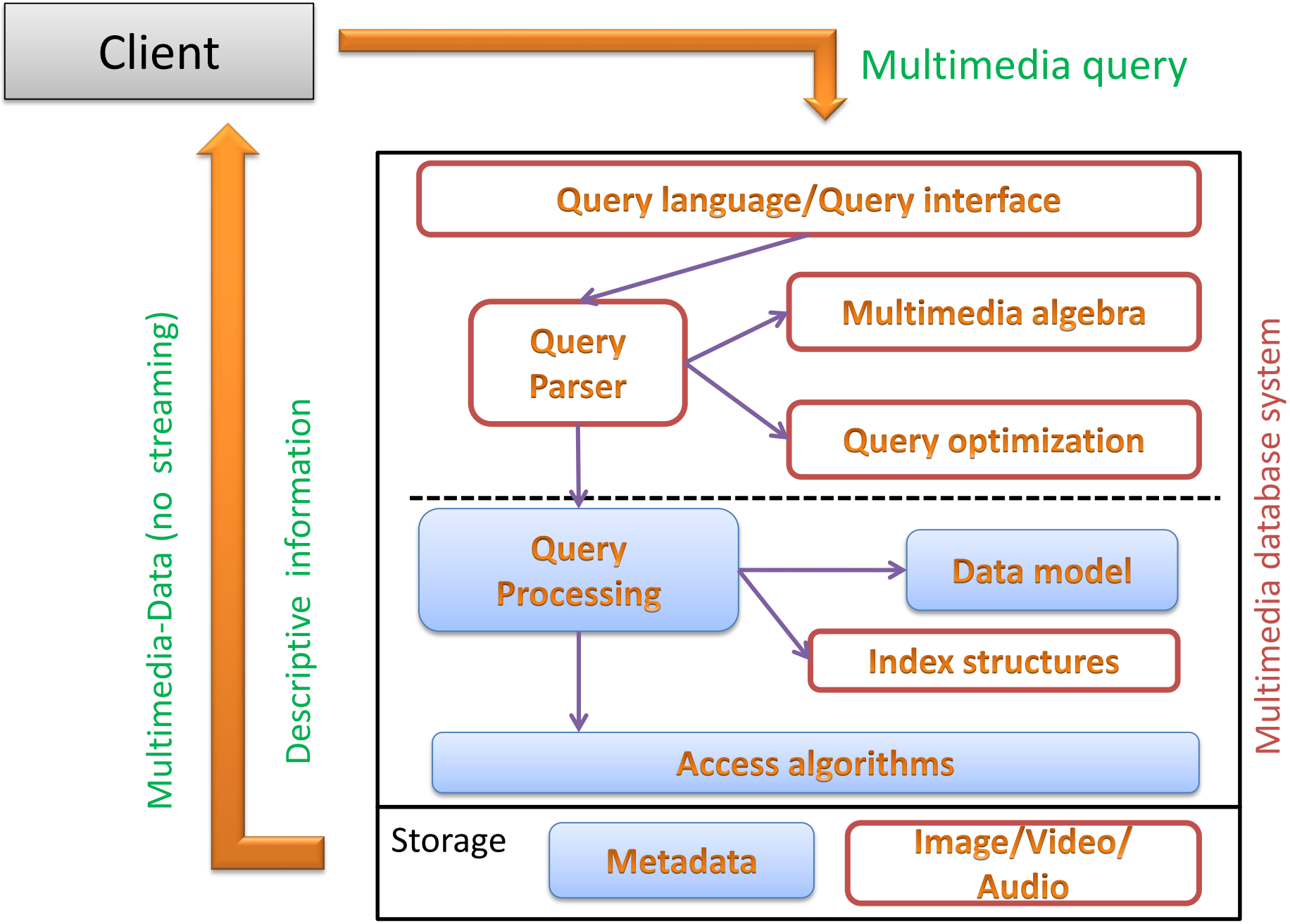
Multimedia Databases
Content Based (Image) Retrieval

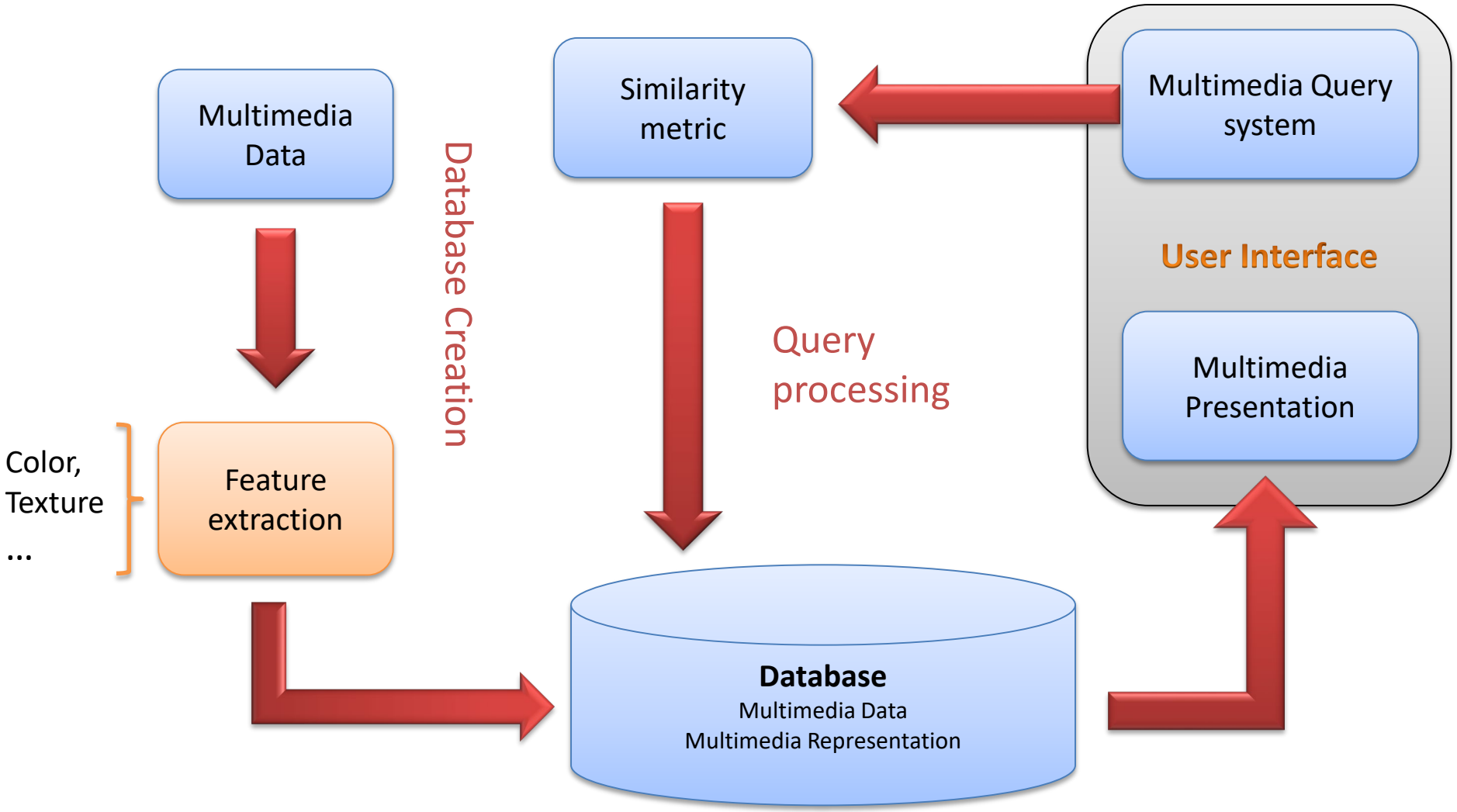
Prof. (FH) PD Dr. Mario Döller

Table of Contents

Content Based (Image) Retrieval


Client
Multimedia query
Multimedia-Data (no streaming)
Descriptive information
Query language/Query interface
Query Parser
Multimedia algebra
Query optimization
Query Processing
Data model
Index structures
Access algorithms
Storage
Metadata
Image/Video/ Audio
Multimedia database system

Content-Based (Image) Retrieval (CBIR) Architecture



CBR and CBIR

- Content-Based Image Retrieval (CBIR) as an example of Content-Based Retrieval (CBR)
- Focus on "low-level„ features (≠ Annotation Based (Image) Retrieval)
- If the features are represented using MPEG-7, XML query languages can be used, however:
  - Specific processes are required to index and compare features.
- Main idea of CBIR:
  - Representation of an image as a set of descriptors.
  - Definition of a similarity measure for the descriptors.
  - When a user submits a query, the system returns images ranked by their similarity to the query.

Motivation: CBIR System for Butterflies

- Should enable non specialists to identify an observed butterfly based on its appearance
- Characterization of appearance:
  - Color, texture, shape

Assumption: users do not (necessarily) have pictures of the observed butterflies

Challenges

- Different perception of different users.
- Users may not exactly remember the appearance of the butterflies.
- Common users lack the expert knowledge to describe butterflies.
- Users do not have the patience to search through a lot of search results.

Solutions

- A user-controlled, interactive query process: QBF/QBE query process
  - "Query By Features" and "Query By Example“
  - Prompt Engineering

- Descriptions of properties for every butterfly are "fuzzy“

- A “What You See Is What You Get” query GUI

- An adapted representation of a summary of the results

Table of Contents

Content Based (Image) Retrieval

QBF/QBE Query Process I

- Query By Feature:
  - In a QBF, some features of the searched butterfly are submitted; the system should return all butterflies matching these features.
  - Example of features:
    - Dominant Color, Texture pattern, Form.
- Query By Example:
  - In a QBE, a sample image is submitted; the system should return all butterflies of the DB that are similar to that image.

QBF/QBE Query Process II

- Characteristics of QBF:
  - Tends to be inaccurate
  - Apply to:
    - The initial query and if the user wants to expand the search space.

- Characteristics of QBE:
  - Accurate and detailed search
  - Apply to:
    - The last query and if the user wants to restrict the search space.

QBF/QBE Query Process III

- Result page: each result page is composed of two parts
  - Result images:
    - DB images of butterflies that satisfy the query conditions.
    - Users may compose further QBE using the images.
  - Similar features:
    - Features of the retrieved images.
    - Users may compose further QBF using the features.

Table of Contents

Content Based (Image) Retrieval

Representation of visual characteristics I

- Description of the properties of a butterfly:
  - Like Metadata which describe the appearance of the butterfly.
  - Enables QBFs.
  - Consists of a set of content descriptors.

- Content descriptor:
  - A tuple of (“Property value” , “similarity level”) pairs or
  - A histogram representing the distribution of Color/Texture/ Shape in the image.

Representation of visual characteristics II

Feature Type	Feature Value	Degree of Match
Color	mixed_with_black_and_orange	52/57
	orange_yellow	12/42
	orange_red	3/38
Texture	many_spots	58/62
	fore_half_different_color	27/33
	horizontal_bands	41/60
	edge_with_different_color	10/74
Shape	wave	98/110

Representation of visual characteristics III

- Color

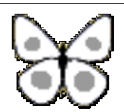
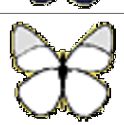
Figure	Feature Value
	black
	brown
	bister
	orange_red
	orange_yellow
	yellow
	green
	blue
	purple

	gray
	white
	mixed_with_black_and_white
	mixed_with_black_and_yellow
	mixed_with_black_and_orange
	mixed_with_black_and_blue
	mixed_with_black_and_red
	mixed_with_wood_and_white
	mixed_with_many_colors

Representation of visual characteristics IV

- Texture

Figure	Feature Value
	vertical_bands
	horizontal_bands
	many_bands
	two_lines
	many_lines
	Obvious_vein
	grids
	eyes

	few_spot
	some_spots
	many_spots
	color_blocks
	grainy
	edge_with_different_color
	starlike
	fore_half_different_color

Representation of visual characteristics V

- Shape

Figure	Feature Value
	swallowlike
	swallowtail
	broken
	wave
	like_leaf
	like_moth
	with_little_tails

Query Processing

- QBF:
  - Query on a single descriptor:
    - Result images: images with a degree of similarity > 0.
    - Sorted according to the degree of similarity.
    - This ordered list is a „feature sequence“.
  - Query on several descriptors:
    - Fusion of the feature sequences.
- QBE:
  - QBF using the feature sequences of the query image

Presentation of the Results

- For a QBF:
  - Characteristic: inaccurate search
  - Presentation: only representative butterflies

- For QBE:
  - Characteristic: detailed search
  - Presentation:
    - for very similar images: display all
    - for less similar images : display only representative images

Feature Indexing

- Goal:
  - Make the retrieval process faster.

- Problems with indexing in CBIR:
  - High dimensionality of the feature space. ("curse of dimensionality")
  - Index structure must support non-Euclidean similarity measures.

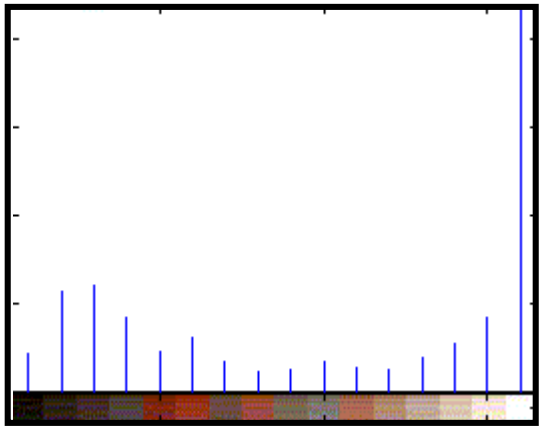
- Solutions:
  - Reduction of dimensionality: KLT, DCT, DWT.
  - Specific indexing structures: R* Tree, SS Tree, SR Tree.

- More on that later…

Color Histogram: Representation

- A list of (Color,Share) pairs:
  - Describes the colors and their proportion in the image.

$$H = \left\{ (I_j, P_j) \mid I_j \in ColorValue, 0 \leq P_j \leq 1, \sum_{1 \leq j \leq N} P_j = 1, and\ 1 \leq j \leq N \right\}$$



General Properties of (Color) Histograms

- A histogram is a discrete representation of the distribution of features

- The distribution depends on pre-defined prototypes: $c_1,\ldots,c_n$
  - Color histogram: n representative colors

- For each element l_j, the best matching prototype is identified = the prototype c_i such that the distance (l_j, c_i) is minimal compared to the distance to all other prototypes.
  - Color histogram: for each pixel, its color is compared to the representative colors and assigned to the closest.

- The set of elements corresponding to a prototype is a “bin” or "container".

Color Quantization I

- Let us consider a JPG with 256 values in each RGB-channel (typical)
- "Color space quantization"
  256 x 256 x 256 = 16.777.216 different colors are distributed into *n* bins, ex. 512 bins

  256x256x256 / 512 =32.768 -> each bin contains 32.768 colors

  For each bin, we compute its bin value, the number of pixels that it contains

  -> ex. 145 (number) or 0.1 (percentage)

The histogram is a vector of bin values.

Color Quantization II

- "Bin quantization" defines the bit-coding of the values in each bin:

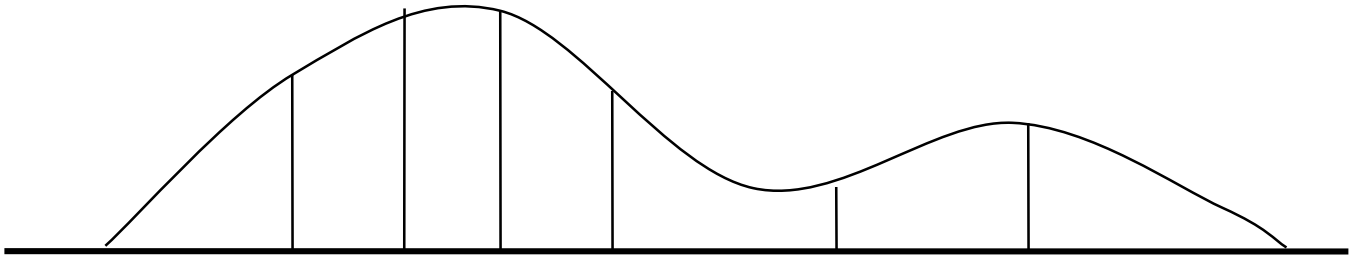
Depending on the desired level of precision, a fixed number of bits is assigned to the representation of the value.

Ex.: Supposing that the image contains 65.536 pixels:

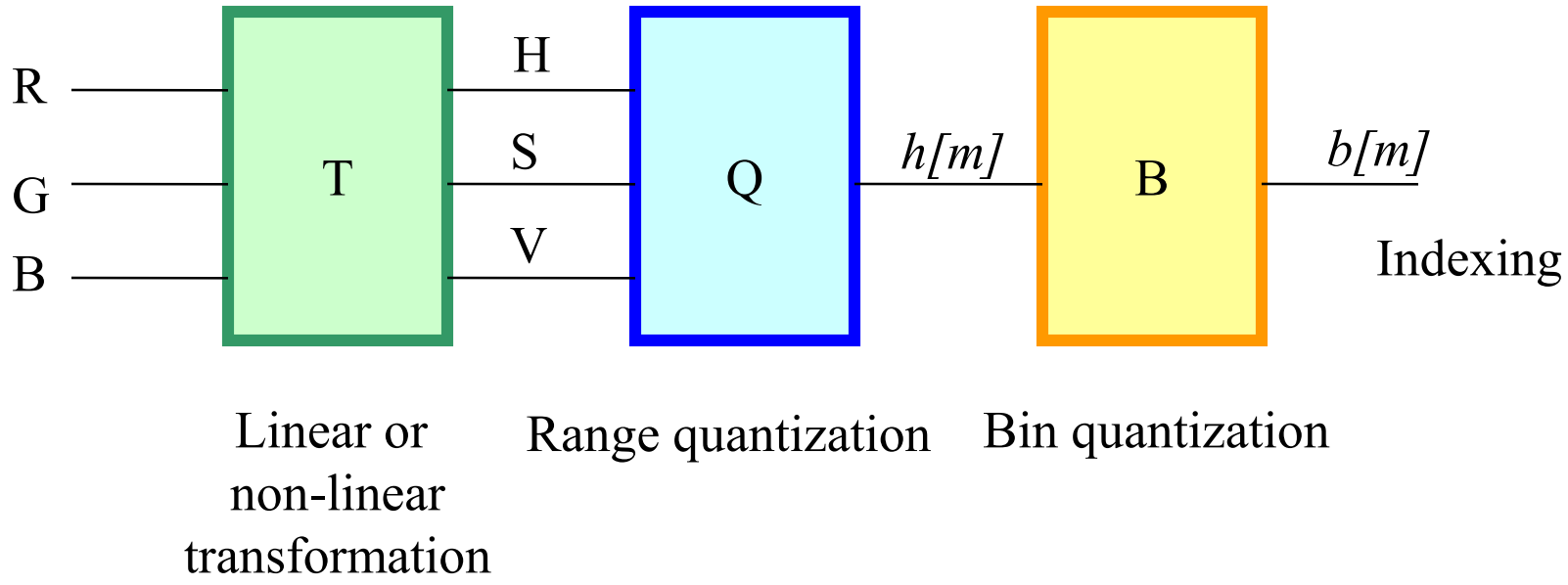
- In theory 16 bits are required to represent the bin values.
- In practice, 15 bits are precise enough -> the bin values are shifted by one bit
- Resulting histogram size: 15 Bits * 512 bins = 7.680 Bits. (50% space gain)

Color Quantization III

- „Adaptive (individual) binning“ (vs. „Regular (uniform) binning”)
  - Adapts the definition of the bins to each image
    - Ex: Several bins with different shades of blue for an image of the sea
  - Used to equalize the distribution of the bin values
  - Tends to provide a more accurate content representation



Color Representation in General



Example I

- Color space selection & Quantization
  - Example with 8x8 24 Bit RGB image, 1 Bit pro channel per bin
    - For each component of each pixel's color, we consider if its value is between 0 and 127 (=0) or above 127 (=1)
    - ➢ 3 Bits defining the bin-number
    - ➢ Ex. RGB color (100, 200, 2) -> (010) -> Bin 2
    - Number of containers = $2^3 = 8$

Example II

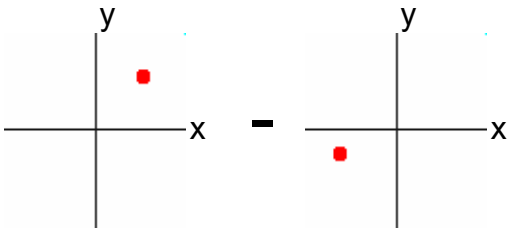
- $H(I)$: Color Histogram for image I
- Three 8x8 pixels, each pixel assigned to one Bin C_1 to C_8
  - image 1 has 8 pixels in each bin
    - $H_1 = (8, 8, 8, 8, 8, 8, 8, 8)$
  - image 2 has 7 pixels in C_1 to C_4, and 9 in C_5 to C_8
    - $H_2 = (7, 7, 7, 7, 9, 9, 9, 9)$
  - image 3 has 2 pixels in C_1 and C_2, and 10 in C_3 to C_8
    - $H_3 = (2, 2, 10, 10, 10, 10, 10, 10)$

- Range quantization to 4 bins ->
    - $H_1 = (16, 16, 16, 16)$
- Bin quantization to 4 Bits ->
    - $H_1 = (8, 8, 8, 8)$

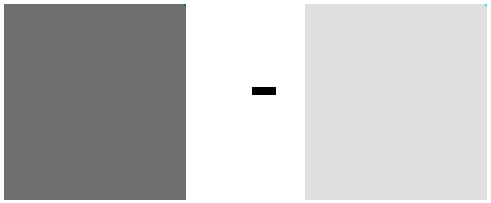
Table of Contents

Content Based (Image) Retrieval

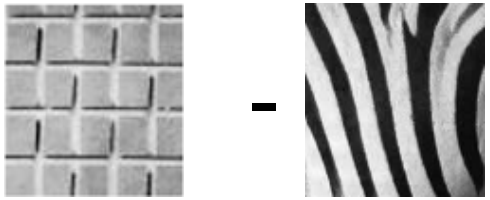
Distance metric



= Euclidian distance of 5



= Grayscale of 50



= ?

