INFO0948
Feature Extraction

Renaud Detry

University of Liège, Belgium

INFO0948
Feature Extraction

Raphaël Marée
http://www.montefiore.ulg.ac.be/~maree/
http://www.cytomine.org

University of Liège, Belgium

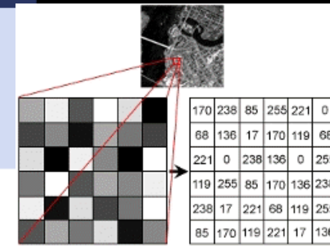
These slides are based on Chapter 13 of the book *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* by Peter Corke, published by Springer in 2011.

The Bag-of-feature section is based on a presentation by Cordelia Schmid `http://www.di.ens.fr/willow/events/cvml2011/materials/CVML2011_Cordelia_bof.pdf`

+ More recent topics:
  - End-to-end learning with tree-based methods and deep learning
  - The need for careful data collection for effective computer vision
  - Intelligent robotics in microscopy/biomedecine

Related topics:
Marc Van Droogenbroeck's "Computer Vision"
and
Louis Wehenkel/Pierre Geurt's "Introduction to Machine Learning"

Motivation



Raw images contain too much data to be of direct practical use for high(er)-level robot vision (object recognition, pose estimation, tracking, ...).

We need to reduce the dimensionality of raw image data, ideally focusing on

- discarding redundant information.
- extracting entities that are invariant to the conditions that typically change while a robot is working (viewpoint, illumination, ...).

Feature extraction is an information concentration step that reduces the data rate from 10^6–10^8 bytes s^{-1} at the output of a camera to something of the order of tens of features (vectors of a few dozen scalars) per frame that can be used as input to a robot's control system.

1. Feature extraction

2. Object Recognition / Image classification
    Challenges
    Bag-of-features
    End-to-end learning
    Dataset quality control

3. Intelligent robotics / AI in Biomedecine

Classification of features

edge

corner

region

Edge refers to pixel at which the image intensities change abruptly. Image pixels are discontinuous at different sides of edges.

Corner refers to the point at which two different edge directions occur in the local neighborhood. It is the intersection of two connected contour lines.

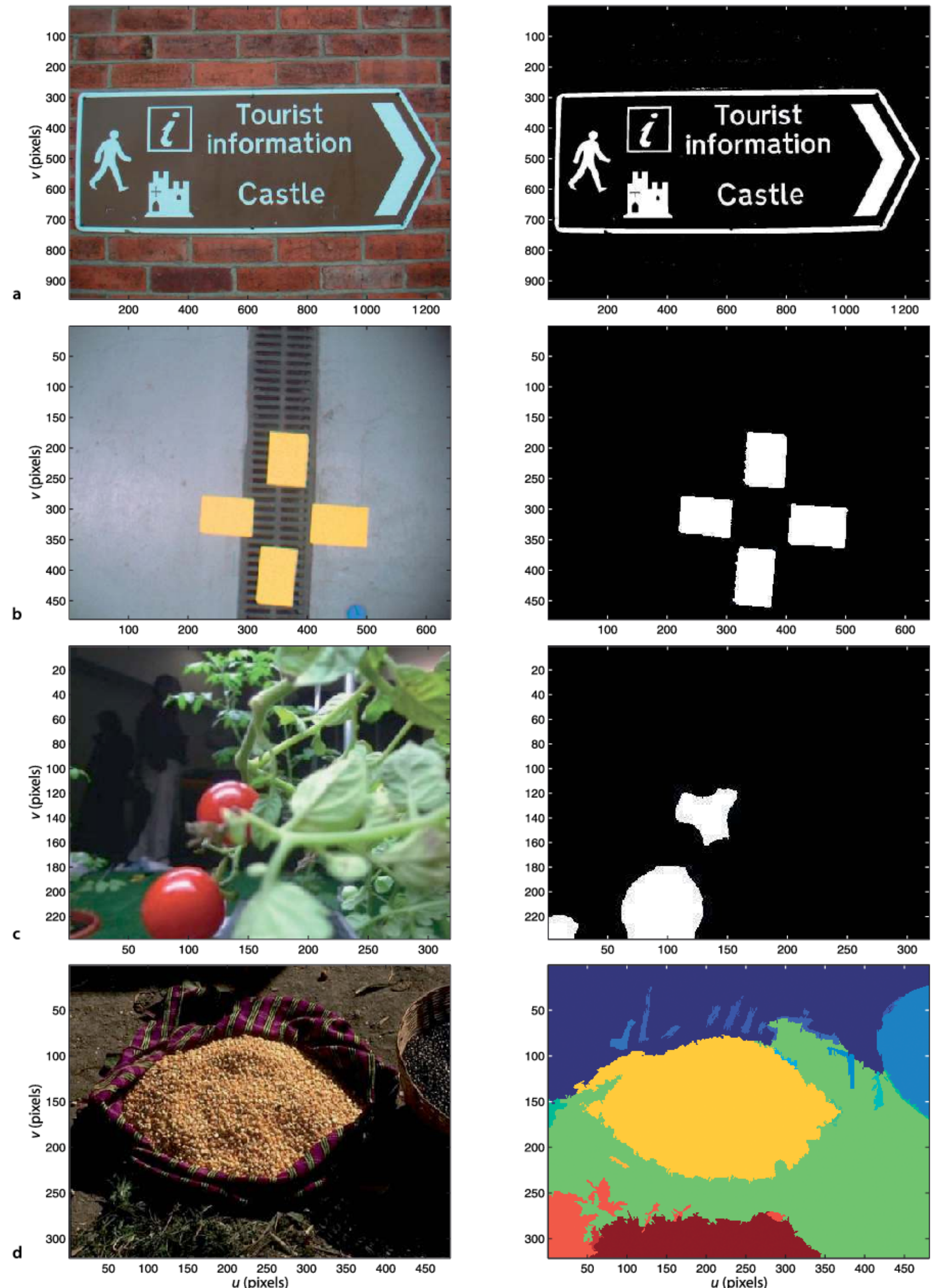
Region refers to a closed set of connected points. Nearby and similar pixels are grouped together to compose the interest region

Others (random, landmarks,...)

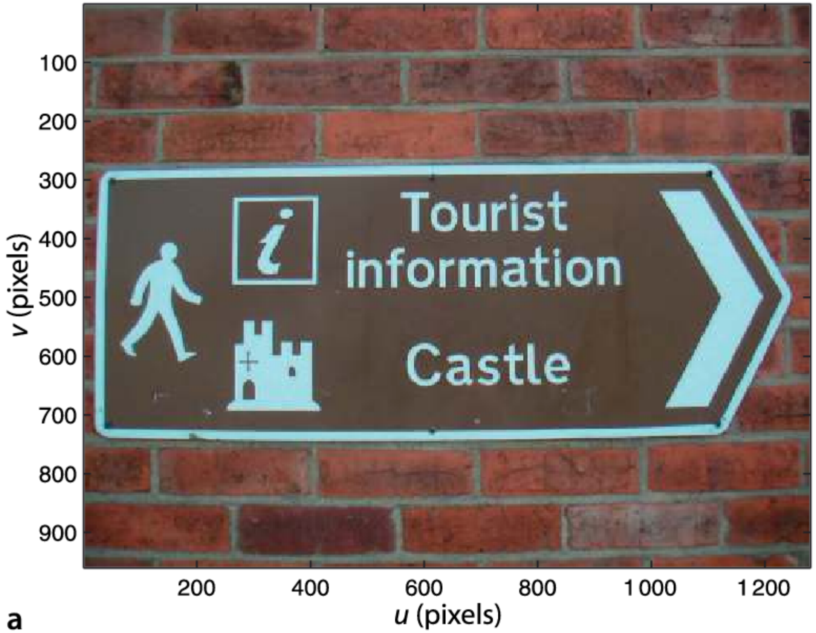
(Y. Li et al., Neurocomputing, 2015)

Region Features

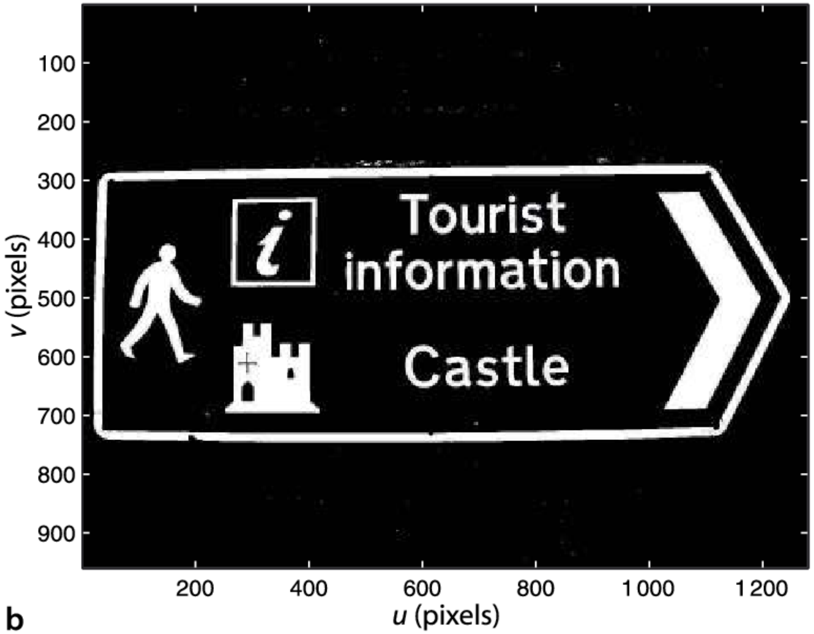
Refer to a closed set of connected points with a similar homogeneity criteria, usually the intensity value



Thresholding

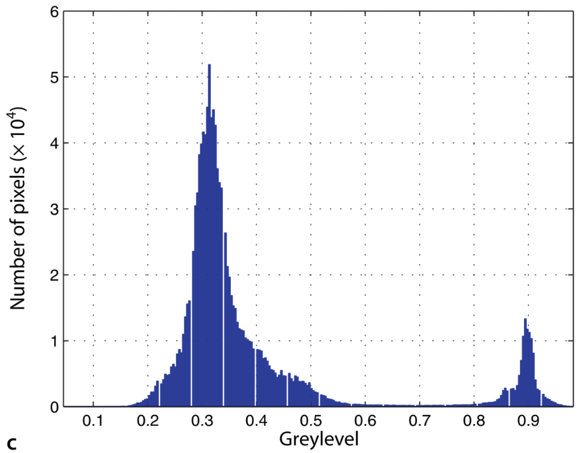


a

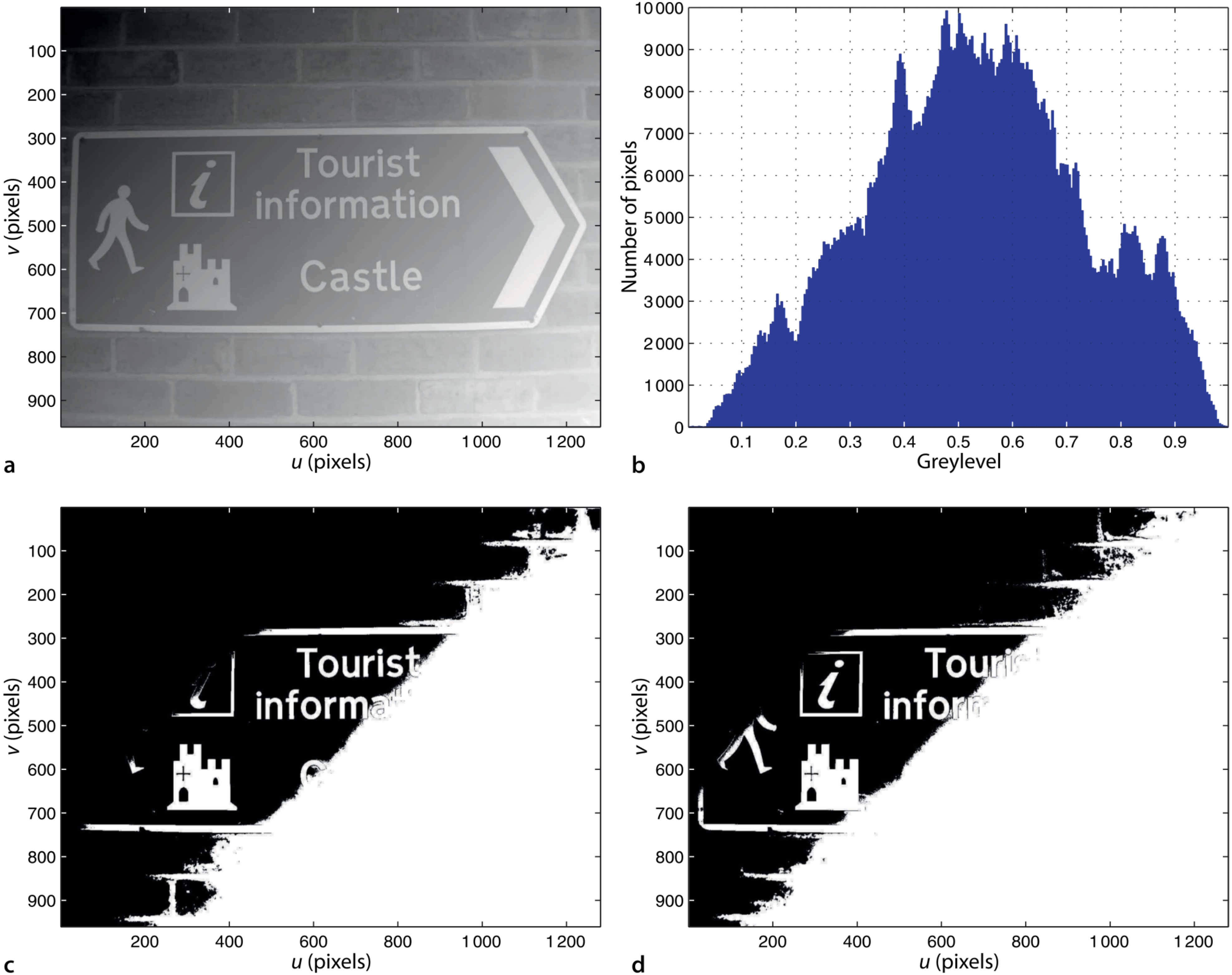


b

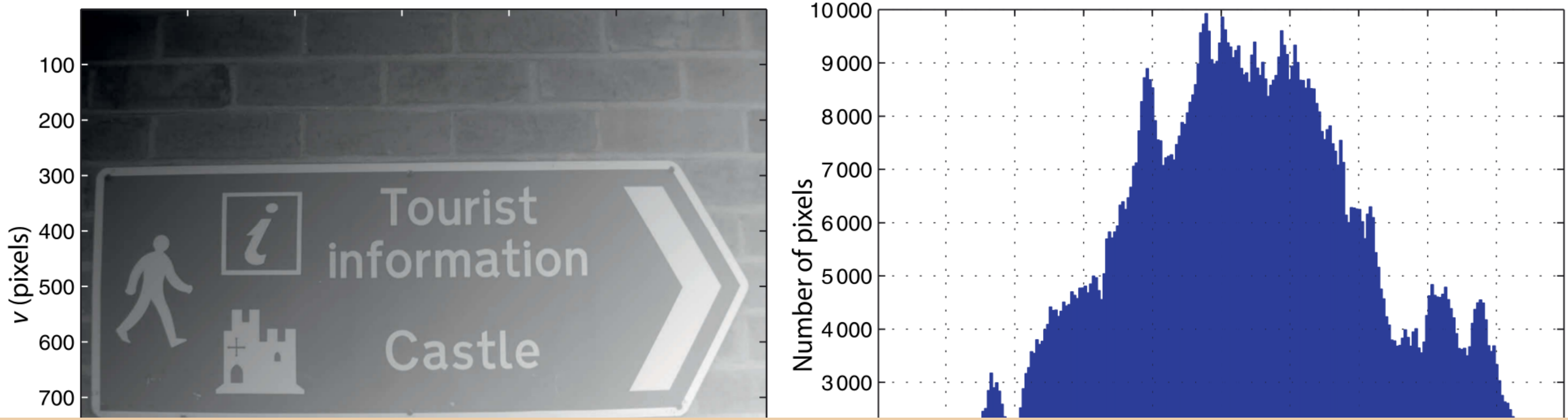
$$c[u,v] = \begin{cases} 0 & I[u,v] < t \\ 1 & I[u,v] \geq t \end{cases} \quad \forall (u,v) \in I$$



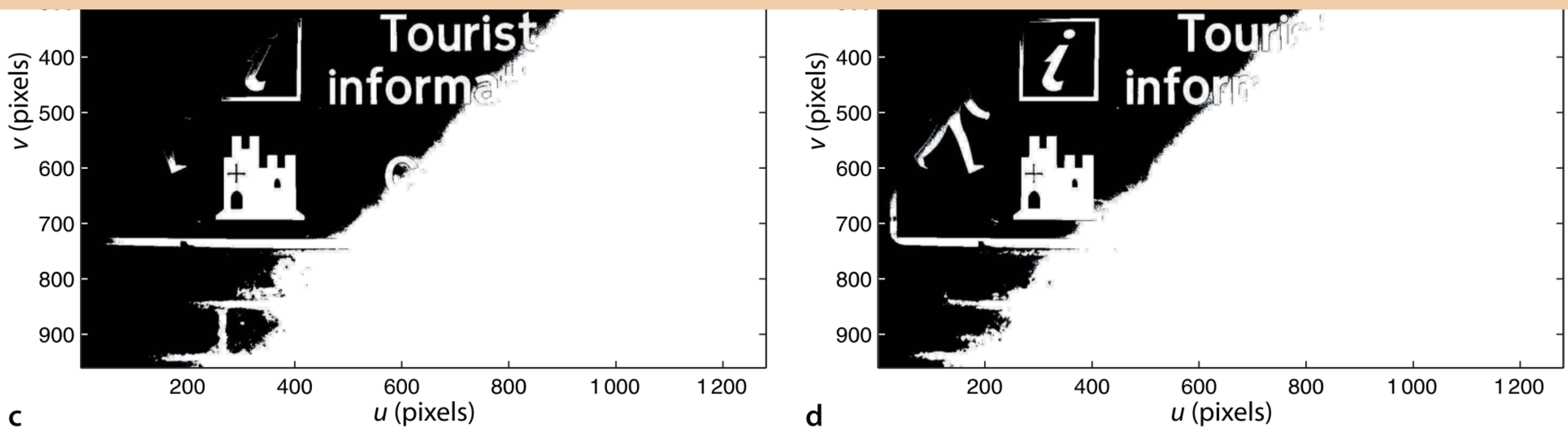
c



a
v (pixels)
u (pixels)
Tourist
information
Castle
i
b
Number of pixels
Greylevel
c
v (pixels)
u (pixels)
Tourist
informa
d
v (pixels)
u (pixels)
Touris
infor



Thresholding-based techniques are notoriously brittle – a slight change in illumination of the scene means that the thresholds we chose would no longer be appropriate. In most real scenes there is no simple mapping from pixel values to particular objects – we cannot for example choose a threshold that would select a motorbike or a duck. Distinguishing an object from the background remains a hard computer vision problem.



