Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

# *Computer Vision - Image Processing Basics*

WS 2019/2020

Prof. Dr. Simone Frintrop

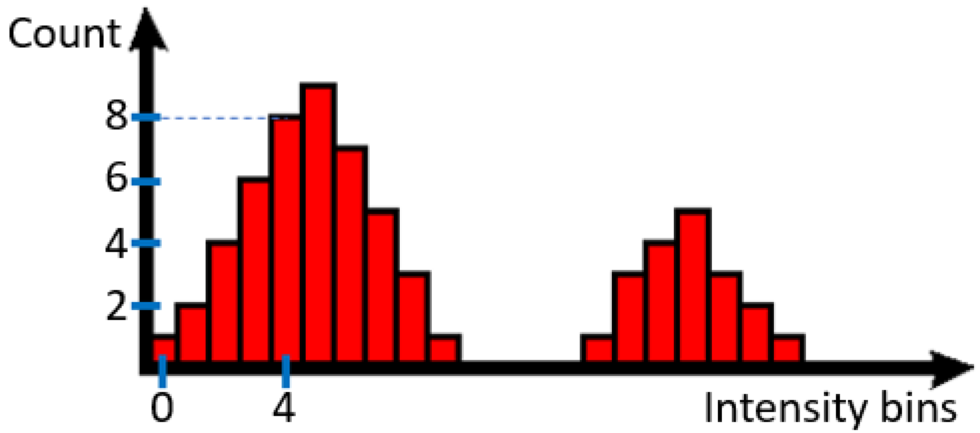
Computer Vision Group, Department of Informatics
University of Hamburg, Germany

# *Outline*

- Histogram (graylevel, normalized, color histograms)
- Metrics for comparing histograms
- Histogram-based recognition
- Evaluations in Computer Vision
- Basic Statistics on Images (Mean, Variance)

# *Histogram*

An *image histogram* is a graphical representation of the distribution of values (like intensity or color) in a digital image (it models a probability distribution if normalized)



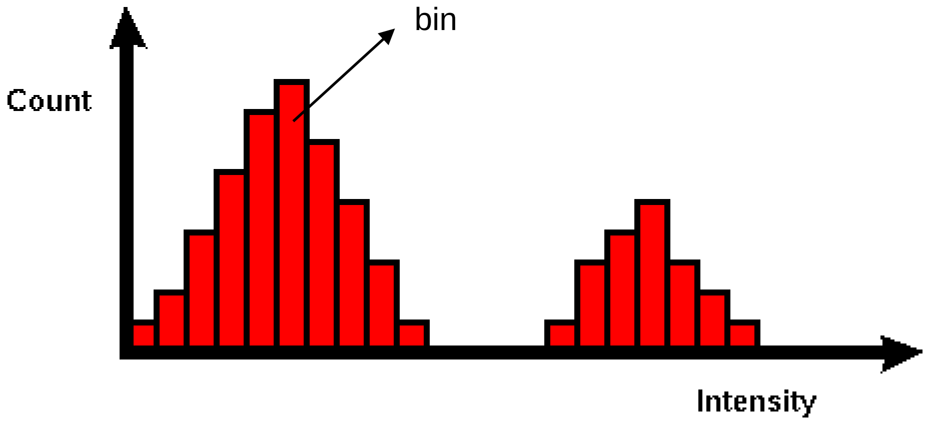

[Image: http://homepages.inf.ed.ac.uk/rbf/HIPR2]

# *Histogram*

**Definition**: Let $r_k$, $k = 0, 1, \ldots, L-1$ denote the intensities of a digital image $f(x, y)$. The unnormalized *gray value histogram* is defined as

$$h(r_k) = n_k$$

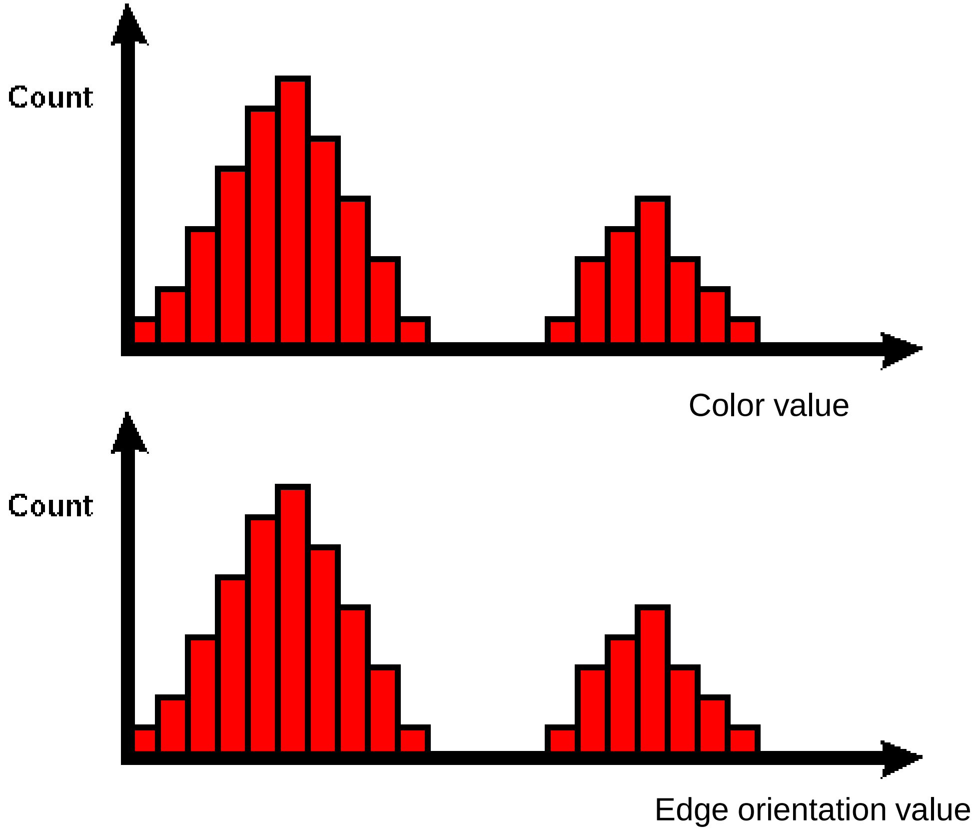
where $r_k$ is the $k$-th intensity value and $n_k$ is the number of pixels in $f$ with intensity $r_k$. The subdivisions of the intensity scale are called *bins*.




[Image: http://homepages.inf.ed.ac.uk/rbf/HIPR2]

# Histogram

Side note: There are also color histograms, orientation histograms, etc.



# Histograms for Two Image Windows


Two 104 × 98 windows in image Yan and corresponding histograms

[Klette 2014]