Let X be a set. An application $d: X\ x\ X \rightarrow \mathbb{R}$ is a metric or distance function if for all $x, y, z \in X$ the following properties are valid:

(1) $d(x,y) = 0 \Leftrightarrow x = y$ **(self-identity)**
(2) $d(x,y) \geq 0$ **(positivity)**
(3) $d(x,y) = d(y,x)$ **(symmetry)**
(4) $d(x,y) \leq d(x,z) + d(z,y)$ **(triangle inequality)**

A metric space is a pair (X,d), where X is a set and $d: X\ x\ X \rightarrow \mathbb{R}$ is a metric.

Distance metric

Properties/Classification

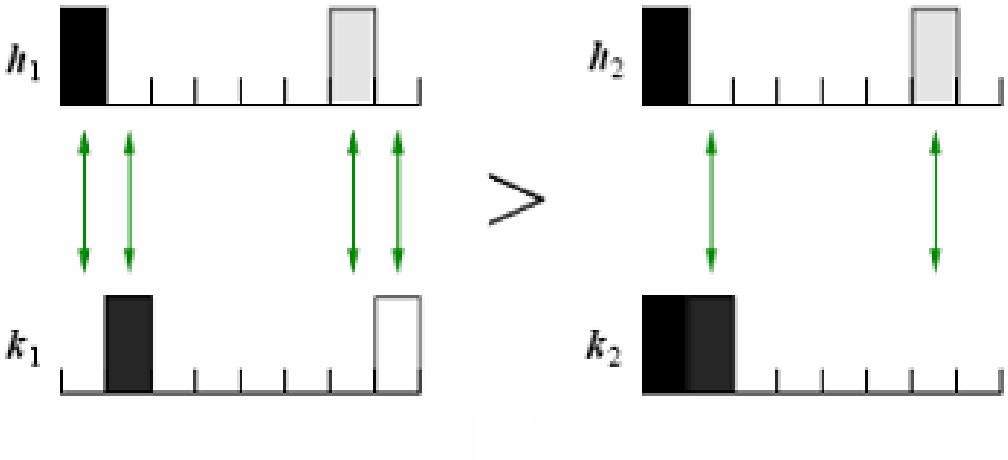
- Invariance to operations
  - Translation
  - Scaling
  - Rotation

- Classification of metrics
  - Pseudo-metric: Distance is not always > 0.
  - Semi-Metric: Does not verify the triangle inequality.
  - Semi-Pseudo-Metric: does neither verify the triangle inequality nor positivity.

Metric	SI	Pos	Sym	Triangle
Metric	YES	YES	YES	YES
Pseudo-Metric	YES	X	YES	YES
Semi-Metric	YES	YES	YES	X
Semi-Pseudo-Metric	YES	X	YES	X

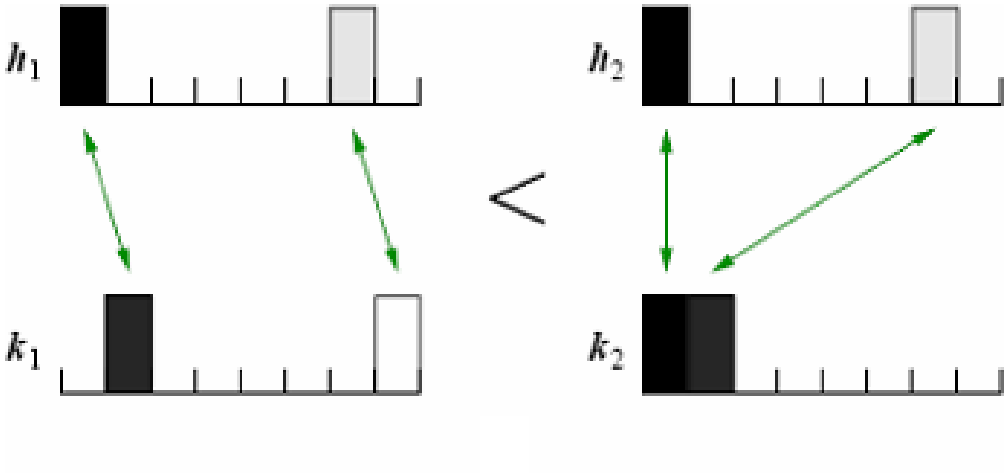
Distance metric

Bin-by-bin Comparison



Bad!

Find the corresponding bins first



Good!

Distance metric

Distances from the literature

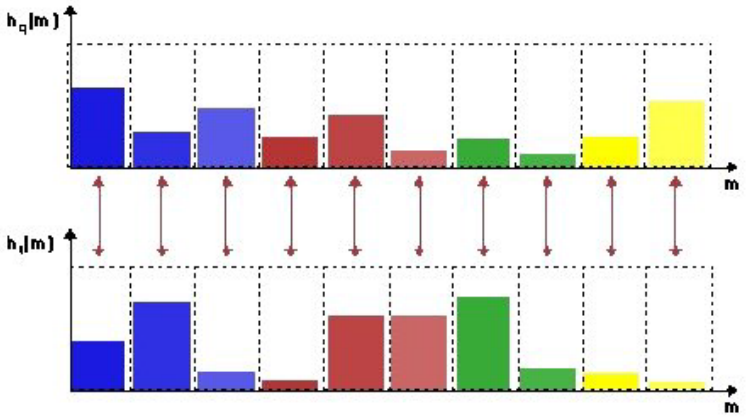
- Heuristics
  - Minkowski distance(s)
  - Weighted-Mean-Variance (WMV)
- Non parametric statistical tests
  - Kolmogorov-Smirnov (KS)
  - Cramer/von Mises (CvM)
  - χ^2 (Chi Square)
- Information Theory distances
  - Kullback-Leibler (KL)
  - Jeffrey-divergence (JD)
- "Ground distance"-measures
  - Histogram intersection
  - Quadratic Form (QF)
  - Earth Movers Distance (EMD)

Distance metric

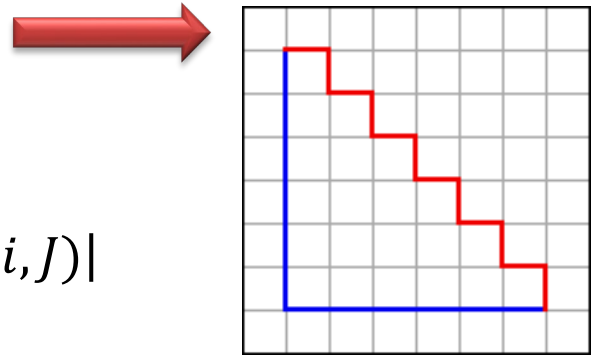
Heuristic Histogram Distance

- Minkowski Distance L_p

$$D(I,J) = \left(\sum_i |f(i,I) - f(i,J)|^p \right)^{1/p}$$



- Special cases:
  - L_1: absolute or Manhattan distance

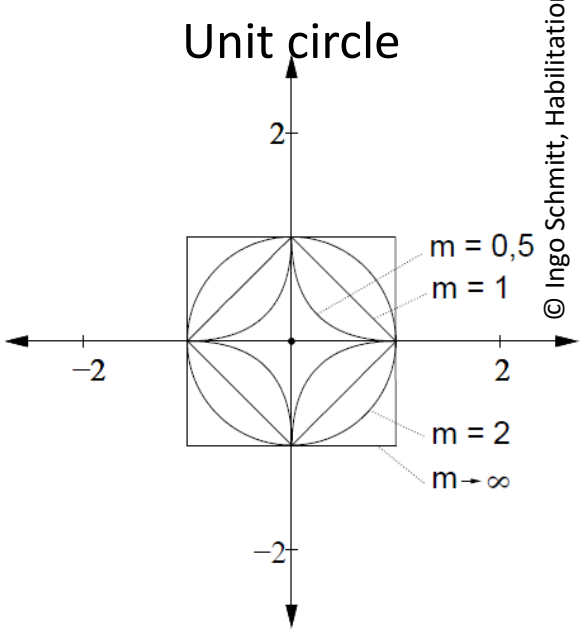


  - L_2: Euclidian distance
  - L_∞: Chebyshev distance $= \max_i |f(i,I) - f(i,J)|$

Distance metric

Limitations of Minkowski

- Ignores similarity of colors
  - Example
    - Two bins
      - Bin-1 Color space: 1 – 10
      - Bin-2 Color space: 11 – 20
    - Three pixels
      - Pixel 1 has Color 10 → Bin-1
      - Pixel 2 has Color 11 → Bin-2
      - Pixel 3 has Color 20 → Bin-2
  - Pixel 2 is actually closer to pixel 1 than to pixel 3 ⇒ not measured by Minkowski!



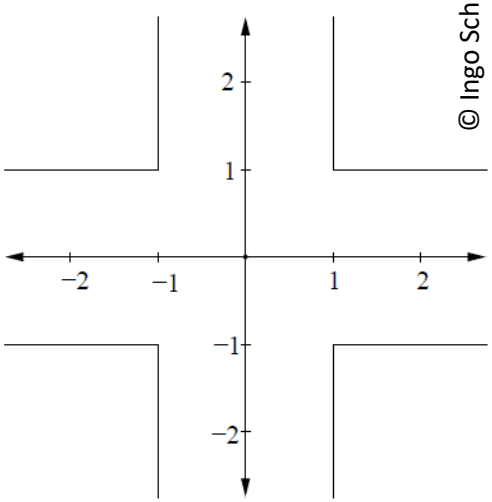
Distance metric

Non-parametric statistical tests

χ^2

- Measures the similarity between two data samples (distributions)
- Considers the pixels as empirical observations
- Bin-by-Bin comparison
- Evaluates how unlikely it is that one distribution of pixels resulted from the other

Unit circle



No positivity

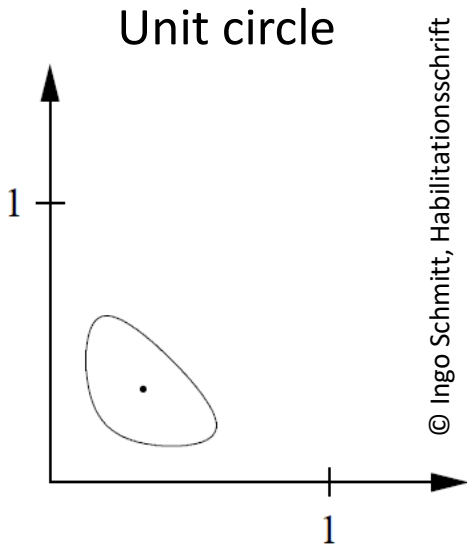
No triangle ineq.

$$D(I,J) = \sum_i \frac{(f(i;I) - f'(i))^2}{f'(i)}, \qquad f' = \frac{[f(i;I) + f(i;J)]}{2}$$

Distance metric

Information theory

- Kullback-Leibler
  - Evaluates the cost of encoding one distribution of pixels (image) from the other
  - Bin-by-Bin distance
  - Measures how inefficient it is on average to generate one histogram from the other.
  - Non-symmetric and sensible to histogram binning
  - Further dev.: Jeffrey-divergence (Symmetric and robust against the number of bins)



No symmetry

No triangle ineq.

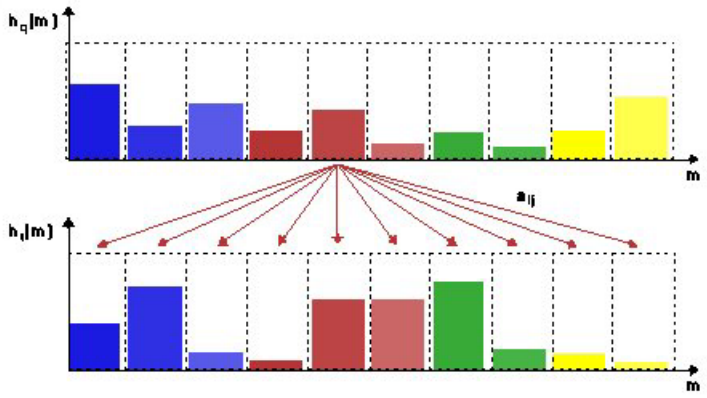
$$D(X, Y) = \sum_i f(i; X) \log \frac{f(i; X)}{f(i; Y)},$$

Distance metric

Ground Distance

- Histogram intersection
  - Ideal for partial matching
  - Bin-by-Bin distance
  - enables the calculation of partial matches when the ranges of two histograms are different

$$d_{\cap}(H,K) = 1 - \frac{\sum_{\mathbf{i}} \min(h_{\mathbf{i}}, k_{\mathbf{i}})}{\sum_{\mathbf{i}} k_{\mathbf{i}}}$$



Distance metric

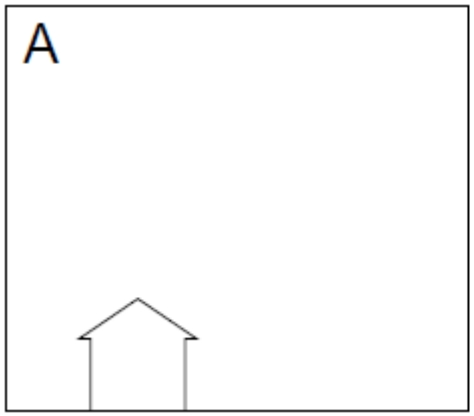
Summary

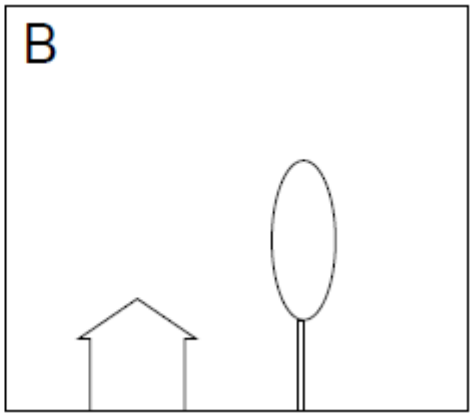
	L^p	*WMV*	*KS/CvM*	χ^2	*KL*	*JD*	*QF*	*EMD*
Symmetrical	+	+	+	+	-	+	+	+
Triangle inequality	+	+	+	-	-	-	+	+
Computation	Medium	Low	Medium	Medium	Medium	Medium	High	High
Ground Distance	-	-	+	-	-	-	+	+
Multivariate	+	-	-	+	+	+	+	+
Individual binning	-	+	-	-	-	-	-	+
Partial Matches	-	-	-	-	-	-	-	+
Non-Parametric	+	-	+	+	+	+	+	+

Distance vs. similarity

Problems

- The properties of distance functions are in general too restrictive for human evaluation of similarity (according to psychological research).
- Example: symmetry
  - The perceived similarity tends to be different depending on which image is the reference.
  - Ex. : if the first image is the object of a QBE, d(A, B) < d(B, A)





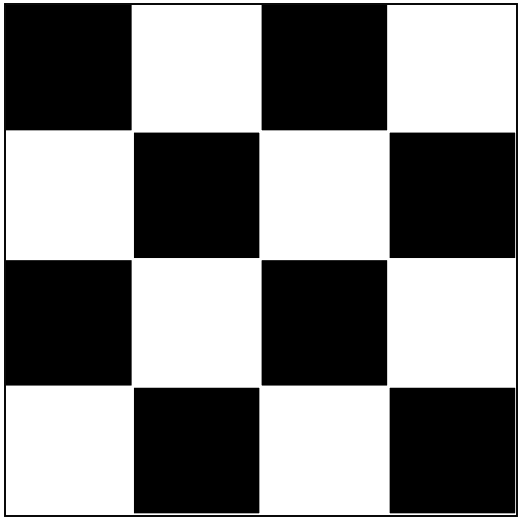
Distance vs. similarity

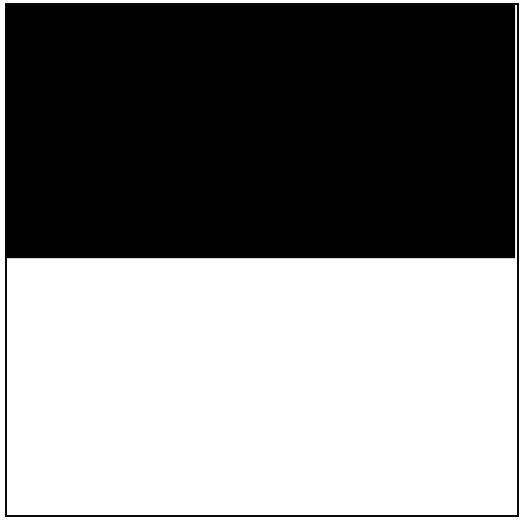
Problems

- Similarity perception is subjective.
  - Depends on knowledge, nationality, education, etc.
  - Example 1: an American will identify photos of two former presidents as similar → no similarity without contextual knowledge
  - Example 2: Comparison of a child's face with that of its parents, opinions may vary a lot → use of different criteria, with different weights etc.
- Multi-criteria queries integrating annotation-based and content-based retrieval are more performant.

Limitations of histograms

- Spatial distribution of pixels is ignored





Different images, same histogram

Spatial Relations between Pixels

- Different approaches:
  - Split the image into regions and compute histograms for each region
  - Remove background and compute histogram for foreground
  - Use of spatial layouts (cf. MPEG-7)

Table of Contents

Content Based (Image) Retrieval

Classical CBIR with Color Descriptors

- CBIR systems have historically focused on the color concept, using descriptors such as:
  - Color histogram
  - Dominant Color
  - Scalable Color, based on the color histogram (locally for a region, globally for the whole image)
  - Color Structure Descriptor (also takes the spatial structure into account)

- In the following we deal with the color theory in CBIR and describe how MPEG-7 handles it.

All MPEG-7 Visual Descriptors:

- Color
  - Color Space, Color Quantization
  - Dominant Color
  - Scalable Color (Histogram), Group of Frames Histogram
  - Color Structure, Color Layout
- Texture
  - Homogeneous Texture
  - Texture Browsing
  - Edge Histogram
- Shape
  - Contour Shape
  - Binary Shape
  - Shape 3D
- Motion
  - Camera Motion
  - Motion Trajectory
  - Parametric Motion
  - Motion Activity

Principles of MPEG-7 Visual Features

- Histogram-based, DCT for compact representation, enable complex filter/manipulation-operations,

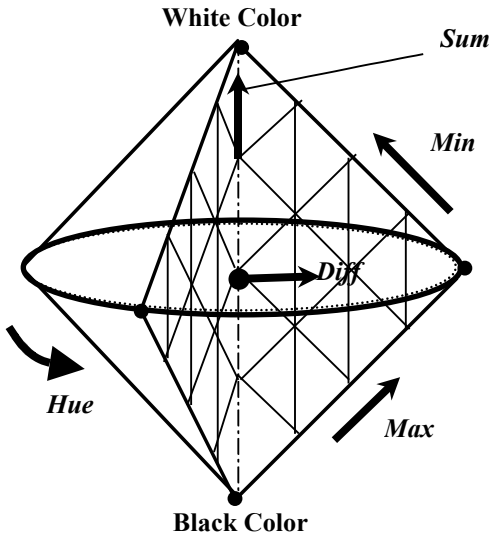
- Good methods of feature extraction are quite complex!
  - Only general information given in the lecture

- Recommended process: segment the image in regions and/or objects and then apply the descriptors
  -> Use of SpatialDecomposition of MPEG-7.

MPEG-7: Color

Color space description in MPEG-7

- Color Space
  - Color space in which the properties of the image are expressed
  - Support for RGB, HSV, HMMD, YC_bC_r, linear transformation matrix with reference to RGB, and monochrome
  - A reference color may be defined
  - Uniform color quantization assumed for histograms (except HMMD)



Example: HMMD

MPEG-7: Color

Dominant Color I

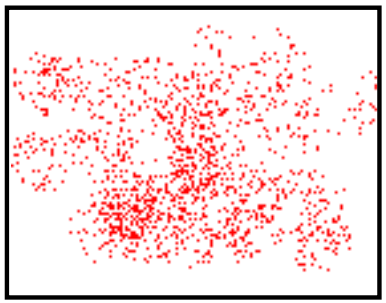
- Dominant Color ("DC")
  - The DC descriptor provides a compact description of the representative colors of an image/image region.
  - The DCD is defined as:
    - $F=\{(c_i,p_i,v_i),s\})$, i=1,..,N
    - with N the number of DCs, ci a vector expressed in a given color space (for example in 3-D for the RGB space), pi the percentage of pixels belonging to ci, and vi the variance of the color values around ci,
    - And s = spatial coherency
    - s and p_i are quantized to 5 bits

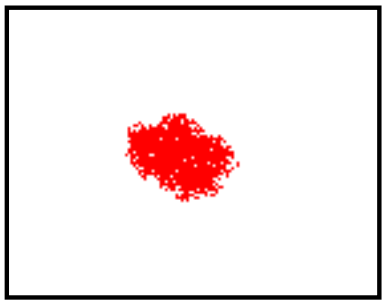
MPEG-7: Color

Dominant Color II

- Extraction algorithm: generalized Lloyd algorithm
  - Recommended: use of perceptually uniform color space such as CIE LUV
  - Computes clusters of pixels based on their color
  - Criterion: minimize cluster distortion defined as $D_i = \Sigma h(n) \, ||x(n)-c_i||^2, x(n) \in C_i$, with c_i the centroid of cluster C_i, x(n) the color vector of pixel n, h(n) the perception weight for n (accounts for different human perception of smooth and textured regions).