Thresholding-based techniques are notoriously brittle – a slight change in illumination of the scene means that the thresholds we chose would no longer be appropriate. In most real scenes there is no simple mapping from pixel values to particular objects – we cannot for example choose a threshold that would select a motorbike or a duck. Distinguishing an object from the background remains a hard computer vision problem.

Many thresholding alternatives (see Sezgin, J. Electronic Imaging 2004):

- Histogram shape-based
  - Convex-hull, peak-and-valley, ...
- Clustering-based
  - Iterative (K-means), Minimum error, …
- Entropy-based
  - e.g. maximize entropy of the thresholded image, minimize the cross-entropy between input & output
- Spatial, locally adaptive thresholding
  - Local variance/contrast, ...

Spatial, locally adaptive thresholding

- A threshold is calculated at each pixel, which depends on some local statistics :

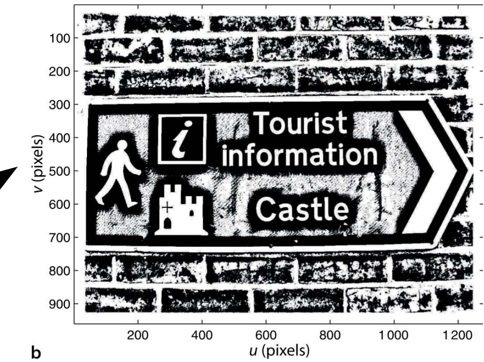


Table 6 Thresholding functions for locally adaptive methods.

variance	Local_Niblack[110]	$T(i,j)=m(i,j)+k.\sigma(i,j)$ where $k=-0.2$ and local window size is $b=15$
variance	Local_Sauvola[111]	$T(i,j)=m(i,j)+\left\{1+k.\left[\frac{\sigma(i,j)}{R}-1\right]\right\}$ where $k=0.5$ and $R=128$
contrast	Local_White[112]	$B(i,j)=\begin{cases}1 & \text{if } m_{w\times w}(i,j)<I(i,j)*\text{bias}\\ 0 & \text{otherwise}\end{cases}$ where $m_{w\times w}(i,j)$ is the local mean over a $w=15$-sized window and bias$=2$.
contrast	Local_Bernsen[113]	$T(i,j)=0.5\{\max_w[I(i+m,j+n)]+\min_w[I(i+m,j+n)]\}$ where $w=31$, provided contrast $C(i,j)=I_{\text{high}}(i,j)-I_{\text{low}}(i,j)\geqslant 15$.
Center-surround	Local_Palumbo[21]	$B(i,j)=1$ if $I(i,j)\leqslant T_1$ or $m_{\text{neigh}}T_3+T_5>m_{\text{center}}T_4$ where $T_1=20$, $T_2=20$, $T_3=0.85$, $T_4=1.0$, $T_5=0$, neighborhood size is 3×3.
Surface-fitting	Local_Yanowitz[115]	$\lim_{n\to\infty} T_n(i,j)=T_{n-1}(i,j)+R_n(i,j)/4$ where $R_n(i,j)$ is the thinned Laplacian of the image.
Center-surround	Local_Kamel[1]	$B(i,j)=1$ if $\{[L(i+b,j)\wedge L(i-b,j)]\vee[L(i,j+b)\wedge L(i,j-b)]\}$ $\{[L(i+b,j+b)\wedge L(i-b,j-b)]\vee[L(i+b,j-b)\wedge L(i-b,j+b)]\}$ where $L(i,j)=\begin{cases}1 & \text{if } [m_{w\times w}(i,j)-I(i,j)]\geqslant T_0\\ 0 & \text{otherwise}\end{cases}$, $w=17$, $T_0=40$
	Local_Oh[13]	Define the optimal threshold value (T_{opt}) by using a global thresholding method, such as the Kapur[53] method, then locally fine tune the pixels between $[T_0-T_1]$ considering local covariance ($T_0<T_{\text{opt}}<T_1$).
contrast	Local_Yasuda[114]	$B(i,j)=1$ if $m_{w\times w}(i,j)<T_3$ or $\sigma_{w\times w}(i,j)>T_4$ where $w=3$, $T_1=50$, $b=16$, $T_2=16$, T_3 128, $T_4=35$

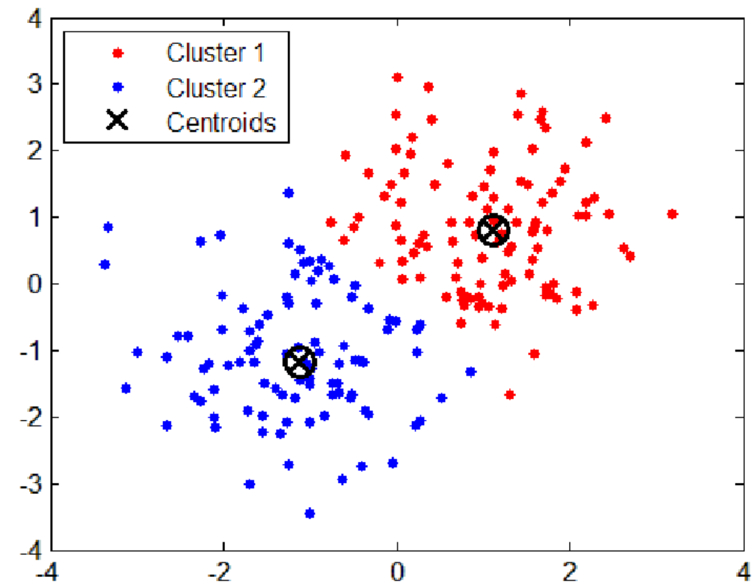
(Sezgin, J. Electronic Imaging 2004)

The k-means Algorithm (see “Introduction to machine learning” course)

Given a set of observations $(x_1, x_2, ..., x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets $(k \leq n)$ $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2$$

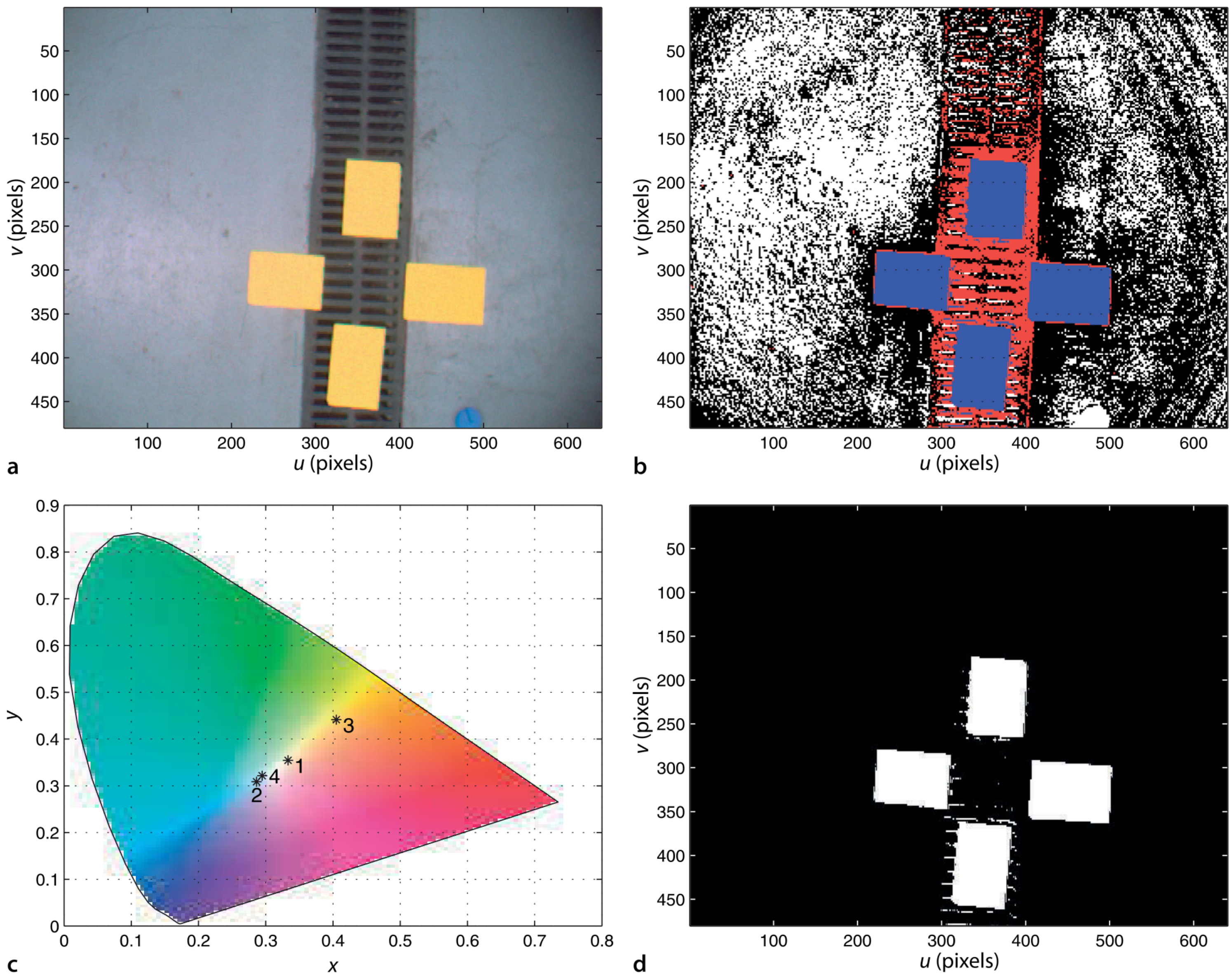
where μ_i is the mean of points in S_i.



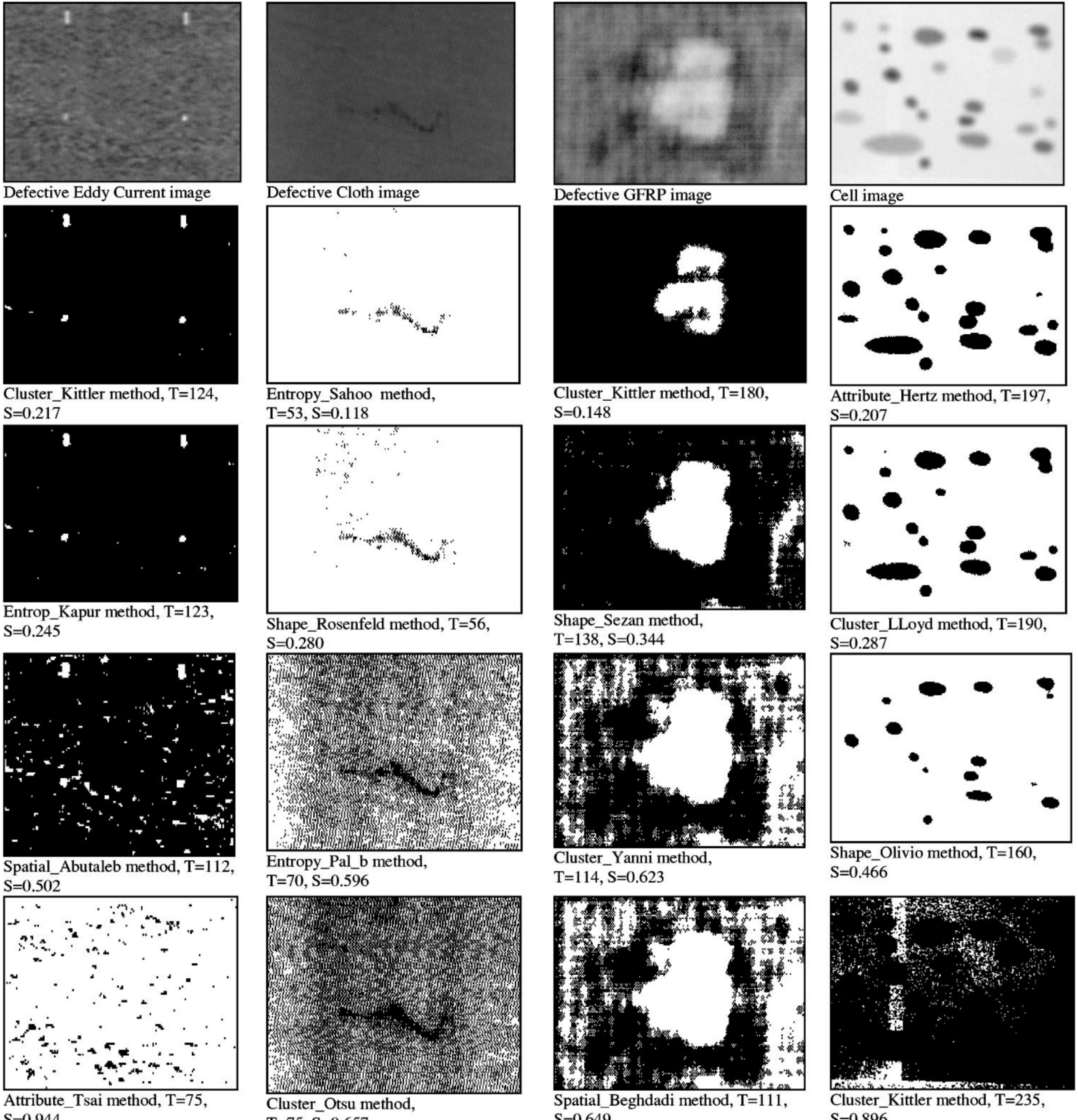
In each round, pixels are partitioned by identifying the best matching cluster, based on Euclidean distance along color dimensions (e.g. R,G,B). Centroids are then updated by re-computing cluster averages.

Color Clustering and Classification

```
>> [cls, cxy, resid] = colorkmeans(im_targets, 4);
```



Each pixel color value is assigned based on its corresponding centroid color value.



