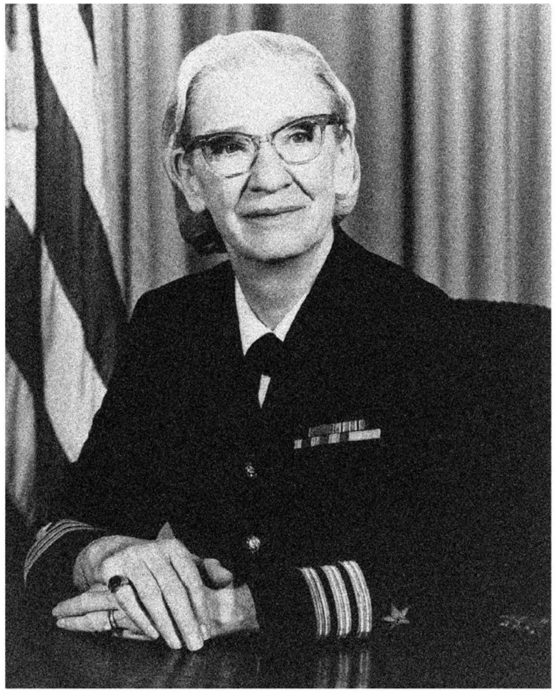
# Smoothing filtering (low-pass filtering): Gaussian filtering. It is also a noise reduction filtering

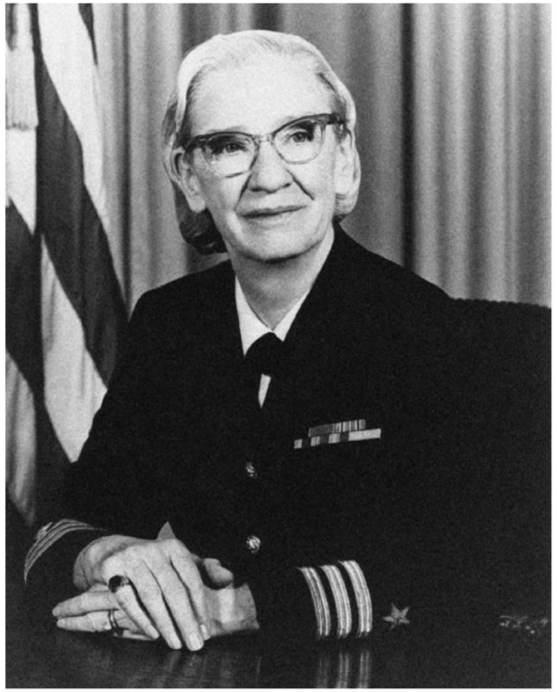
**Smoothing:** Smoothing reduces <u>noise</u> and removes sharp intensity variations by averaging pixel intensities.

**Gaussian filtering:**

- Uses a Gaussian-shaped kernel for weighted averaging, preserving edges better than the mean filter

- **Use Case**: Preserves edges and can also remove the noise.

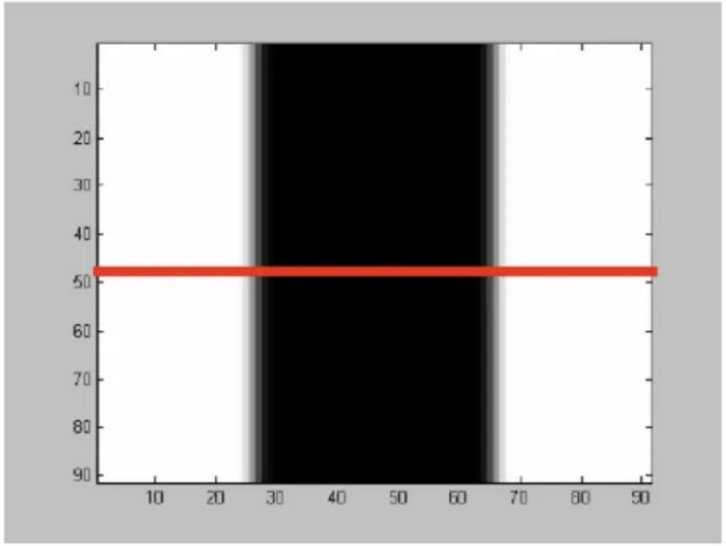




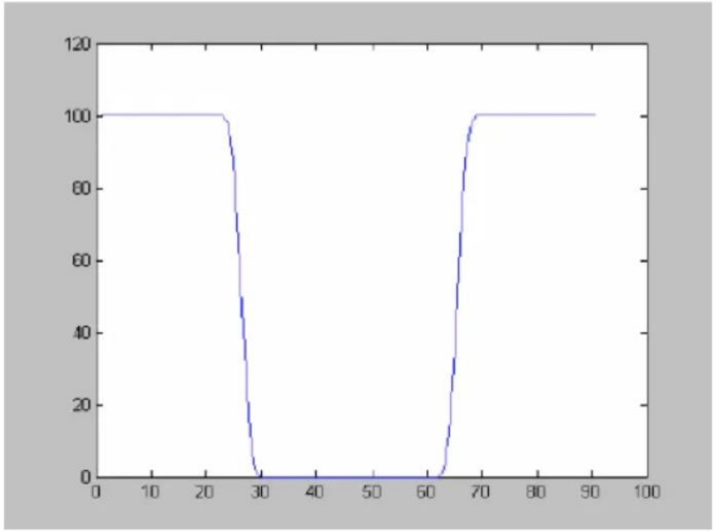
# Sharpening filtering (high-pass filtering): Gradient