- SpatialCoherency: describes the spatial homogeneity of the colors





MPEG-7: Color

Dominant Color III

- Value: specifies an array of elements that hold percentages and values of colors in a visual item. The array elements consist of Percentage, ColorValueIndex and ColorVariance.
- Percentage: specifies the percentage of pixels that have the associated color value. The percentage value is uniformly quantized to 5 bits with 0 corresponding to 0 percentage and 31 corresponding to 100%.
- **Index**: specifies the index of the dominant color in the selected color space as defined in ColorQuantization. The number of bits for each component is derived from the ColorQuantization element
- **ColorVariance**: specifies an integer array containing the value of the variance of color values of pixels corresponding to the dominant color in the selected color space,i.e.

MPEG-7: Color

Example: a completely red image

```
<Image>
        <MediaLocator>
                <MediaUri> file://red.jpg</MediaUri>
        </MediaLocator>
        <TextAnnotation>
                <FreeTextAnnotation>
                        A completely red image :-)
                </FreeTextAnnotation>
        </TextAnnotation>
        <VisualDescriptor xsi:type="DominantColorType">
                <SpatialCoherency>31</SpatialCoherency>
                <Value>
                        <Percentage>31</Percentage>
                        <Index>255 0 0</Index>
                        <ColorVariance>0 0 0</ColorVariance>
                </Value>
        </VisualDescriptor>
</Image>
```

MPEG-7: Color

Similarity matching based on DCD

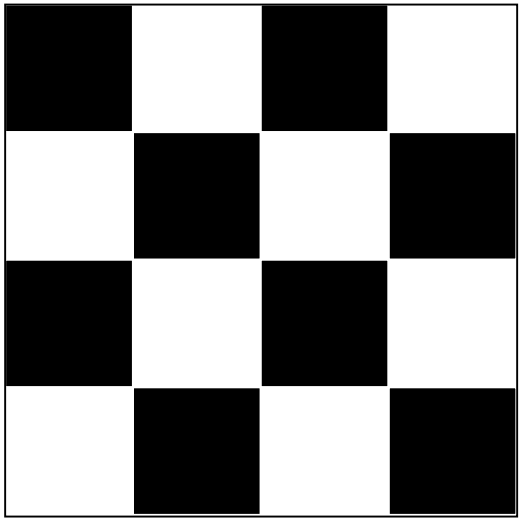
- Assuming two DCs defined as:
  - $F_1=\{(c_{1i},p_{1i},v_{1i}),s_1\})$, $i=1,..,N_1$
  - $F_2=\{(c_{2i},p_{2i},v_{2i}),s_2\})$, $i=1,..,N_2$

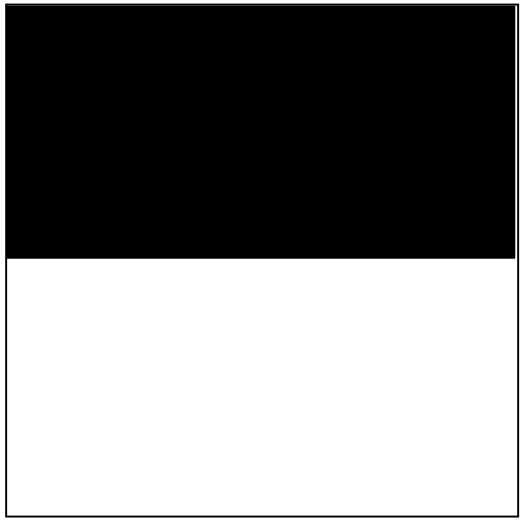
- In the simplest form, spatial coherency and variance are ignored. The Dissimilarity $D(F_1,F_2)$ is then defined as:
  - $D^2(F_1,F_2)= \sum p^2_{1i} + \sum p^2_{2j} - \sum\sum 2a_{1i,2j}p_{1i}p_{2j}$
  - Where $a_{k,l}$ is the similarity coefficient of colors c_k and c_l:
    - $a_{k,l}$ is 0 if the distance between the colors has a given property in a given color space, ex. with CIE LUV between 10 and 20

MPEG-7: Color

Color Structure Descriptor (CSD)

- The CSD represents an image through both color distribution and local structure.





Unlike a classical global histogram,
the CSD can distinguish these 2 images!

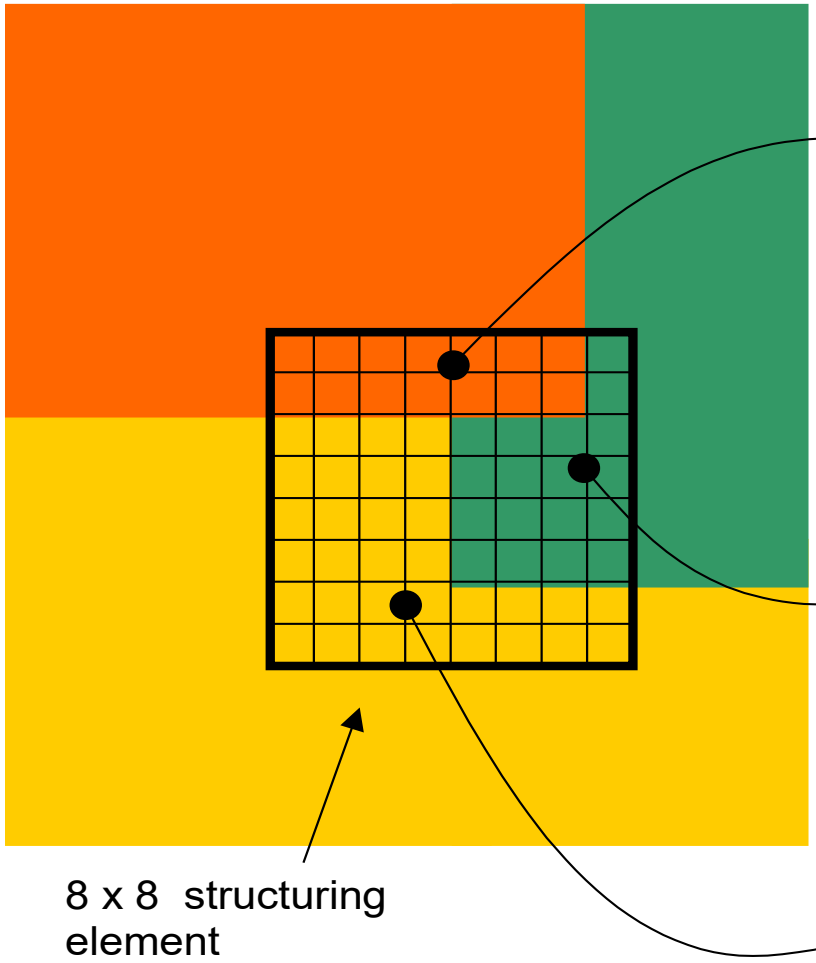
MPEG-7: Color

Color Structure Descriptor (CSD)

- The CSD is identical to a 1-D color histogram but has different semantics:
  - CSD = h(m), m ∈ {0, 1, ..., M-1},
  - M is selected out of {256, 128, 64, 32}.
  - The Bin value is quantized to 8 Bits.
  - The histogram values h(m) count how many colors appear in a so-called "structuring element".
  - The structuring element is an 8x8 element moved across the image. At each position the bin value is increased depending on whether the color appears.
  - The color space is the HMMD color space.

MPEG-7: Color

Color Structure Descriptor (CSD) - Example

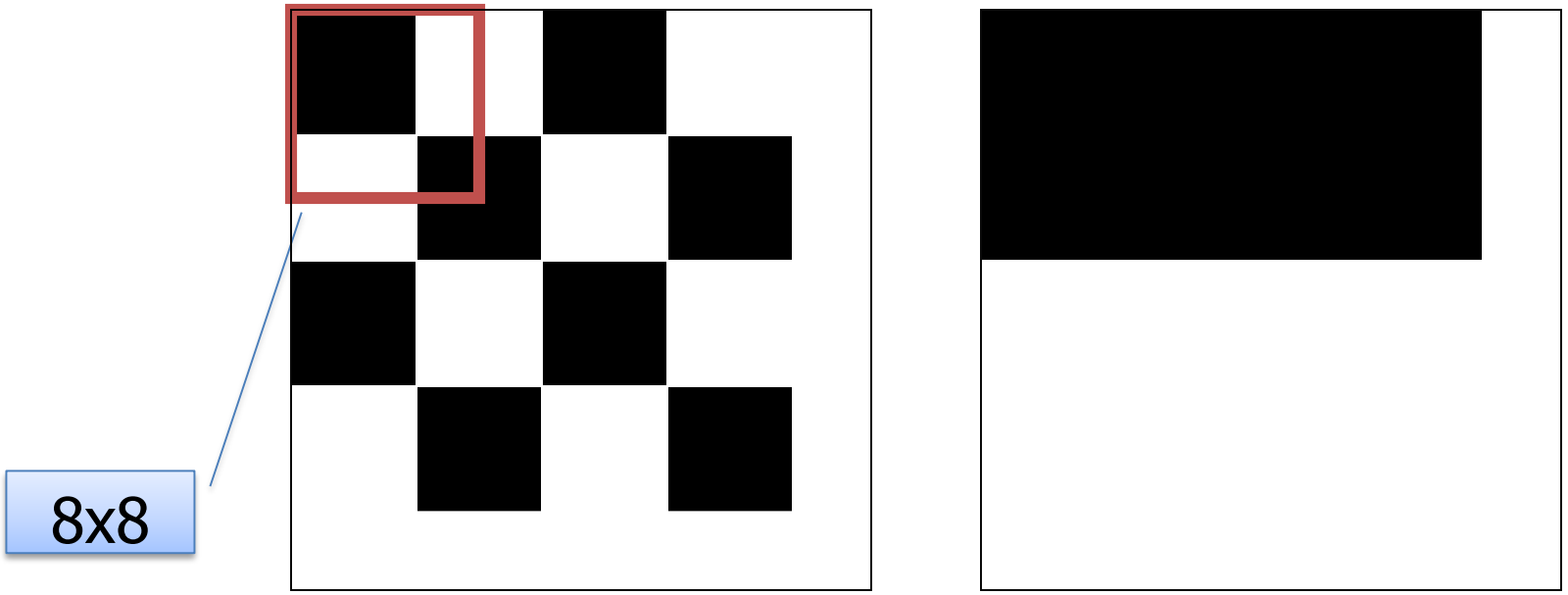


COLOR	BIN
C0	
C1	+1
C2	
C3	+1
C4	
C5	
C6	
C7	+1

MPEG-7: Color

Example: Color Structure Descriptor I

- Example images



MPEG-7: Color

Example: Color Structure Descriptor II

- XML file for the left image

```
<Image>
    <MediaLocator>
      <MediaUri> file://left.jpg</MediaUri>
     </MediaLocator>
     <TextAnnotation>
          <FreeTextAnnotation>
               The left black and white image
          </FreeTextAnnotation>
     </TextAnnotation>
     <VisualDescriptor xsi:type="ColorStructureType"
                       colorQuant="128">
          <Values>  9 0 0 0 ... 0  9 </Values>
     </VisualDescriptor>
</Image>
```

MPEG-7: Color

Example Color Structure Descriptor III

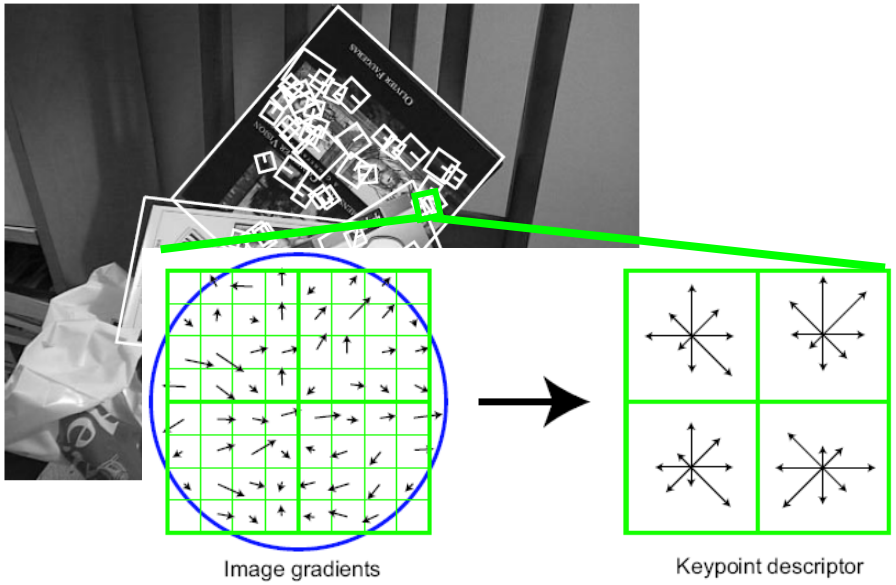
- XML file for the right image

```
<Image>
    <MediaLocator>
     <MediaUri> file://right.jpg</MediaUri>
    </MediaLocator>
    <TextAnnotation>
     <FreeTextAnnotation>
           A black and white image
     </FreeTextAnnotation>
    </TextAnnotation>
    <VisualDescriptor xsi:type="ColorStructureType"
                      colorQuant="128">
       <Values>  6 0 0 0 ... 0  7 </Values>
    </VisualDescriptor>
</Image>
```

Current Features from the literature

SIFT, PCA-SIFT, SURF, …

- SIFT (Scale Invariant Feature Transform)
  - Detects interest points -> local feature
  - Characterizes the image in a scale, angle, … independent way
  - Similar to primate visual system



Important Links:

http://www.cs.ubc.ca/~lowe/keypoints/
http://www.vision.ee.ethz.ch/~surf

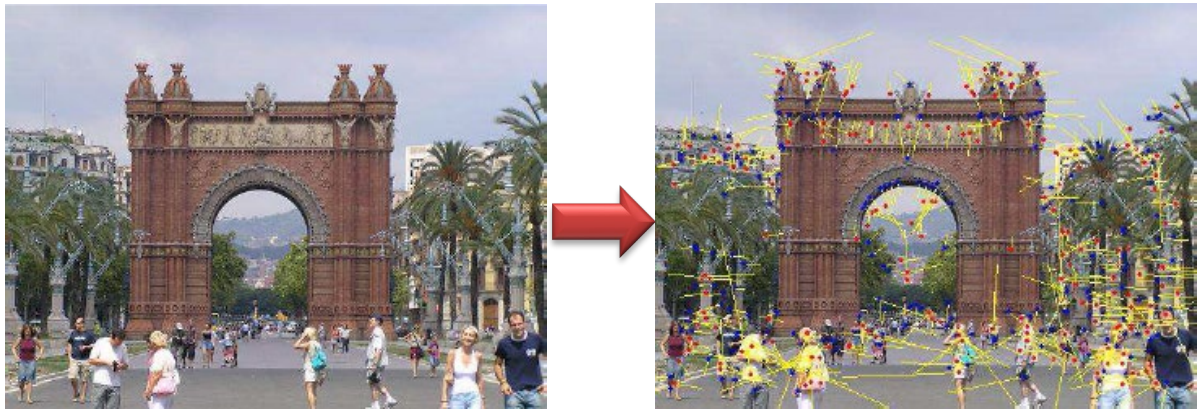
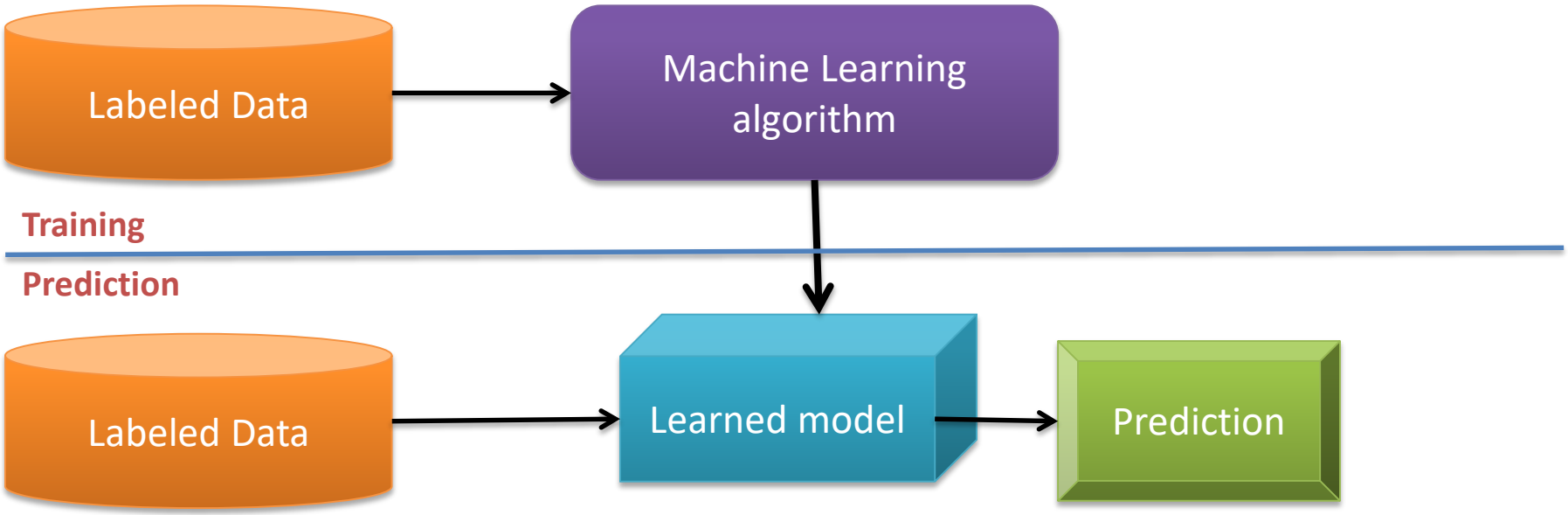


Table of Contents

Content Based (Image) Retrieval

Machine Learning Basics

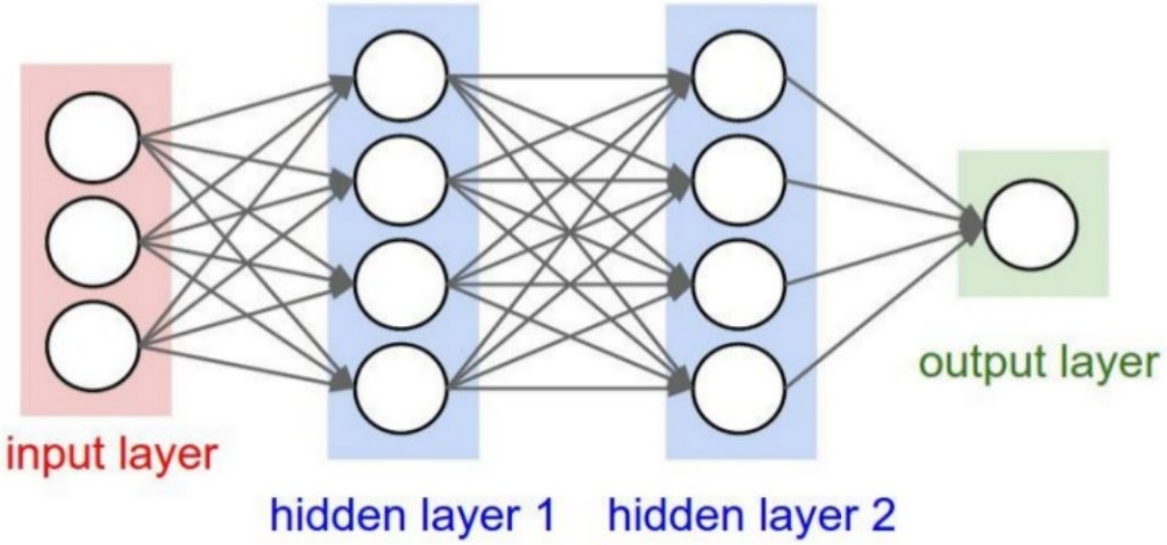
Machine learning is a field of computer science that gives computers the ability to **learn without being explicitly programmed**



Methods that can learn from and make predictions on data

Deep Learning - Basics

- Consists of one input, one output and multiple fully-connected hidden layers inbetween . Each layer is represented as a series of neurons and progressively extracts higher and higher-level features of the input until the final layer essentially makes a decision about what the input shows. The more layers the network has, the higher level it will learn.