Defective Eddy Current image
Defective Cloth image
Defective GFRP image
Cell image

Cluster_Kittler method, T=124, S=0.217
Entropy_Sahoo method, T=53, S=0.118
Cluster_Kittler method, T=180, S=0.148
Attribute_Hertz method, T=197, S=0.207

Entrop_Kapur method, T=123, S=0.245
Shape_Rosenfeld method, T=56, S=0.280
Shape_Sezan method, T=138, S=0.344
Cluster_LLoyd method, T=190, S=0.287

Spatial_Abutaleb method, T=112, S=0.502
Entropy_Pal_b method, T=70, S=0.596
Cluster_Yanni method, T=114, S=0.623
Shape_Olivio method, T=160, S=0.466

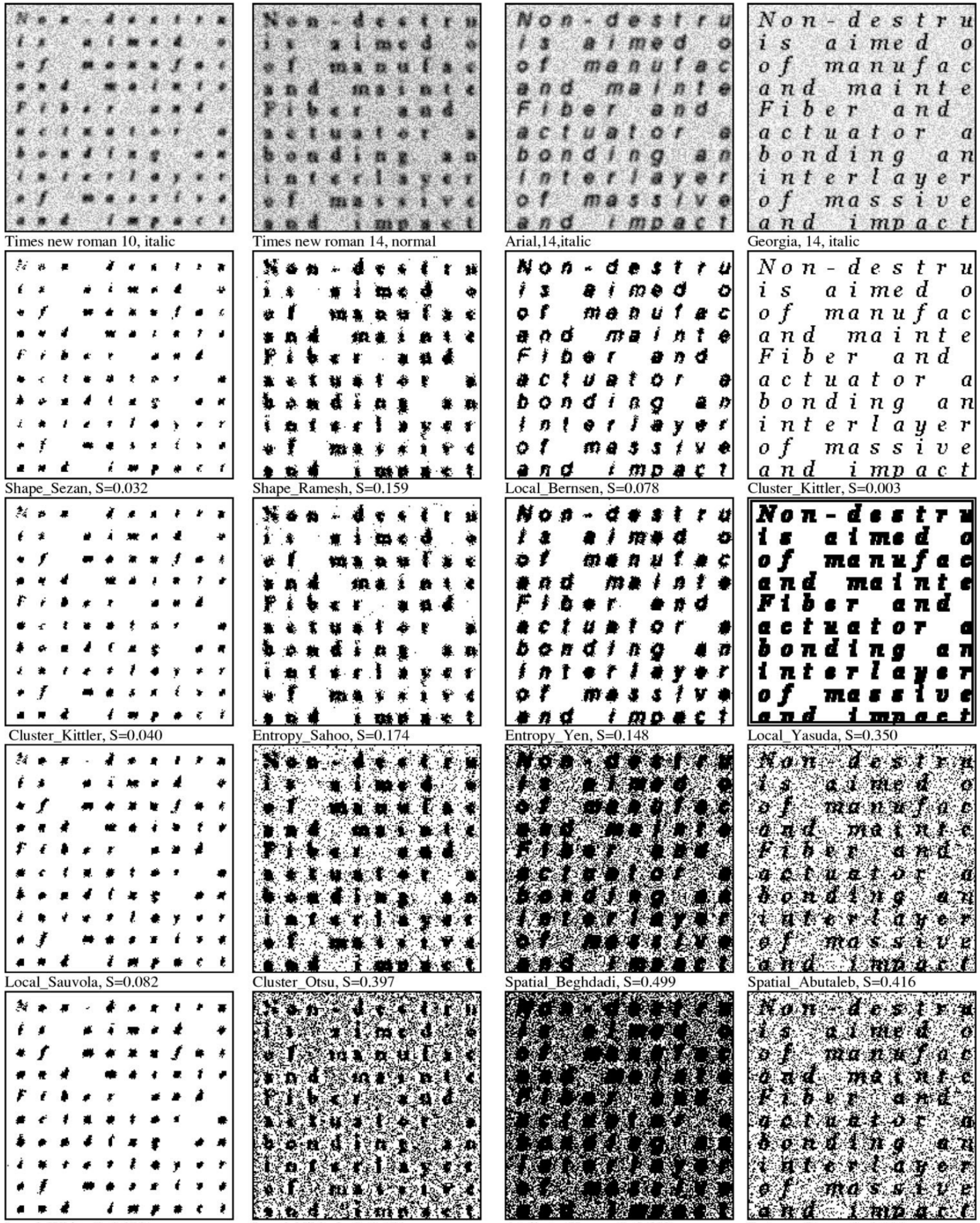
Attribute_Tsai method, T=75, S=0.944
Cluster_Otsu method, T=75, S=0.657
Spatial_Beghdadi method, T=111, S=0.649
Cluster_Kittler method, T=235, S=0.896

(Sezgin, J. Electronic Imaging 2004)

S is the arithmetic averaging of:

- ME (misclassification error)
- EMM (edge mismatch)
- NU (region non uniformity)
- RAE (relative foreground area error)
- NMHD (Hausdorff distance)

Other metrics, see M. Van Droogenbroeck , 2005



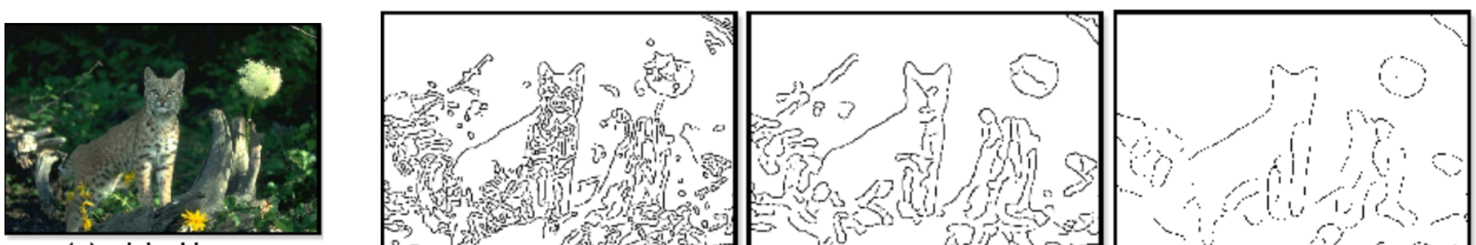
(Sezgin, J. Electronic Imaging 2004)

Edges, corners

Category	Classification	Methods and Algorithms
Edge-based	Differentiation based	Sobel, Canny
Corner-based	Gradient based	Harris (and its derivatives), KLT, Shi-Tomasi, LOCOCO, S-LOCOCO
Corner-based	Template based	FAST, AGAST, BRIEF, SUSAN, FAST-ER
Corner-based	Contour based	ANDD, DoG-curve, ACJ, Hyperbola fitting, etc.
Corner-based	Learning based	NMX, BEL, Pb, MS-Pb, gPb, SCG, SE, tPb, DSC, Sketch Tokens, etc.

(Y. Li et al., Neurocomputing, 2015)

- Edges: *refer to pixel patterns at which the intensities abruptly change (with a strong gradient magnitude)*



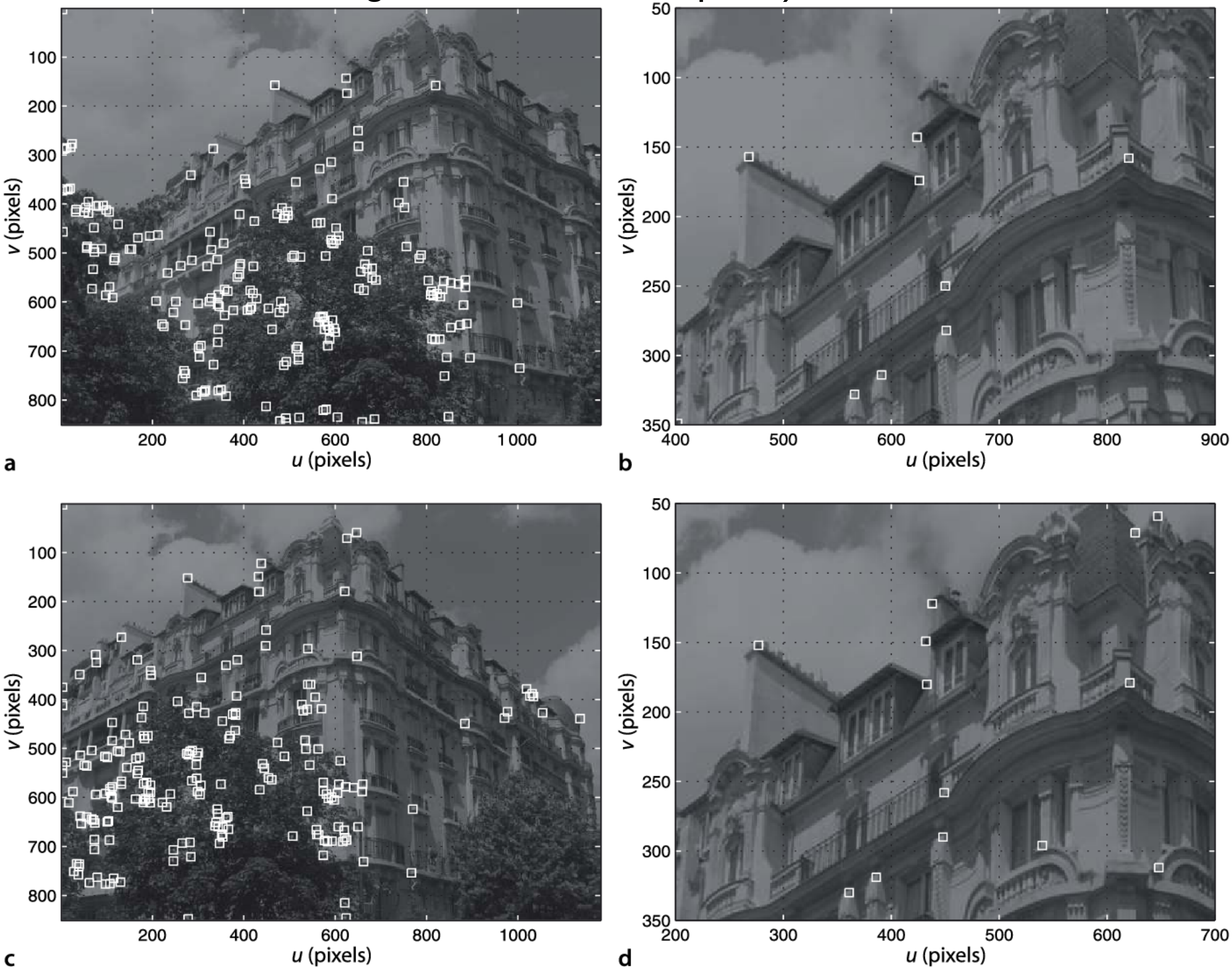
(a) original image (g) Canny: $\sigma = 2$ (h) Canny: $\sigma = 4$ (i) Canny: $\sigma = 8$

Parameter: standard deviation of the Gaussian function. Small value to detect sharp intensity transitions, large value to detect gradual transitions.

- Corners: *refer to the point at which two (or more) edge intersect in the local neighborhood*

Corner detection

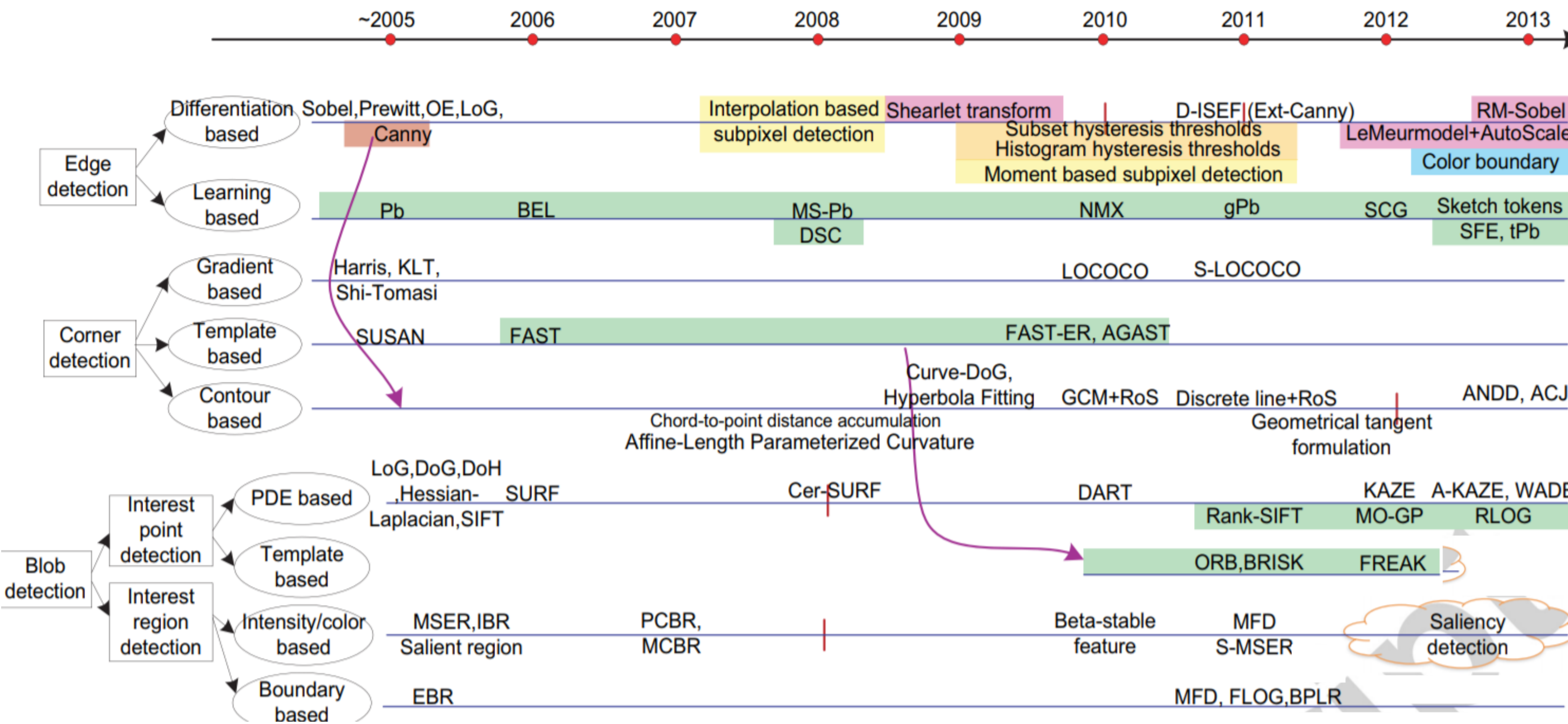
(a point for which there are two dominant and different edge directions in a local neighbourhood of the point)



Harris detector:

- Computed from image gradients
- Robust to illumination offsets and rotation
- No scale invariance (gradients around the corner point become lower)
- Responds strongly to fine texture, not to large structures

Interest point detection



SIFT: maxima in a difference of Gaussian sequence (patented)
SURF: maxima in an approximate Hessian of Gaussian sequence (patented)
ORB: FAST keypoint detector and BRIEF descriptor
…

(Y. Li et al., Neurocomputing, 2015)
(S. Krig, Computer vision metrics, 2014)

Interest point detection

Interest Point detection

Features Detector	Invariance			Qualities			
	Rotation	Scale	Affine	Repeatability	Localization	Robustness	Efficiency
Harris	■	-	-	+ + +	+ + +	+ + +	+ +
Hessian	■	-	-	+ +	+ +	+ +	+
SUSAN	■	-	-	+ +	+ +	+ +	+ + +
Harris-Laplace	■	■	-	+ + +	+ + +	+ +	+
Hessian-Laplace	■	■	-	+ + +	+ + +	+ + +	+
DoG	■	■	-	+ +	+ +	+ +	+ +
Salient Regions	■	■	■	+	+	+ +	+
SURF	■	■	-	+ +	+ + +	+ +	+ + +
SIFT	■	■	-	+ +	+ + +	+ + +	+ +
MSER	■	■	■	+ + +	+ + +	+ +	+ + +

(T. Tuytelaars et al., Foundations and trends in computer graphics and vision, 2008)

Invariance: in scenarios where a large deformation is expected (scale, rotation, etc.), the detector algorithm should model this deformation mathematically as precisely as possible so that it minimizes its effect on the extracted features.

Repeatability: given two frames of the same object (or scene) with different viewing settings, a high percentage of the detected features from the overlapped visible part should be found in both frames.

Efficiency: features should be efficiently identified in a short time that makes them suitable for real-time (i.e. time-critical) applications.

Locality: features should be local so as to reduce the chances of getting occluded as well as to allow simple estimation of geometric and photometric deformations between two frames with different views.

Robustness: not too much sensitive to small deformations (noise, blur, discretization effects, compression artifacts, etc.)