**Sharpening filters (gradient):** detect edges by computing intensity gradients along image (x,y) direction (or a combination of both).

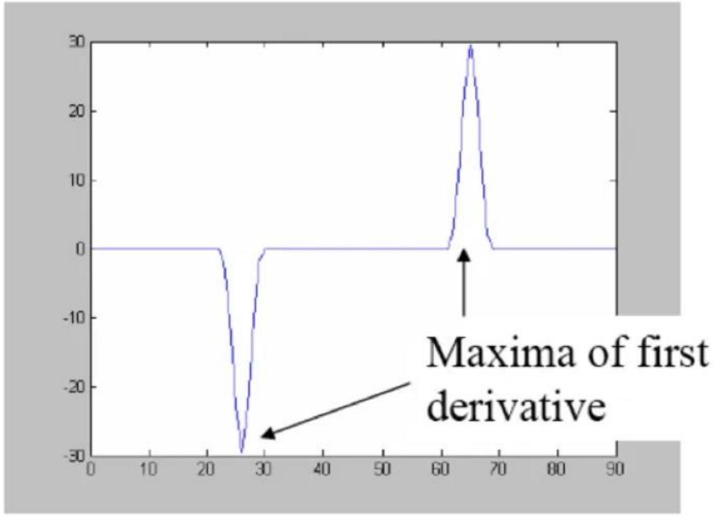
**Used for Edges detection (or features detection)**



**Image**



**Intensities of pixels on the red line**




**1st derivative of pixel intensities**

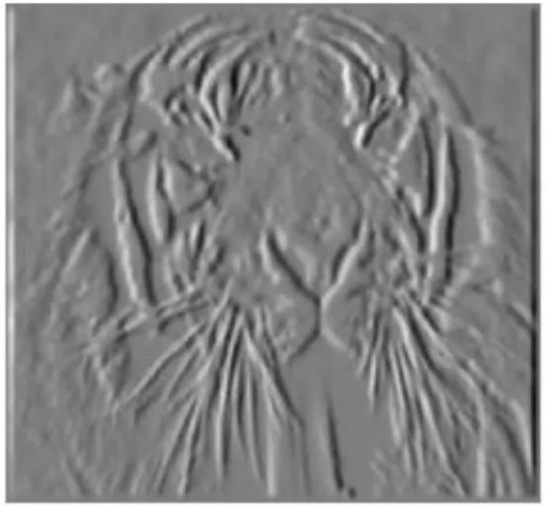
https://medium.com/jun94-devpblog/cv-3-gradient-and-laplacian-filter-difference-of-gaussians-dog-7c22e4a9d6cc

# Sharpening filtering (high-pass filtering): Gradient

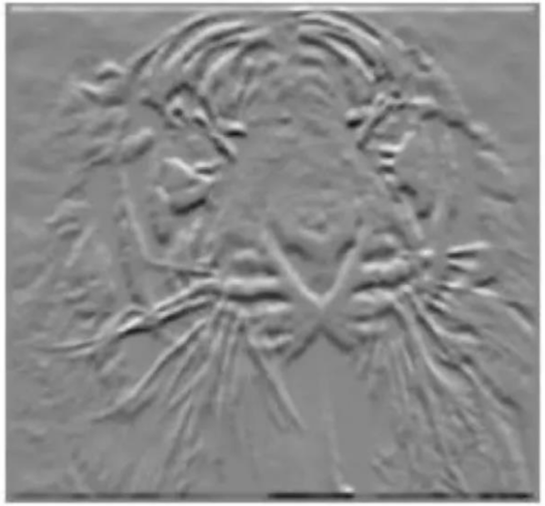
**Sharpening filters (gradient):** detect edges by computing intensity gradients along image (x,y) direction (or a combination of both).