# Histograms

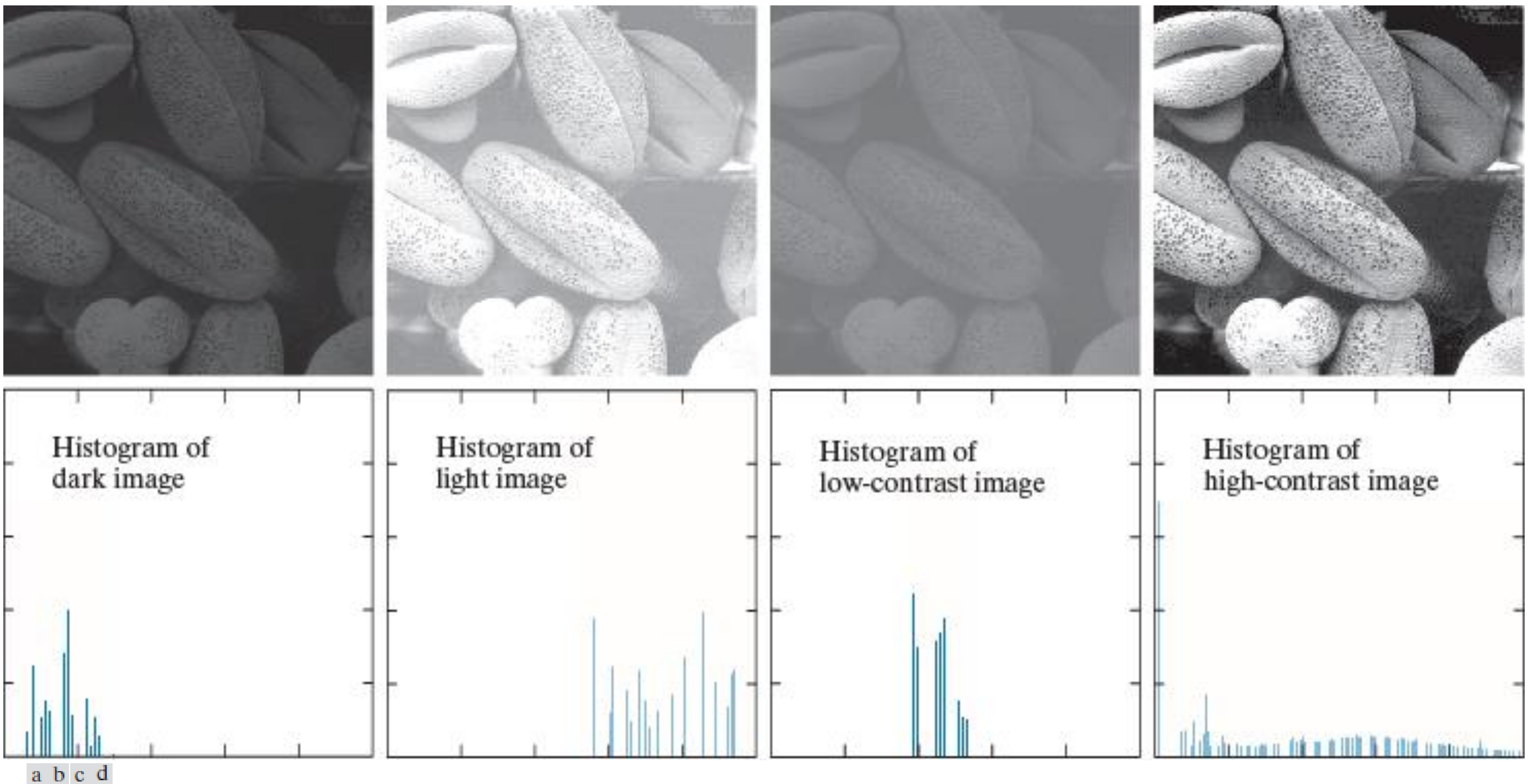
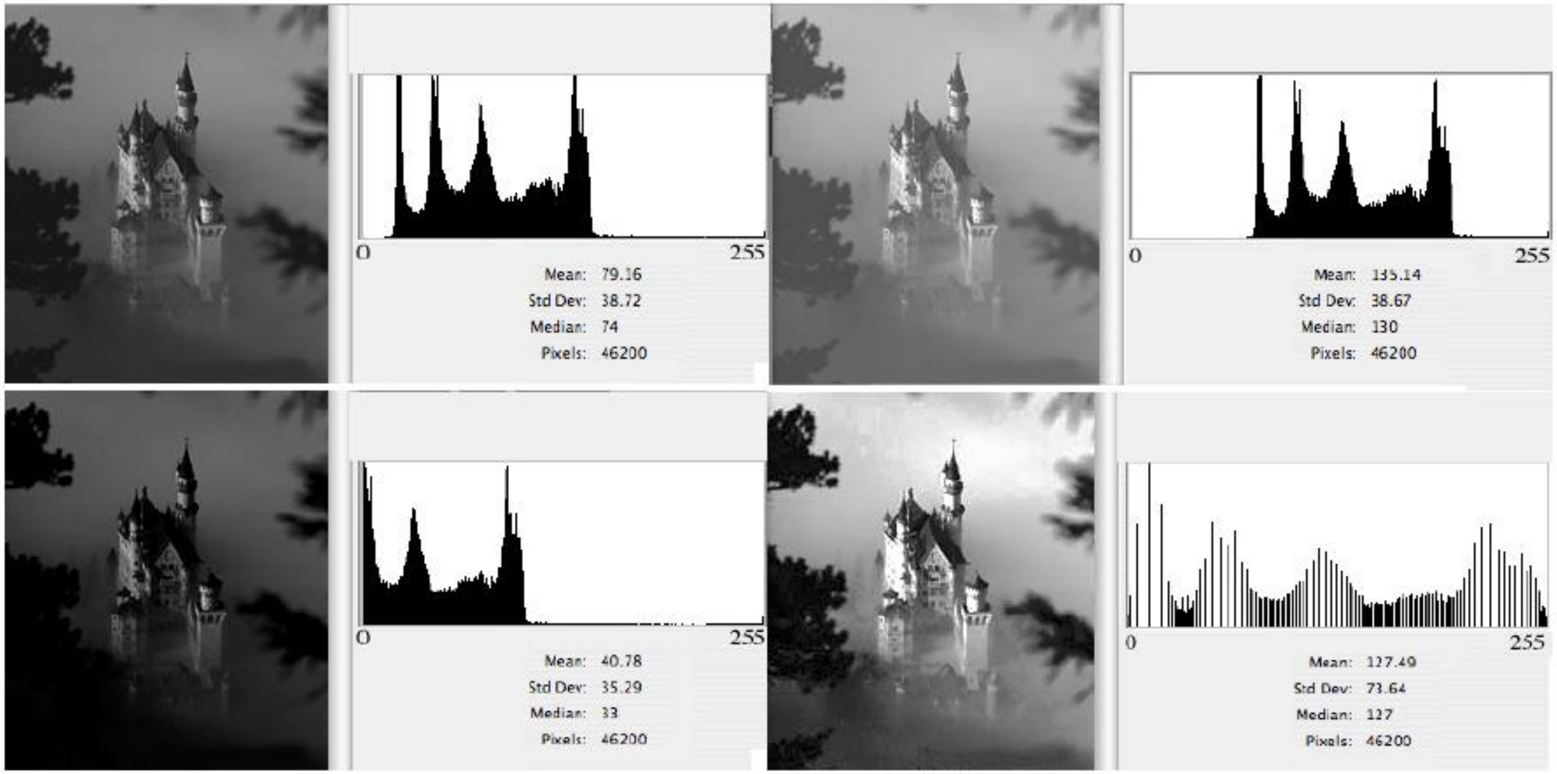



**FIGURE 3.16** Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of $r_k$ and the vertical axis are values of $p(r_k)$.

[Gonzales/Woods]

# Histograms




Histograms for a $200 \times 231$ image `Neuschwanstein`

*Upper left*: Original image. *Upper right*: Brighter version. *Lower left*: Darker version. *Lower right*: After histogram equalization

[Klette]

# Histogram

A *normalized histogram* is a histogram in which the value $n_k$ in each bin is divided by the number of pixels in the image:

**Definition**: Let $r_k$, $k = 0, 1, ..., L-1$ denote the intensities of a digital image $f(x, y)$. The *normalized histogram* of $f$ is defined as

$$p(r_k) = \frac{n_k}{MN}$$

where $n_k$ is the number of pixels in $f$ with intensity $r_k$ and $M$ and $N$ are the number of rows and columns respectively.

http://homepages.inf.ed.ac.uk/rbf/HIPR2

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

# Histograms

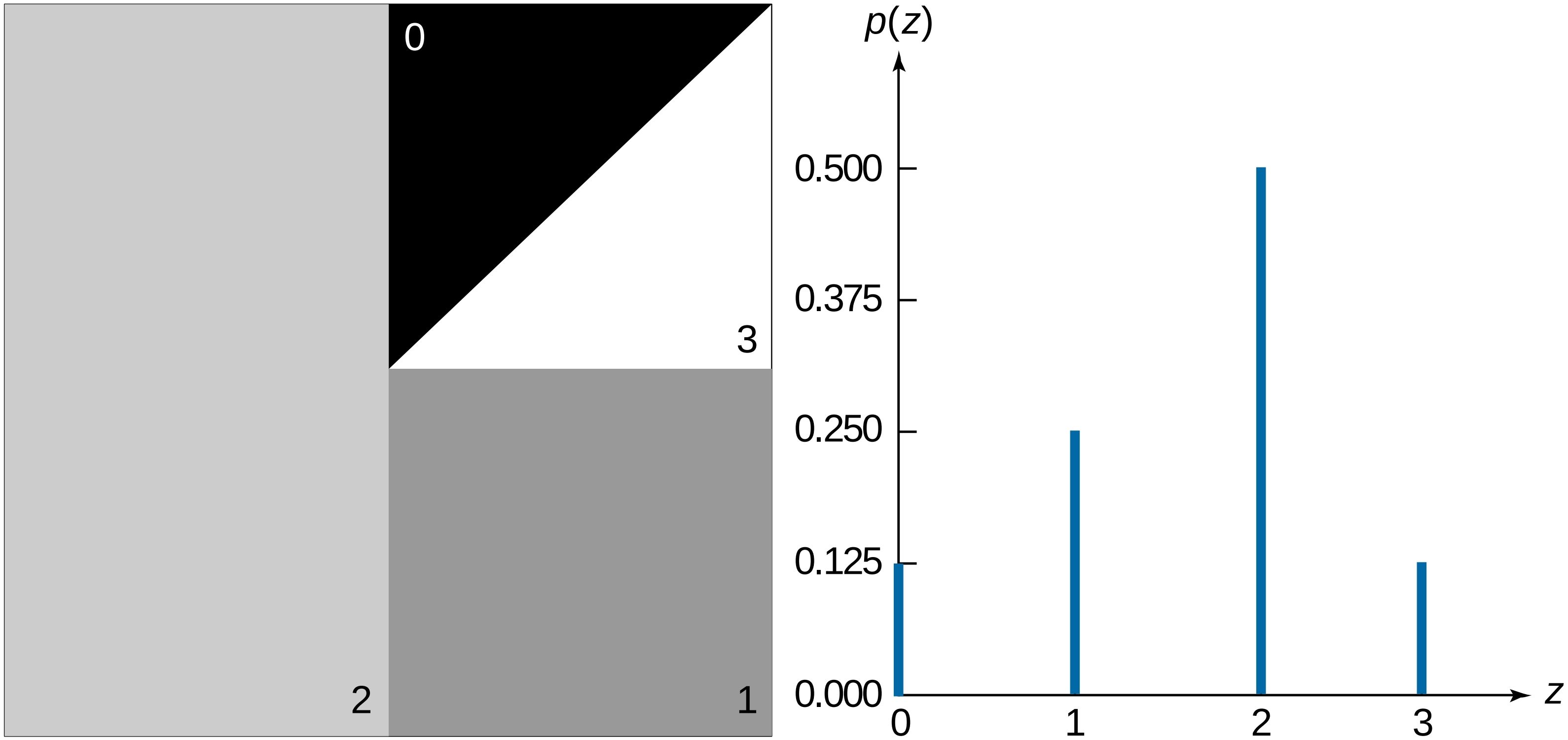
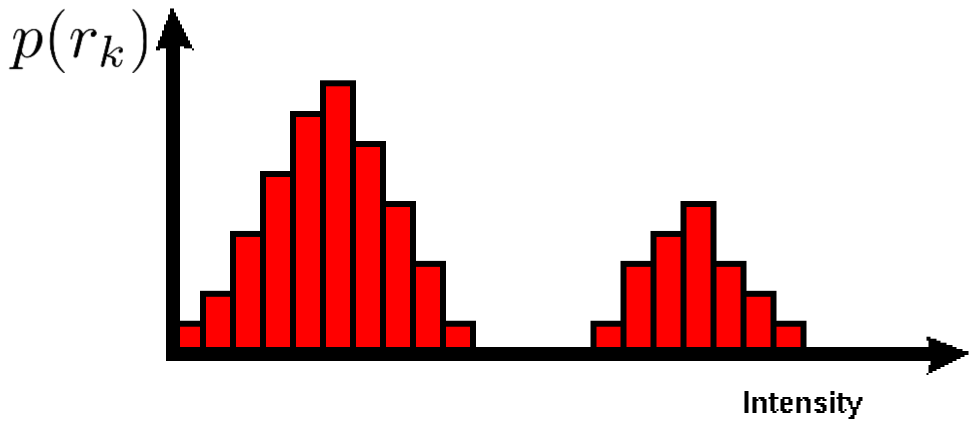