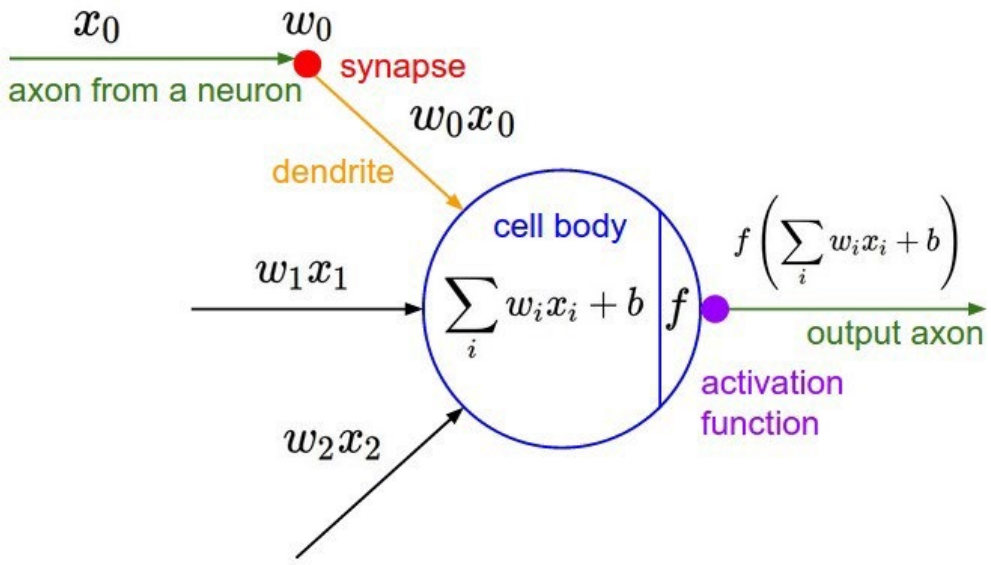


Deep Learning - Basics

- Forward Propagation
  - Input neurons receive the data features of the object. After processing the data the output is forward to the first hidden layer
  - Every hidden layer processes the individual input and sends the result to the next hidden layer
  - Finally the data reaches the output layer where the output value determines the object‘s classification
- Backward Propagation
  - To train a neural network over a large set of labelled data, you must continuously compute the difference between the network predicted output and the actual output
  - This difference is called cost and the process for training a net is known as backpropagation
  - During backprop, weights and biases are tweaked slightly until the lowest possible cost is achieved.

Deep Learning – Basics
The Neuron

- An artifical neuron contains a nonlinear activation function and has several incoming and outgoing weighted connections
- Neurons are trained to filter and detect specific features or patterns (e.g., edge, nose) by receiving weighted input, transforming it with the activation function and passing it to the outgoing connections.



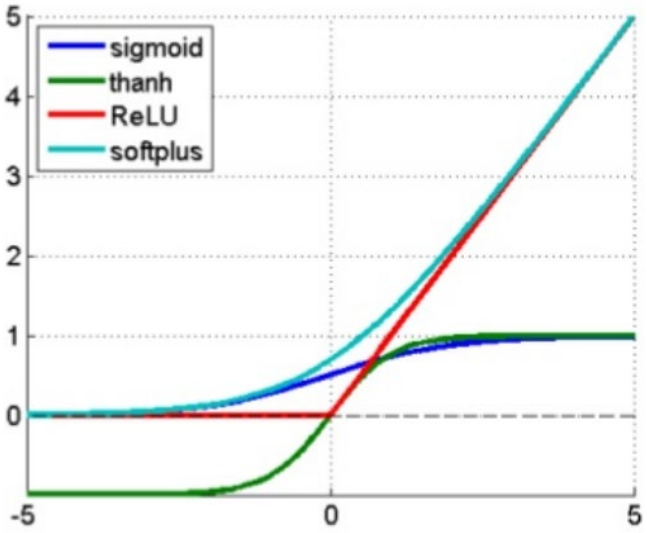
© https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc

Deep Learning – Basics
Non linear Activation Function

- Most deep networks use ReLU – max(0,x) – nowadays for hidden layers, since it trains much faster, is more expressive than logistic function and prevents the gradient vanishing problem
- Non-linearity is needed to learn complex (non-linear) representations of data, otherwise the NN would be just a linear function

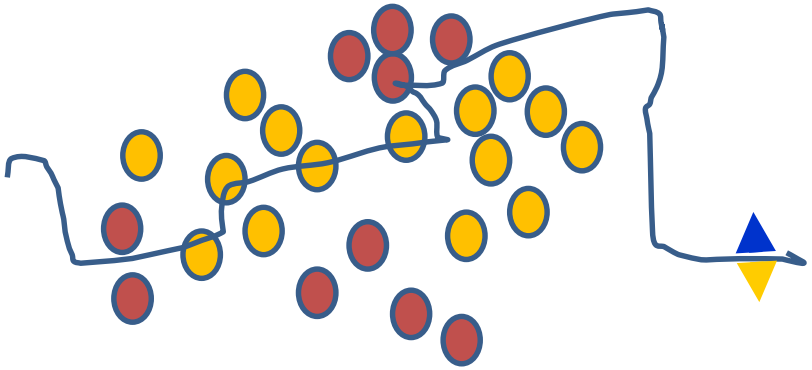
Rectified Linear Unit (ReLU)

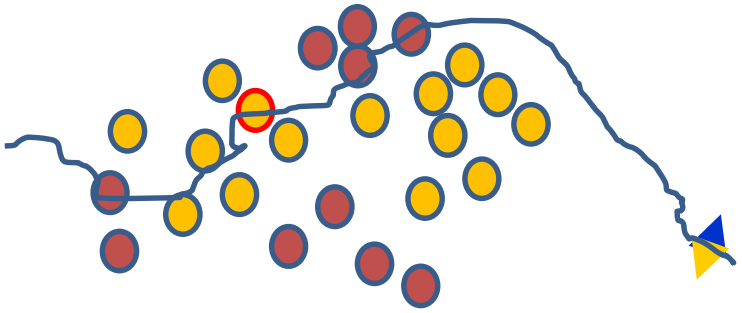
$$f(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$$



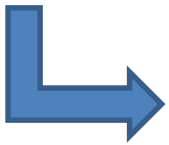
Deep Learning – Basics
Non linear Activation Function

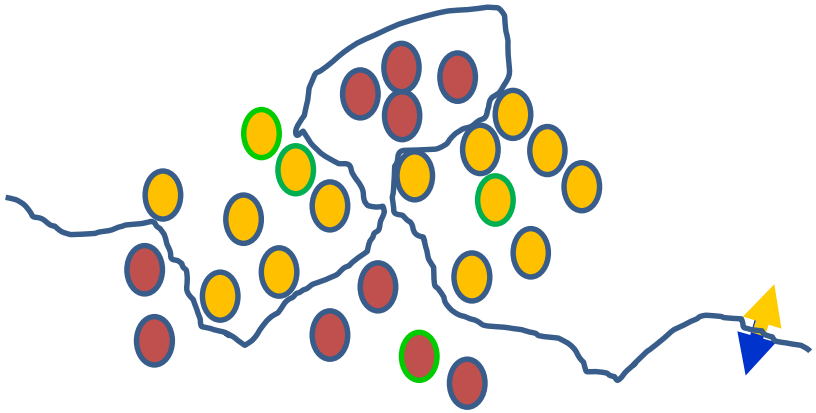
- Principial Idea of Training phase and its correlation with the activation function





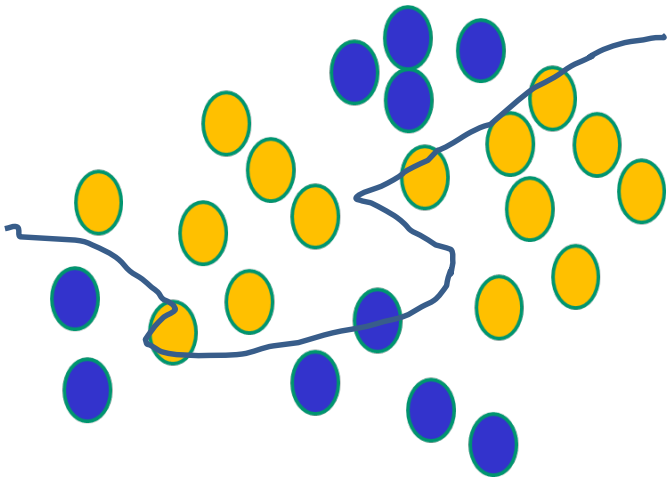
Result of training multiple epochs



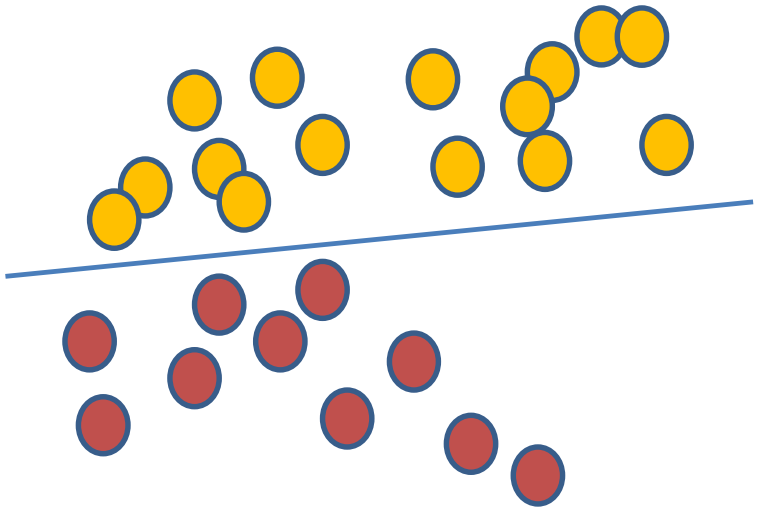


Deep Learning – Basics
Non linear Activation Function

NNs use nonlinear *f*(x) so they can draw complex boundaries, but keep the data unchanged



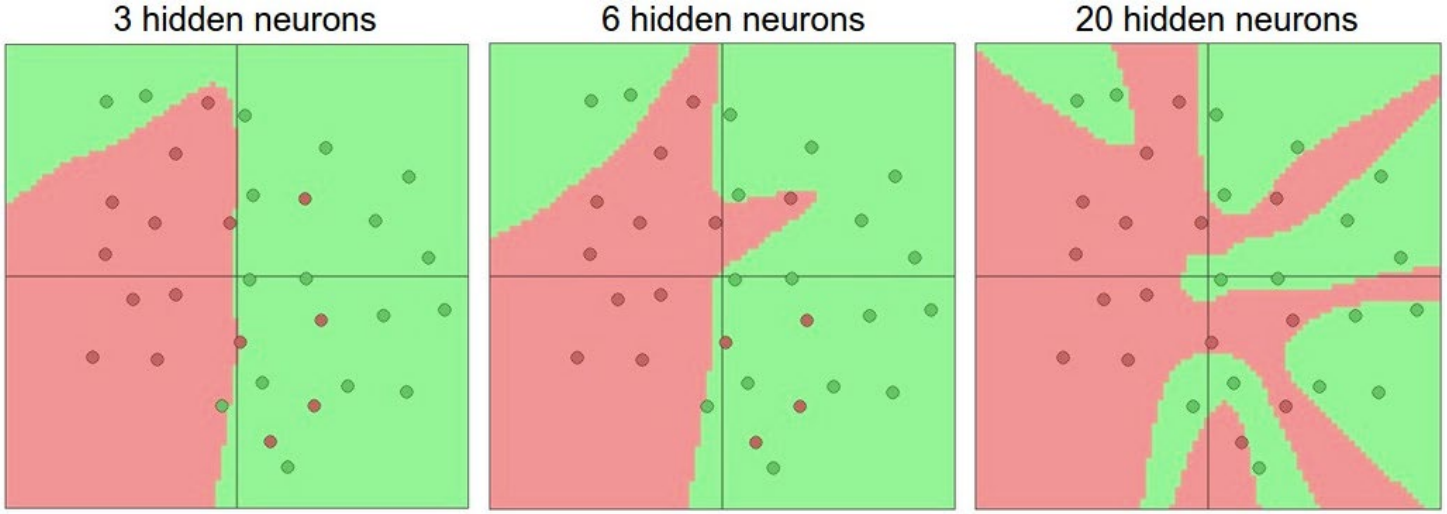
SVMs only draw straight lines, but they transform the data first in a way that makes that OK



If *f*(*x*) is linear, the NN can **only** draw straight decision boundaries (even if there are many layers of units)

Deep Learning – Basics
Non linear Activation Function

Non-linearities needed to learn complex (non-linear) representations of data, otherwise the NN would be just a linear function $W_1 W_2 x = Wx$

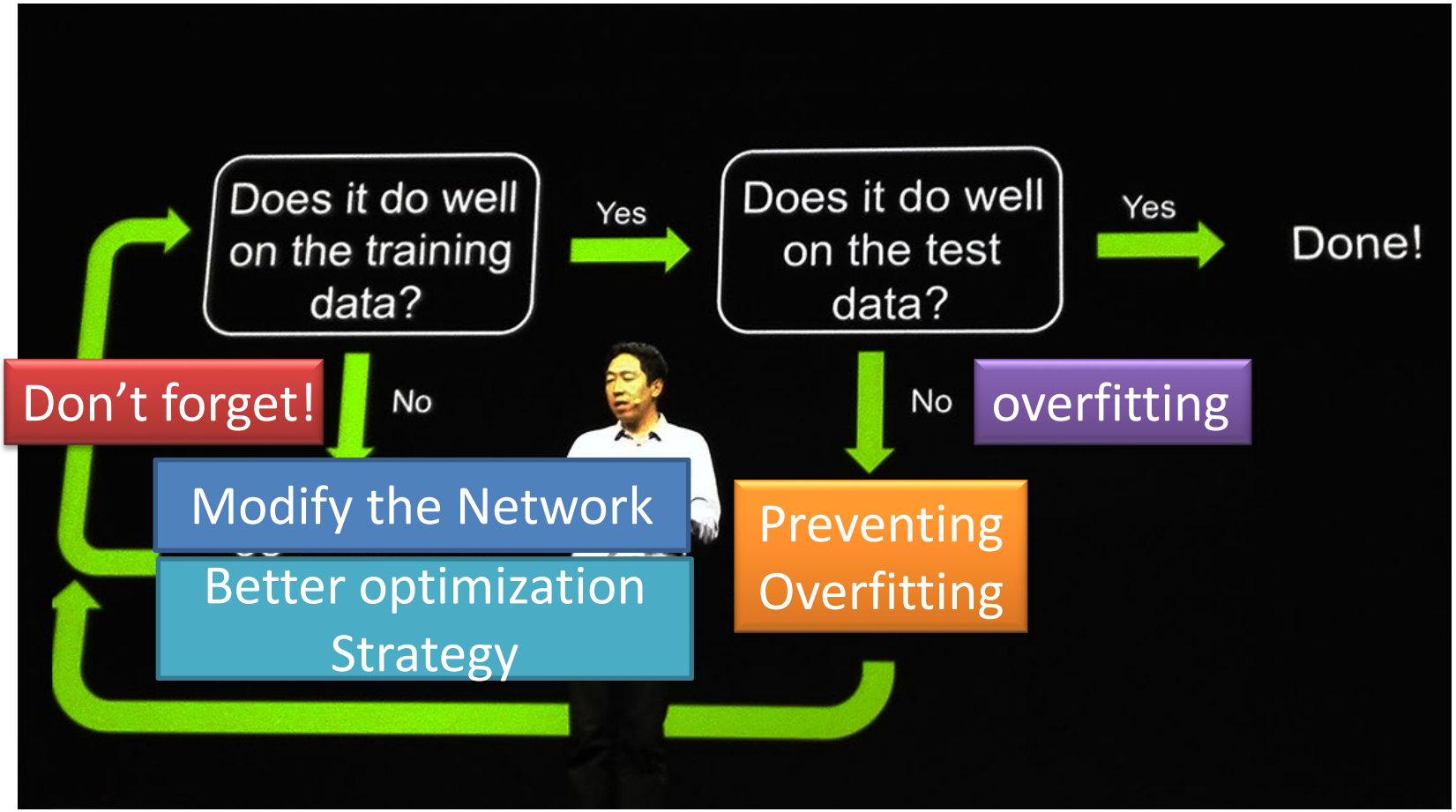


http://cs231n.github.io/assets/nn1/layer_sizes.jpeg

More layers and neurons can approximate more complex functions LIVE Demo

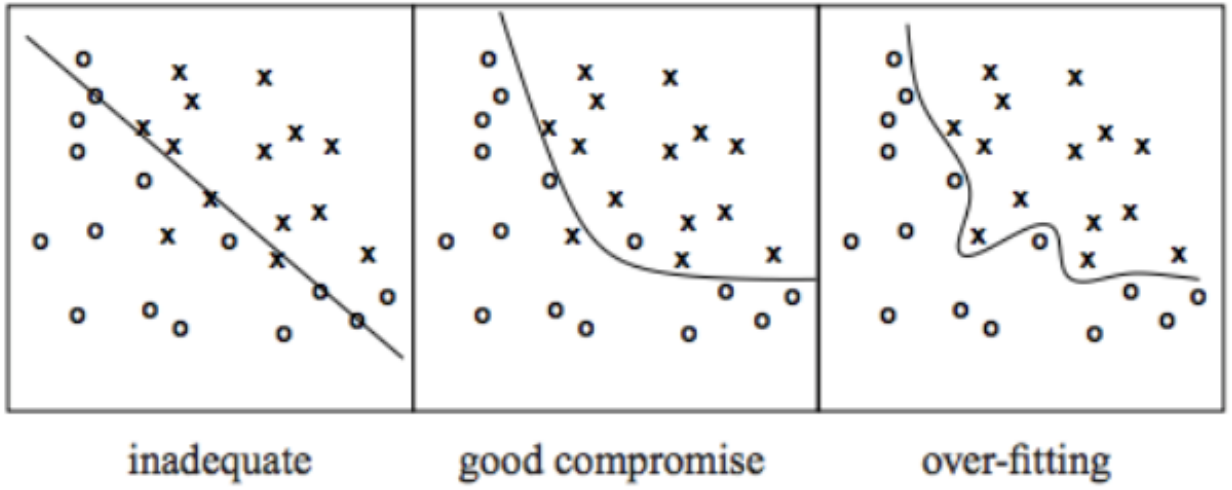
http://playground.tensorflow.org/

Deep Learning – Basics
Recipe for Learning

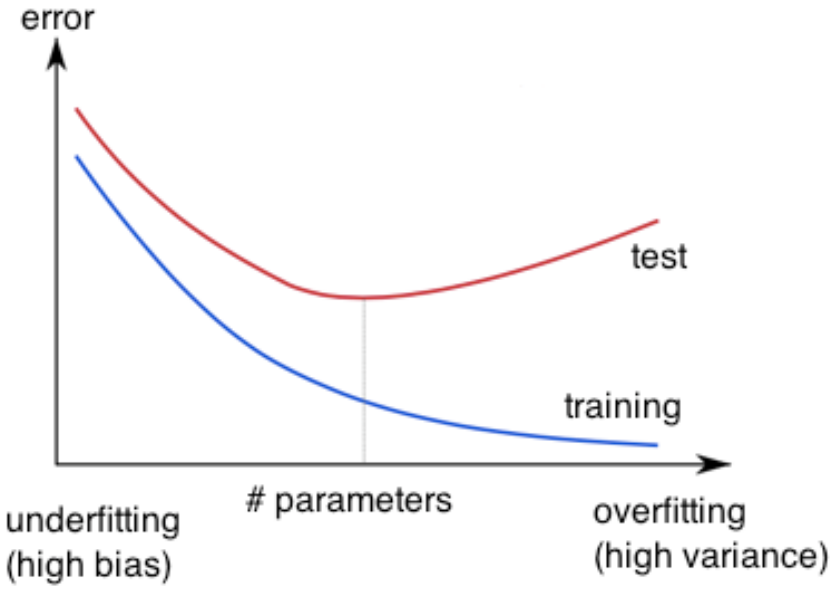


http://www.gizmodo.com.au/2015/04/the-basic-recipe-for-machine-learning-explained-in-a-single-powerpoint-slide/

Overfitting



http://wiki.bethanycrane.com/overfitting-of-data

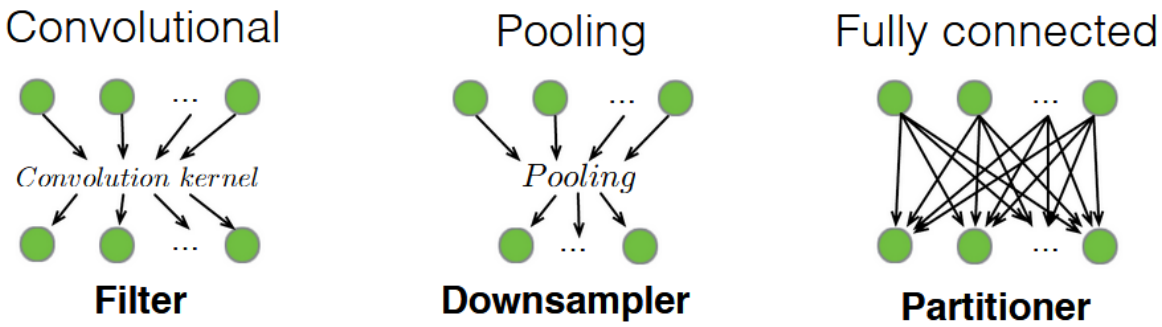


https://www.neuraldesigner.com/images/learning/selection_error.svg

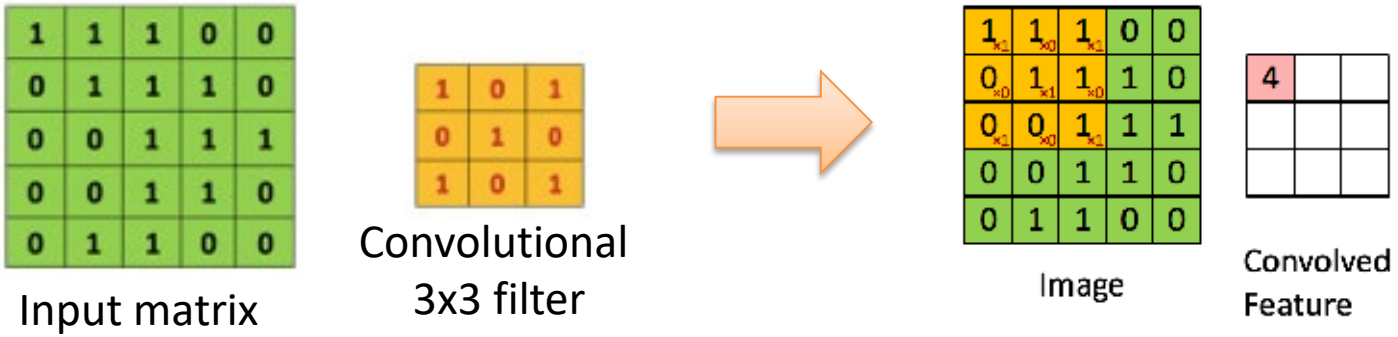
Learned hypothesis may **fit** the training data very well, even outliers (**noise**) but fail to **generalize** to new examples (test data)