**Used for Edges detection (or features detection)**

**Image *f***



$$\frac{\partial f(x,y)}{\partial x}$$



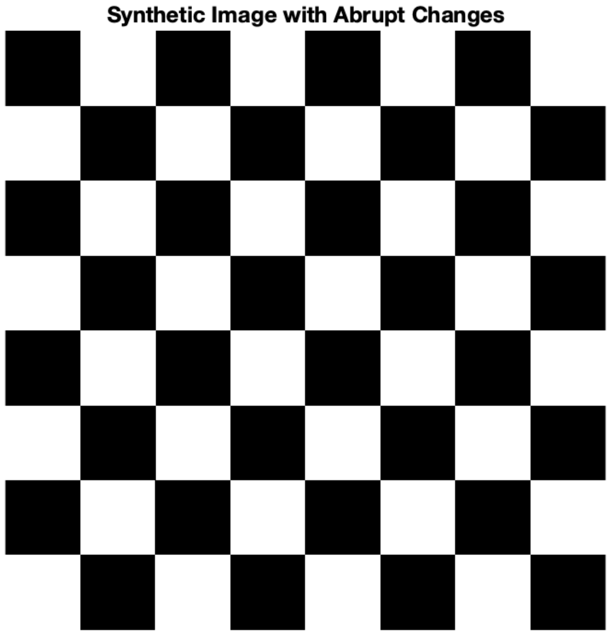
$$\frac{\partial f(x,y)}{\partial y}$$

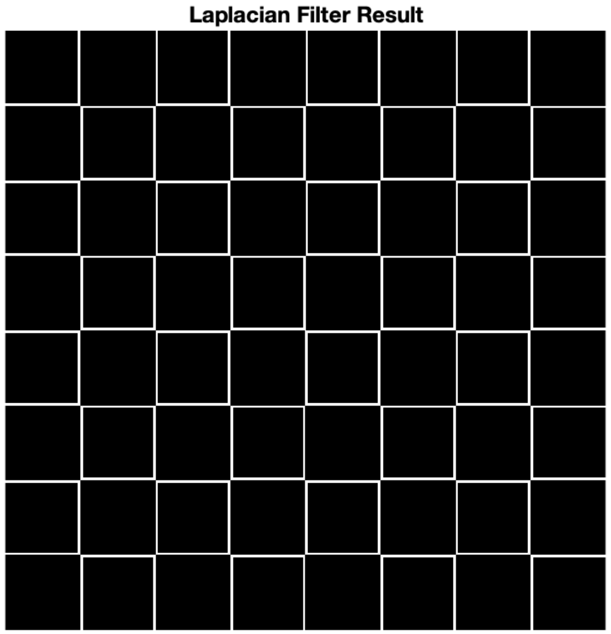
# Sharpening filtering (high-pass filtering): Laplacian

**Sharpening filters (Laplacian):**
another edge detector, it highlights regions where pixels intensity dramatically change.

**Matrix kernel:**

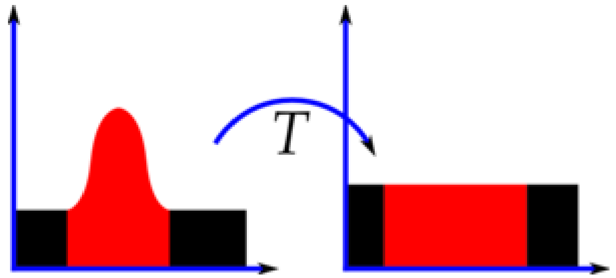
| | | |
|---|---|---|
| 0 | −1 | 0 |
| −1 | 4 | −1 |
| 0 | −1 | 0 |



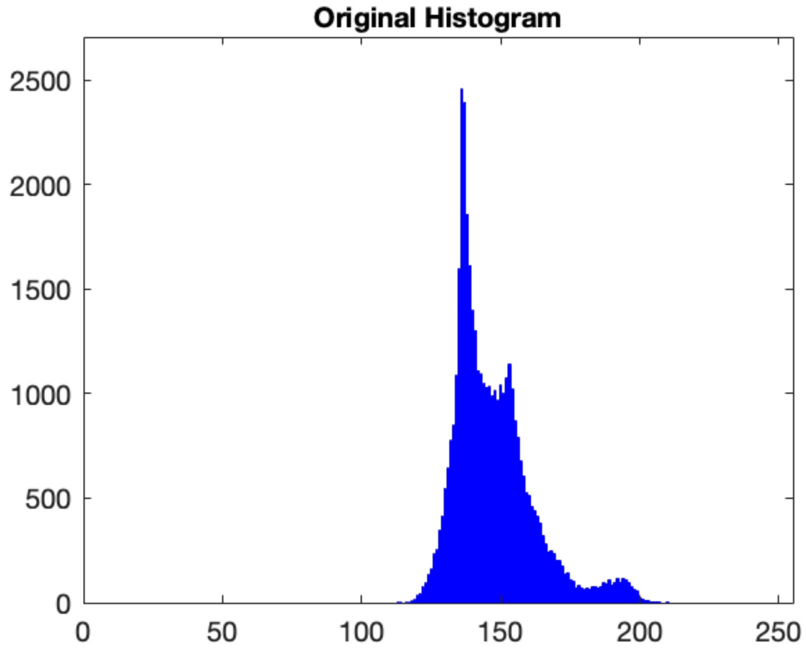



# Contrast enhancement via histogram equalization

**Enhancement:** we redistribute the pixel intensity over the entire available range. To do so, we map the comulative distribution of the intensity values to a linear distribution. **This enhances (increase the contrast) of the image.**



