Applications of & Introduction to Artificial Intelligence

Image Processing Basics

Technische Hochschule Rosenheim

Sommer 2020

Prof. Dr. J. Schmidt

Image Recognition is complicated …

Chihuahua oder Muffin?

https://twitter.com/teenybiscuit/status/707727863571582978

Applications – Examples

Quality Inspection

From: S.R. Dubey and A. S. Jalal. "Application of image processing in fruit and vegetable analysis: A review." *Journal of Intelligent Systems* 24(4): 405-424, 2015.

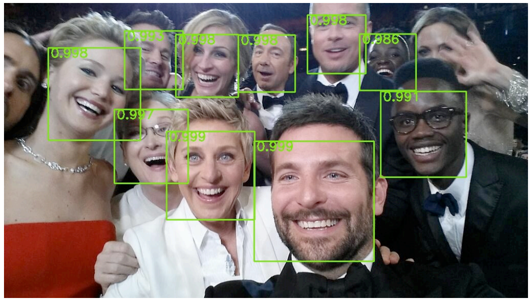
License-Plate Recognition

Brenner toll station, © Peter Müller, CC BY-SA 3.0 DE

Optical Character Recognition (OCR)

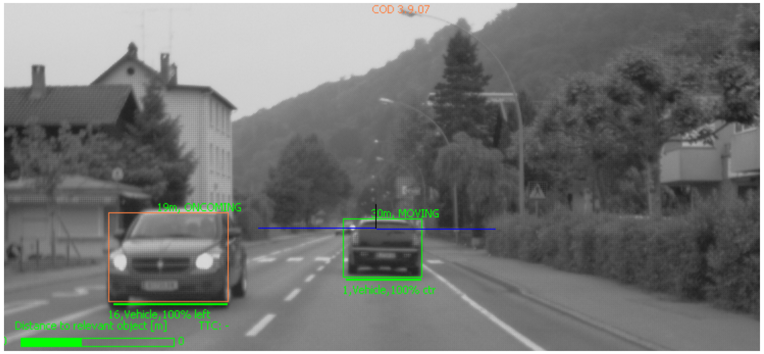
MNIST

Face Recognition



S.S. Farfade, M. J. Saberian, and L.-J. Li. *Multi-view face detection using deep convolutional neural networks.* Proc. 5th ACM International Conference on Multimedia Retrieval. ACM, 2015.

Autonomous Driving



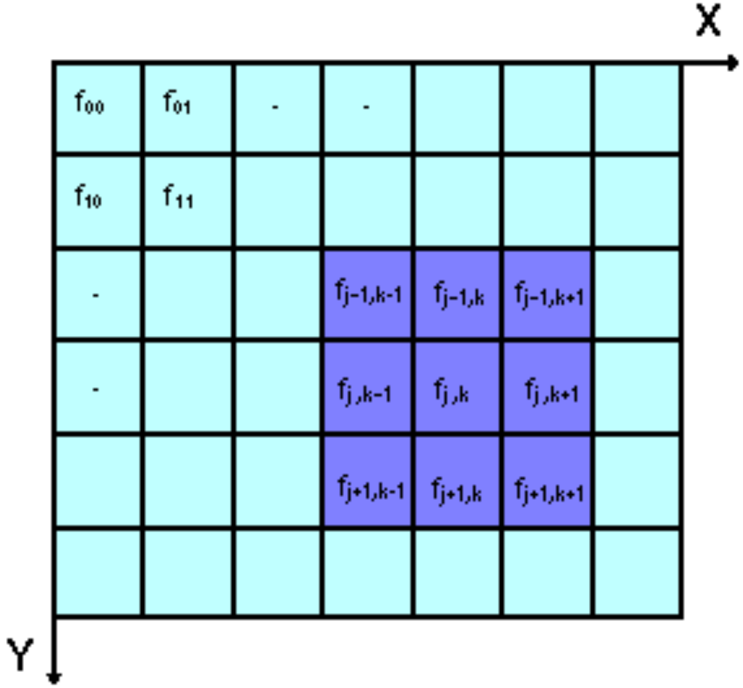
Images: Continental, A.D.C. GmbH, Lindau

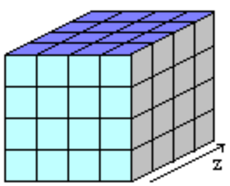
Medical

Some APIs

- scikit-image
  https://scikit-image.org/
- OpenCV
  https://opencv.org/
- Matlab
  https://de.mathworks.com/products/matlab.html