**FIGURE 2.49**
(a) A synthetic image, and (b) its histogram. (c) A natural image, and (d) its histogram.

[Gonzales/Woods Fig. 2.49]

# Histogram

- We work mostly with normalized histograms: if nothing is mentioned explicitely, the histogram is normalized
- $p(r_k)$ is an estimate of the probability distribution of the intensity levels in image $f$ (it tells us how intensity values are distributed in the image).



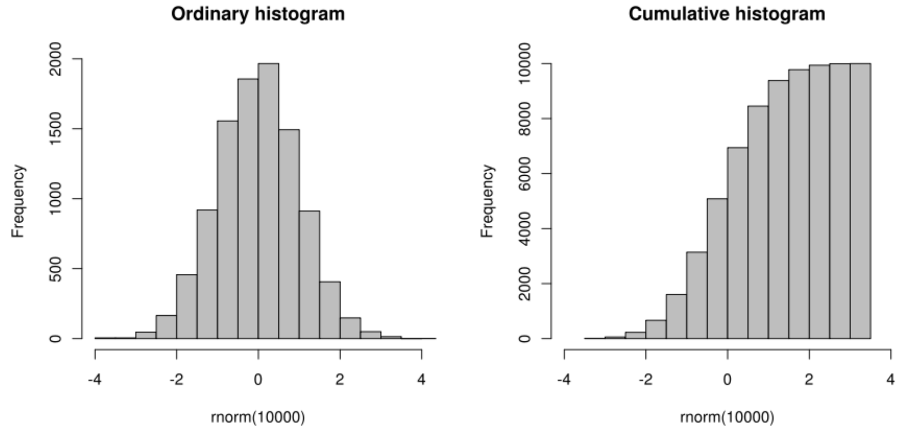

- What is the sum of all values in a normalized histogram?
- Answer: 1

http://homepages.inf.ed.ac.uk/rbf/HIPR2

# Histogram

**Definition**: a *cumulative histogram* $c$ for histogram $h$ is a histogram that counts in each bin $c(i)$ the number of occurrences in all bins $j \leq i$ :

$$c(i) = \sum_{j=1}^{i} h(i)$$




[Image: by Kierano, Wikimedia Commons]

# Purpose of histograms

Histograms are very useful representations of images:

1. They can be manipulated to modify (usually to improve) the image appearance (e.g. histogram equalization)
2. They can be compared to determine the similarity of image regions (e.g. for object recognition etc.)

Here we will mainly cover the 2nd aspect.
For the 1st aspect, see Gonzales/Woods, chapter 3.

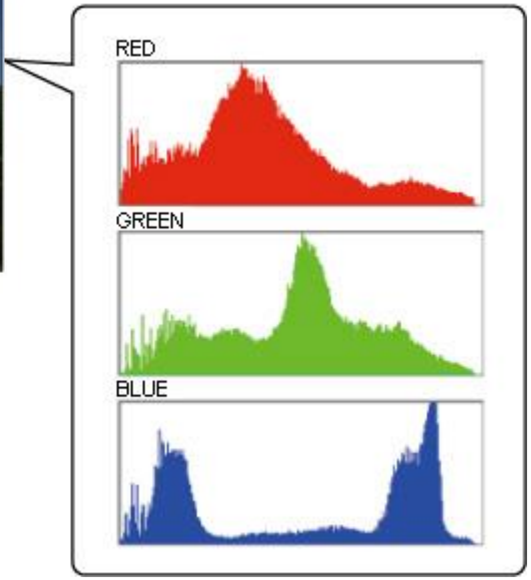
# Color Histograms

How could we deal with color images?

# Color Histograms

Color histograms, simple solution:

- Create a histogram for each channel of the color image
- An example for the RGB space:




- The same is possible for arbitrary color spaces

[Image: Canon, Canada]

# Color Histograms

In HSV space, we need to model only one channel: the Hue channel


(but this gives us only the pure color value, for complete description of colors also saturation and intensity are important)

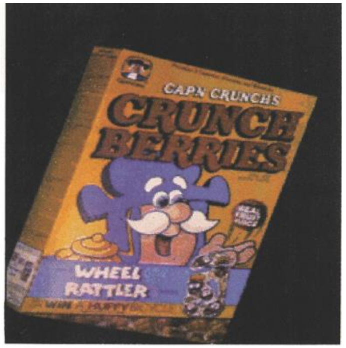
[Image: Canon, Canada]

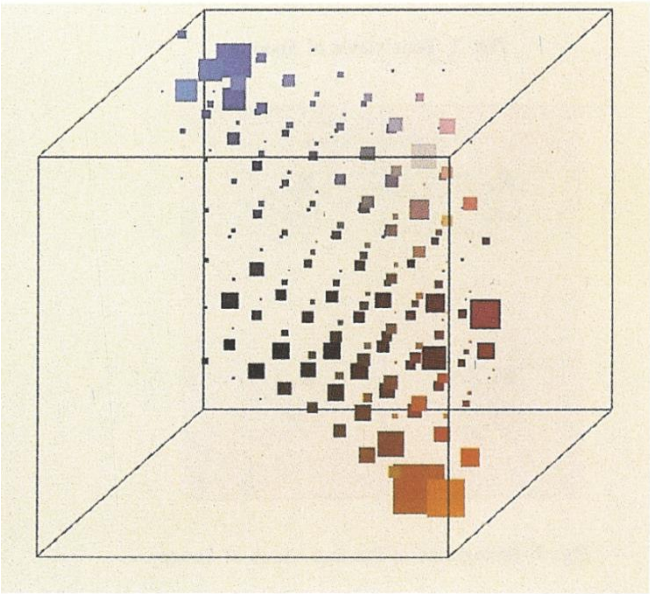
# Color Histograms

More sophisticated color representation:

Color histograms in a joint 3D histogram

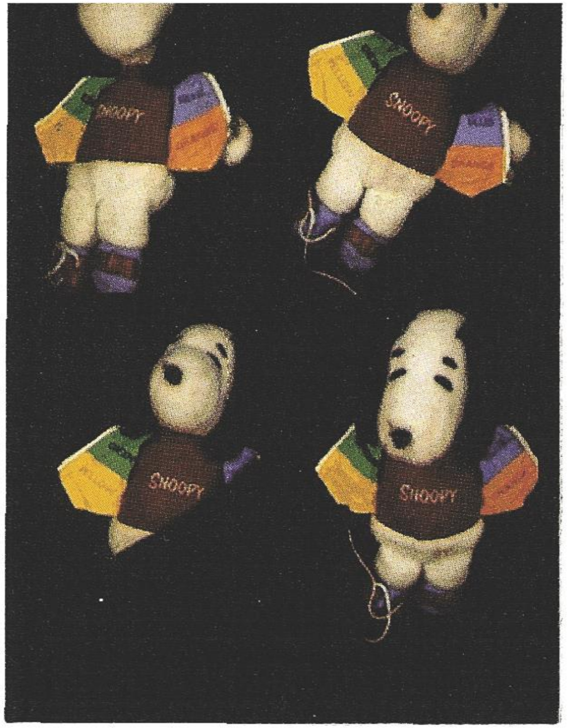
- Here: RGB as an example
- Compute 3D histogram
  $H(R, G, B)$= #(pixels with color $(R, G, B)$)
- Gain robustness by normalizing each channel by intensity:
  divide each channel by $I = R + G + B$

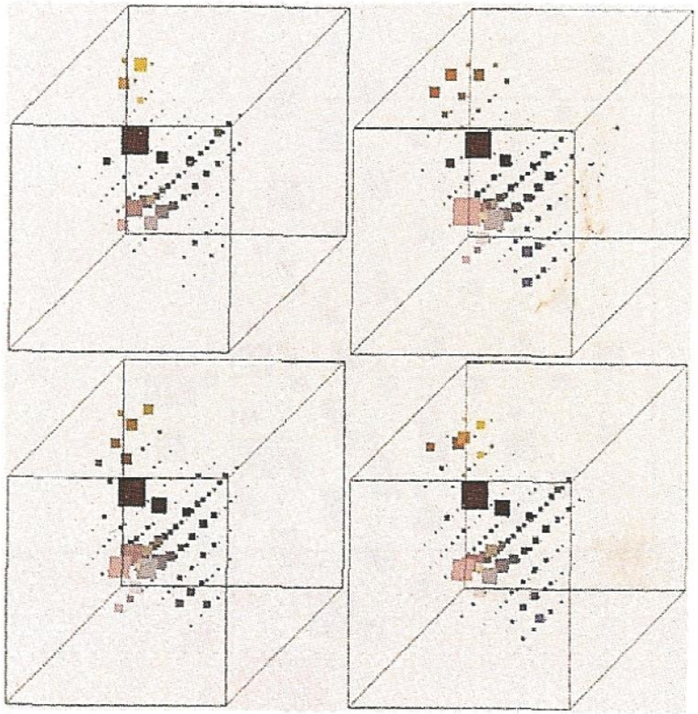


[Swain & Ballard 1991]

# Color Histograms





Although the images differ, the (normalized) color histograms are very similar (robust representation)

[Swain & Ballard 1991]

# *Histogram Comparison*

How similar are these histograms?