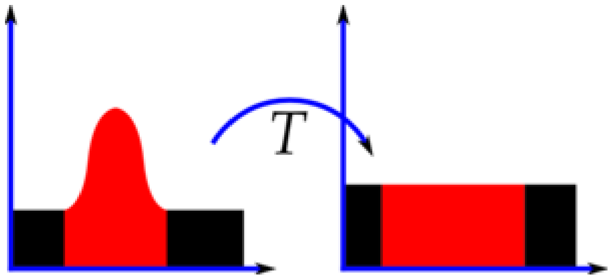




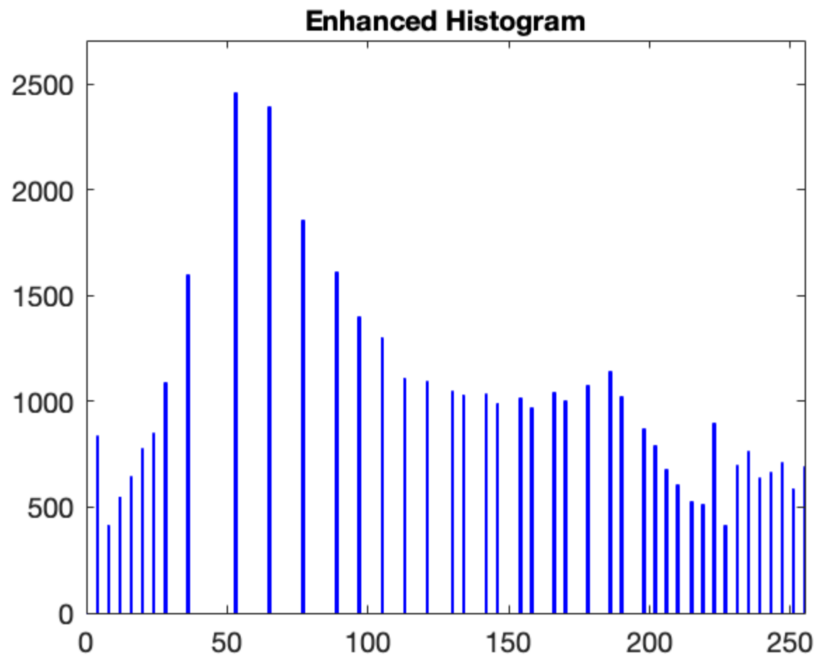

https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html

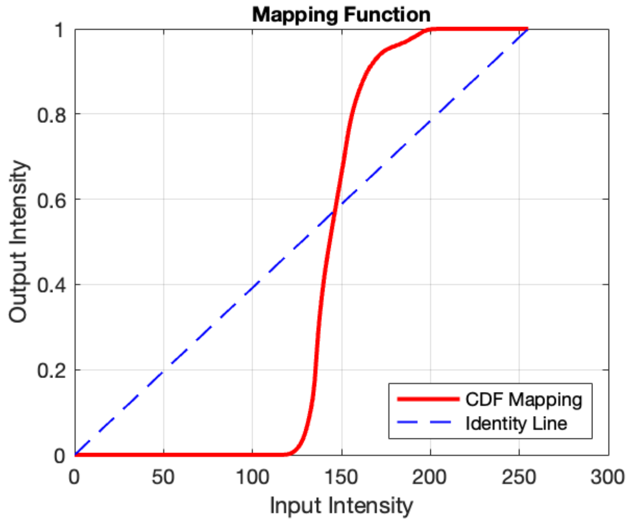
# Image preprocessing techniques: enhancement

**Enhancement:** we redistribute the pixel intensity over the entire available range. To do so, we map the comulative distribution of the intensity values to a linear distribution. **This enhances (increase the contrast) of the image.**