Line features

- Convolution (see chapter 11)

a)

-1	-1	-1
2	2	2
-1	-1	-1

b)

-1	2	-1
-1	2	-1
-1	2	-1

c)

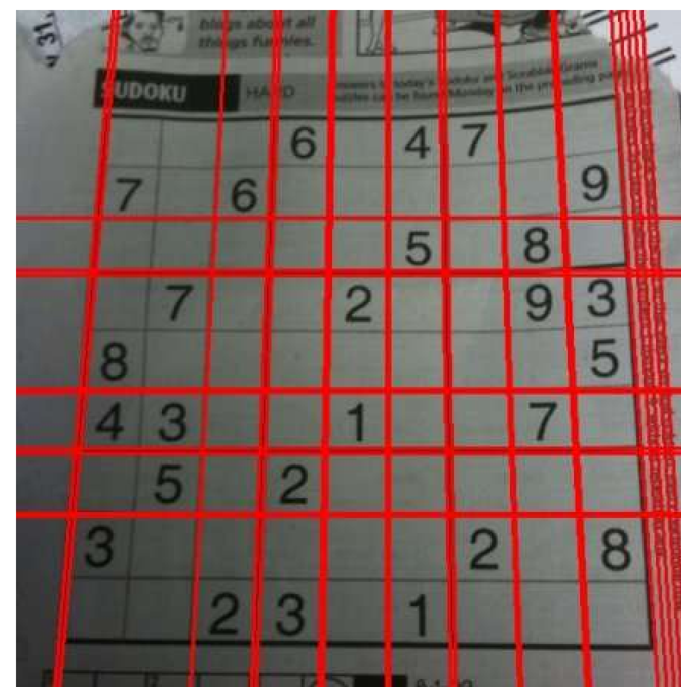
-1	-1	2
-1	2	-1
2	-1	-1

d)

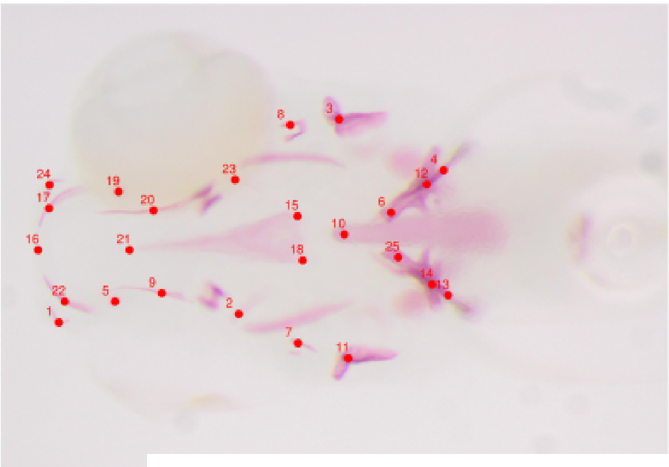
2	-1	-1
-1	2	-1
-1	-1	2

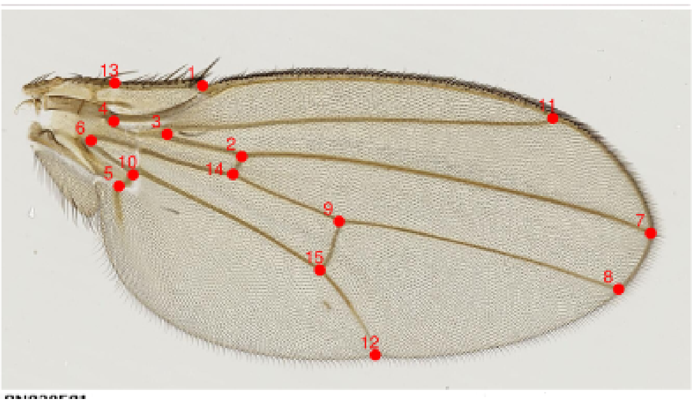
Four line detection kernels which respond maximally to horizontal, vertical and oblique (+45 and - 45 degree) single pixel wide lines.

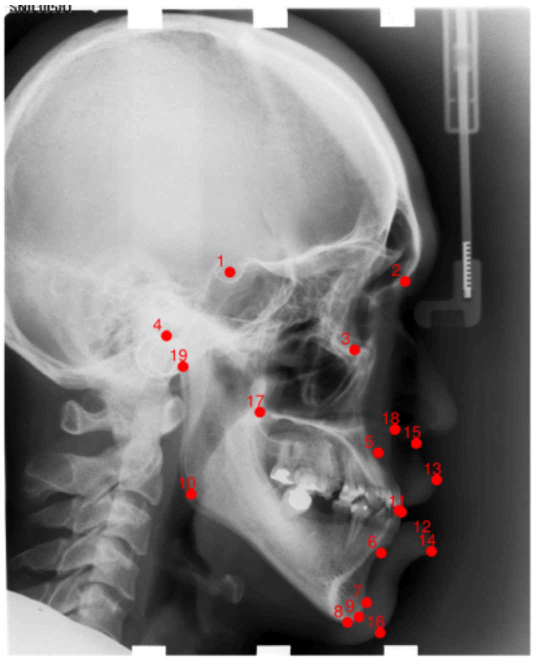
- Hough transform for lines, circles, ellipses
  (requires that the desired features be specified in some parametric form)



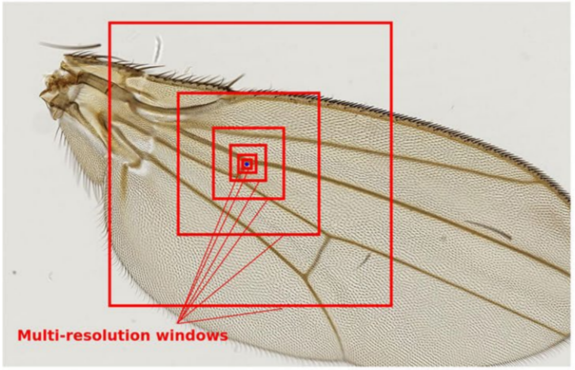
Specific (supervisely learned) landmarks

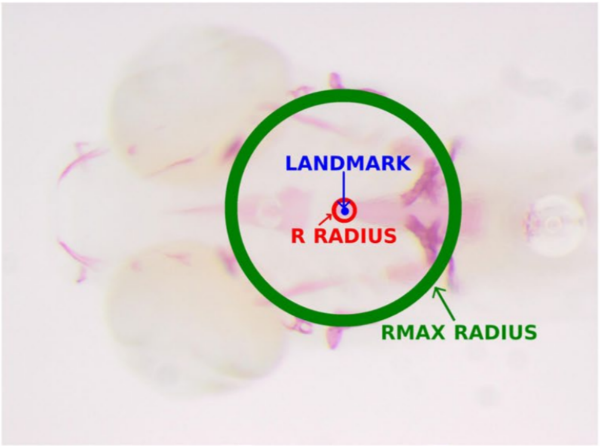


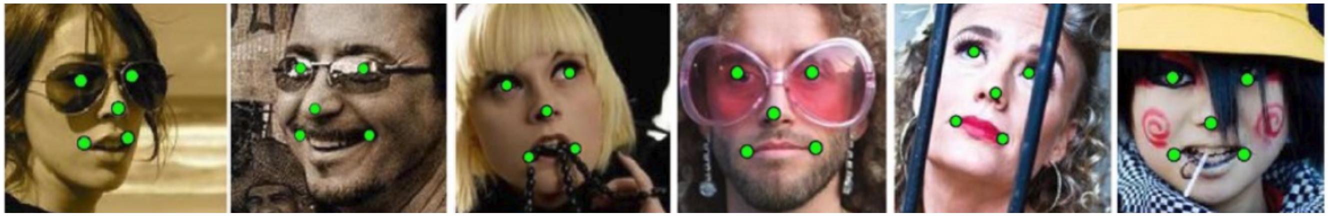




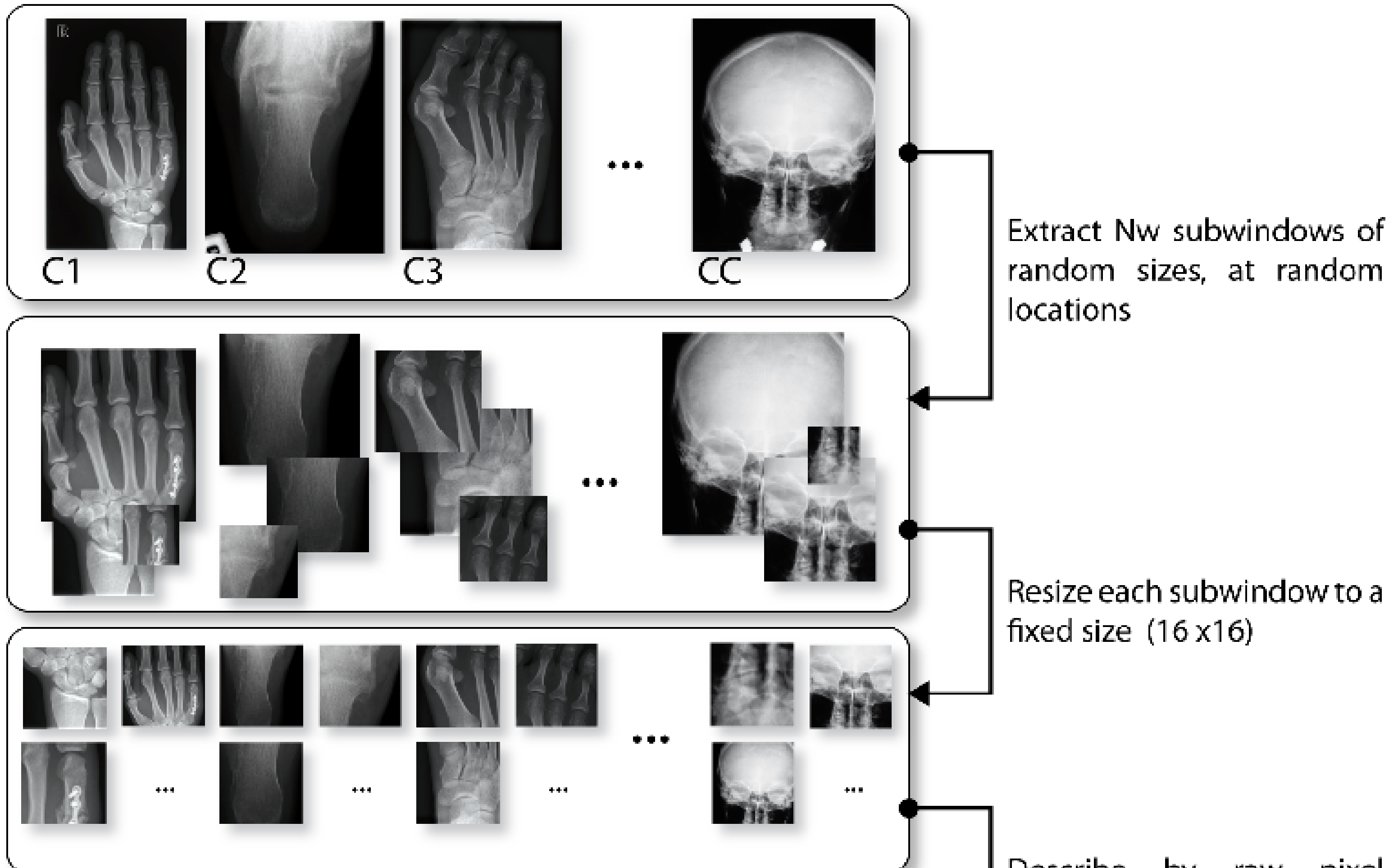
(Vandaele et al., Nature Scientific Reports, 2017)







Random features (patches)

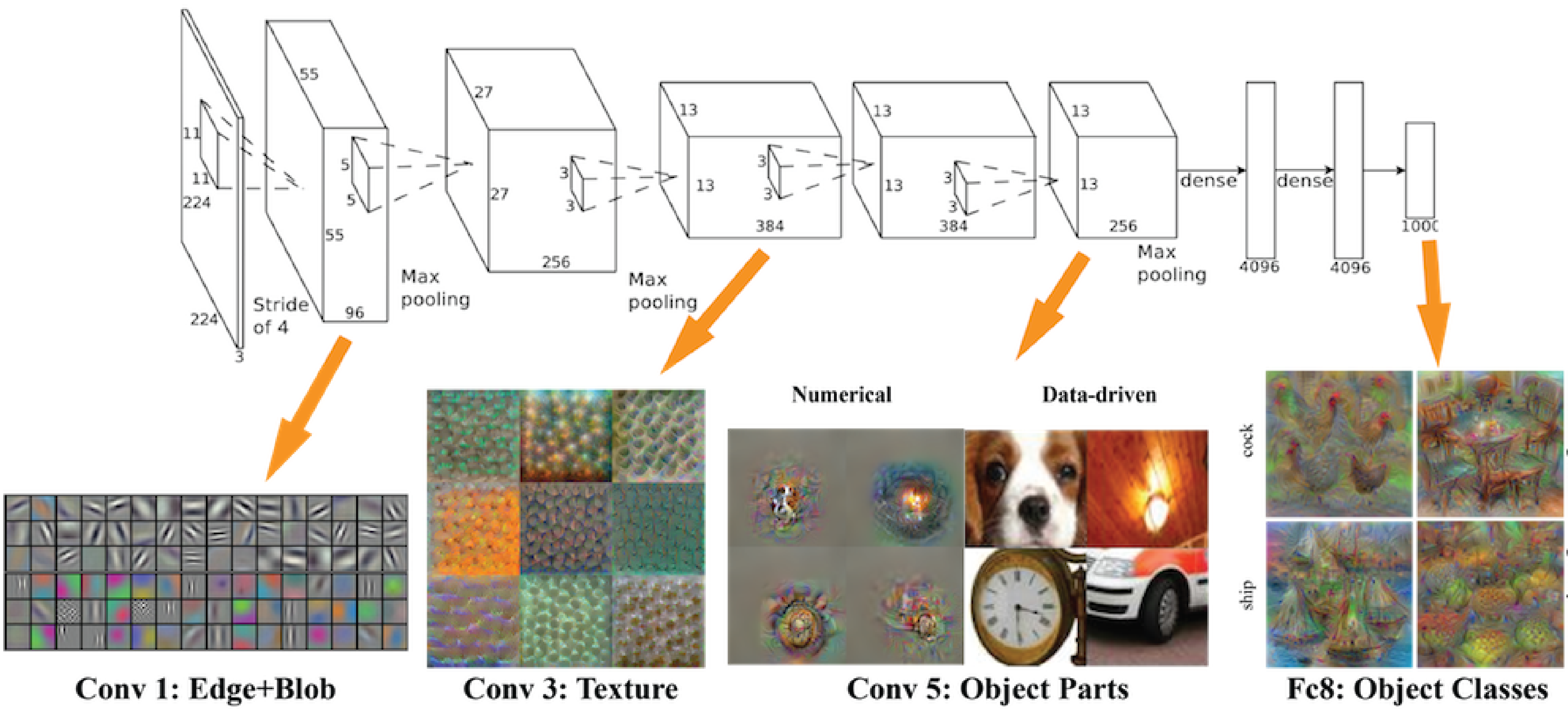


	Pixel-001	Pixel-002		Pixel-256	CLASS
Subwindow-00	34	201		255	C1
Subwindow-01	78	31		204	C1
Subwindow-02	145	167		32	C2
...					
Subwindow-Nw	14	4		134	CC

Parameters :
Nsw = nb subwindows
MinSize = [0%-100%]
MaxSize = [0%-100%]
Resize = 16x16
Colorspace = HSV/GRAY

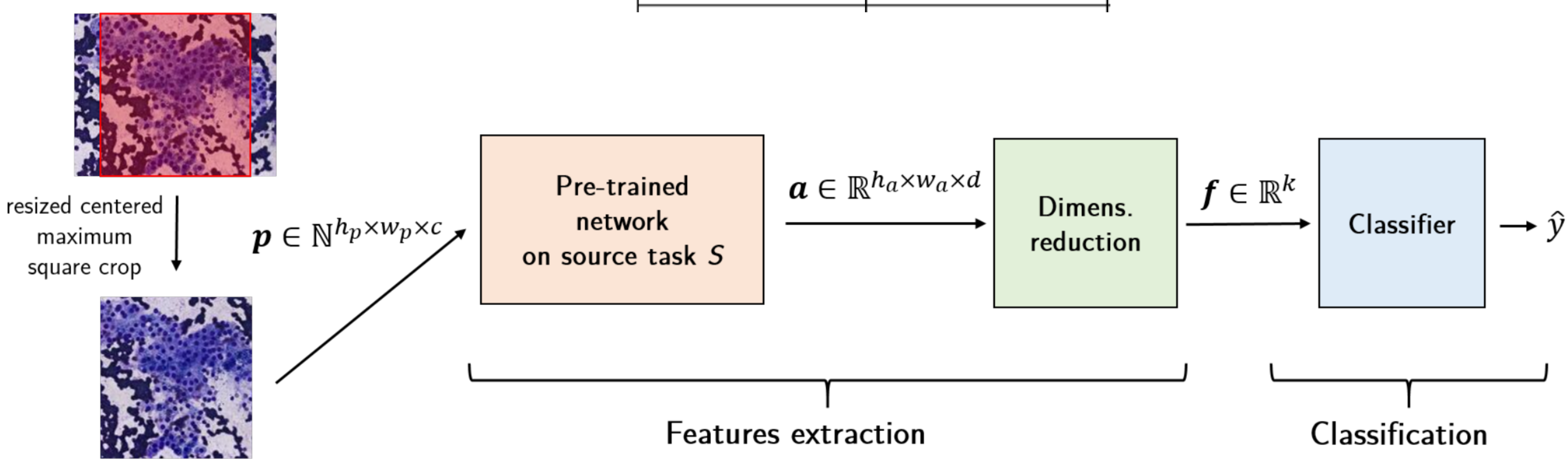
Deep-network based features

- A pre-built deep network can be seen as a feature extractor



Deep-network based features

$\mathcal{N}$	**Last layer**
	# feat.
Mobile	1024
DenseNet	1920
IncResV2	1536
ResNet	2048
IncV3	2048
VGG19	512
VGG16	512



1. Feature extraction

2. Object Recognition / Image classification
   - Challenges
   - Bag-of-features
   - End-to-end learning
   - Quality control

3. Intelligent robotics / AI in Biomedecine

Object/Category Recognition

Image classification: assigning a class label to the image

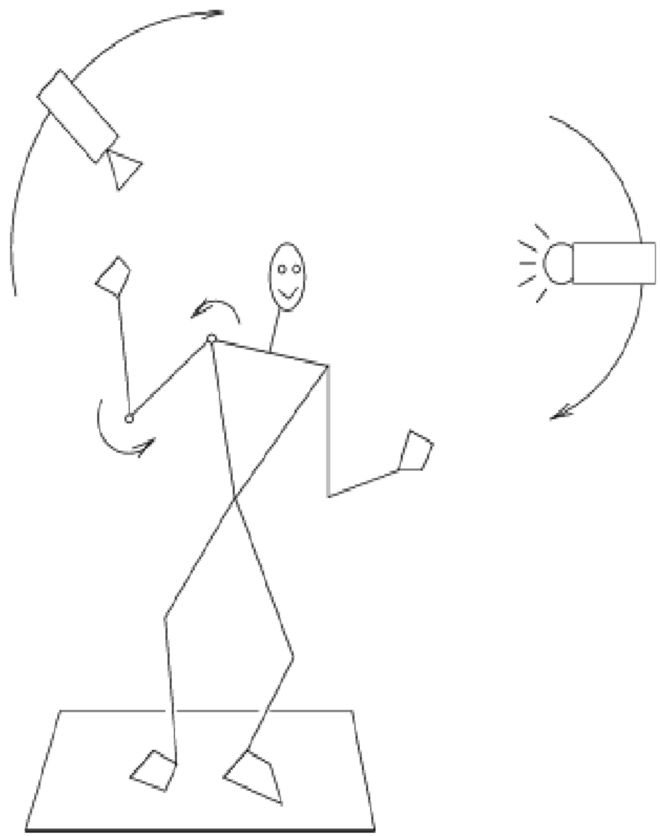
Car: present
Cow: present
Bike: not present
Horse: not present
...