# Histogram Comparison

There are plenty of methods to compare histograms:

- $L_x$ Distance ($L_0$: Hellinger, $L_1$: Manhattan, $L_2$: Euclidean)
- Histogram Intersection
- Earth Movers Distance
- Kullback-Leibler Divergence
- Jeffreys Divergence
- Mahalanobis Distance
- Chi-Square distance ($\chi^2$)
- Bhattacharyya Distance
- and many more…

# Histogram Comparison

There are plenty of methods to compare histograms:

- $L_x$ Distance ($L_0$ : Hellinger, $L_1$ : Manhattan, $L_2$ : Euclidean)
- Histogram Intersection
- Earth Movers Distance
- Kullback-Leibler Divergence
- Jeffreys Divergence
- Mahalanobis Distance
- Chi-Square distance ($\chi^2$)
- Bhattacharyya Distance
- and many more…

covered in this lecture

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

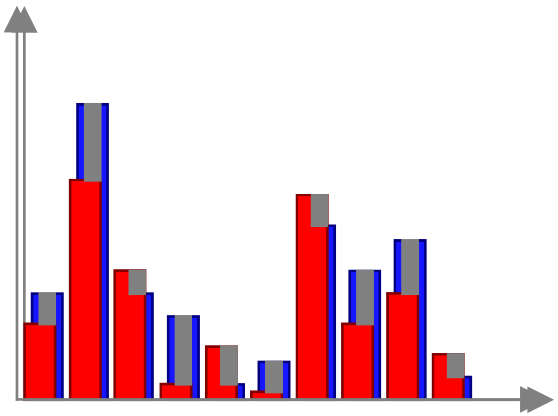
# $L_x$ Distance

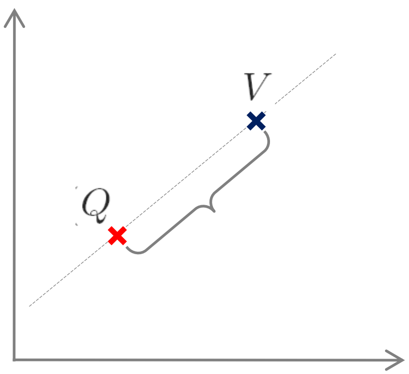
- The $L_x$ Distance focuses on the differences between the histograms $h_1$ and $h_2$:
  - Interpretation: distance in feature space
  - All cells are weighted equally.
  - Not very robust to outliers!
- Popular versions:
  - $L_1$: Manhattan distance

$$D_{L_1} = \sum_i |h_1(i) - h_2(i)|$$

  - $L_2$: Euclidean distance

$$D_{L_2} = \sqrt{\sum_i (h_1(i) - h_2(i))^2}$$



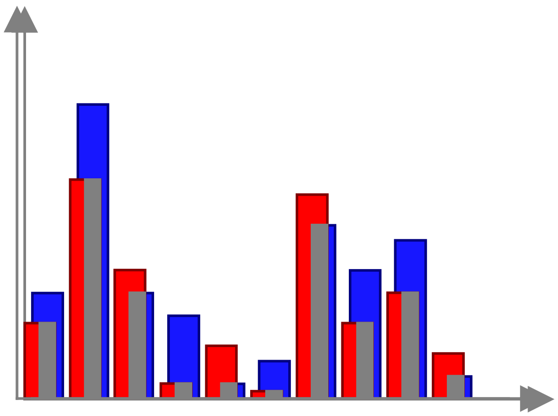



[Images: Bastian Leibe]

# Histogram Intersection

- The histogram intersection of (normalized) histograms $h_1$ and $h_2$ is defined as:

$$D_\cap = \sum_i min(h_1, h_2)$$



- Motivation
  - Measures the common part of both histograms
  - Range: [0,1]
  - For unnormalized histograms, use the following formula

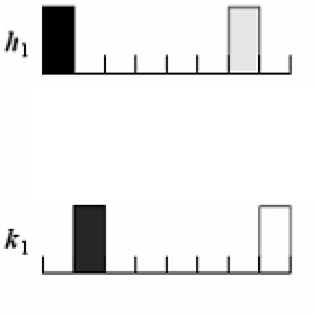
$$D_\cap = \frac{1}{2}\left(\frac{\sum_i min(h_1,h_2)}{\sum_i h_1} + \frac{\sum_i min(h_1,h_2)}{\sum_i h_2}\right)$$

[Image: Bastian Leibe]

# Histogram Comparison

- What could be a problem with the above distance measures?

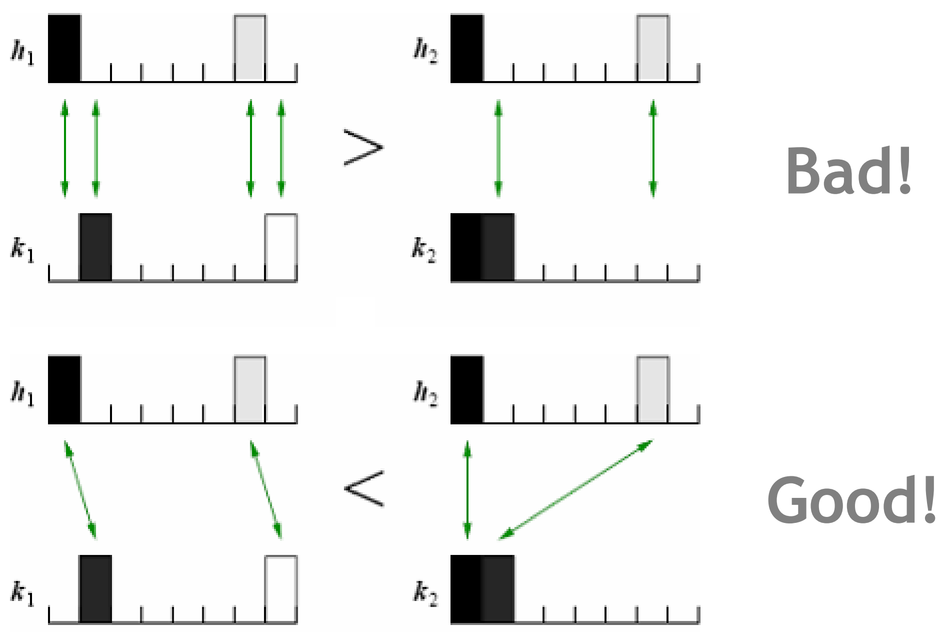
Similar?




- Not according to $L_x$ Distance or Histogram Intersection!
- What could cause such kind of a change?
- Illumination changes

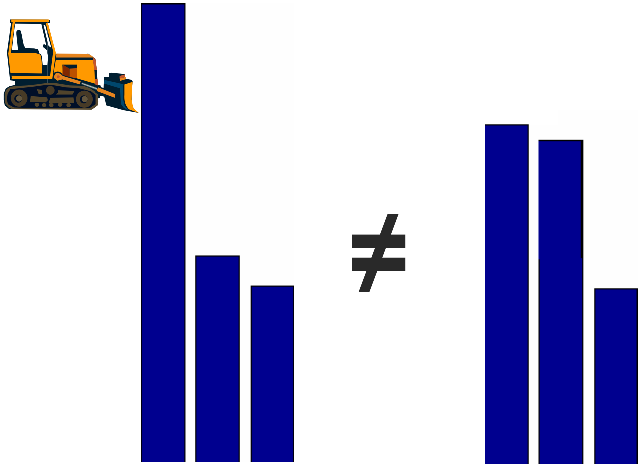
# Histogram Comparison

- What could be a problem with the above distance measures?