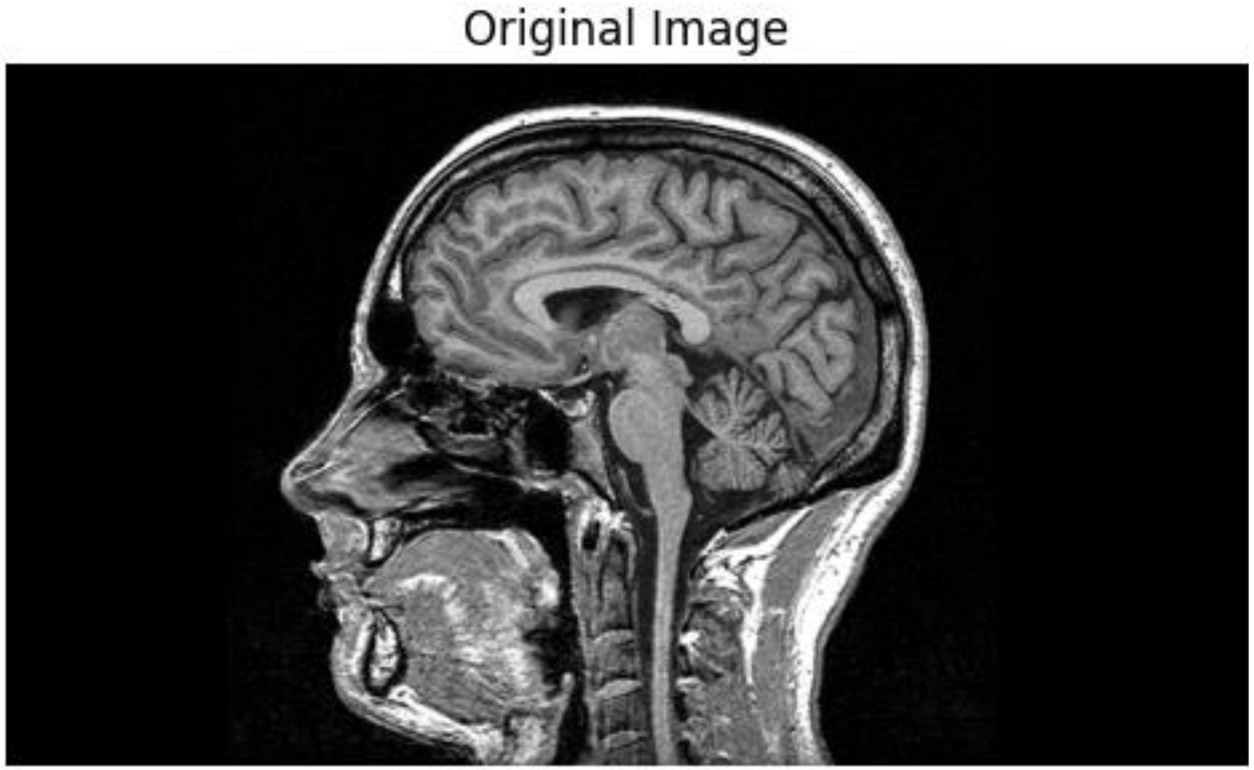



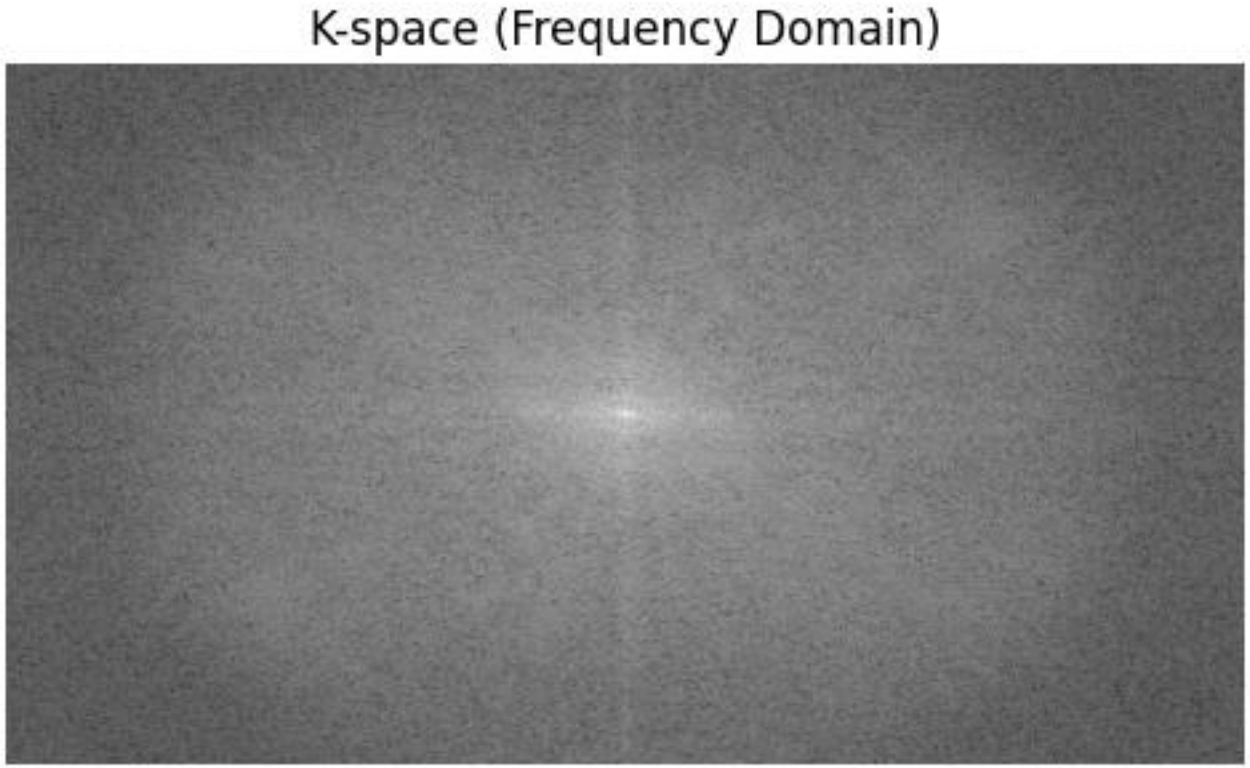

https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html

# 4. FILTERING IN THE FREQUENCY DOMAIN

# Frequency domain analysis: recap FT on images

**<u>Low frequencies</u>**: responsible for the slowly changing patterns, such as the average values and smooth transitions.
**<u>Higher (increasing) frequencies</u>**: sharp changes (edges)



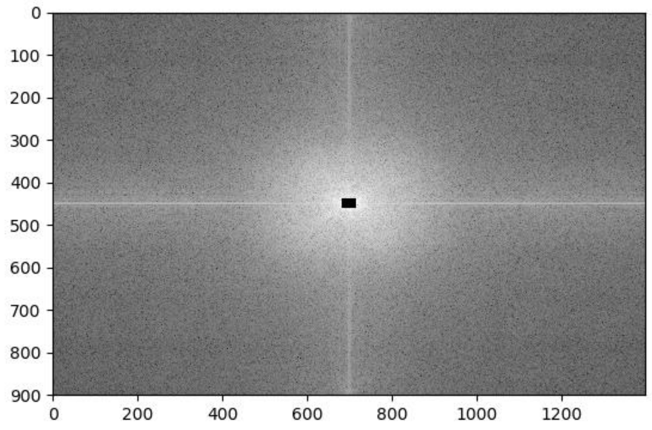