Digital Images





Volume Image

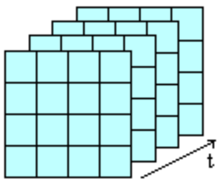
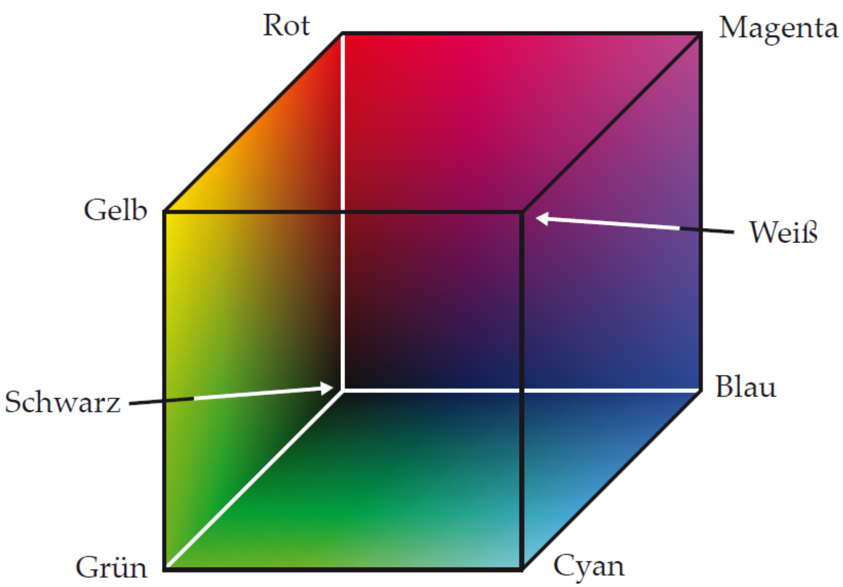


Image Sequence (Video)

- Binary Image
  1 Bit / Pixel
- Grayscale Image
  8 Bit / Pixel
  10, 12, 16 Bit / Pixel
- Color Image (RGB)
  8 Bit / channel
  10, 12, 16 Bit / channel

RGB Color Space



Beyerer, J., Puente Leon, F., Frese, Ch.: ***Automatische Sichtprüfung***, Springer Vieweg, 2012

- **additive** Mixing: Red (R), Green (G), Blue (B)
- Range: 0 – 1
- Applications: Monitors, cameras, projectors
- RGB-color spaces and color range different from device to device
- device-independent: sRGB
- many other color spaces in use: YUV, YC_BC_R, HSV, HSL, HSI, CIE XYZ, CIELUV, CIELAB, …
- **subtractive** Mixing: CMYK
  - Printers

Color – Grayscale Conversion

Standard Test Image: Lena Söderberg, © Playboy Nov. 1972

- if possible as a first step: Color → Grayscale image
  - even better: camera delivers grayscale image directly
  - less data to process
  - less problems with many algorithms (e.g. filtering)

LINEAR FILTERS

Basic Principle

original image

+ filter mask

convolution

filtered image

$$f'(x,y) = \sum_{x} \sum_{y} f(x,y) g(i-x, j-y)$$

filtered image original image filter mask

- mirror filter mask horizontally and vertically,
- move filter mask over image,
- compute weighted sum.

$$f' = f * g$$

Low-pass Filter

- removes high frequencies
- reduces image noise
- results in smoothing of image

3x3 Mean $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

3x3 Gaussian $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Low-pass Filter

5x5 Mean $\frac{1}{25}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

5x5 Gaussian (with $\sigma = 1$) $\frac{1}{273}\begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}$

Influence of Filter Size

Mean filter, Sizes: 3x3, 5x5, 11x11, 21x21

Original

Caution with Color Images!

- filter each RGB-channel separately
  - but: RGB is unsuitable for linear interpolation
- better: use CIELUV or CIELAB color space
  - but: Conversion from RGB is computationally expensive (non-linear)

Mean 11x11

Gaussian 11x11, $\sigma = 5$

High-pass Filter

- removes low frequencies
- edge detection
- widely used:
  - Sobel
    - based on computing the first order partial derivatives
    - result: two edge images (horizontal and vertical direction)
    - edges = large values (maxima of derivative)
  - Laplace
    - based on computing the second order derivatives (the Laplace operator)
    - edges = zero crossings
    - more subjective to noise compared to Sobel

Sobel

- first order partial derivatives

$$f_x = \frac{\partial f(x,y)}{\partial x} \qquad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad f_y = \frac{\partial f(x,y)}{\partial y} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Range?

- gradient strength:

$$s = \sqrt{f_x^2 + f_y^2}$$

- gradient direction:

$$\theta = \arctan\left(\frac{f_y}{f_x}\right)$$

Note: use atan2(y, x)

High-pass Filter – Sobel

horizontal

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

vertical

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

(images inverted)

Laplace

- Laplace-operator (from 2. order partial derivatives)

$$\Delta f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

- different variants

8-neighborhood

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

4-neighborhood

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

with Gaussian smoothing

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

Range?