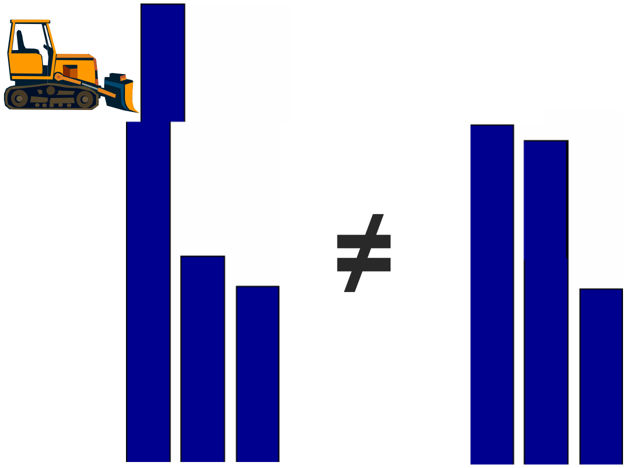
# Earth Movers Distance

- Motivation: Moving Earth



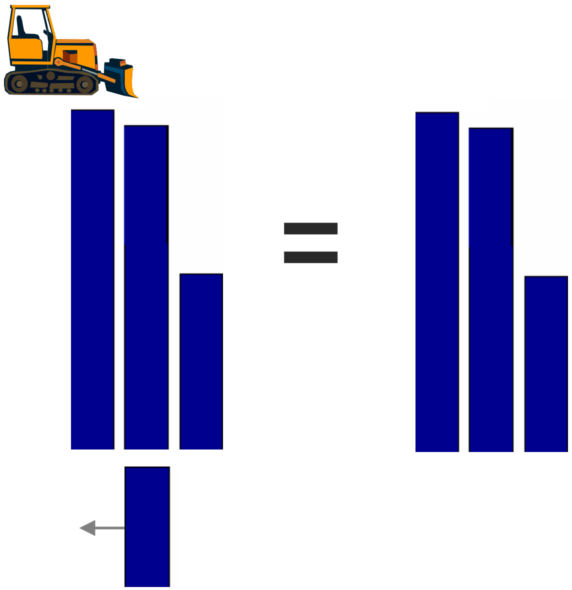
# Earth Movers Distance

- Motivation: Moving Earth



# Earth Movers Distance

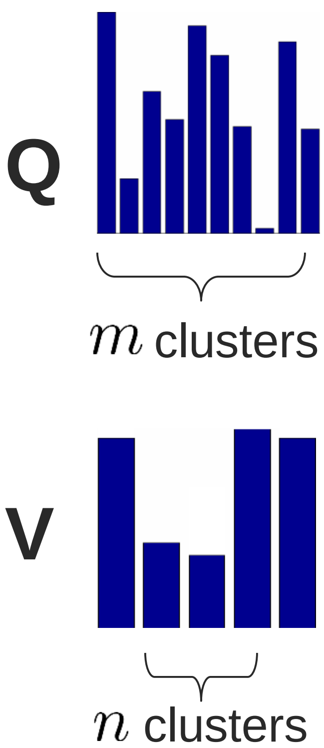
- Motivation: Moving Earth

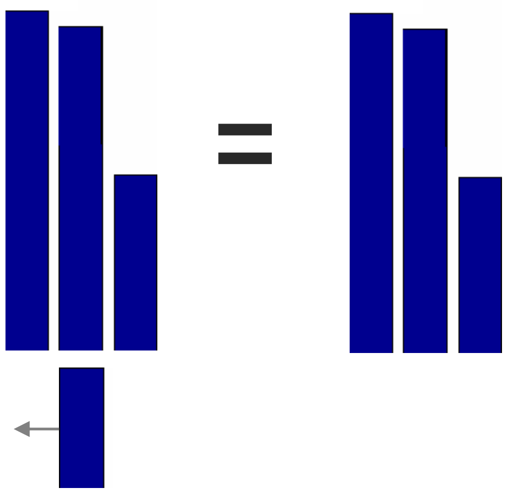


(distance moved) * (amount moved)

# Earth Movers Distance

- Motivation: Moving Earth



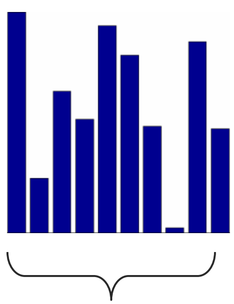



$$\sum_{\text{All movements}} \text{(distance moved)} * \text{(amount moved)}$$

# Earth Movers Distance

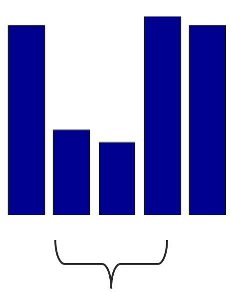
- Motivation: Moving Earth

**Q**

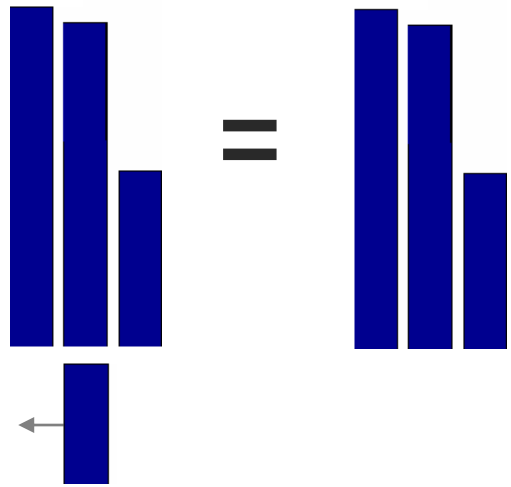


$m$ clusters

**V**



$n$ clusters



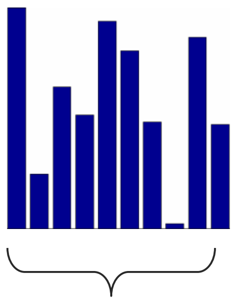

$$\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} * (amount\ moved)$$

All movements

# Earth Movers Distance

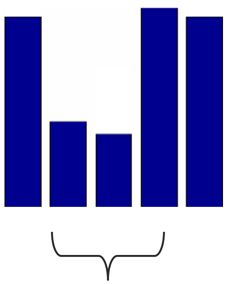
- Motivation: Moving Earth

Q

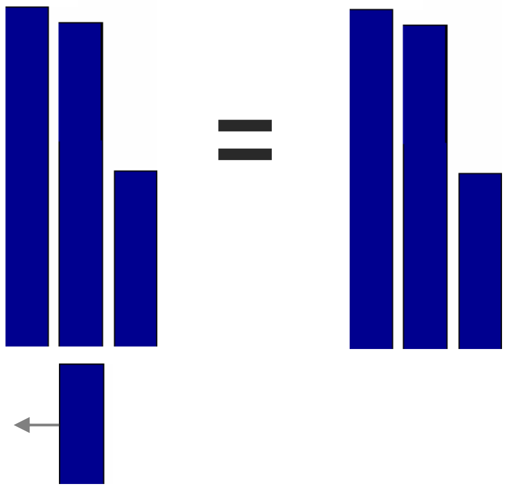


$m$ clusters

V



$n$ clusters




$$\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij} = WORK$$

All movements

# Histogram Comparison

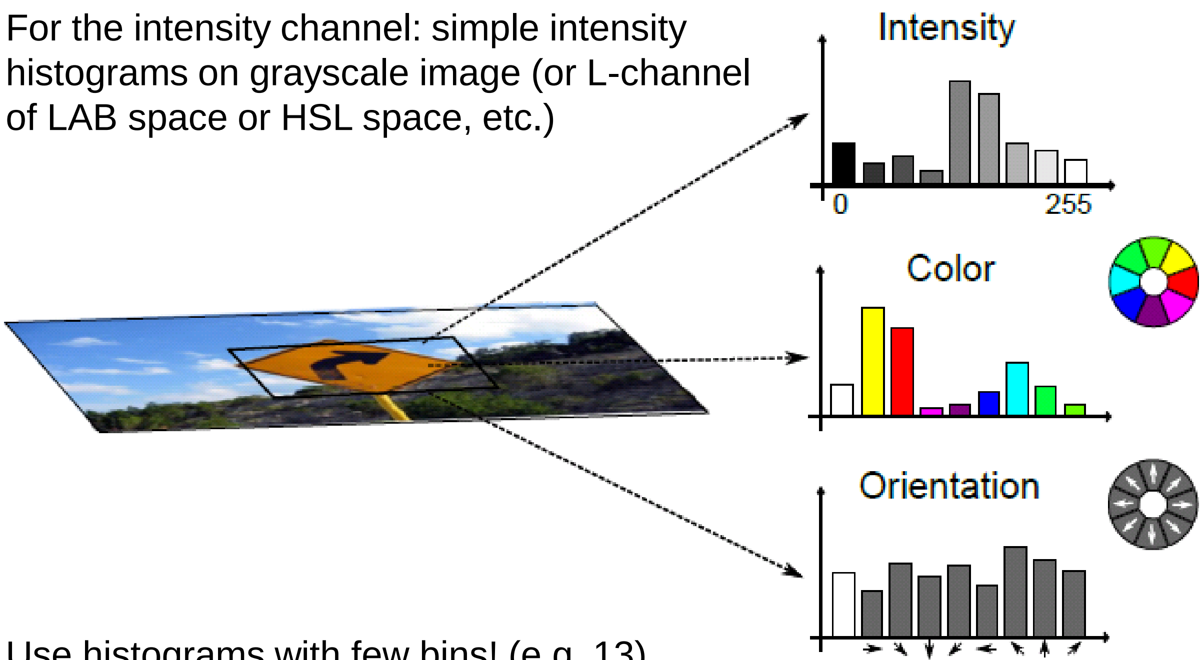
- Which measure is best?
  - Depends on the application…
  - Euclidean distance is often not robust enough.
  - Intersection usually gives good performance for histograms.
  - KL/Jeffrey works sometimes very well, but is “expensive” (slow)
  - EMD is most powerful, but also quite expensive

If you use histogram comparisons, it is often useful to compare the performance for several different comparison measures

# Histograms: Application

In the BITS saliency system, different histogram types (intensity, color, edge orientation) are used to represent and compare image patches (compare with KLD)

For the intensity channel: simple intensity histograms on grayscale image (or L-channel of LAB space or HSL space, etc.)