[Go back on the DFT tutorial from Mauro!]

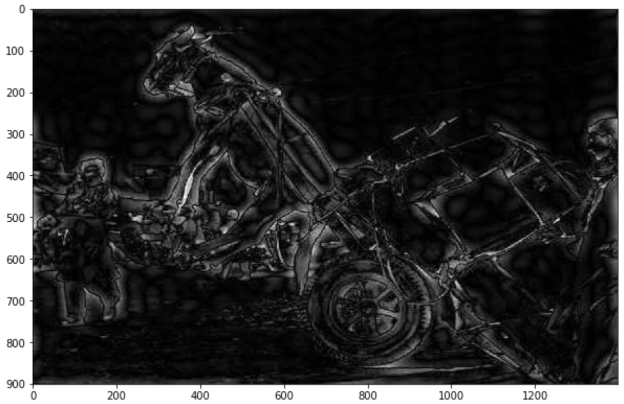
# Low-pass and high-pass filtering (here high-pass)

IMAGE → DFT → Frequency Domain → MASK → Masked Freq. Domain

We shift 0 frequency to center

Inverse DFT


https://sbme-tutorials.github.io/2018/cv/notes/3_week3.html#ideal-lpf-example

# Code break (20 minutes)

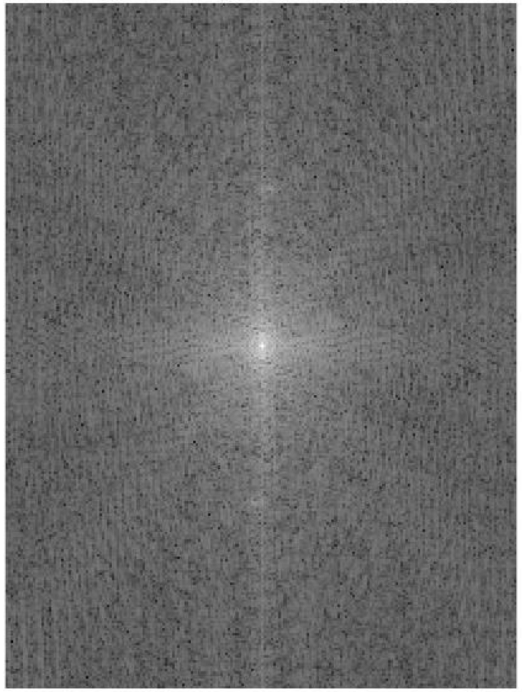
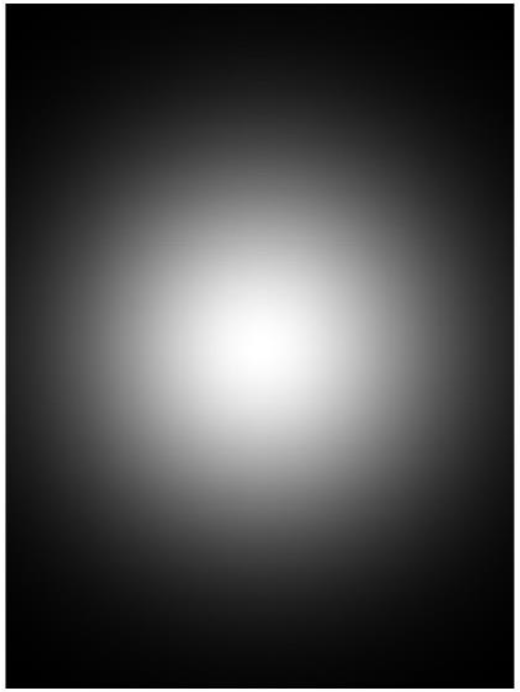
**Implement:**