Object localization: define the location and the category

Location

Category

Challenge 1: Intra-instance Variations

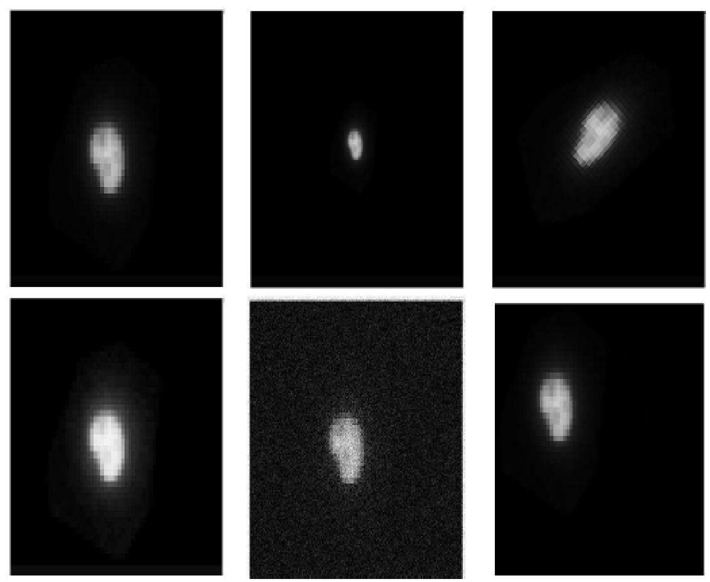


Viewpoint, illumination, kinematic configuration, ...

Variations in “biomedecine”

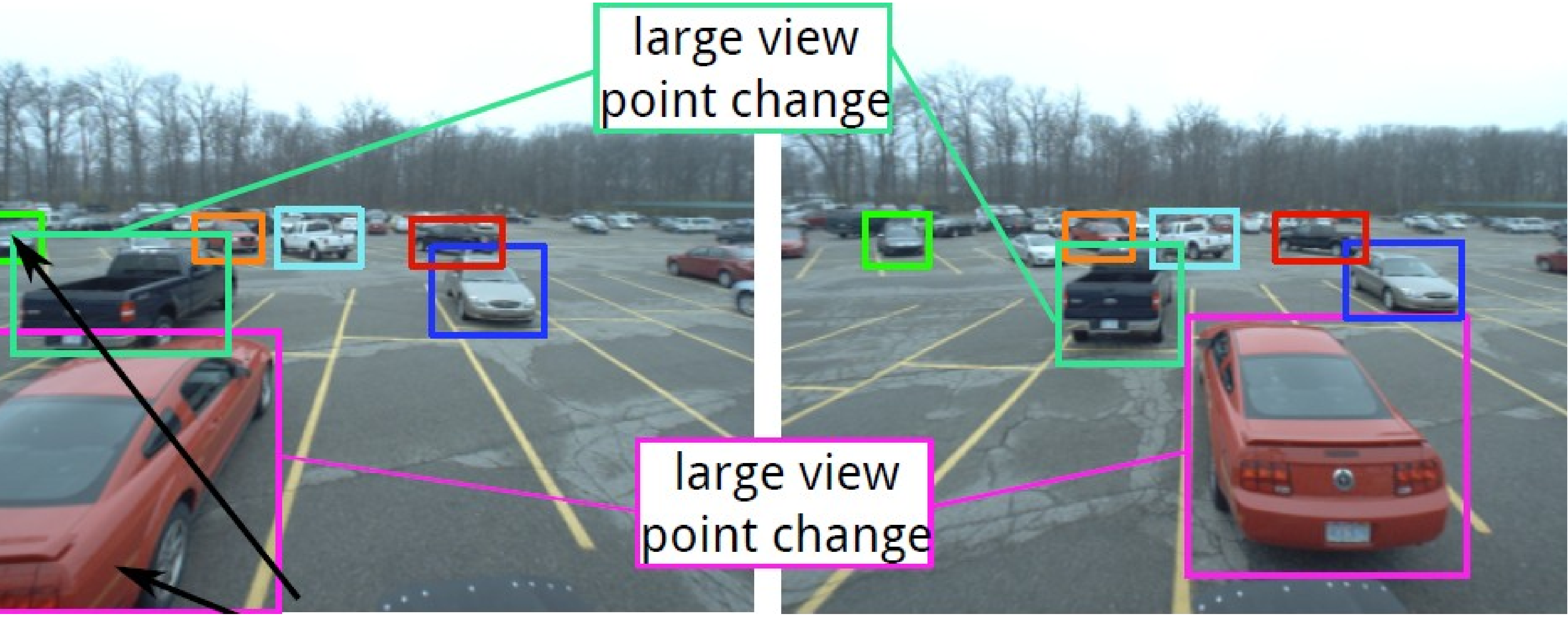
Image acquistion conditions can not be fully controlled

- Illumination, viewpoint, position, scale changes
- Noise, cluttered background, occlusions
- Staining, imaging kits, microscopes, ...



These simple variations yield different image matrices

Challenge 1: Intra-instance Variations



Viewpoint, illumination, kinematic configuration, ...

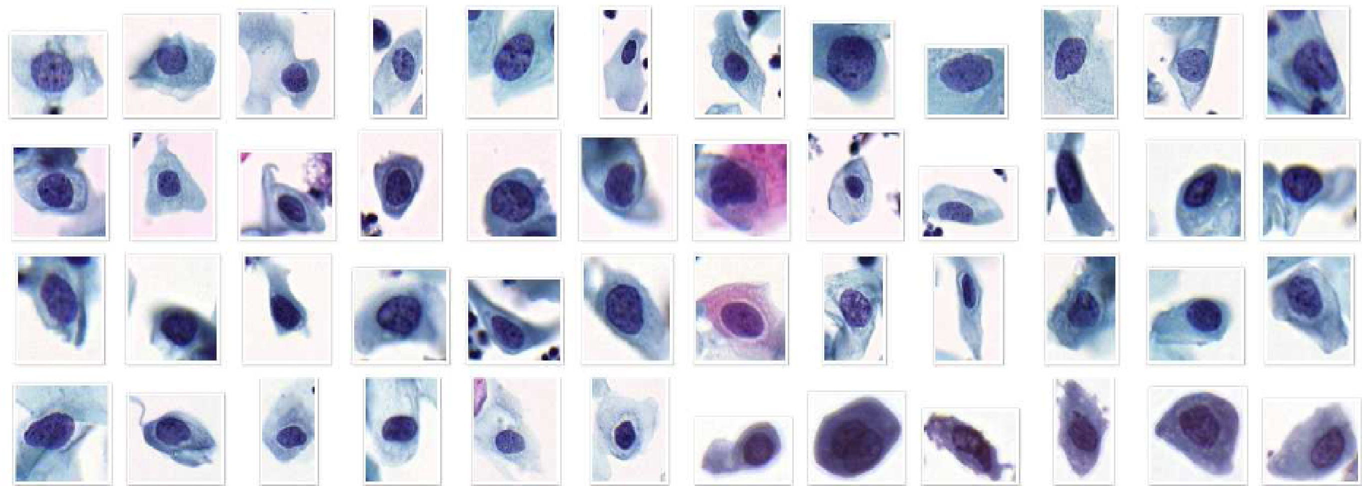
Challenge 2: Intra-class Variations

Challenge 2: Intra-class Variations

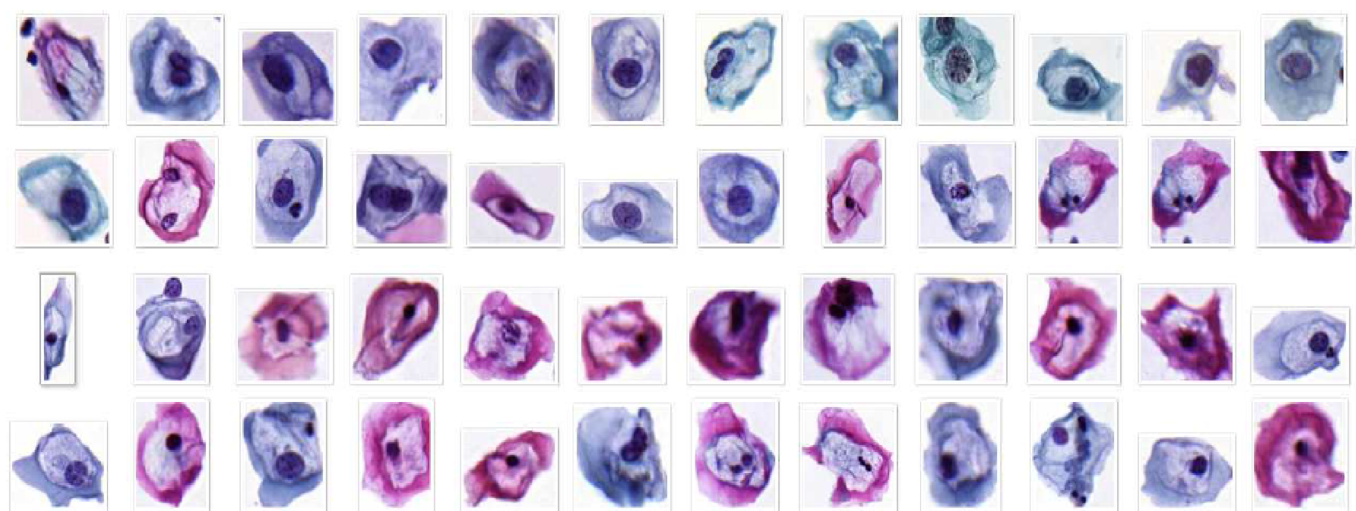
(DOTA dataset)

Variations in « biomedecine »

Intra-class / inter-class variations



vs



1. Feature extraction

2. Object Recognition / Image classification
   Challenges
   Approaches:
      Bag-of-features,   End-to-end learning
   Dataset quality control

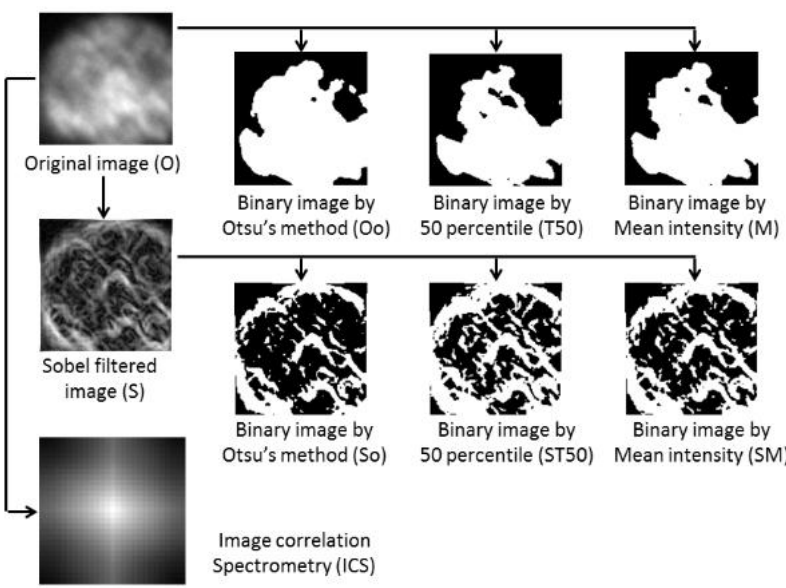
3. Intelligent robotics / AI in Biomedecine

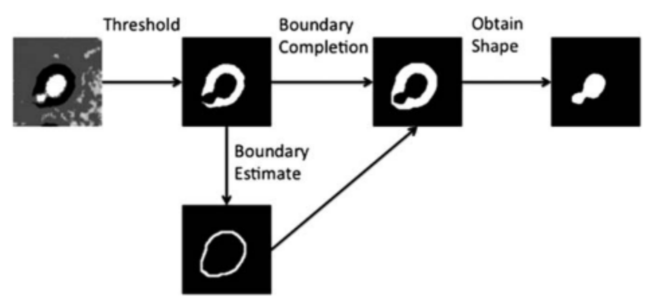
Computer vision approaches

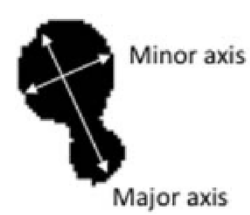
- Traditional : hand-crafted, specific, features +learning

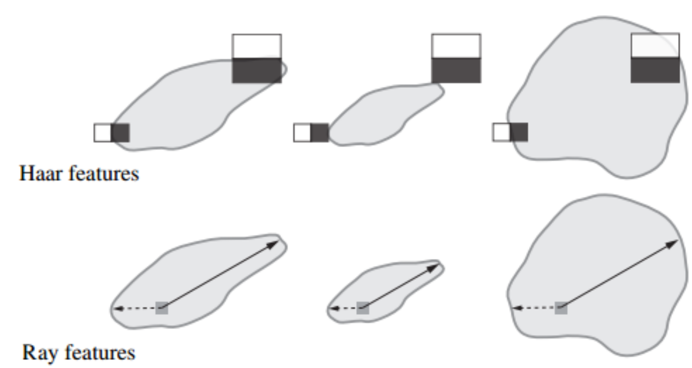
- *Hypothesis : the researcher is very imaginative, and smart*
- *Pros : exploitation of domain knowledge*
- *Cons : need to be adapted when the problem changes*

researchers are indeed imaginative
limited evaluation
} which features to choose ?