Use histograms with few bins! (e.g. 13)

More in “Computer Vision 2” lecture

[Klein/Frintrop 2011]

# Mean

We can compute *image statistics* (e.g. mean and variance) of an image region based on its histogram:

**Definition (mean):**

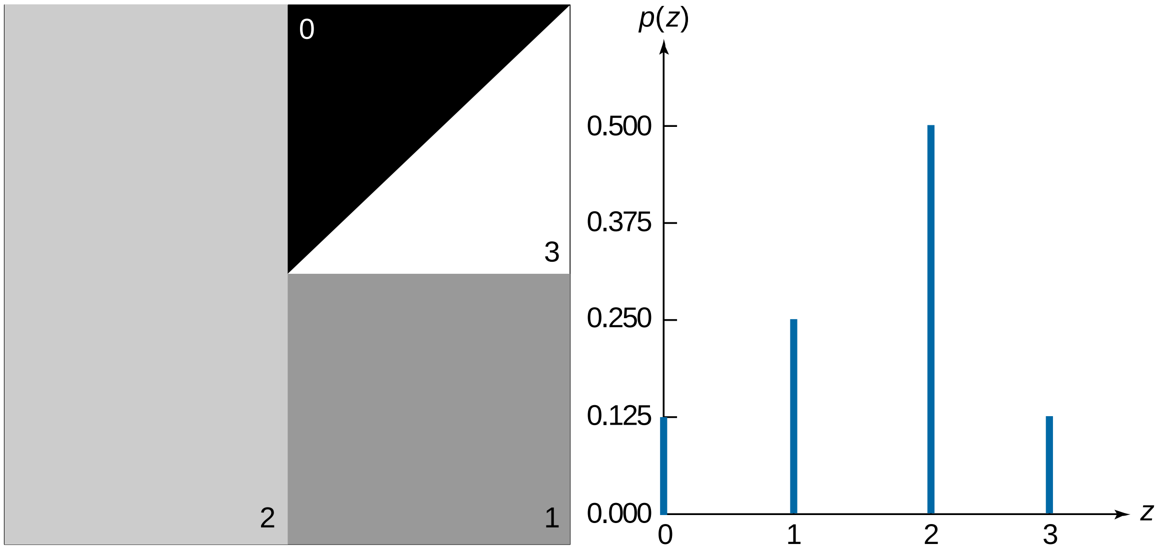
The mean $m$ (average intensity value) of a digital image $f$ with intensity levels $r_k$ and the normalized histogram $p(r_k)$ is defined as:

$$m = \sum_{i=0}^{L-1} r_i p(r_i)$$

# Mean

a b
c d

**FIGURE 2.49** (a) A synthetic image, and (b) its histogram. (c) A natural image, and (d) its histogram.




$$\bar{z} = E[z] = \sum_{z \in \{0,1,2,3\}} z p(z) = (0)p(0) + (1)p(1) + (2)p(2) + 3p(3)$$

$$= (0)(0.125) + (1)(0.250) + (2)(0.500) + (3)(0.125) = 1.625$$

[Gonzales/Woods]

# Variance

## Definition (variance):

The variance of a digital image $f$ with $L$ intensity levels $r_k$ and the normalized histogram $p(r_k)$ and mean $m$ is defined as

$$\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$$

The variance measures how much pixel intensities vary from the mean. This is a measure of *image contrast*.

# Mean and Variance

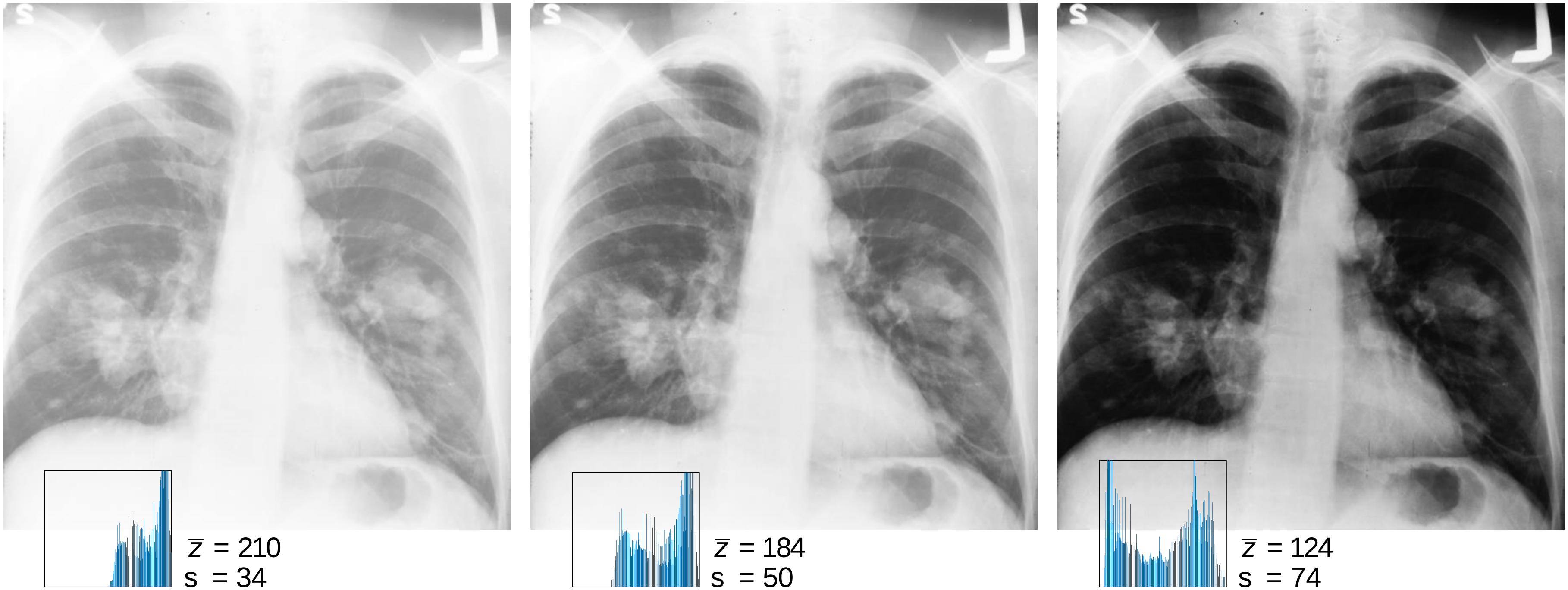
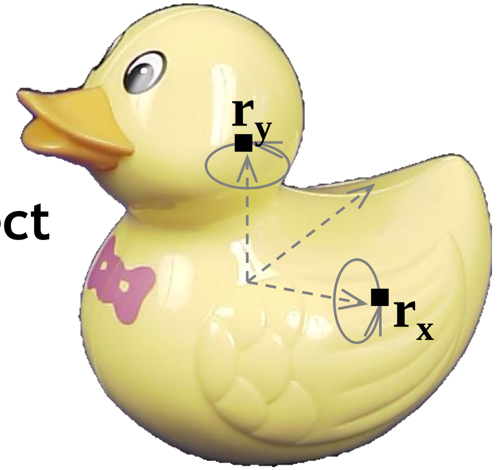



**FIGURE 2.51** Illustration of the mean and standard deviation as functions of image contrast. (a)-(c) Images with low, medium, and high contrast, respectively. (Original image courtesy of the National Cancer Institute.)

# Appearance-Based Recognition

- Our first application: appearance-based object recognition
- Basic assumption
  - Objects can be represented by a set of images (“appearances”).
  - For recognition, it is sufficient to just compare the 2D appearances.
  - No 3D model is needed.

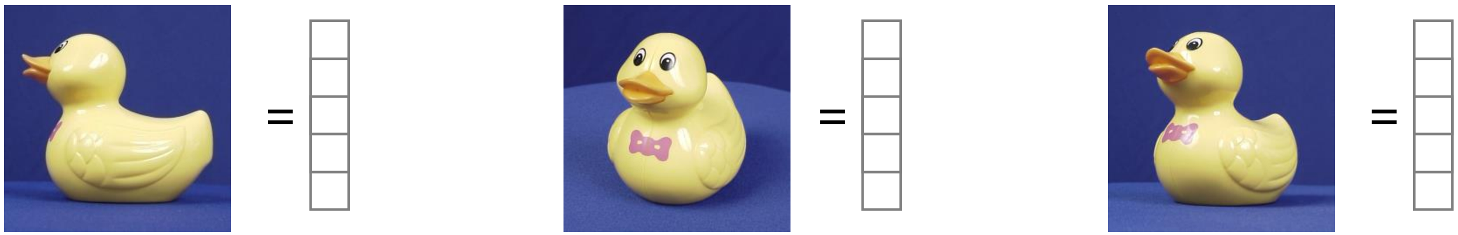
3D object




**⇒ Fundamental paradigm shift in the 90s**

# Global Representation

- Idea
  - Represent each object (view) by a global descriptor.