High-pass Filter – Laplace

Laplace $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

combined Sobel
(gradient strength image)

(images inverted)

Remarks

- Filter masks typically have odd size (3x3, 5x5, …) → symmetric about the current pixel
- they are not necessarily square
- there are other widely used standard masks that are not presented here (Prewitt, Harris, Robert‘s Cross, …)

- Design of special-purpose filters is possible
- filters can be concatenated, resulting a single new linear filter mask (cf. convolution equation)

- in image processing, convolution is usually computed in image space
- convolution using FFT makes sense with large filter sizes only

NON-LINEAR FILTERS

Rank-Order Operations

- use a structuring element
  - defines a window for neighborhood
  - comparable to filter mask for linear filters

- **sort** gray-values by size (within the neighborhood)

  $$R_{ij} = \{r_1, r_2, \ldots, r_M \mid r_k \leq r_{k+1} \forall k = 1, \ldots, M\}$$

- result: ranking of gray-values

- Rank-order operation: Function on R_{ij}

  $$h_{ij} = g(R_{ij})$$

Rank-Order Operations – Examples

- Median $h_{ij} = r_{(M+1)/2}$
- kNN-Mean h_{ij} = Mean of k nearest values around Median
- Erosion $h_{ij} = r_1$
- Dilation $h_{ij} = r_M$
- Contour extraction $h_{ij} = r_M - r_1$
- Contrast enhancement $h_{ij} = \begin{cases} r_1 & \text{if } f_{ij} - r_1 < r_M - f_{ij} \\ r_M & \text{otherwise} \end{cases}$

Comparison Median – Gaussian

5x5 Median

5x5 Gaussian

Median preserves edges!

Comparison Median – Gaussian

5x5 Median

5x5 Gaussian

Median preserves edges!

Comparison Median – Gaussian

5x5 Gaussian

Original

noisy image

5x5 Median

Contour Extraction (5x5)

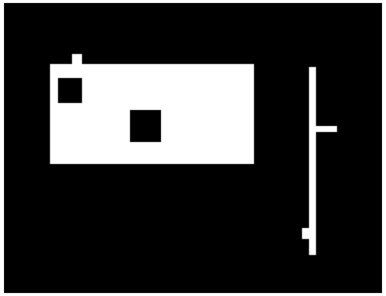
$$h_{ij} = r_M - r_1$$

(inverted)

Erosion/Dilation

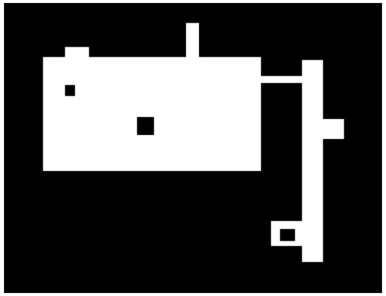
”morphological operations”

Erosion

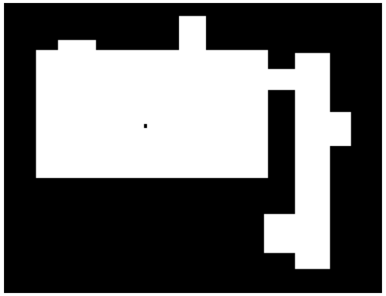


$$h_{ij} = r_1$$

Original



Dilation

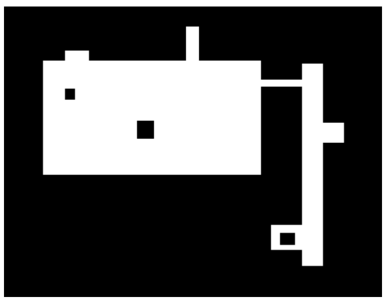


$$h_{ij} = r_M$$

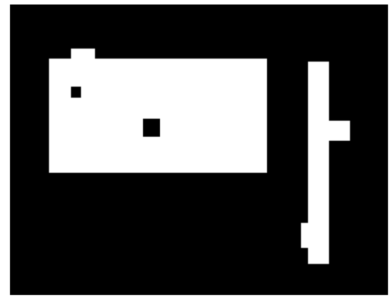
Technische
Hochschule
Rosenheim

Opening & Closing

Original

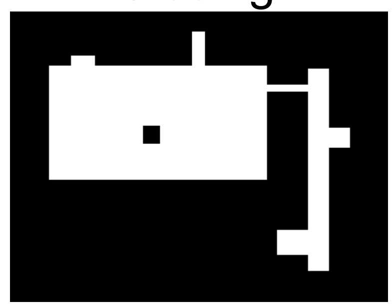


Opening



Erosion followed by dilation

Closing



Dilation followed by erosion

Color Images?

- Low-pass filter
  - filter channels separately
    - better use color spaces like CIELAB or CIELUV instead of RGB
  - combine all channels using tensors ($\rightarrow$ CNN)
    - does not really solve the RGB-problem (linear operation)
    - but results in single value combining all channels (no color image)

- High-pass filter
  - filtering channels separately does not really make sense
  - combine all channels using tensors ($\rightarrow$ CNN)

- Rank-order operations/Median: complicated, because
  - channels cannot be handled independently
    (this would not be a Median anymore)
  - vectors do not have an ordering relation