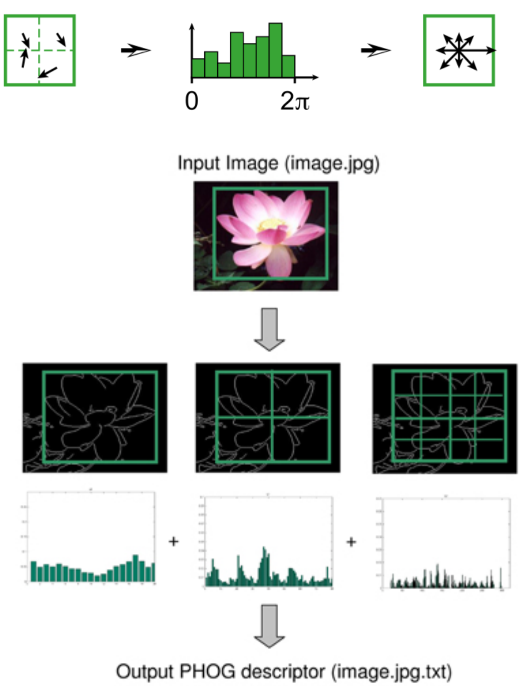
Computer vision approaches

- *Traditional : hand-crafted, specific, features +learning*

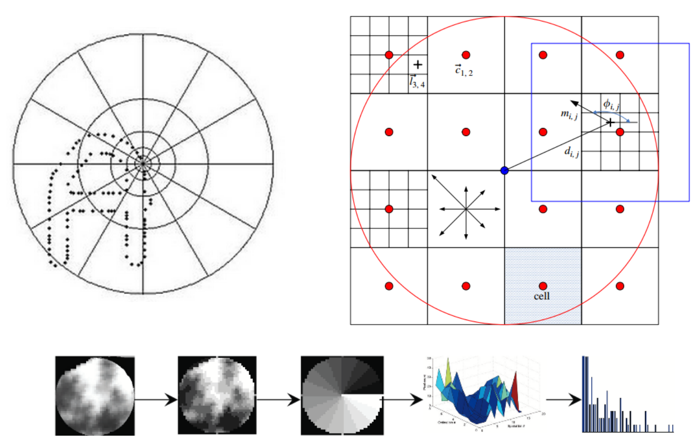
- *Hypothesis : the researcher is very imaginative, and smart*
- *Pros : exploitation of domain knowledge*
- *Cons : need to be adapted when the problem changes*

researchers are indeed imaginative
limited evaluation

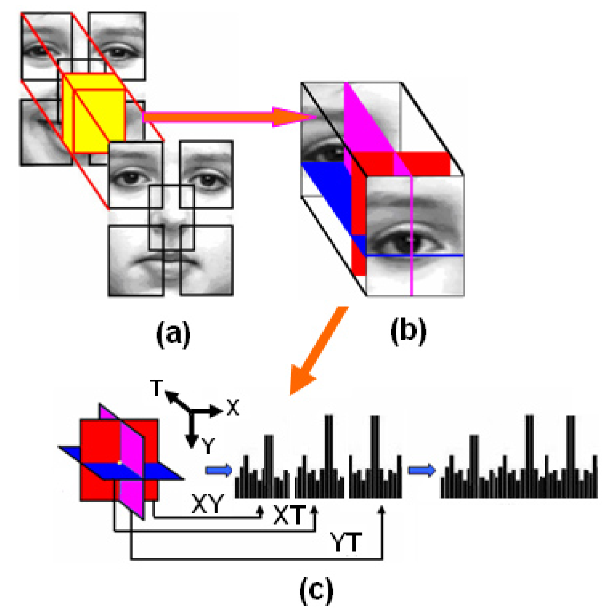
which features to choose ?



Harris-Affine, Hessian-Affine, EBR, IBR, MSER, SFOP,DAISY, GIST, GLOH, LBP, OSID, PHOG, PHOW, SIFT, RIFT, PCA-SIFT, Spin Image, SURF, VLAD, Shape contexts, Textons, ...



Li & Allison, Neurocomputing 2008

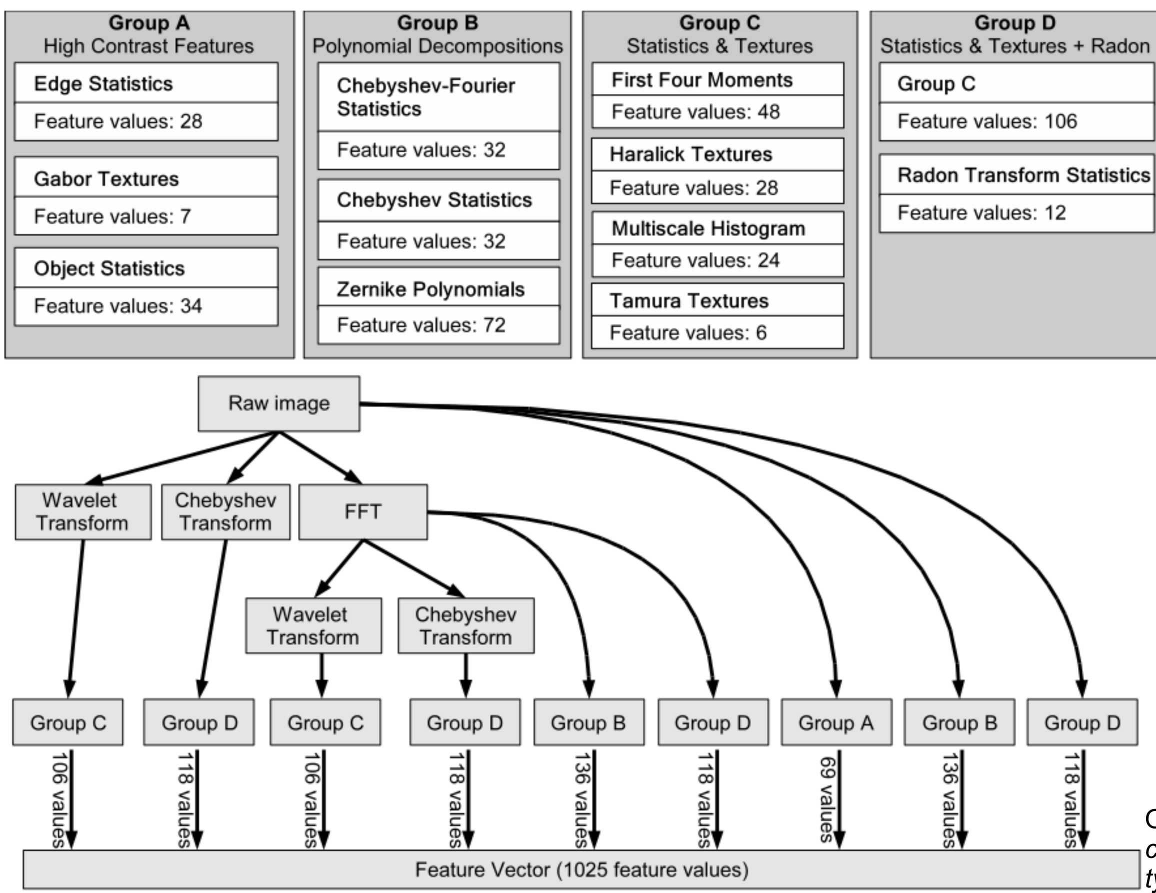


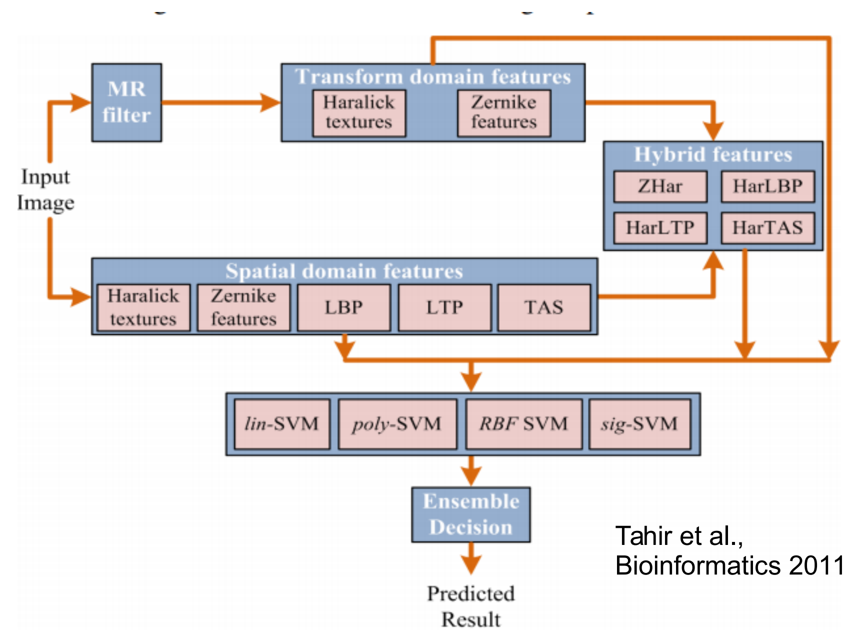
Scholarpedia

Computer vision approaches

- *Recent : Combine many features + learning*

- *Hypothesis : the good features should be among them*
- *Pros : take advantage of previous research efforts*
- *Cons : computationally intensive*





Tahir et al., Bioinformatics 2011

Orlov et al., Pattern Recognition letters, 2008 : « *...poor performance in terms of computational complexity, making this method unsuitable for real-time or other types of applications in which speed is a primary concern*. »

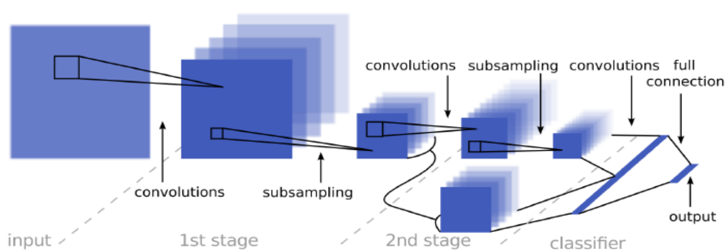
Computer vision approaches

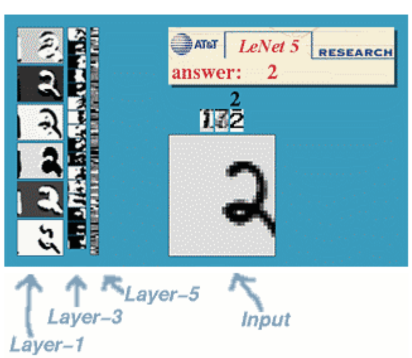
- *Generic : « end-to-end » learning*

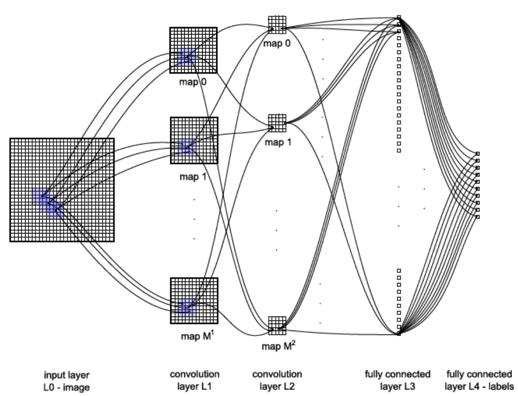
- *Hypothesis : human brain learn from raw data, let's design such an algorithm*
- *Pros : it should work on everything with minimal tuning*
- *Cons : <> architectures*

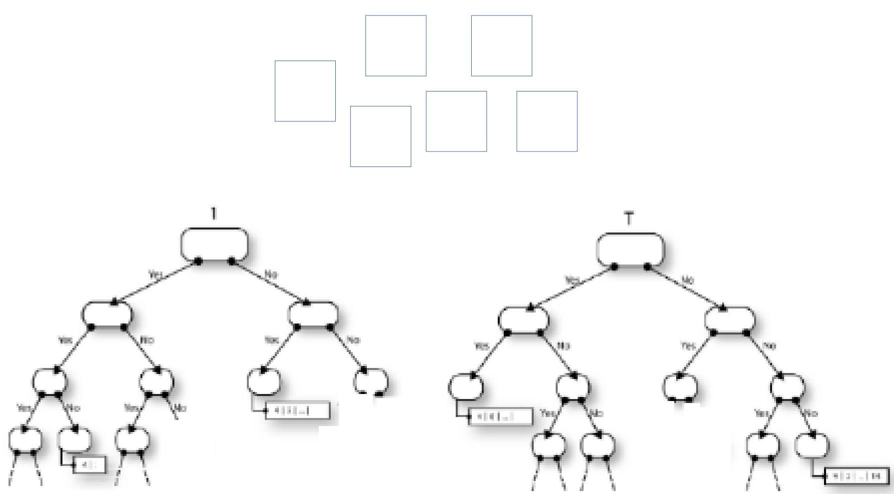
  many parameters to optimize: need large training data, time-consuming

  does it work ? Is it generic ?





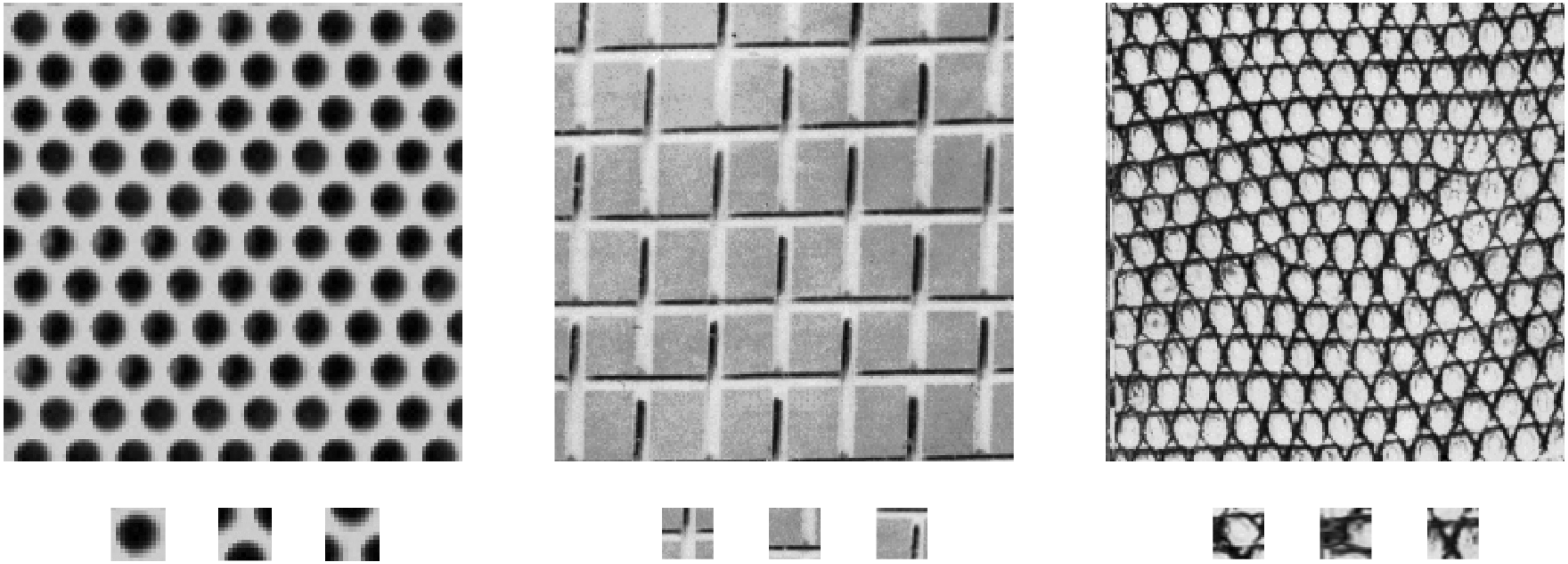




Marée, Geurts, Wehenkel, et al. 2003 ...

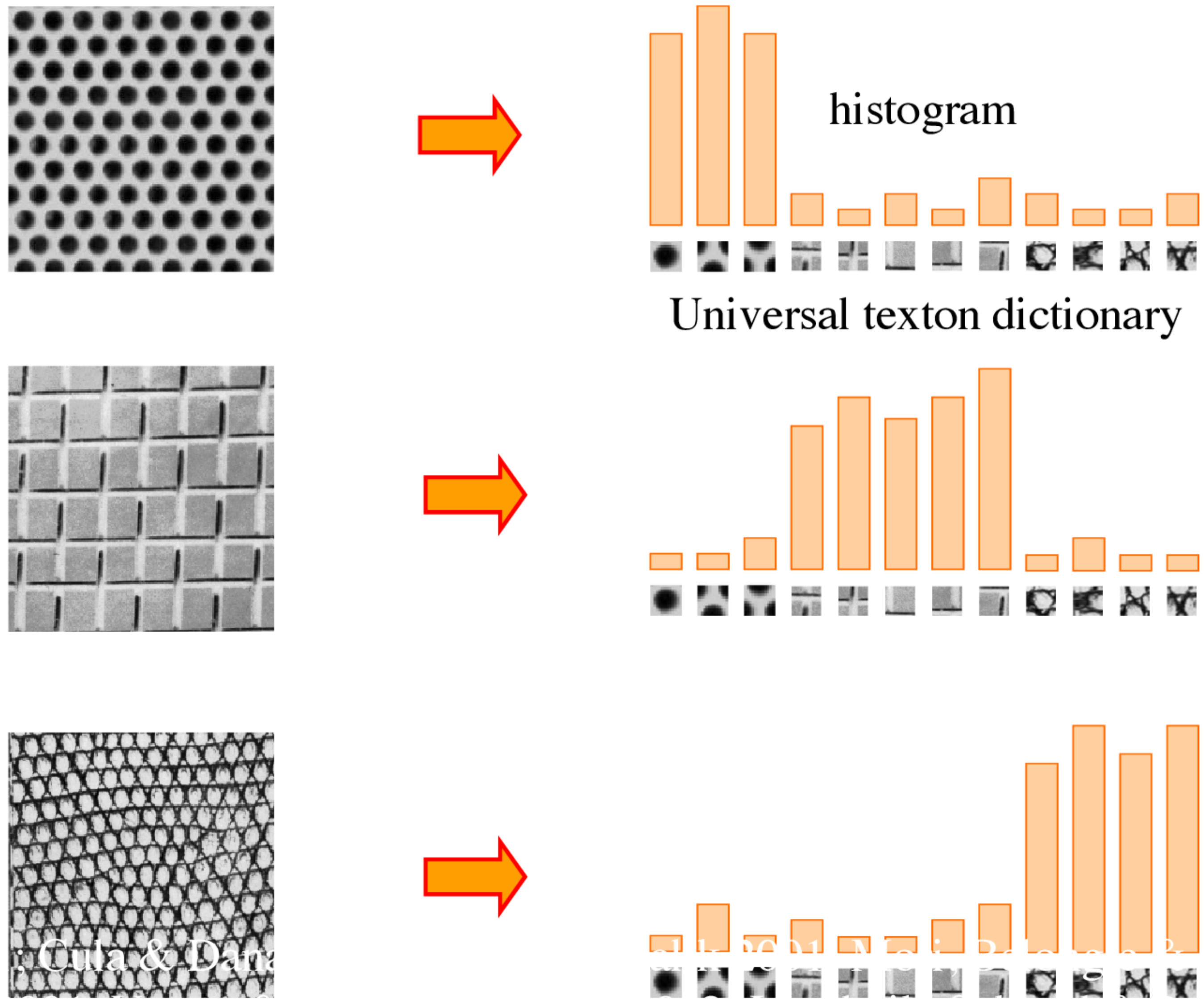
Lecun et al. 1989..., Hinton et al., Ciresan et al. (GPU) 2011