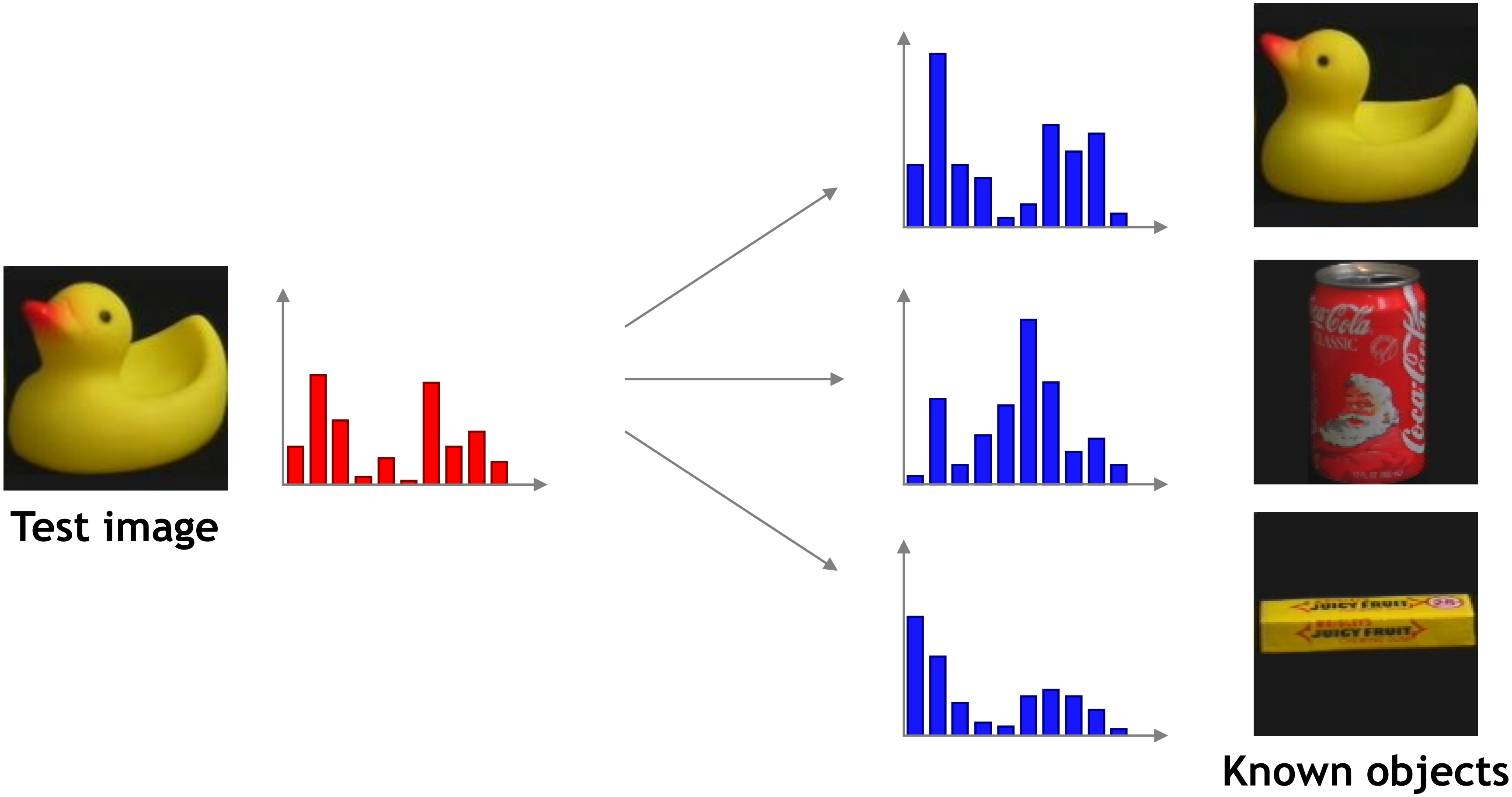


  - For recognizing objects, just match the descriptors.
  - Variations are incorporated in the training data.
    - (partly) invariant to rotations.
    - Other variations:

| | |
|---|---|
| **Viewpoint changes** | **Illumination** |
| – **Translation** | **Noise** |
| – **Scale changes** | **Clutter** |
| – **Out-of-plane rotation** | **Occlusion** |

# Recognition Using Histograms

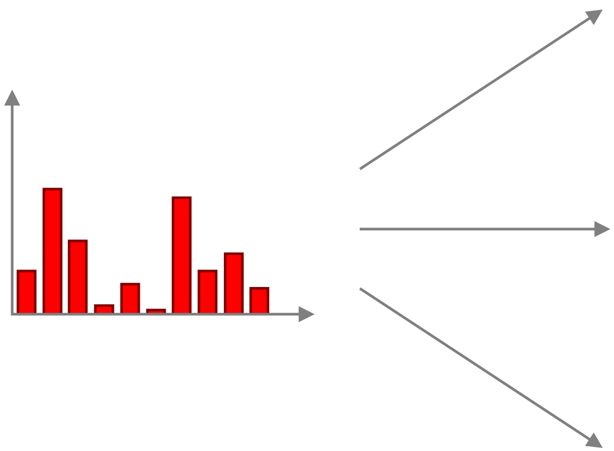
- Histogram comparison

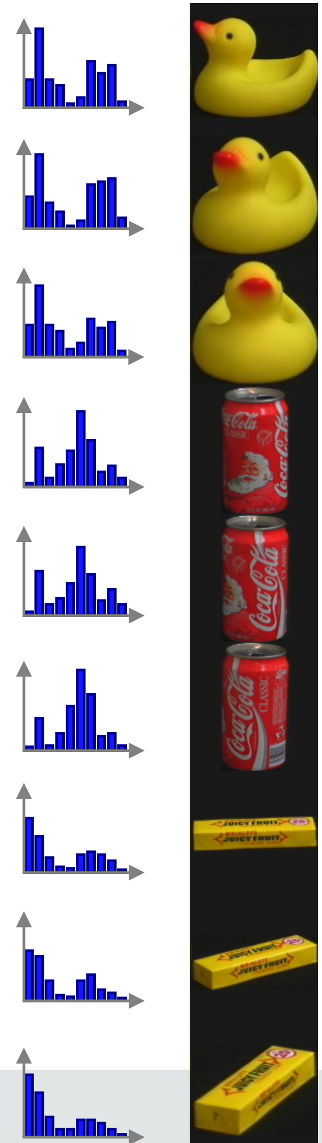


# Recognition Using Histograms

- With multiple training views

Test image





# Recognition Using Histograms

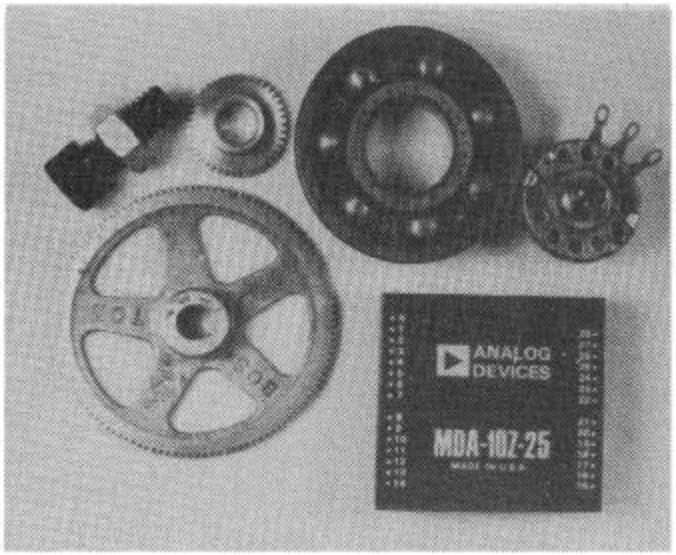
## Simple algorithm

1. Build a set of histograms $H = \{h_i\}$ for each known object
   - More exactly, for each *view* of each object
2. Build a histogram $h_t$ for the test image.
3. Compare $h_t$ to each $h_i \in H$
   - Using a suitable comparison measure
4. Select the object with the best matching score
   - Or reject the test image if no object is similar enough.

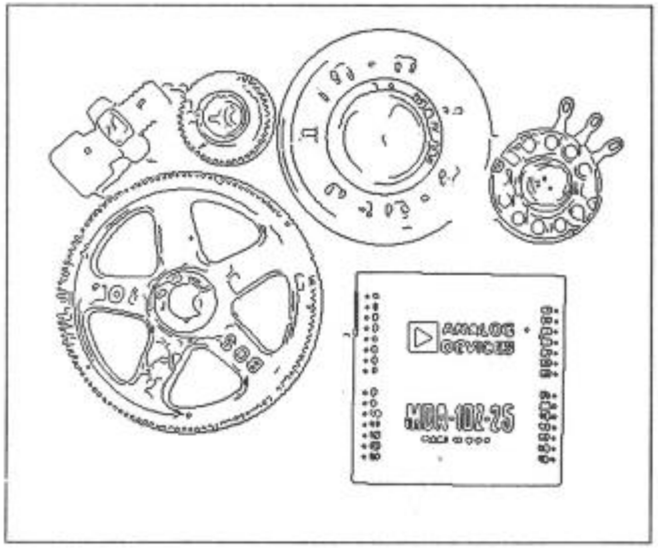
**"Nearest-Neighbor" strategy**

# Evaluations in CV

- In the early days of CV:
- Show on one or a few images that your algorithms works
- E.g.: the Canny edge detector (1986):




(a)




(d)

[Canny 1986]

# Evaluations in CV

- Today: evaluations of algorithms usually by benchmarking:
- Benchmarking: compare performance of algorithm to ground truth from some benchmark dataset
- E.g.: ImageNet (the largest image dataset for computer vision): (http://image-net.org/)

# Evaluations in CV

- Today: evaluations of algorithms usually by benchmarking:
- Benchmarking: compare performance of algorithm to ground truth from some benchmark dataset
- Benchmark datasets:
  - use existing datasets from the web (there are plenty available). Advantage: comparable to other approaches. Can be used by anyone. Other people know how to rate your results. Avoids bad scientific behavior like hand-picking examples that work well. Disadvantages: dataset might not fit your needs perfectly.
  - or create your own one Advantage: you can create a dataset that fits exactly your problem. Disadvantage: opposite of above (not comparable) If ever possible, make your dataset at least online available for others
  - Good solution in case existing datasets do not fit perfectly: pick the best-fitting existing dataset and create additionally an own one

# Evaluations in CV

## Ground truth (gold standard):

- Usually hand-labeled by humans (but can be also the output of another system/sensor/method)

# Evaluations in CV

Ground truth varies depending on the application.