- A low-pass filtering in the previous image
- A Gaussian shaped filter that attenuates higher frequency (and that **allows** low frequency to pass)
- A **band-pass filter**: meaning we allow frequencies within a specific range to pass

# Code break (20 minutes)

**Implement:**

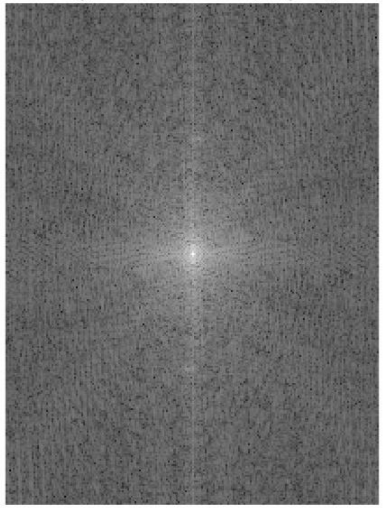
- A Gaussian shaped filter that attenuates higher frequency (and that **allows** low frequency to pass)

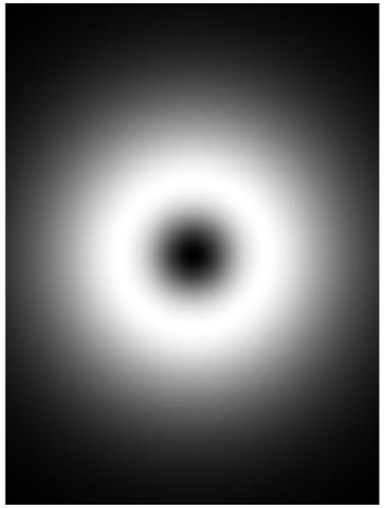

**Implement:** a **band-pass filter**: meaning we allow frequencies within a specific range to pass

Original Image

Magnitude Spectrum (Original)

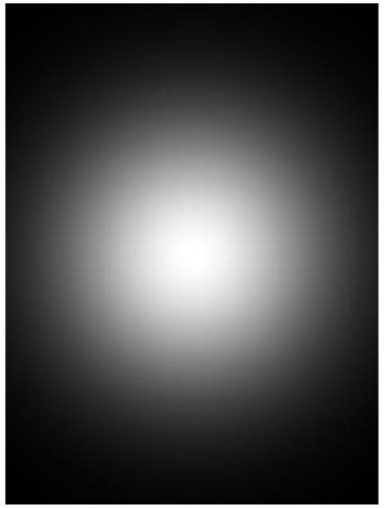


Band-Pass Filter

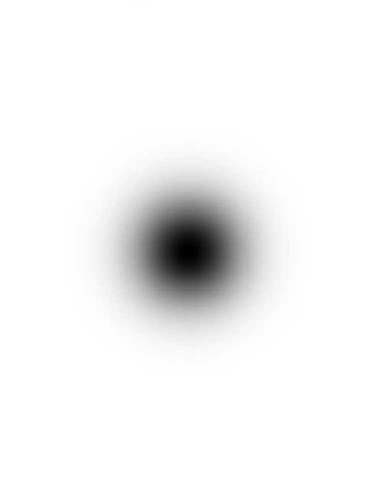


Filtered Image (Band-Pass)

Low-Pass Filter



High-Pass Filter



# Spatial vs. Frequency domain processing comparison

| **Aspect** | **Spatial Domain** | **Frequency Domain** |
| --- | --- | --- |
| Operation | Local filtering with kernels (e.g., convolution). | Global filtering of frequency components. |
| Focus | Works on pixel intensities. | Works on frequency representation of the image. |
| Applications | Smoothing, sharpening, edge detection. | Noise reduction, smoothing, periodic pattern removal. |
| Advantages | Easier to implement and interpret. | Allows precise control over global features. |
| Limitations | Ineffective for global patterns or periodic noise. | Computationally intensive for large images. |

# 5. FEATURES EXTRACTION

# What does it mean

**Features extraction:** the process of identifying and extracting relevant information (features) from an image to facilitate understanding, analysis, or classification. Features are specific attributes or patterns within the image, such as edges, corners, or textures.