BoF Origin: Texture Classification



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Texture Classification: Histograms over Textons



Bag-of-features for Image Classification

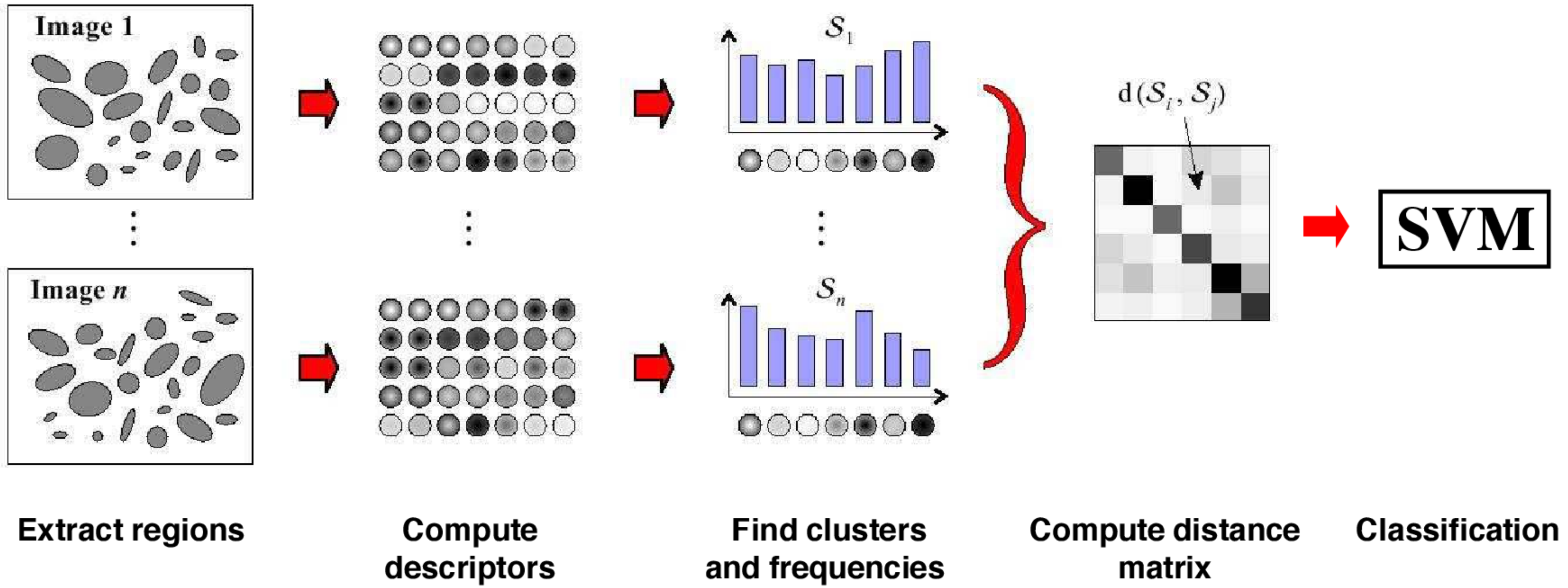
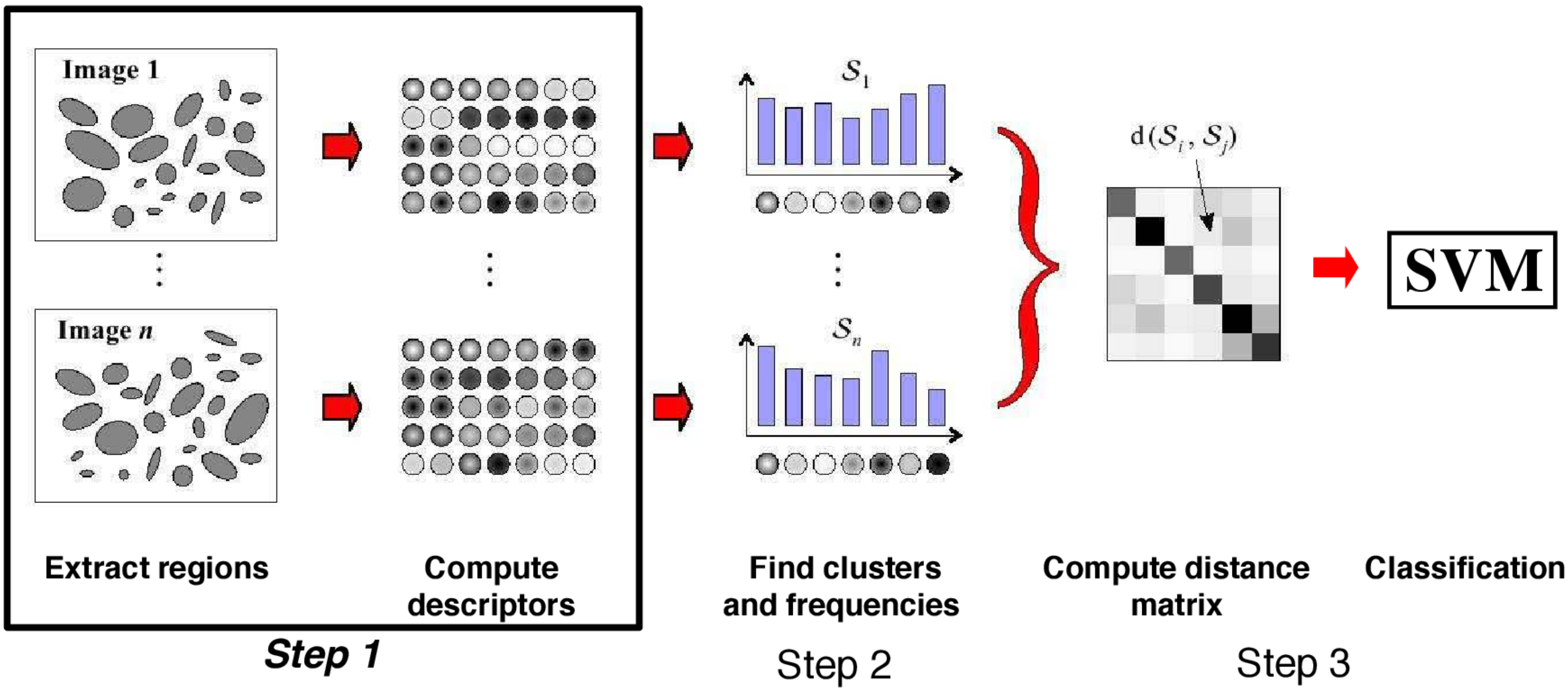


Image 1 contains a bike, image 2 contains a horse, image 3 contains a car, etc...

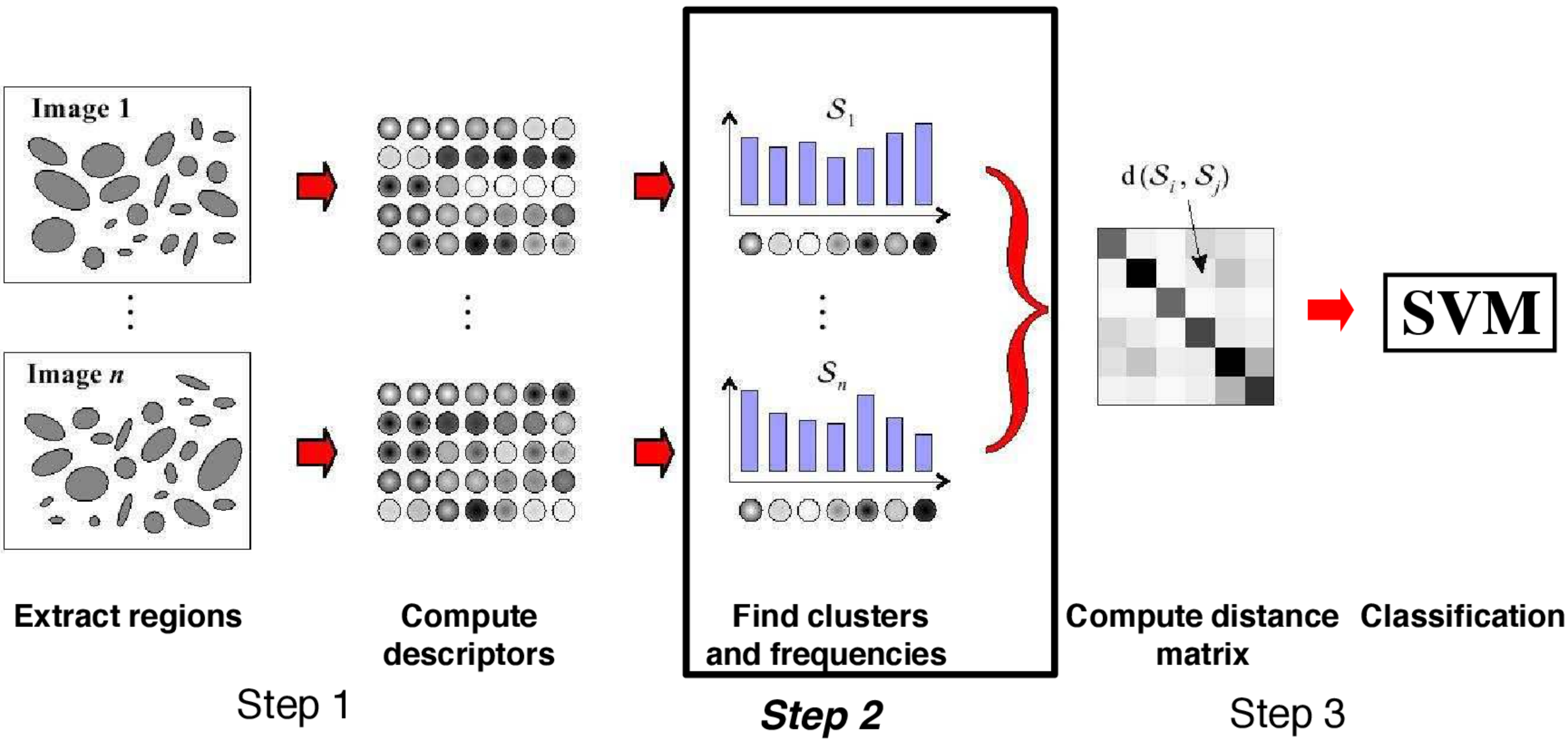
[Csurka et al., ECCV Workshop'04], [Nowak, Jurie, Triggs, ECCV'06], [Zhang, Marszalek, Lazebnik, Schmid, IJCV'07]

Step 1: Extract Features



Corners / Point / Random / …
+
Descriptors (statistics, pixels,...)

Step 2: Cluster Features, Compute Feature Frequencies



Unsupervised
(e.g. k-means)

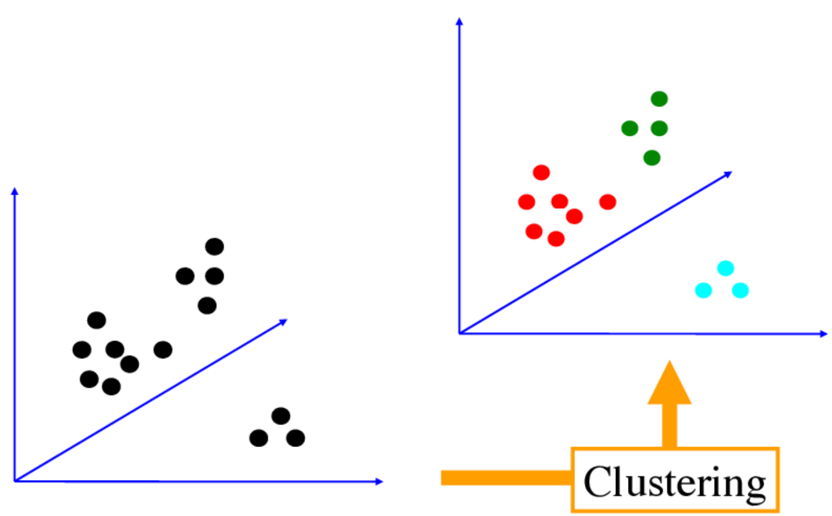
Supervised
(e.g. trees)

Clustering Features (k-means)

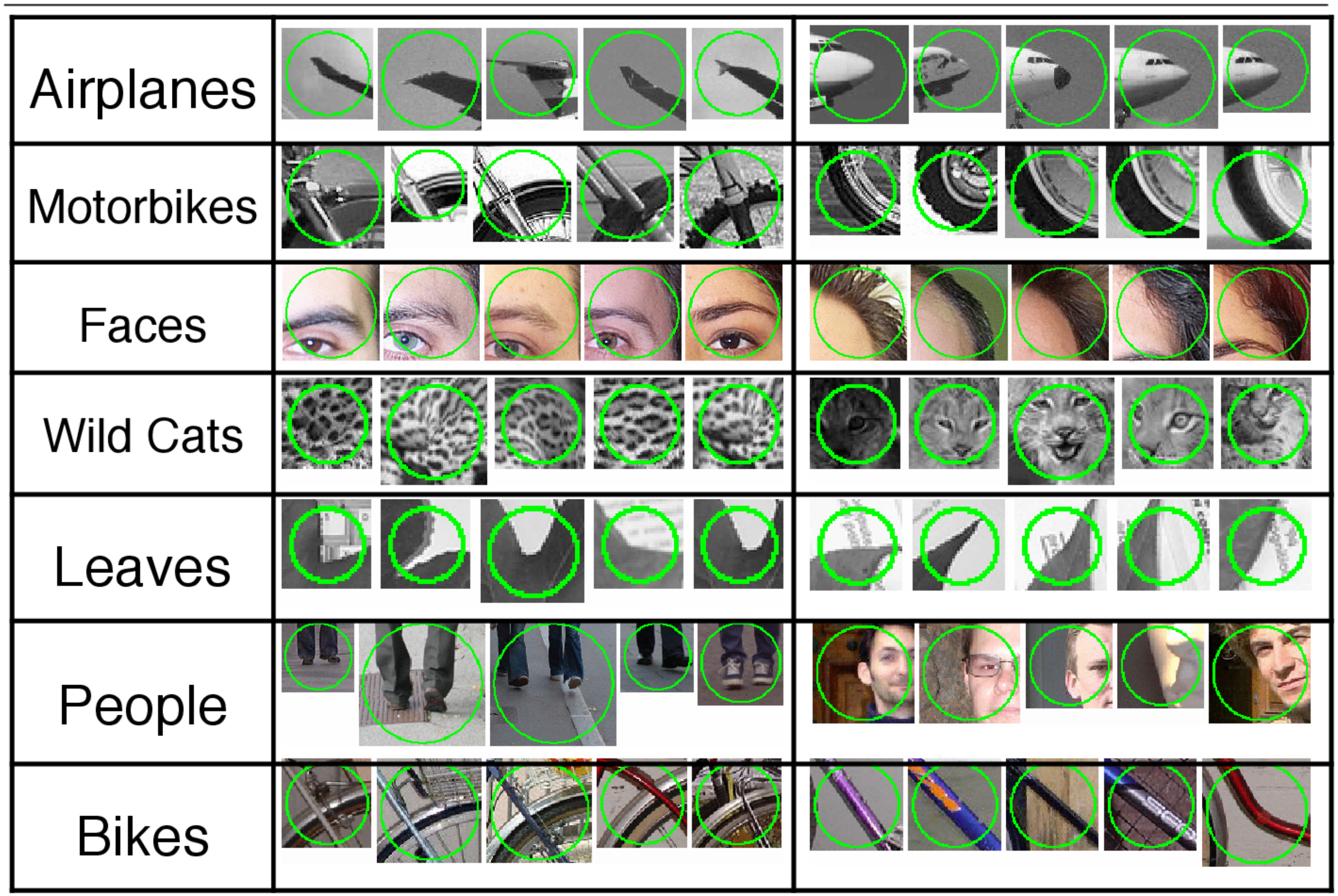
Because of viewpoint and lighting changes, it is unlikely that two images of even the same object will produce exactly the same features. This gets even worse when working with different instances of a class (e.g., different cars).