Convolutional Neural Networks (CNNs)

Different types of layers:



Example:

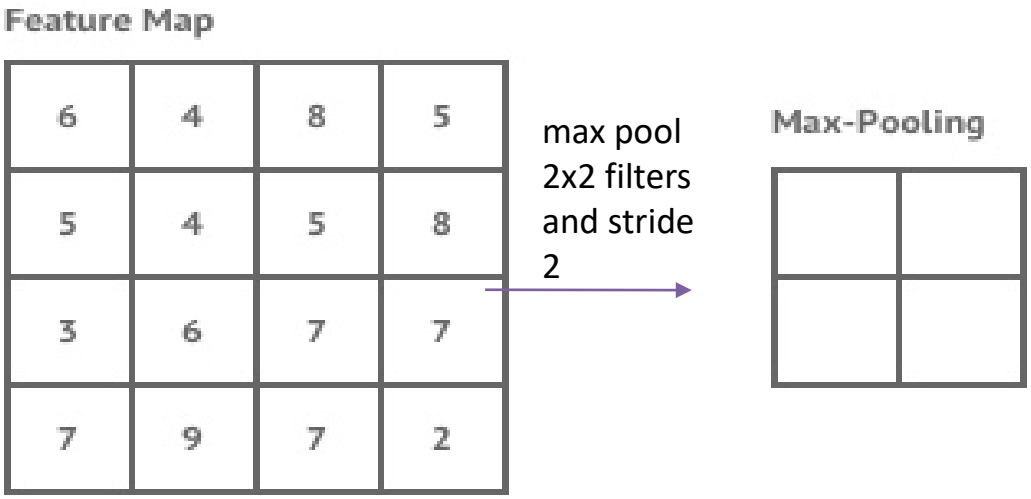


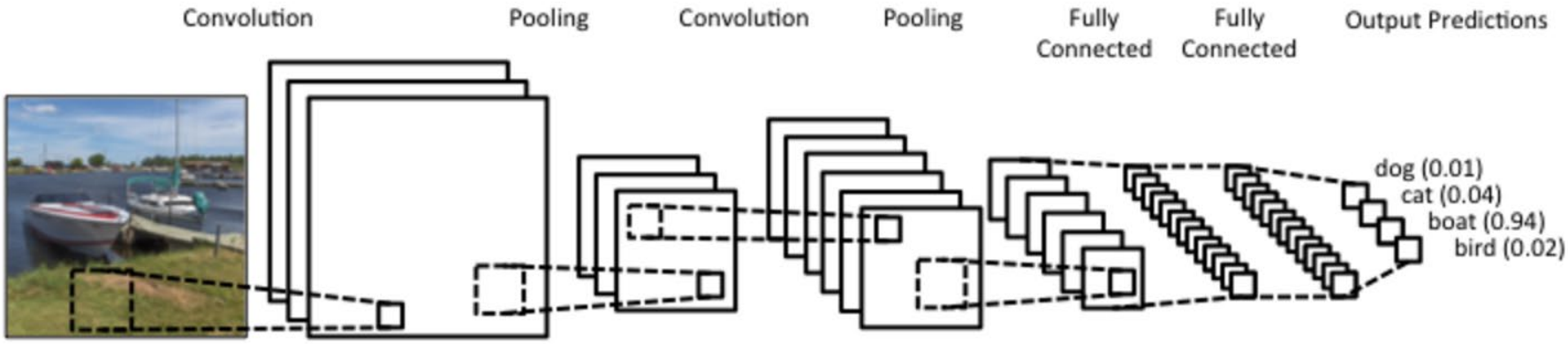
http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

DEMO: http://setosa.io/ev/image-kernels/

Convolutional Neural Networks (CNNs)

Example:



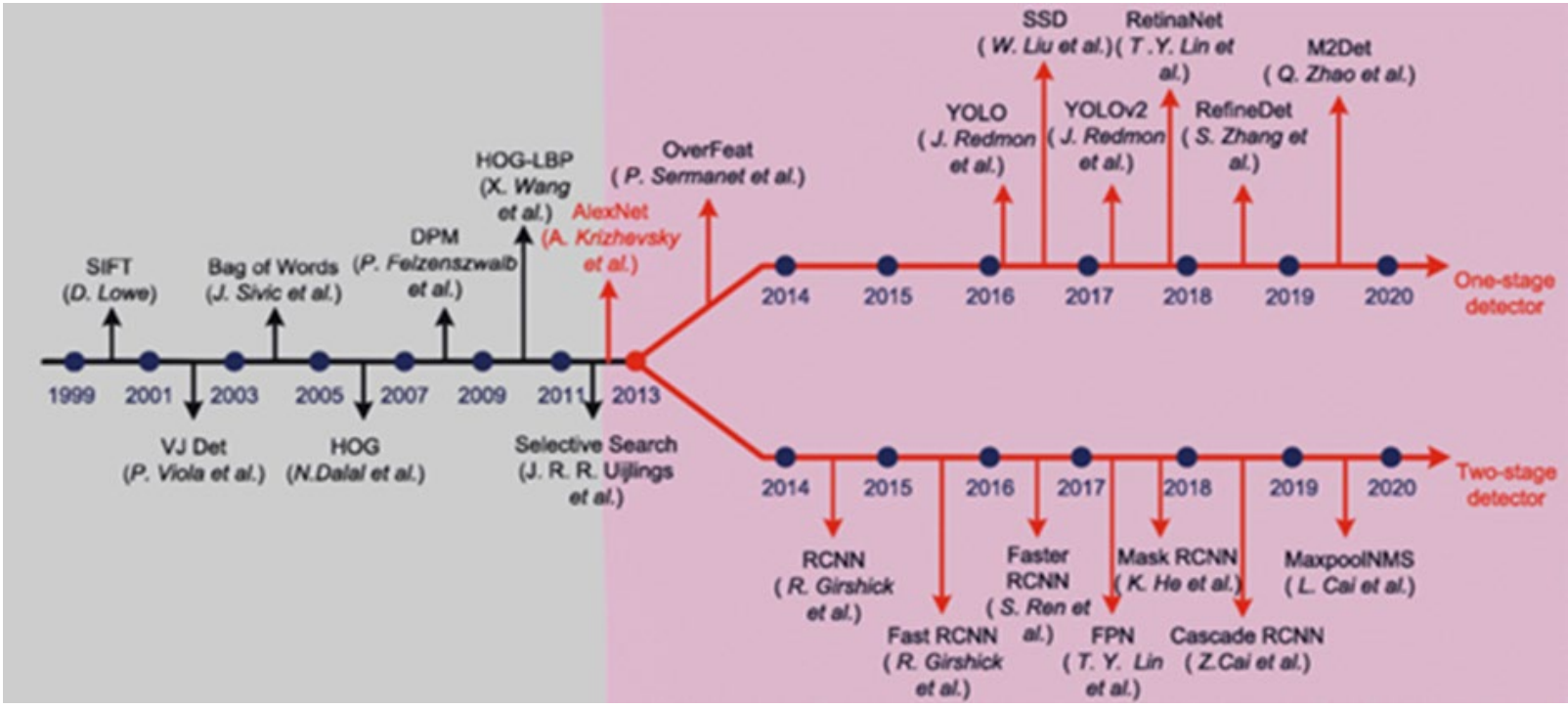


https://shafeentejani.github.io/assets/images/pooling.gif

Convolutional Neural Networks (CNNs)

- Good overview given in Figure below [12] by 2020. But still does not reflect latest findings such as Yolo4 [13]
- Same for GAN where the GAN Zoo reports more than 500 different networks

History & Evolution CNN for object classification [12]



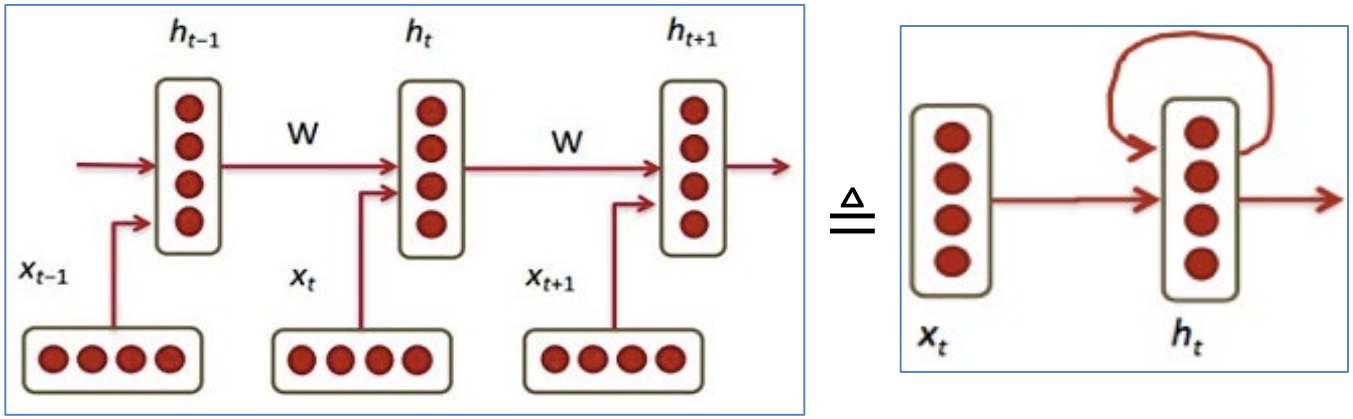
CNN: Application

- Classify a scene in an image
  - Image classifier Demo (NYU): http://horatio.cs.nyu.edu
- Describe or understanding an image
  - Toronto Deep Learning Demo: http://deeplearning.cs.toronto.edu/i2t
  - MIT Scene Recognition Demo: http://places.csail.mit.edu/demo.html
- Handwritten digits recognition: http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html
- Video classification
  - Large-scale Video Classification with Convolutional Neural Networks http://cs.stanford.edu/people/karpathy/deepvideo

Recurrent Neural Networks (RNNs)

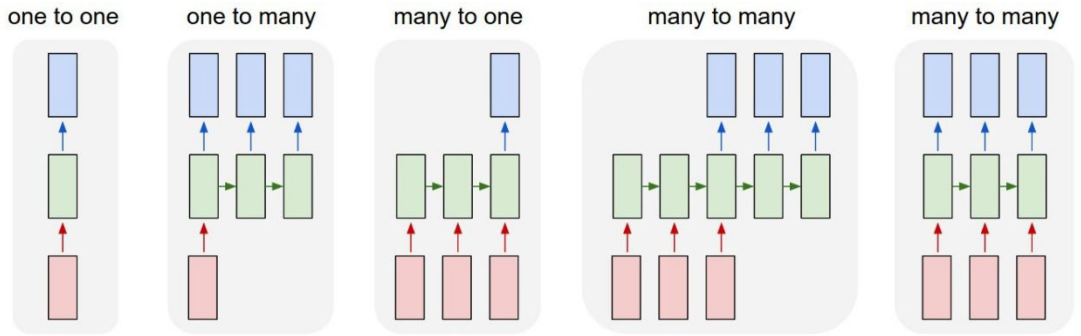
Main RNN idea for text:
Condition on **all previous words**
Use same set of weights at all time steps

$$h_t = \sigma(W^{(hh)} h_{t-1} + W^{(hx)} x_t)$$



https://pbs.twimg.com/media/C2j-8j5UsAACgEK.jpg

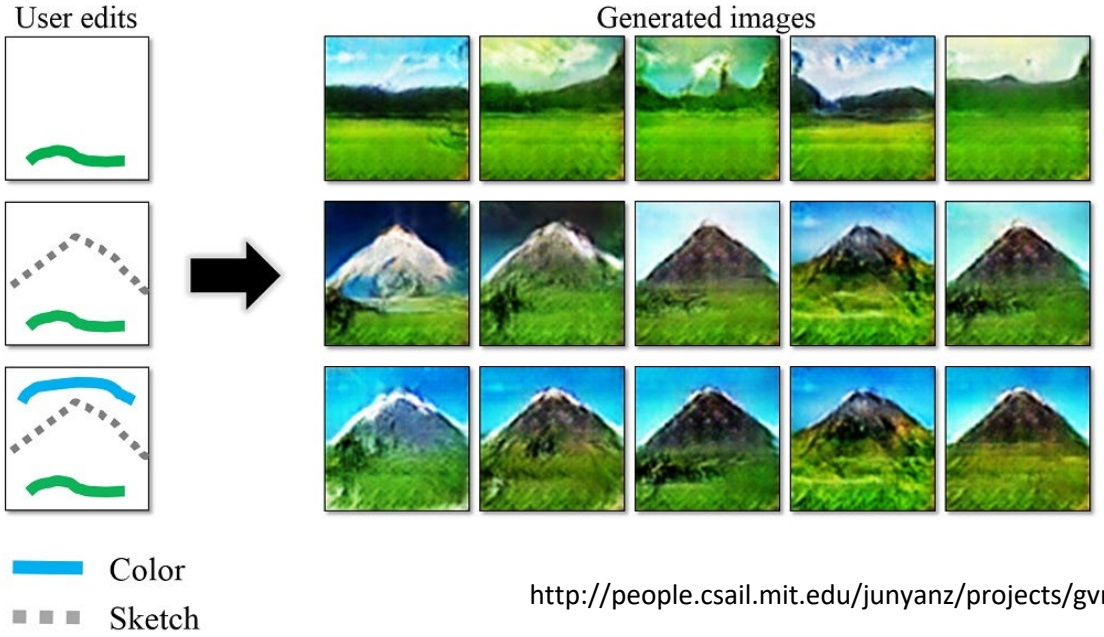
- Stack them up, Lego fun!
- Vanishing gradient problem



https://discuss.pytorch.org/uploads/default/original/1X/6415da0424dd66f2f5b134709b92baa59e604c55.jpg

Generative Adversarial Networks (GANs)

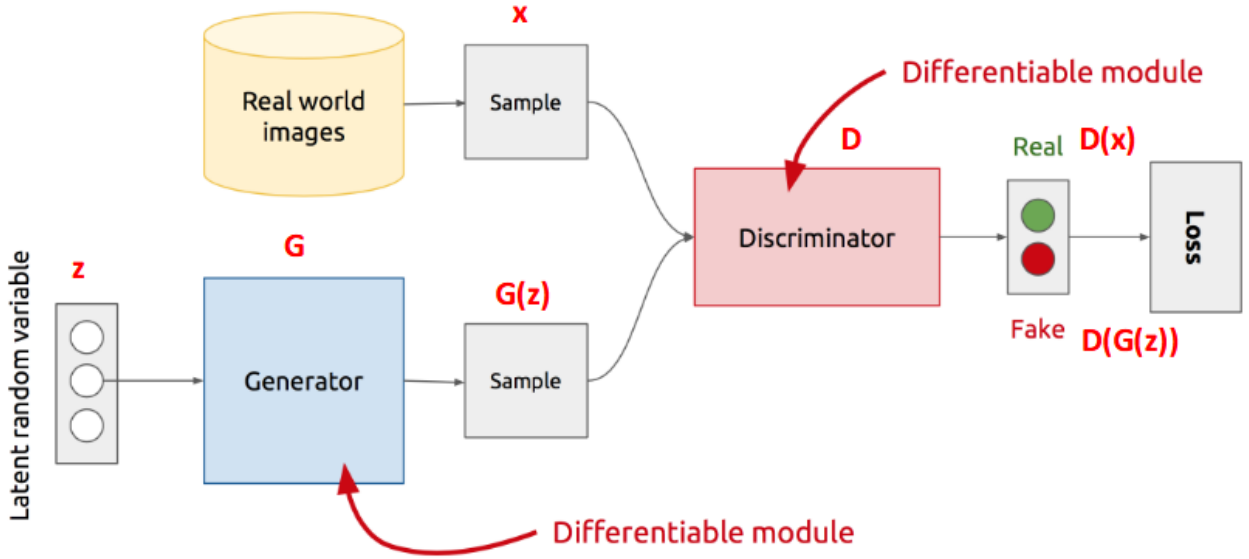
- What are Generative Models?
  - Key idea: the model cares about what distribution generated the input data points, and we want to mimic it with our probabilistic model. Our learned model should be able to make up new samples from the distribution.



http://people.csail.mit.edu/junyanz/projects/gvm/

Generative Adversarial Networks (GANs)

- GAN architecture
  - We have a pair of networks: Generator (G) and Discriminator (D):
  - They „fight“ against each other during training => Adversarial Training
  - G mission: make its Pmodel as much similar as possible to our training set distribution Pdata => Try to make predictions so realistic that D can‘t distinguish
  - D mission: distinguish between G samples and real samples

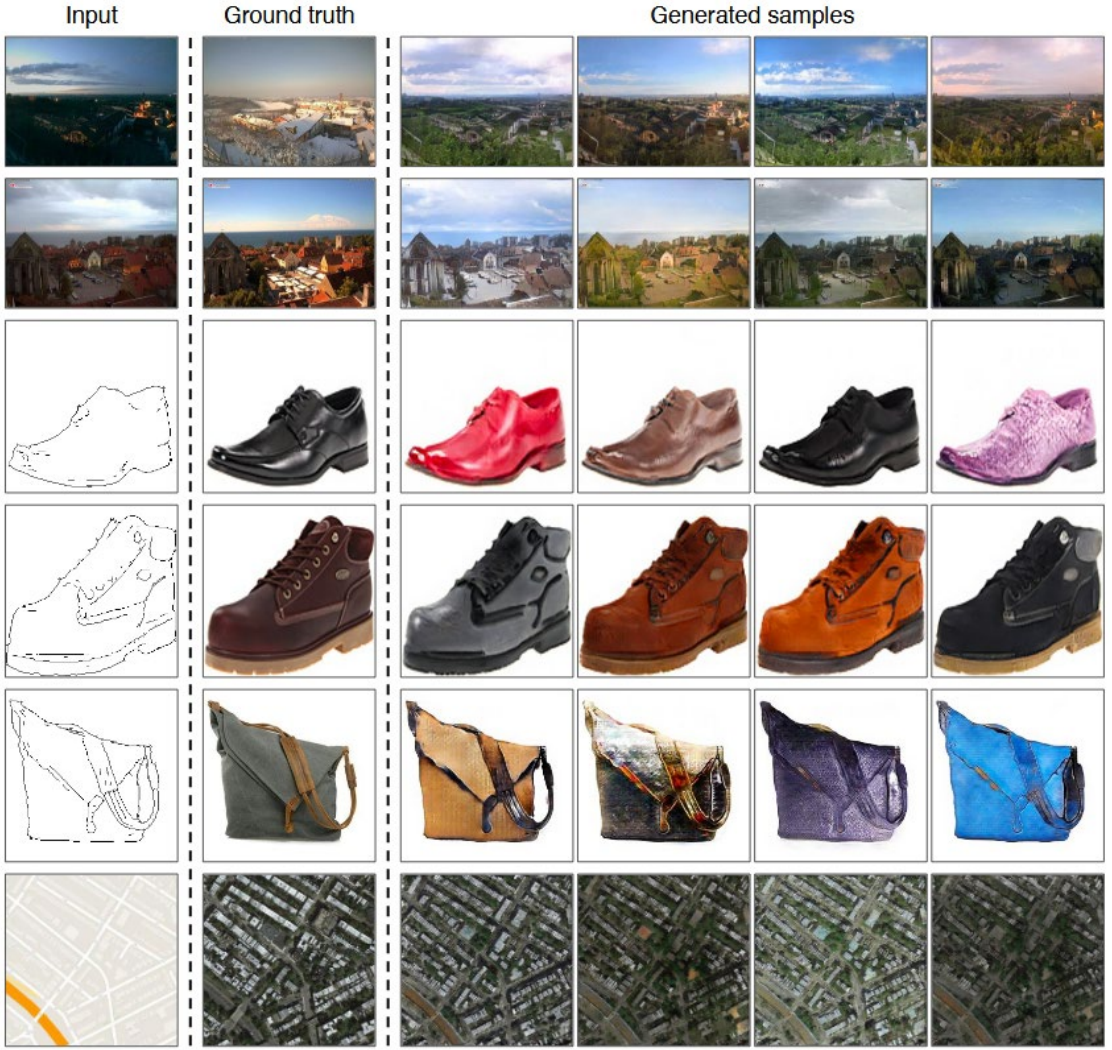


Generative Adversarial Networks (GANs)