Some types of Ground truth:

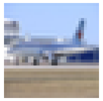
- Word-labels e.g. “cat” (this image contains a cat)
- Sentence descriptions, e.g. “A man with a red cap plays basketball”
- Bounding boxes
- Contours of objects
- Pixel-precise binary maps (object/non-object)
- Pixel-precise labeled maps (one label per class)

# Evaluations in CV

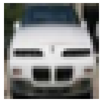
Types of Ground truth:

- Example 1: word-labels in CIFAR 10 dataset (https://www.cs.toronto.edu/~kriz/cifar.html)
  (60000 32x32 colour images in 10 classes, with 6000 images per class)

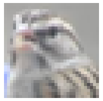
airplane



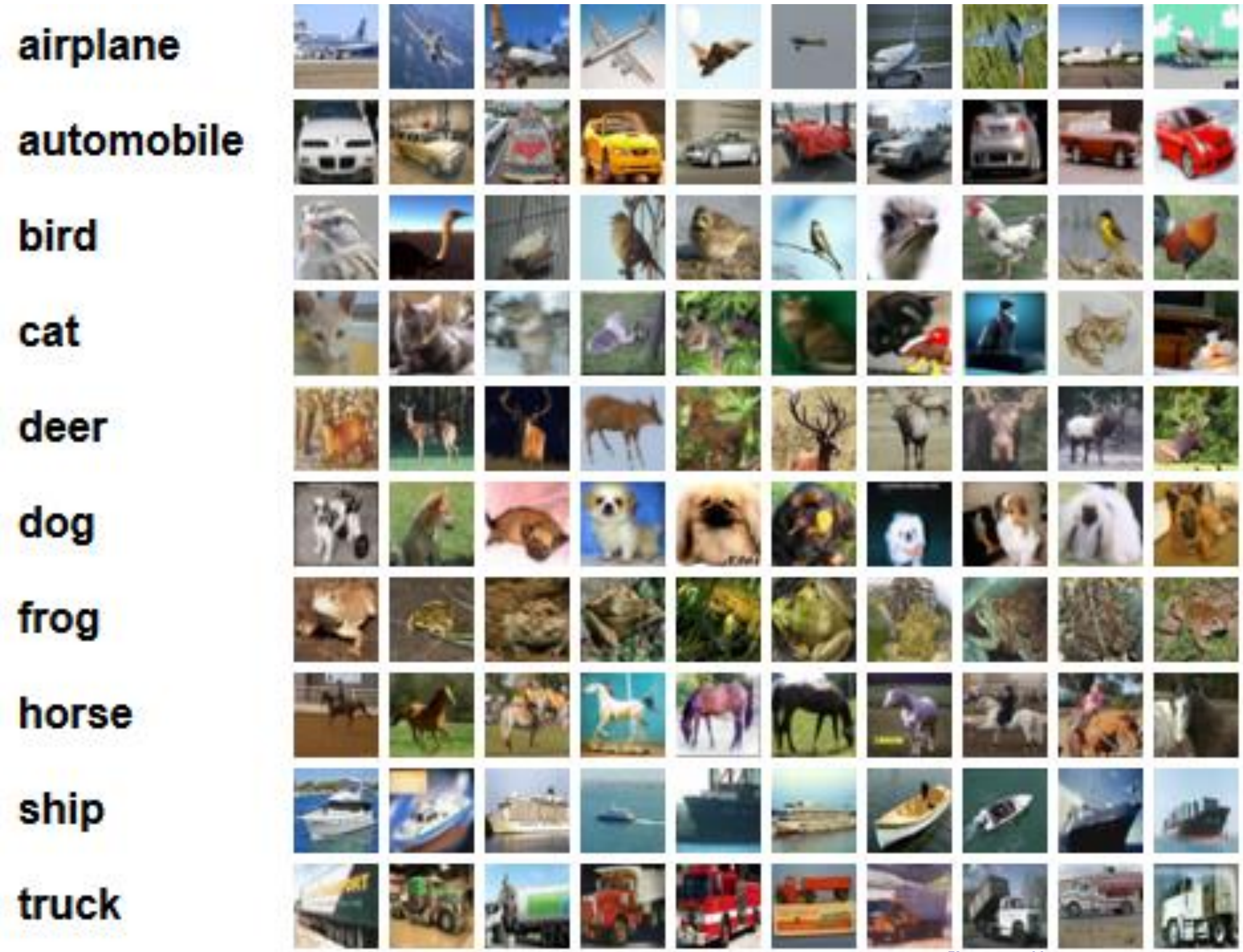
automobile



bird



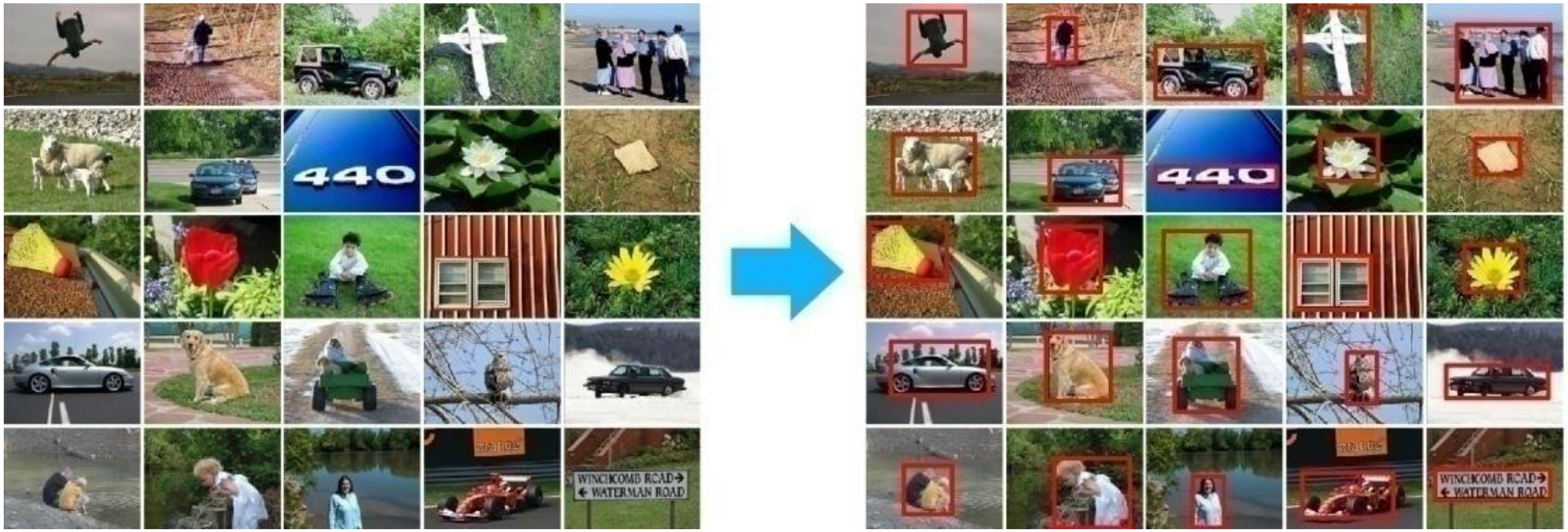
# CIFAR 10




[https://www.cs.toronto.edu/~kriz/cifar.html]

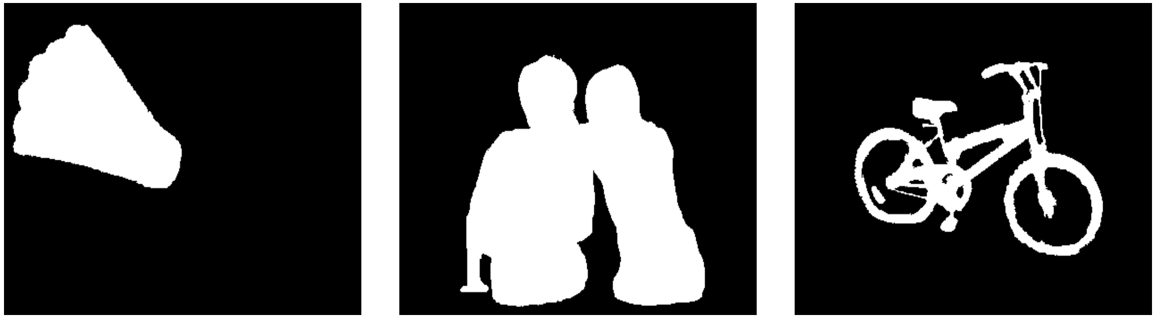
# Evaluations in CV

- Example 2: bounding box ground truth in MSRA Salient object dataset:



# Evaluations in CV

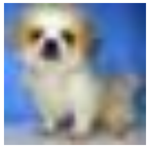
- Example 3: pixel-precise ground truth:
  (from Achanta’s subset of MSRA dataset)



# Evaluations in CV

- Compare output of system with ground truth
- Dimensions to compare: quality (e.g. detection rate, precision-recall, F-measure, etc.), time, …

"Plane" => correct



"Cat" => wrong

- For now only the most simple measure:
- Detection rate: $\frac{number\ or\ correct\ samples}{number\ of\ all\ samples}$

# Primary Literature

- Rafael C. Gonzalez and Richard E. Woods: Digital Image Processing,
  Addison-Wesley Publishing Company, 4th edition: 2017.
  (parts from chapters 1 and 2)
- R. Klette: Concise Computer Vision: An Introduction into Theory and Algorithms, Springer 2014
  (parts form chapters 1 and 2)

# Secondary Literature


- Canny, John. "A computational approach to edge detection." *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986): 679-698.
- D.A. Forsyth, J. Ponce: Computer Vision, A Modern Approach (2nd edition), Prentice-Hall 2012
- Klein, Dominik and Frintrop, Simone: Center-surround Divergence of Feature Statistics for Salient Object Detection, *Proc. of the International Conference on Computer Vision (ICCV), Barcelona, Spain, Nov. 2011*
- Swain, Michael J., and Dana H. Ballard. "Color indexing." *International journal of computer vision* 7.1 (1991): 11-32.
- Viola, Paul, and Michael J. Jones. "Robust real-time face detection. "*International journal of computer vision* 57.2 (2004): 137-154.