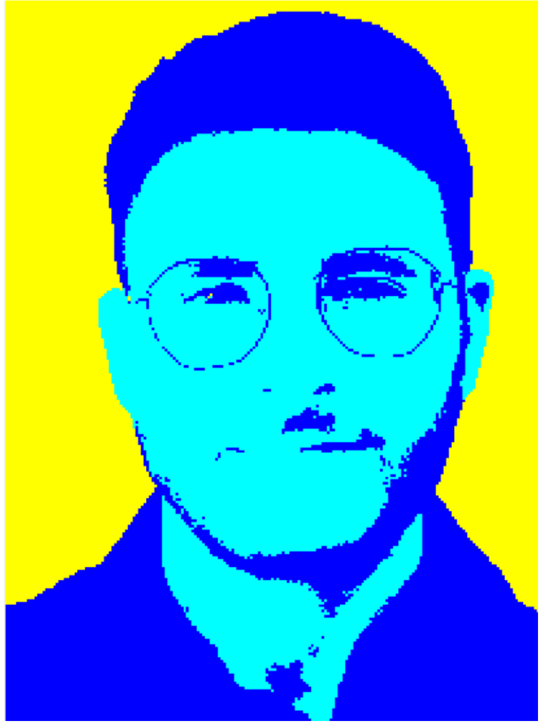
Example: Edge detection as feature detection (**Canny – popular filter**)

Original Grayscale Image

Canny Edge Detection

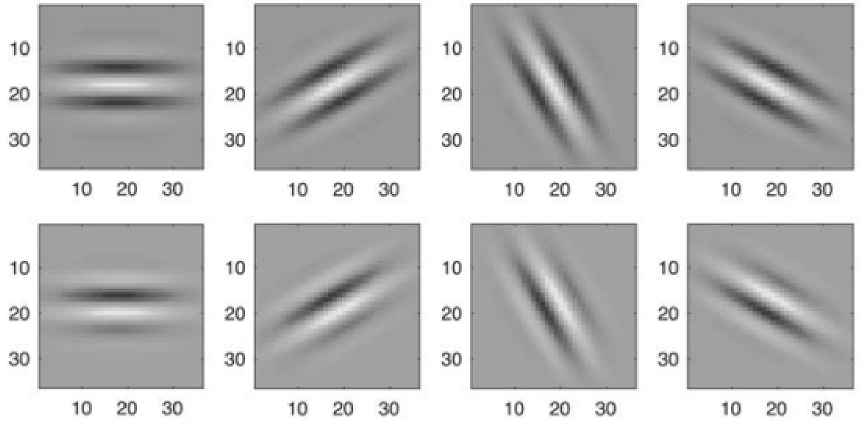
# Segmentation

**Jaime's tutorial**



# Gabor filters: extracting orientation

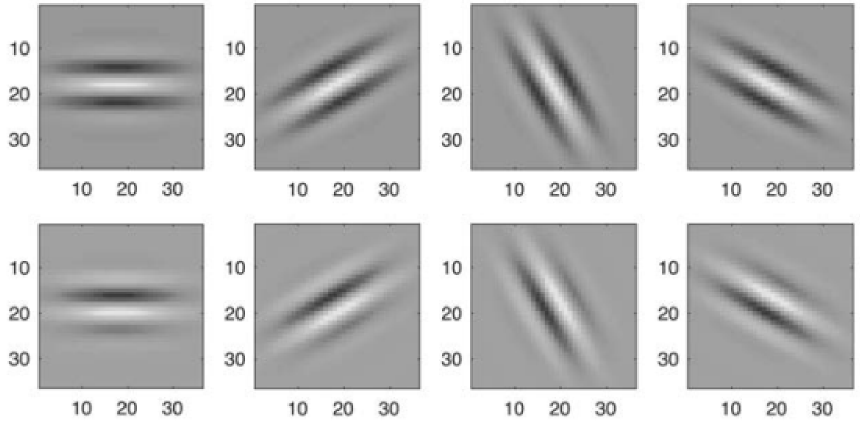
Gabor filter code: [link]




In each image: (top) even part of Gabor filters (real part), (bottom) odd one (imaginary part). Corresponding orientation from left to right: $\theta = 0$, $\theta = \pi/6$, $\theta = 2\pi/3$, $\theta = 5\pi/6$, with $\sigma = 4.48$ pixels

# Gabor filters: extracting orientation

Gabor filter code: link




In each image: (top) even part of Gabor filters (real part), (bottom) odd one (imaginary part). Corresponding orientation from left to right: $\theta = 0$, $\theta = \pi/6$, $\theta = 2\pi/3$, $\theta = 5\pi/6$, with $\sigma = 4.48$ pixels