- GAN examples

Jun-Yan Zhu, et. Al; Toward Multimodal Image-to-Image Translation; In 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA

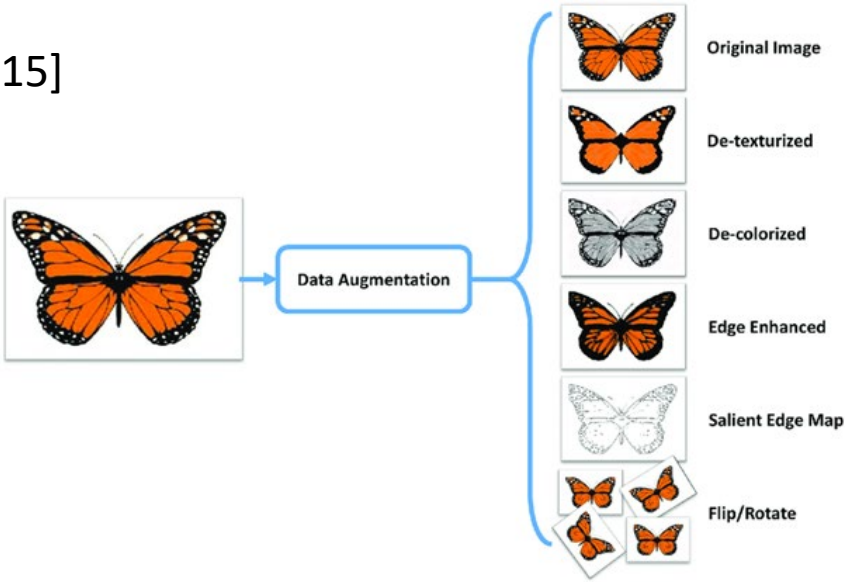
https://stablediffusionweb.com/



Data Augmentation

- Data augmentation is used to enlarge the data set for training of neuronal networks.
- Many approaches available: vertical flip, horizontal flip, rotation, blurring, etc.
- CLoDSA is a tool for image recognition tasks involving augmentations [17]

Stacked Geometric and Photometric Augmentations [15]



Augmentation techniques [16]

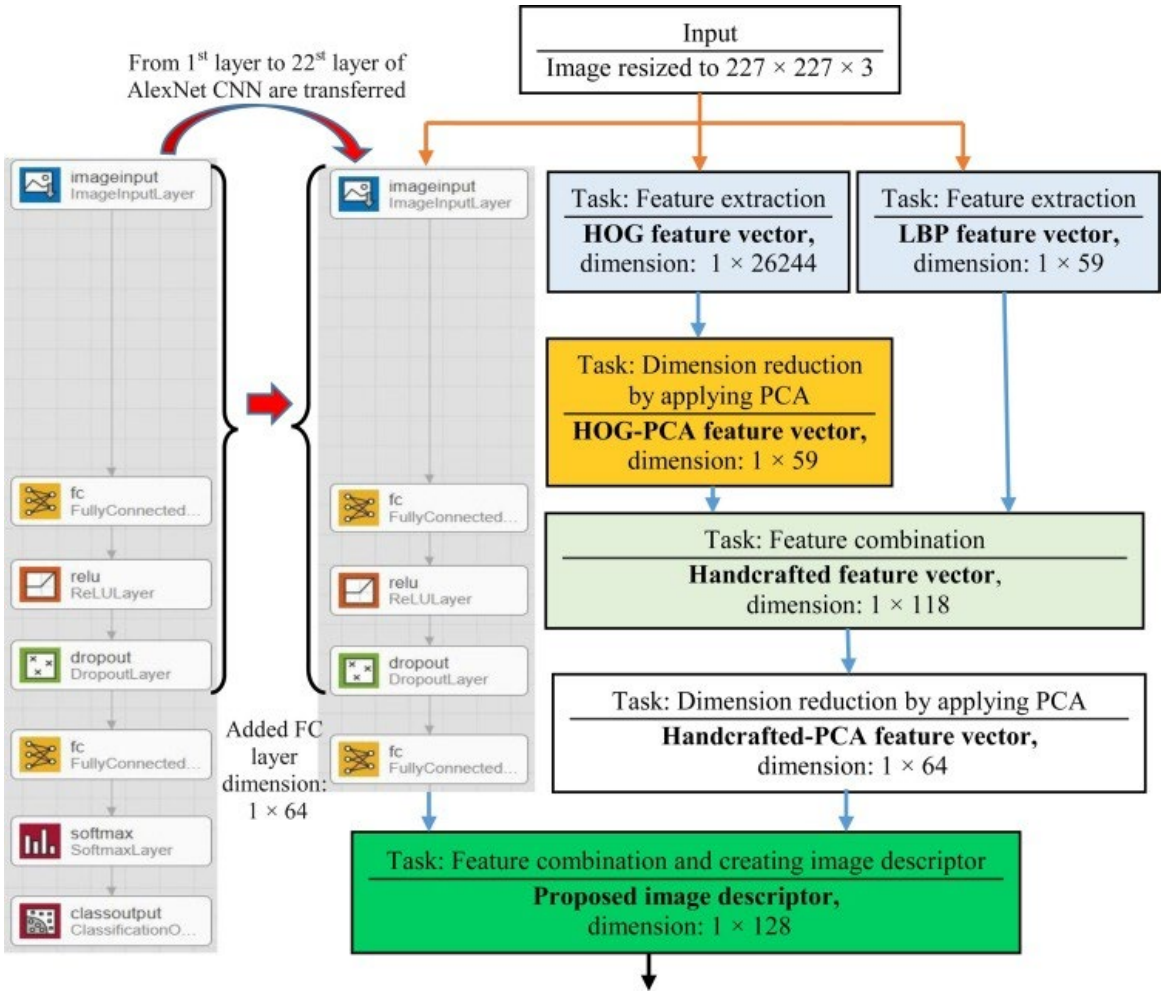
Deep Learning Benefits

- Robust
  - No need to design the features ahead of time – features are automatically learned to be optimal for the task at hand
  - Robustness to natural variations in the data is automatically learned
- Generalizable
  - The same neural net approach can be used for many differend applications and data types
- Scalable
  - Performance improves with more data, method is massively parallelizable

Deep learning Weaknesses

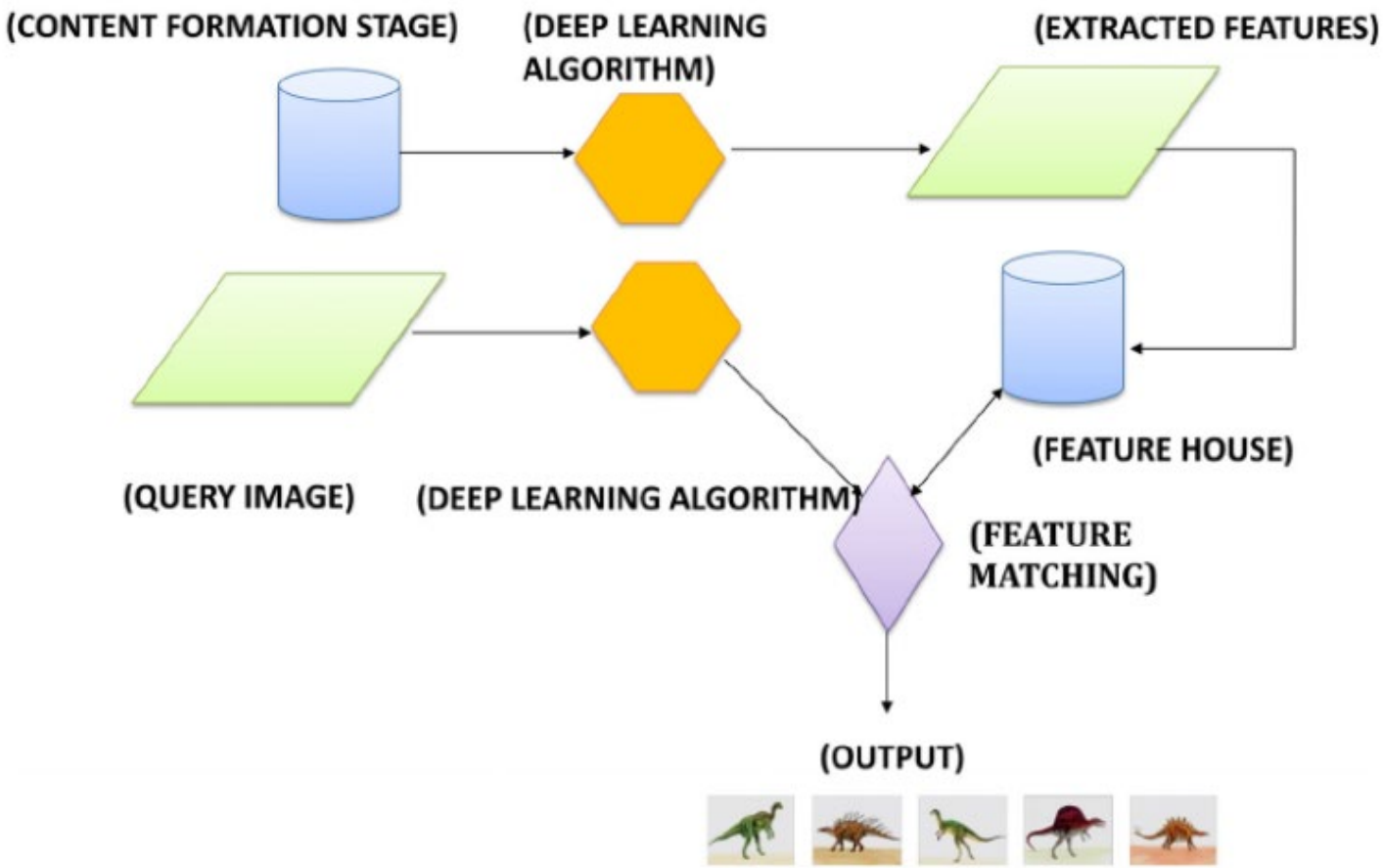
- Deep learning requires a large dataset, hence long training period
- In term of cost, machine learning methods like SVMs and other tree ensembles are very easily deplayed even by relative machine learning novices and can usually get your reasonably good results
- Deep learning methods tend to learn everything. It`s better to encode prior knowledge about structure of images
- The learned features are often difficult to understand. Many vision features are also not really human understandable (e.g. combination of different features)

Deep learning Integration into Feature Extraction



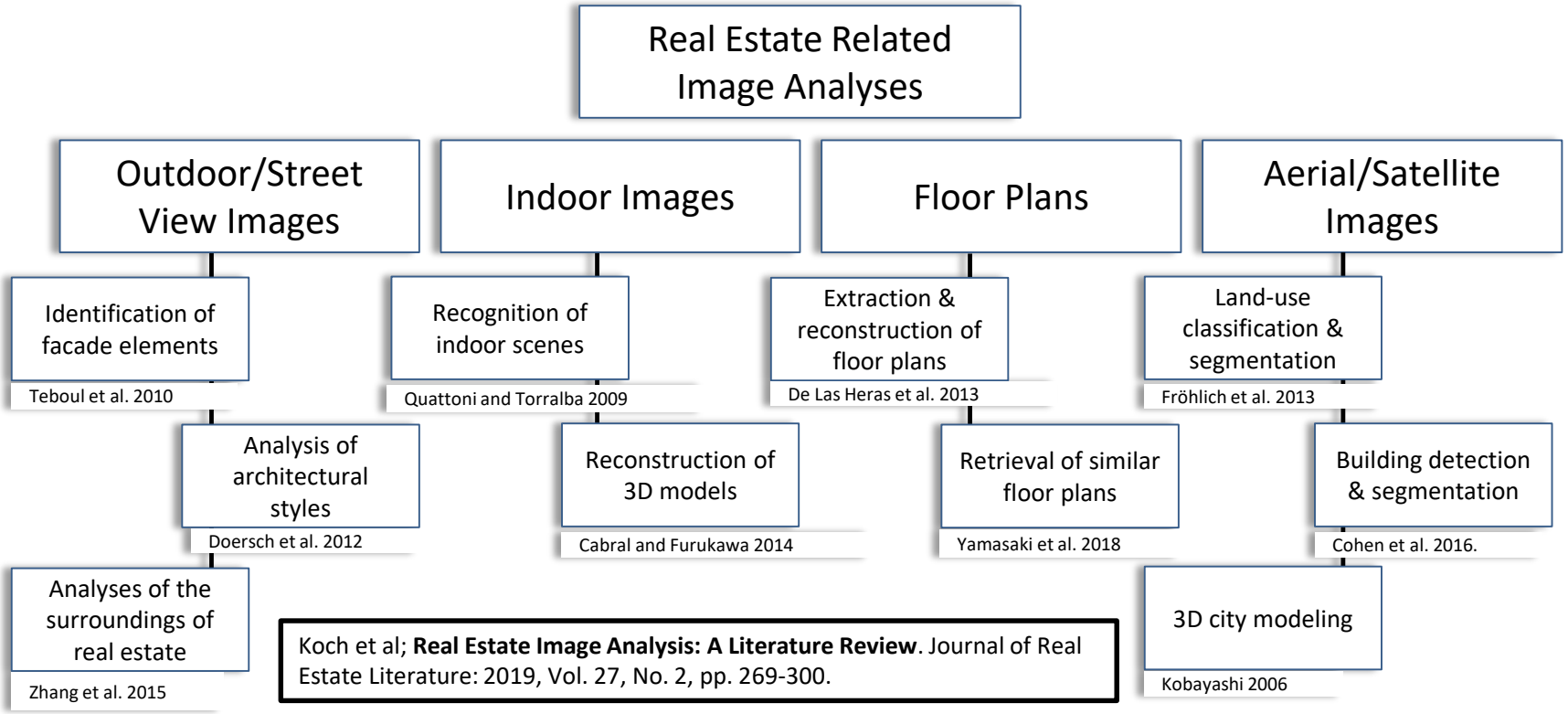
Ashkan Shakaramia,Hadis Tarrah; An efficient image descriptor for image classification and CBIR; Optik (Stuttg) 2020 Jul; 214: 164833. Published online 2020 doi: 10.1016/j.ijleo.2020.164833; PMCID: PMC7198219

Deep learning Integration into Query Process

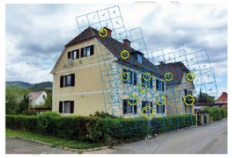


R. Rani Saritha, Varghese Paul, P. Ganesh Kumar; Content based image retrieval using deep learning process, Cluster Computing The Journal of Networks, Software Tools and Applications, 2018, DOI 10.1007/s10586-018-1731-0

Use Case Real Estate Image Analysis: Related Work



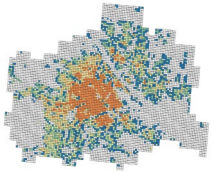
Some Contributions of the team



- Despotovic et al; **Prediction and analysis of heating energy demand for detached houses by computer vision**; Energy & Building; 193, 29–35, 2019

» Zeppelzauer et al; **Automatic Prediction of Building Age from Photographs**, In ACM International Conference on Multimedia Retrieval (ICMR) 2018 Conference, Japan, 2018 awarded by MIT Technology Review in the category Most Thought-Provoking Papers 2018

» Zeppelzauer et al; **Visual Estimation of Building Condition with Patch-level ConvNets**, In Workshop on Multimedia for Real Estate Tech at ACM International Conference on Multimedia Retrieval (ICMR) 2018



» Koch et al; **Real Estate Image Analysis: A Literature Review**. Journal of Real Estate Literature: 2019, Vol. 27, No. 2, pp. 269-300.

» Ziaee et al; **A Novel Adaptive Deep Network for Building Footprint Segmentation**; https://arxiv.org/abs/2103.00286; 2021

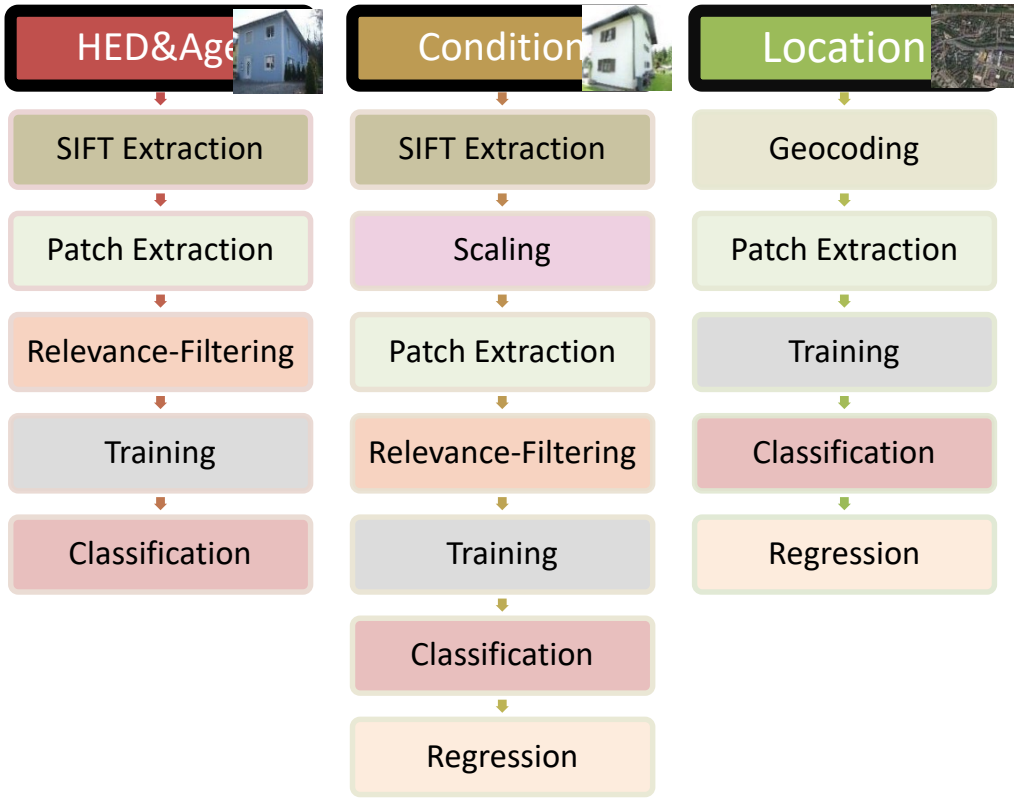
Process chain of approaches

Amount of images for analysis

	TRAINING	VALIDATION	TEST
HED & Age	**2898**	**387**	**580**
Condition	**7367**	**1086**	**2183**
Location	**1400**	**168**	**1714**

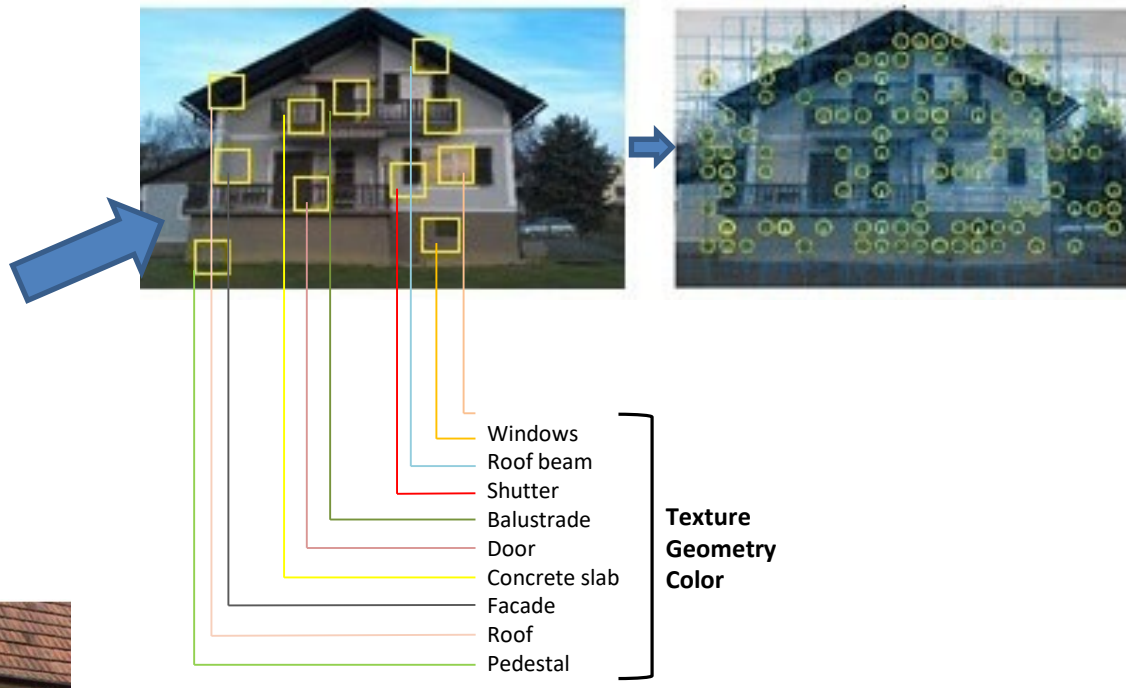
Amount of extracted patches for e.g. HED&Age Analysis:

- **114.561** for Training
- **13.384** for Validation



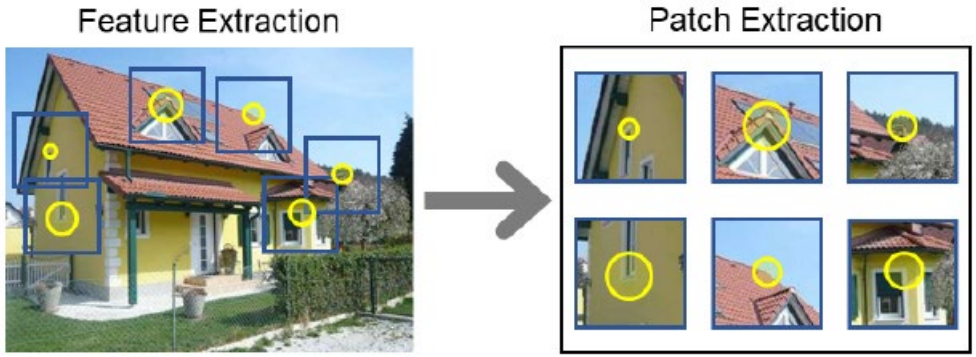
Prediction of HED and Age: Approach Setup

- 2 Observations are the basis for our approach:
  - On a patch wise level the condition of a fascade can be characterized
  - specific constructions demand a specific epoch



Prediction of HED and Age: Approach Setup

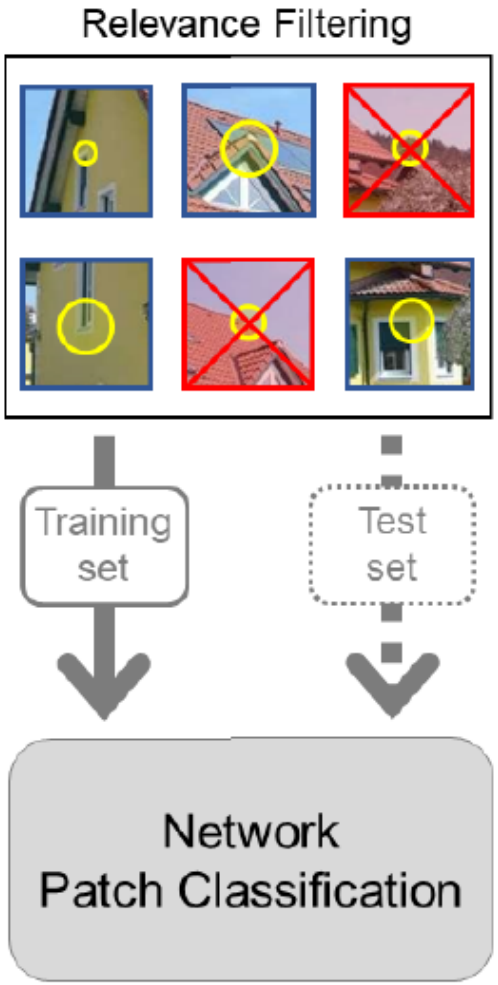
- Apply DSIFT feature extraction algorithm to identify interesting area
- To reduce amount of patches:
  - Apply k-means cluster for all patches of input image
  - Select the patch which is near to the centroid and process further



- Sort the k patches according to its norm and select t (fixed percentage) as representatives
- Filter irrelevant patches by a trained CNN (AlexNet) for non-Building objects (tree, car, etc.)

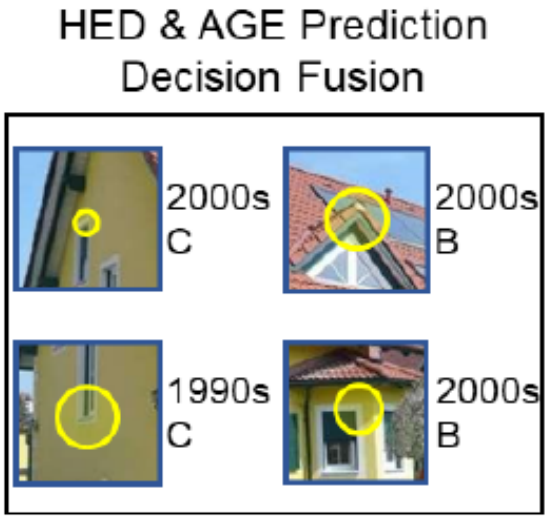
Prediction of HED and Age: Approach Setup

- Relevance Filtering
  - Filter irrelevant patches by a trained CNN (AlexNet) for non-Building objects (tree, car, etc.)
- HED, Age, Condition Prediction
  - Training and prediction bases on a residual network (ResNet-50)
  - Only horizontal flipping as augmentation technique has been used
  - ResNet better speed during training due to less parameters as e.g. AlexNet

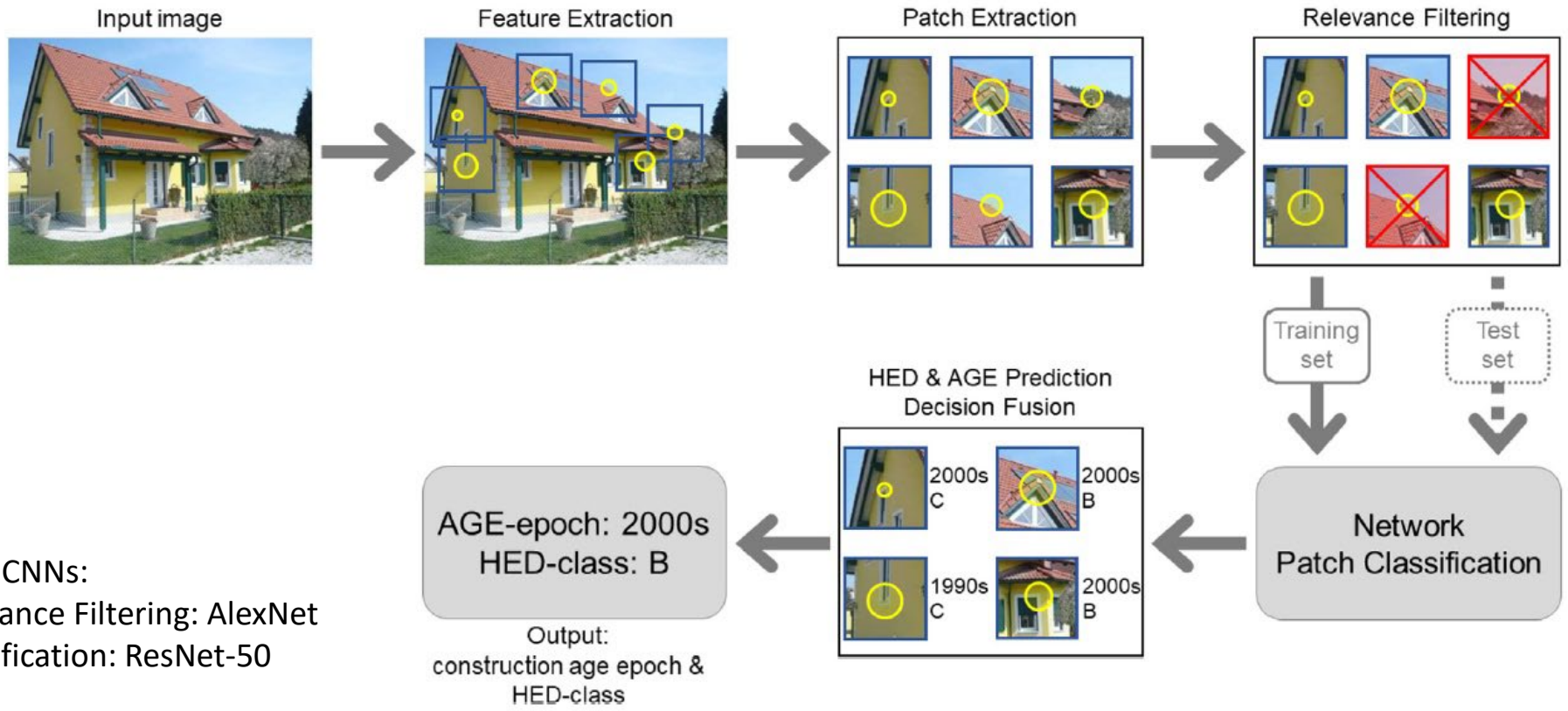


Prediction of HED and Age: Approach Setup

- Prediction Decision Fusion
  - After the prediction every patch of an input image is assigned with a certain HED, Age and Condition class
  - For final prediction two approaches have been evaluated:
    - Majority Voting (MV) on the path wise predictions
      - MV performed better in our tests.
    - averaging the class likelihoods (LH), i.e. the outputs of the softmax layer of the individual patches.

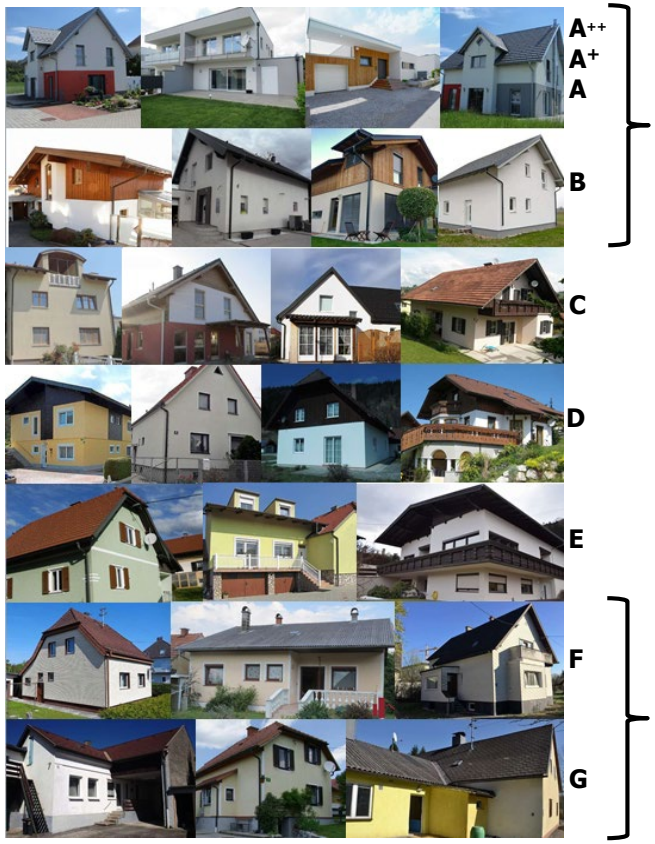


Prediction of HED and Age: Approach Setup



Used CNNs:
Relevance Filtering: AlexNet
Classification: ResNet-50

Prediction of HED and Age: Data set

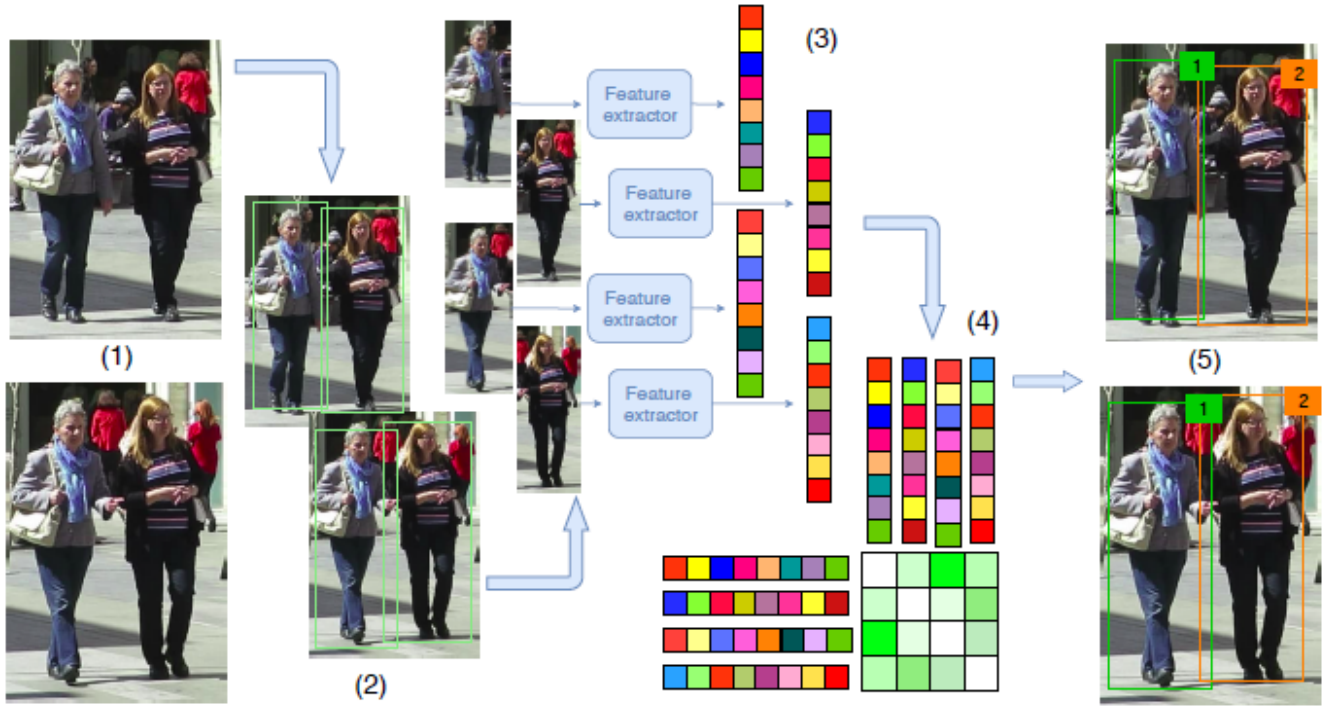


- 3.865 images of 2.065 detached houses in Austria
- Appying of Augmentation techniques (mirroring, etc.)
- For every building the HED category is available (A to G)
- Based on the amount of images and buildings, 5 classes have been created:
    - A (A++, A+, A) and B; C; D; E; F and G
- Three data sets
    - 75% Training
    - 10% Validierung
    - 15% Test

Use Case Multi Object Tracking (MOT)

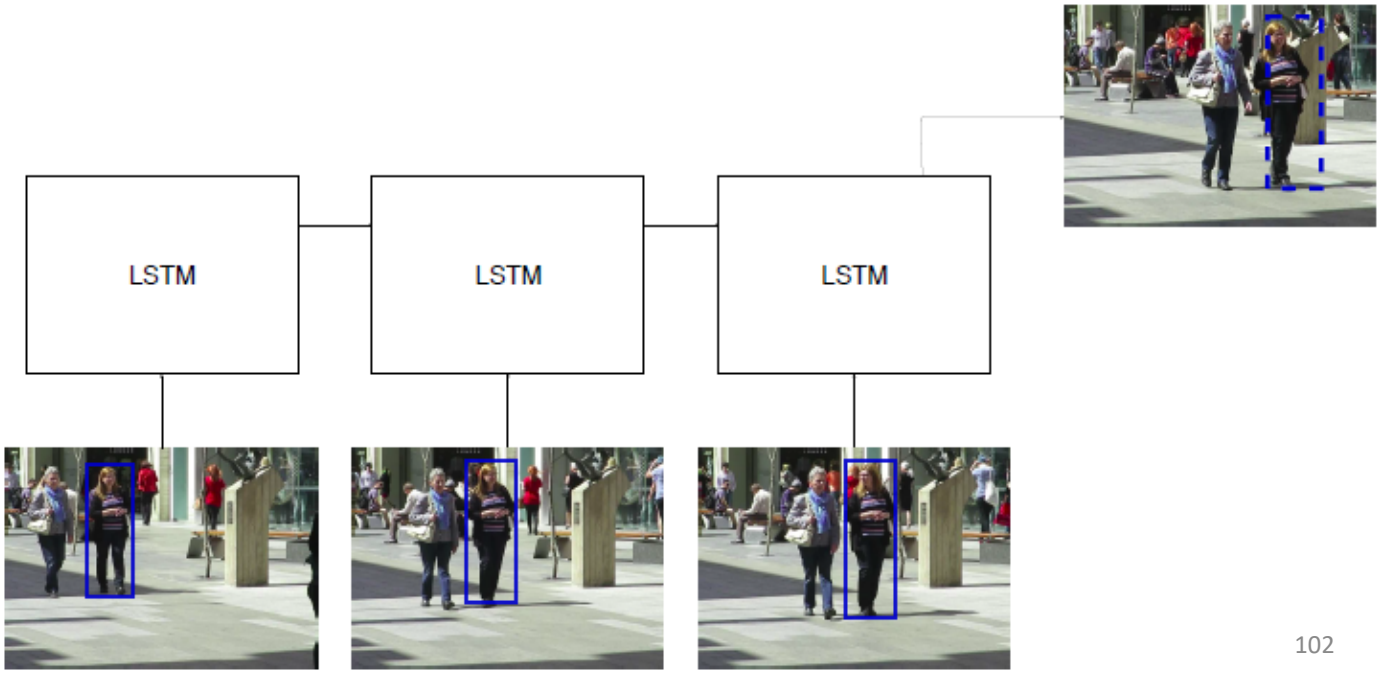
- Classification in batch and online mode [10]; > 50 approaches between 2015-
- Four man steps: (1) Detection, (2) Feature extraction, (3) Affinity Distances, (4) Association
- Datasets: e.g. MOT challenge (https://motchallenge.net/)

Steps of MOT approach [10]



Use Case Multi Object Tracking (MOT)

- Detection step: SSD, YOLOv4/5, etc.
- Feature Extraction: CNNs in various settings (e.g., ResNet-50, etc.)
- Association step: many variants of Recurrent Networks are used: LSTM, etc. or Multilayer Perceptron or Reinforcement Learning



Vision based Railtrack Monitoring

Mirjam Klammsteiner, Mario Döller, Patrik Golec, Michael Kohlegger, Ehtsham Rashid, Stefan Mayr, Vision based stationary railway track monitoring system, In Proceedings of the 33th FRUCT conference, IEEE Explore, Zilina, Slovakia, 2023

Fig. 1: Rail and Obstacle Detection used by [1]

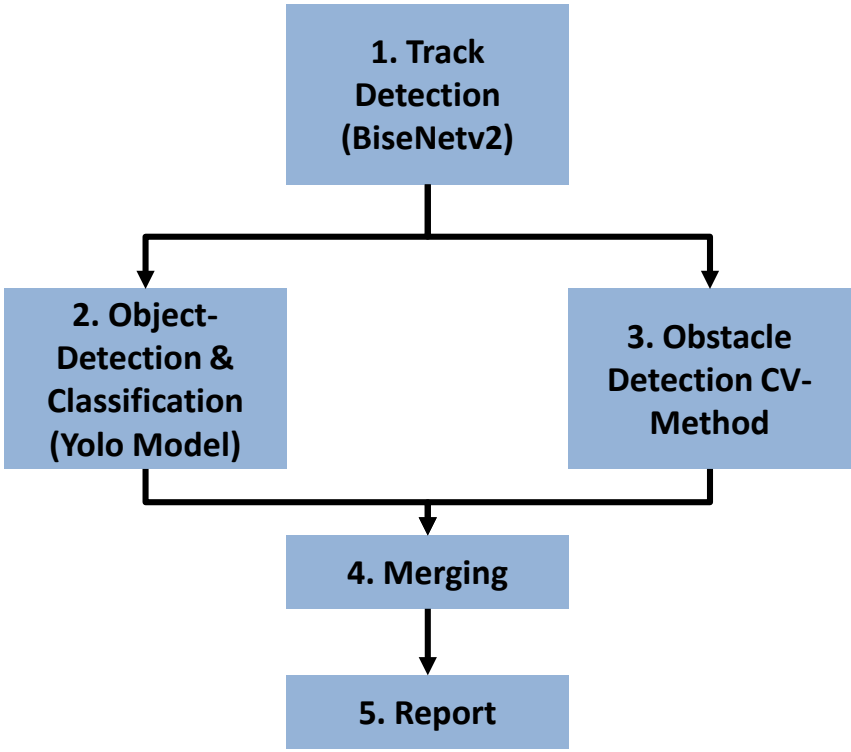
- Traditional Approaches (Computer Vision based)
  - Identifying rail track and its environment (e.g., based on Hough Transform [18])
  - Detecting Obstacles based on Optical Flow approach [20]

- AI based approaches
  - Identifying rail track and its environment (e.g., based on CNN [22]), a well workng approach currently is BiSeNet V2 [21]
  - Obstacle detection focused on AI approaches based on YOLOx version [23] or predefined data sets with obstacles such as trains, people or animals [24]

- Very good recent survey is given in [19], focuses on obstacle detection, rail track detection, distance estimation (evaluated 49 papers) in 2021

Combined Approach: Track detection and Object classification

- Main differences of related work to our contribution:
  - Using YOLO and BiSeNet v2 for rail track detection and well known objects,
  - focusing on specific obstacles (mud, stones, trees, etc.),
  - classifying allowed objects and disallowed objects,
  - Providing a monitoring system for specific dangerous track areas and not a drivers eye system
  - Reference image comparison for obstacle detection including



- Approach split into 5 subtasks

1. Track Detection

- Using BiSeNet V2 [4] to create mask of region of interest

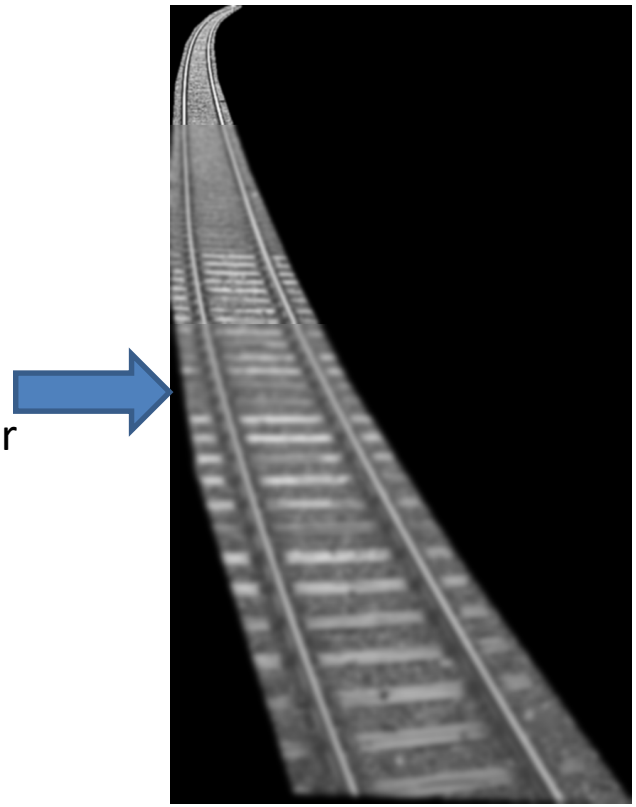
(a) (b) (c)

2. Object Detection & Classification:

- Using YOLOv5 to classify well known objects (people, car, train, etc.)
- Evaluated on RailSem Data set (https://wilddash.cc/railsem19 and Youtube videos)

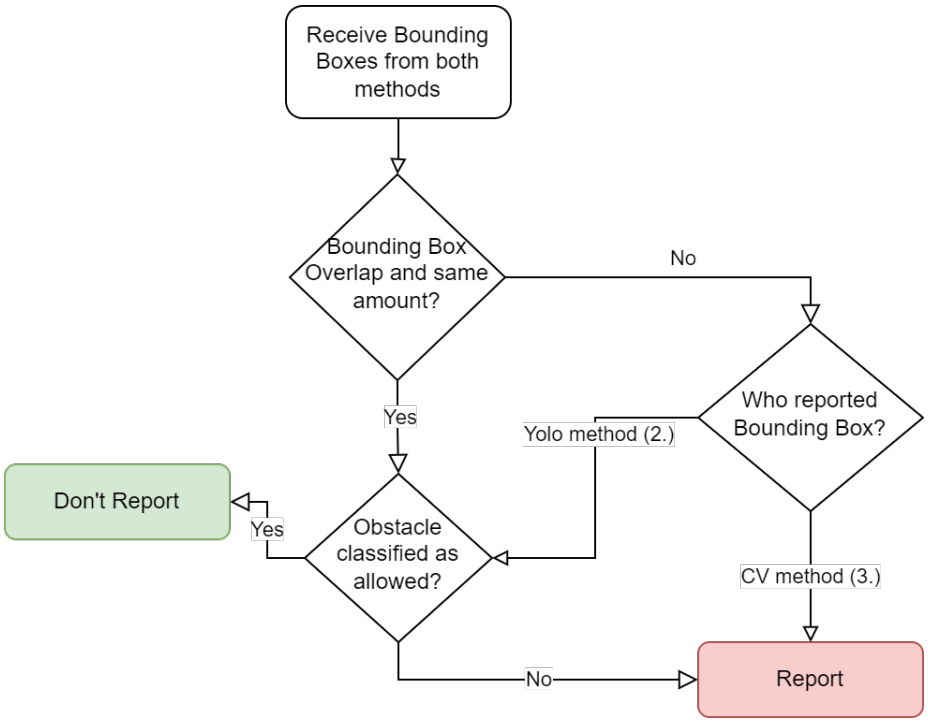
3. Obstacle Detection:

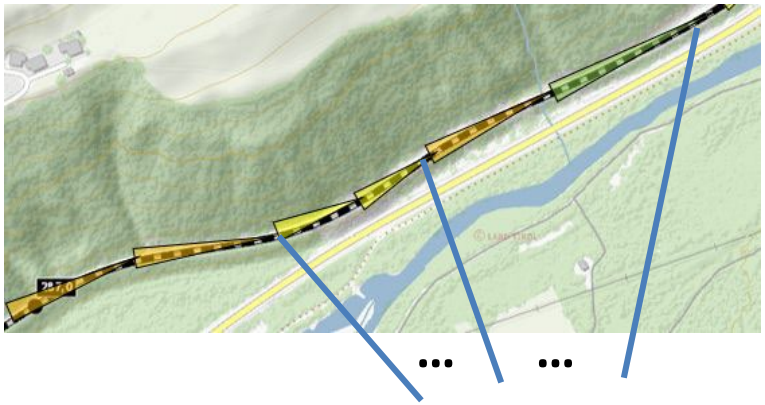
- Comparison of current track image with an empty track image
  - Preprocessing Steps:
    - Greyscaling
    - Adjusting Contrast and Brightness
    - Histogram Matching
  - Noise Removal by applying blur with different kernel size depending on distance from camera (closer parts of the image are blurred with a bigger kernel)
  - Pixel for Pixel Comparison of both images
  - Enlargen possible obstacles by applying dilation operation
  - Calculate area of obstacles → if bigger than threshold draw bounding box



4. Merging

- How to deal with different results from steps 2. and 3.?





Monitoring system
View of individual sensors

5. Report

- Images with bounding boxes highlighting detected obstacles are reported
- Red areas are forwarded to the monitoring rail track system for human in the loop check

Limitations & Outlook

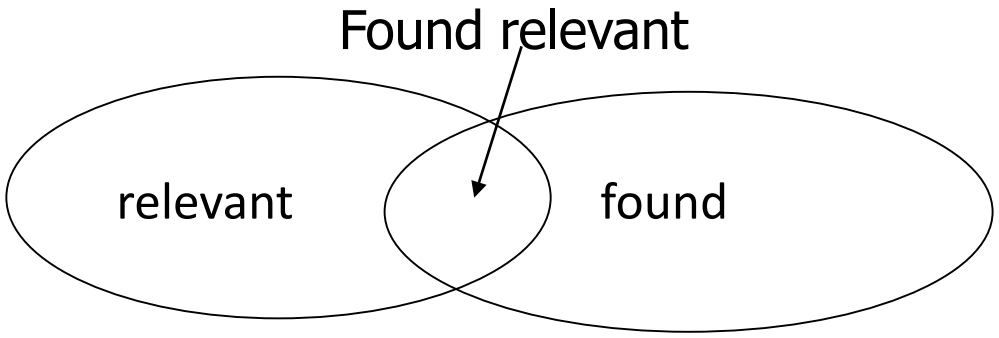
- Limitations
  - Camera Lens Obstruction
  - Shadows being detected as obstacles (see yellow area)
  - Obstacle Detection during the night
- Outlook
  - Synthetic Data Generation for rare situations (mud, stones, trees, etc.)
  - Optimize Shadow-Detection during reference image selection

Table of Contents

Content Based (Image) Retrieval

System Evaluation

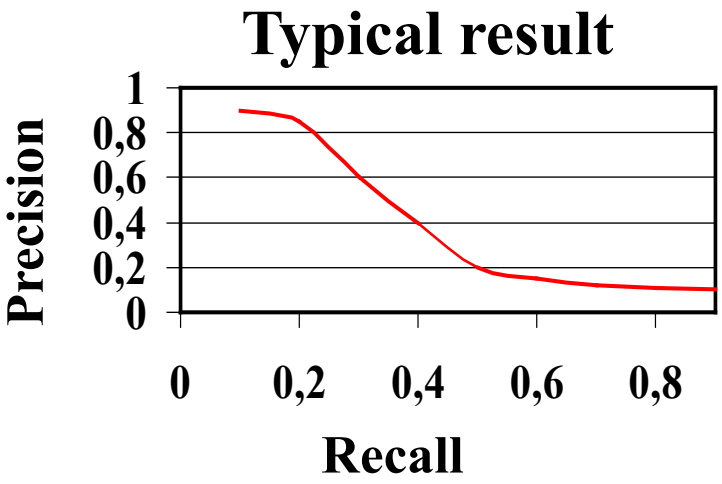
- Efficiency: time, space
- Effectiveness:
  - How well does the system retrieve relevant documents?
  - Is a system better than another?
- Most commonly used pair of metric:
  - Precision = number of found relevant docs / number of found docs
  - Recall = number of found relevant docs / number of relevant docs

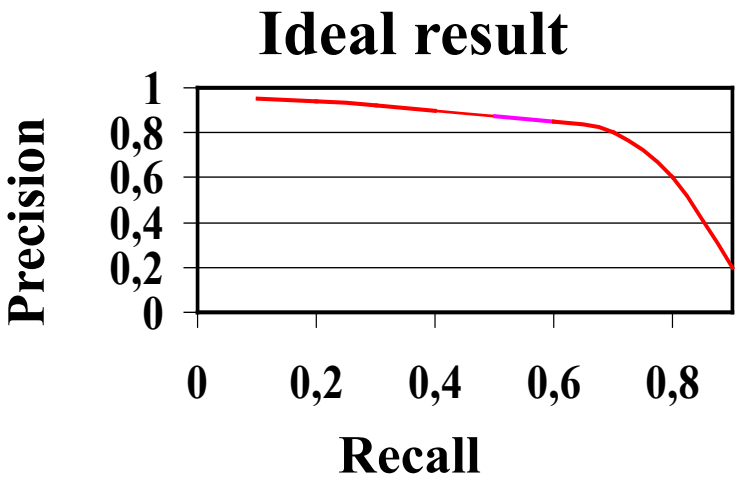


System Evaluation

General Shape of Precision/Recall

- Precision and recall are not independent
- Systems cannot be compared at a given precision/recall point





System Evaluation

MAP (Mean Average Precision)

- Take the rank position into account
- Formal:
  - r_{ij} = Rank of j-th relevant document for Q_i
  - $|R_i|$ = number rel. Doc. for Q_i
  - n = number of test-queries

$$MAP = \frac{1}{n} \sum_{Q_i} \frac{1}{|R_i|} \sum_{D_j \in R_i} \frac{j}{r_{ij}}$$

- Example:

1 **5** **10**	**4** **8**

1st rel. Doc.
2nd rel. Doc.
3rd rel. Doc.

$$MAP = \frac{1}{2}\left[\frac{1}{3}\left(\frac{1}{1} + \frac{2}{5} + \frac{3}{10}\right) + \frac{1}{2}\left(\frac{1}{4} + \frac{2}{8}\right)\right]$$

System Evaluation

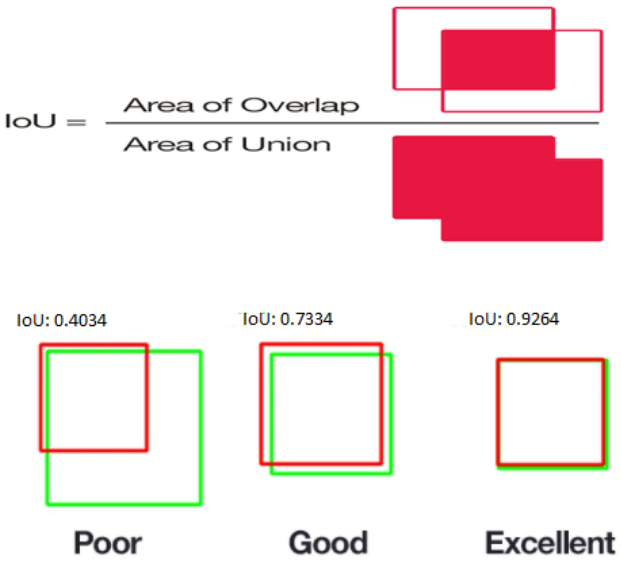
Other metric

- Noise = found irrelevant docs / found docs
- Silence = not found relevant docs / relevant docs
  - Noise = 1 – Precision; Silence = 1 – Recall
- Fallout = found irrelevant docs / irrelevant docs
- Single value metric:
  - F-measure = 2 P * R / (P + R)
  - Average Precision = Average at 11 recall points
  - Precision at n-th document (often used for Web IR)
  - Expected search duration (number of irrelevant documents to be considered in the result list until n relevant documents are identified)

System Evaluation

Object Detection Metrice:

- IuO (Intersection over Union) [14]: is also referred to as Jaccard index, it is used to measure the ratio between common of two areas and the total of two areas



System Evaluation

Test Corpus

- Comparison of different IR systems with the same test corpus
- A test corpus consists of:
  - multiple documents
  - multiple queries
  - Evaluation of relevance for each document-query pair (i.e. expected results for each query)
- The performance of a system is evaluated by comparing its output with the expected results.

Known Benchmarking Events:

- TRECVID: http://www-nlpir.nist.gov/projects/trecvid/
- MediaEval: http://www.multimediaeval.org/

System Evaluation

Test Corpus - TRECVID

- Text REtrieval Conference (TREC)
  - Organized by National Institute of Standards (NIST)
  - Support from interested government agencies
  - Annual evaluation (NOT a competition)
  - Different “tracks” over the years, e.g.
    - web retrieval, email spam filtering, etc., video (standalone conference from 2001)

- TREC Video Retrieval Evaluation (TRECVID)
  - Promote progress in content-based analysis and retrieval from digital videos

System Evaluation

Test Corpus - TRECVID

- Evaluation is driven by participants
- The collection is fixed, available in the spring
  - 50% data used for development, 50% for testing
- Test queries available in July, 1 month to submission
- More details:
  - http://www-nlpir.nist.gov/projects/trecvid

References

- *Srivastava, Nitish, et al. "**Dropout: a simple way to prevent neural networks from overfitting.**" Journal of machine learning research (2014)*
- *Bergstra, James, and Yoshua Bengio. "**Random search for hyper-parameter optimization.**" Journal of Machine Learning Research, Feb (2012)*
- *Kim, Y. "**Convolutional Neural Networks for Sentence Classification**", EMNLP (2014)*
- Severyn, Aliaksei, and Alessandro Moschitti. "**UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification.**" *SemEval@ NAACL-HLT* (2015)
- *Cho, Kyunghyun, et al. "**Learning phrase representations using RNN encoder-decoder for statistical machine translation**." EMNLP (2014)*
- Ilya Sutskever et al. "**Sequence to sequence learning with neural networks.**" *NIPS (*2014)
- *Bahdanau et al. "**Neural machine translation by jointly learning to align and translate.**" ICLR (2015)*
- *Gal, Y., Islam, R., Ghahramani, Z. "**Deep Bayesian Active Learning with Image Data**." ICML (2017)*
- *Nair, V., Hinton, G.E. "**Rectified linear units improve restricted boltzmann machines**." ICML (2010)*
- Ronan Collobert, et al. "**Natural language processing (almost) from scratch**." *JMLR* (2011)

- *Kumar, Shantanu. "A Survey of Deep Learning Methods for Relation Extraction." arXiv preprint arXiv:1705.03645 (2017)*
- *Lin et al. "Neural Relation Extraction with Selective Attention over Instances" ACL (2016) [code]*
- *Zeng, D.et al. "Relation classification via convolutional deep neural network". COLING (2014)*
- *Nguyen, T.H., Grishman, R. "Relation extraction: Perspective from CNNs." VS@ HLT-NAACL. (2015)*
- *Zhang, D., Wang, D. "Relation classification via recurrent NN." -arXiv preprint arXiv:1508.01006 (2015)*
- *Zhou, P. et al. "Attention-based bidirectional LSTM networks for relation classification . ACL (2016)*
- Mike Mintz et al. "Distant supervision for relation extraction without labeled data." *ACL- IJCNLP (2009)*

References & Resources

- http://web.stanford.edu/class/cs224n
- https://www.coursera.org/specializations/deep-learning
- https://chrisalbon.com/#Deep-Learning
- http://www.asimovinstitute.org/neural-network-zoo
- *http://cs231n.github.io/optimization-2*
- *https://medium.com/@ramrajchandradevan/the-evolution-of-gradient-descend-optimization-algorithm-4106a6702d39*
- https://arimo.com/data-science/2016/bayesian-optimization-hyperparameter-tuning
- http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow
- http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp
- https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8
- *http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/*
- *http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/*
- *http://colah.github.io/posts/2015-08-Understanding-LSTMs*
- https://github.com/hyperopt/hyperopt
- https://github.com/tensorflow/nmt

References

[1] A. Geiger, P. Lenz, C. Stiller, R. Urtasun; Vision meets robotics: The KITTI dataset; In The International Journal of Robotics Research; 2013; https://doi.org/10.1177/0278364913491297
[2] Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, Mingkui Tan; Multi-marginal Wasserstein GAN; In Proceedings of Advances in Neural Information Processing Systems 32 (NeurIPS 2019)
[3] Yunpeng Zhang, Jiwen Lu, Jie Zhou; Objects Are Different: Flexible Monocular 3D Object Detection; In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 3289-3298
[4] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, Yi Yang; Random Erasing Data Augmentation; In Proceedings of the AAAI Conference on Artificial Intelligence; Vol. 34 No. 07; ISSN 2374-3468; 2021
[5] Qinghe Zheng, Mingqiang Yang, Jiajie Yang, Qingrui Zhang, Xinxin Zhang; Improvement of Generalization Ability of Deep CNN via Implicit Regularization in Two-Stage Training Process; In IEEE Access (Volume: 6); pp:15844 – 15869; 2018; DOI: 10.1109/ACCESS.2018.2810849
[6] Binod Bhattarai, Seungryul Baek, Rumeysa Bodur, Tae-Kyun Kim; Sampling Strategies for GAN Synthetic Data; In Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2020; DOI: 10.1109/ICASSP40776.2020.9054677

References

[7] Vinay Kukreja, Deepak Kumar, Amandeep Kaur, Geetanjali, Sakshi; GAN-based synthetic data augmentation for increased CNN performance in Vehicle Number Plate Recognition; In Proceedings of the4th International Conference on Electronics, Communication and Aerospace Technology (ICECA) ; 2020; DOI: 10.1109/ICECA49313.2020.9297625

[8] João Paulo Lima, Rafael Roberto, Lucas Figueiredo, Francisco Simões, Veronica Teichrieb; Generalisable Multi-Camera 3D Pedestrian Detection. In Proceeding of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop. 2021.

[9] Vladislav Sovrasov and Dmitry Sidnev. Building computationally efficient and well-generalizing person re-identification models with metric learning. arXiv:2003.07618, 2020.

[10] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, Francisco Herrera; DEEP LEARNING IN VIDEO MULTI-OBJECT TRACKING: A SURVEY; In arXiv:1907.12740v4; 2019

[11] Ramão Tiburski, Carlos Roberto Moratelli, Sergio Johann Filho, Marcelo Veiga Neves, Everton de Matos, Leonardo Albernaz Amaral, F. P. Hessel; Lightweight Security Architecture Based on Embedded Virtualization and Trust Mechanisms for IoT Edge Devices; IEEE Communications Magazine 57(2):67 – 73; 2019; DOI:10.1109/MCOM.2018.1701047

References

[12] Youzi Xiao et al.; A review of object detection based on deep learning.; In Multimedia Tools and Applications 79(33): 23729–23791; 2020.
[13] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao; YOLOv4: Optimal Speed and Accuracy of Object Detection; arXiv:2004.10934; 2020.
[14] Rafael Padilla et al. “A comparative analysis of object detection metrics with a companion open-source toolkit.” In: Electronics 10.3 (2021), p. 279
[15] Golnaz Ghiasi et al. “Simple copy-paste is a strong data augmentation method for instance segmentation.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 2918–2928
[16] Jamil Ahmad, Khan Muhammad, and Sung Wook Baik. “Data augmentation assisted deep learning of hand-drawn partially colored sketches for visual search.” In: *PloS one* 12.8 (2017), e0183838
[17] Ángela Casado-García et al. “CLoDSA: a tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks.” In: BMC bioinformatics 20.1 (2019), pp. 1–14

References

[18] L. A. Fonseca Rodriguez, J. A. Uribe, and J. F. Vargas Bonilla, “Obstacle detection over rails using hough transform,” in Proceedings of the XVII Symposium of Image, Signal Processing, and Artificial Vision (STSIVA), 2012, p. 317–322.

[19] D. Ristic-Durrant, M. Franke, and K. Michels, “A review of vision based on-board obstacle detection and distance estimation in railways,” vol. 21, no. 10, 2021

[20] J. A. Uribe, L. Fonseca, and J. F. Vargas, “Video-based system for railroad collision warning,” in Proceedings of the IEEE International Carnahan Conference on Security Technology (ICCST), 2012, p. 280–285.

[21] D. Ristic-Durrant, M. A. Haseeb, M. Franke, M. Banic, M. Simonovic,´ and D. Stamenkovic,´ Artificial Intelligence for Obstacle Detection in Railways: Project SMART and Beyond. Springer, 2020.

[22] Z. Wang, X. Wu, G. Yu, and M. Li, “Efficient rail area detection using convolutional neural network,” IEEE Access, vol. 6, p. 77656–77664, 2018.

[23] D. Ristic-Durrant, M. A. Haseeb, M. Franke, M. Bani´c, M. Simonovi´c, and D. Stamenkovi´c, Artificial Intelligence for Obstacle Detection in Railways: Project SMART and Beyond. Springer, 2020, iSBN: 978-3-030-58462-7.

[24] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in CoRR abs/1506.01497, 2015.

The end