As a result, it is not a good idea to model an object (or object class) with a histogram over all the features that the object produces.

Instead, we cluster *all* the features that come from the training images (of all classes), and keep only the cluster centers. The set of cluster centers is called a codebook, or dictionary. The elements of the codebook are called codewords.



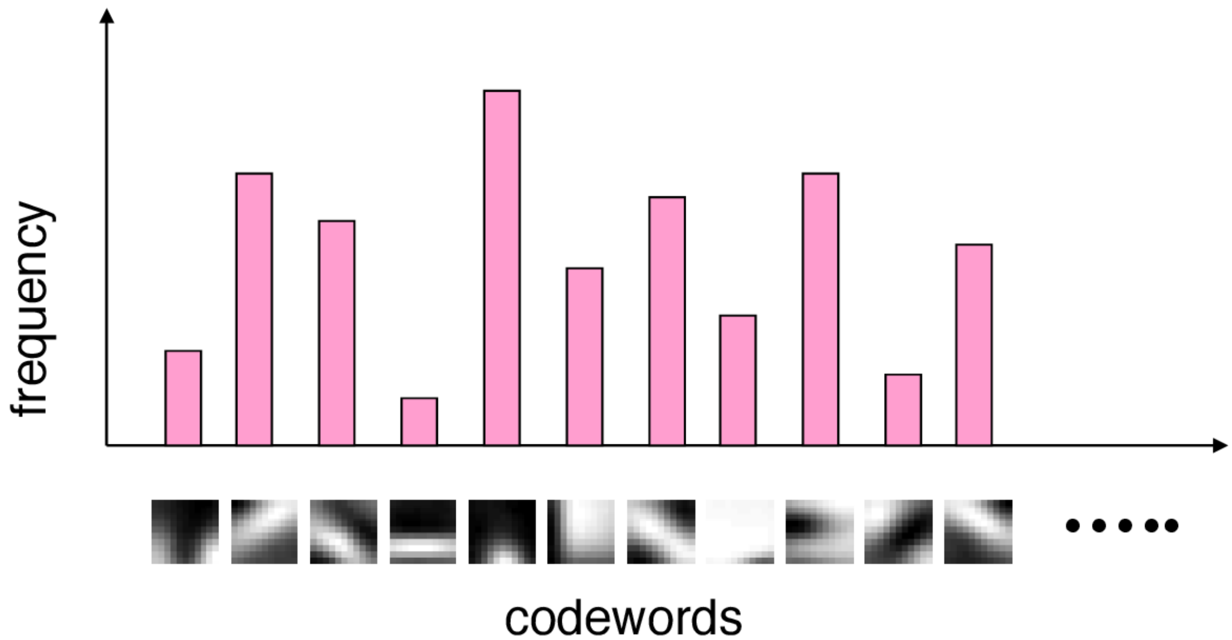
Examples of Clusters of Features



↓ Average of these becomes one codeword

↓ Average of these becomes one codeword

Object/Class Instance Representation: Codeword Frequencies

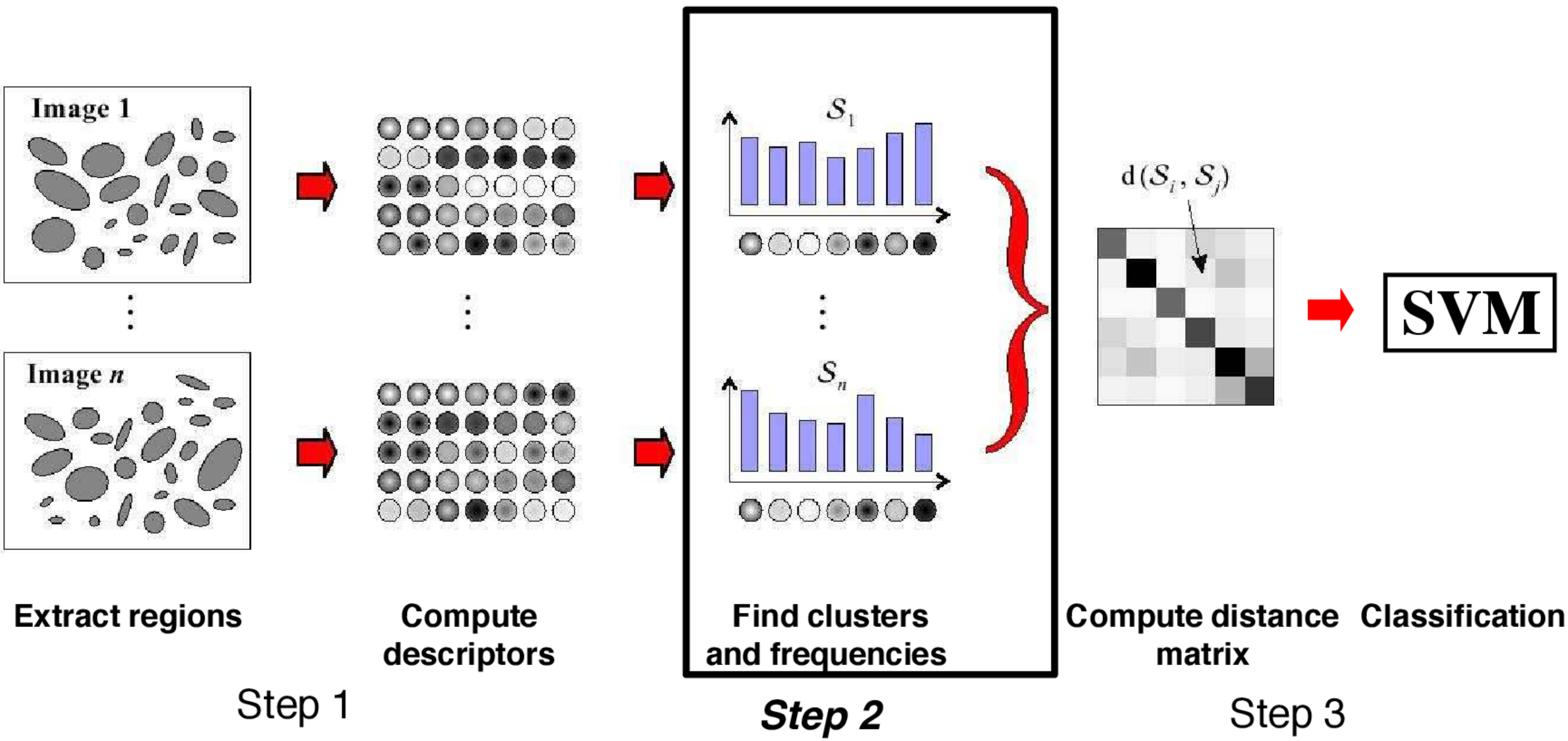


Typically: 1000–4000 codewords:

- More codewords: towards object representation
- Less codewords: towards object class representation

One image of an instance of an object/class is represented with a vector V of frequencies of the codewords. (L1/L2 normalization)

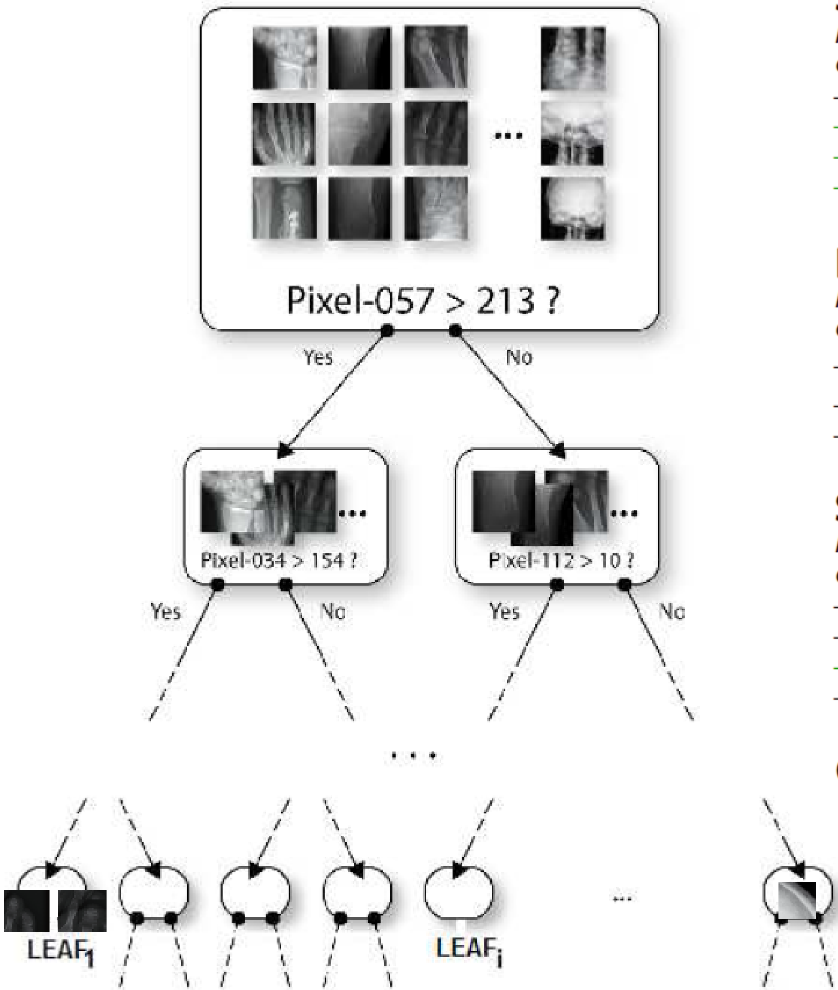
Step 2: Cluster Features, Compute Feature Frequencies



Unsupervised
(e.g. k-means)

Supervised
(e.g. trees)

Extra-Trees for Feature Learning : training



Split_a_node(S)

Input: the local learning subset S corresponding to the node we want to split
Output: a split $[a < a_c]$ or leaf
- If **Stop_split**(S) is TRUE then attach predictions (ET-DIC) or nothing (ET-BOF).
- Otherwise select randomly K attributes $\{a_1, \ldots, a_K\}$ among all non constant (in S) candidate attributes;
- Draw K splits $\{s_1, \ldots, s_K\}$, where $s_i =$ **Pick_a_random_split**$(S, a_i), \forall i = 1, \ldots, K$;
- Return a split s_i such that $\text{Score}(s_i, S) = \max_{j=1,\ldots,K} \text{Score}(s_j, S)$.

Pick_a_random_split(S,a)

Inputs: a subset S and an attribute a
Output: a split
- Let $a^S_{\max}$ and $a^S_{\min}$ denote the maximal and minimal value of a in S;
- Draw a random cut-point a_c uniformly in $]a^S_{\min}, a^S_{\max}]$;
- Return the split $[a < a_c]$.

Stop_split(S)

Input: a subset S
Output: a boolean
- If $|S| < n_{\min}$, then return TRUE;
- If if all attributes are constant in S, then return TRUE;
- If the output is constant in S, then return TRUE;
- Otherwise, return FALSE.

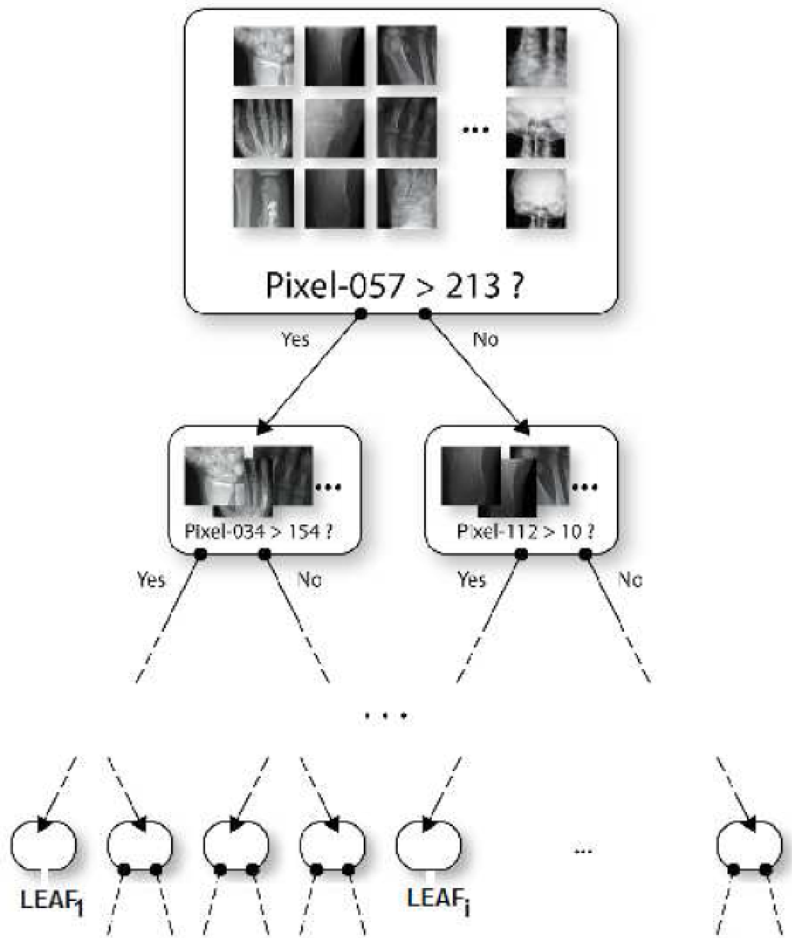
Complexity: $O(TKN\log_2(N))$

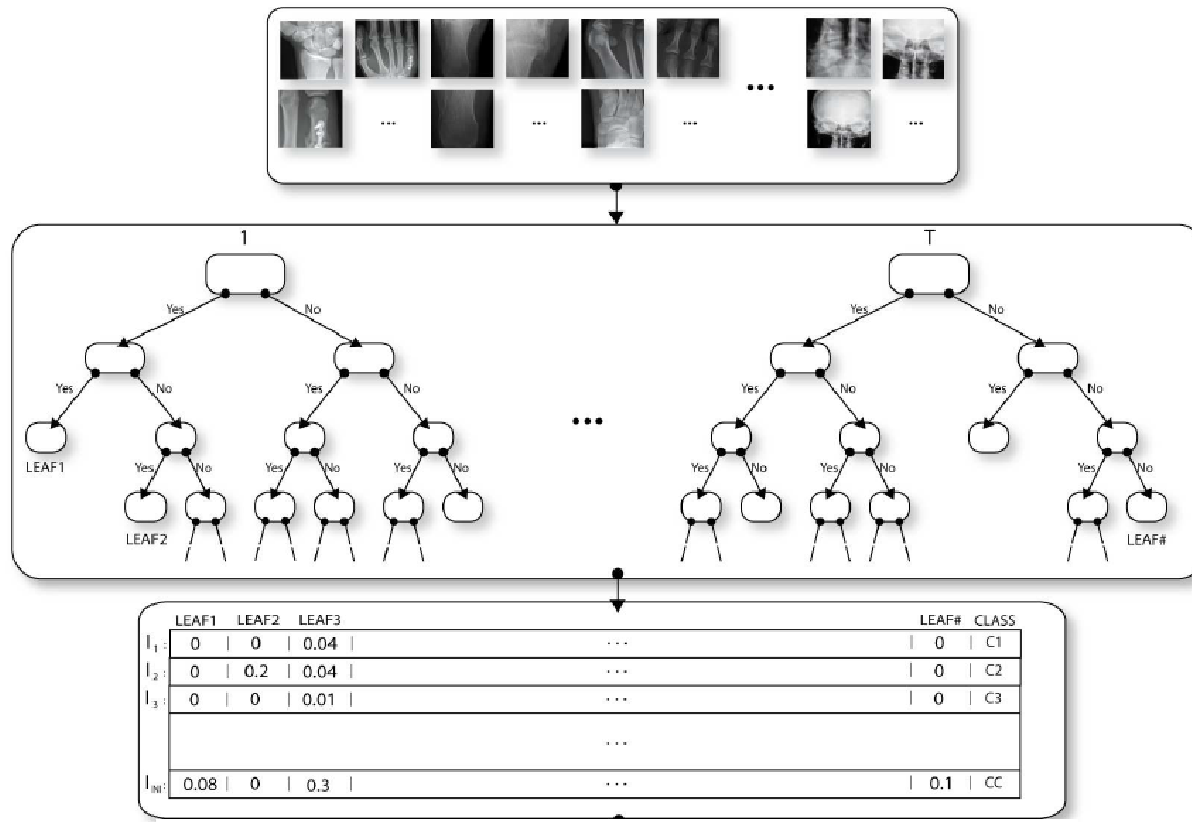
Parameters :

K = nb random tests
Nmin = minimum node size

Marée, Geurts, Wehenkel, 2016

Extra-Trees for Feature Learning : training





Parameters :
T= nb trees
K = nb random tests
Nmin = minimum node size
Coding = binary/frequency
FinalC = liblinear

Marée, Geurts, Wehenkel, 2016

Step 3: Image Classification

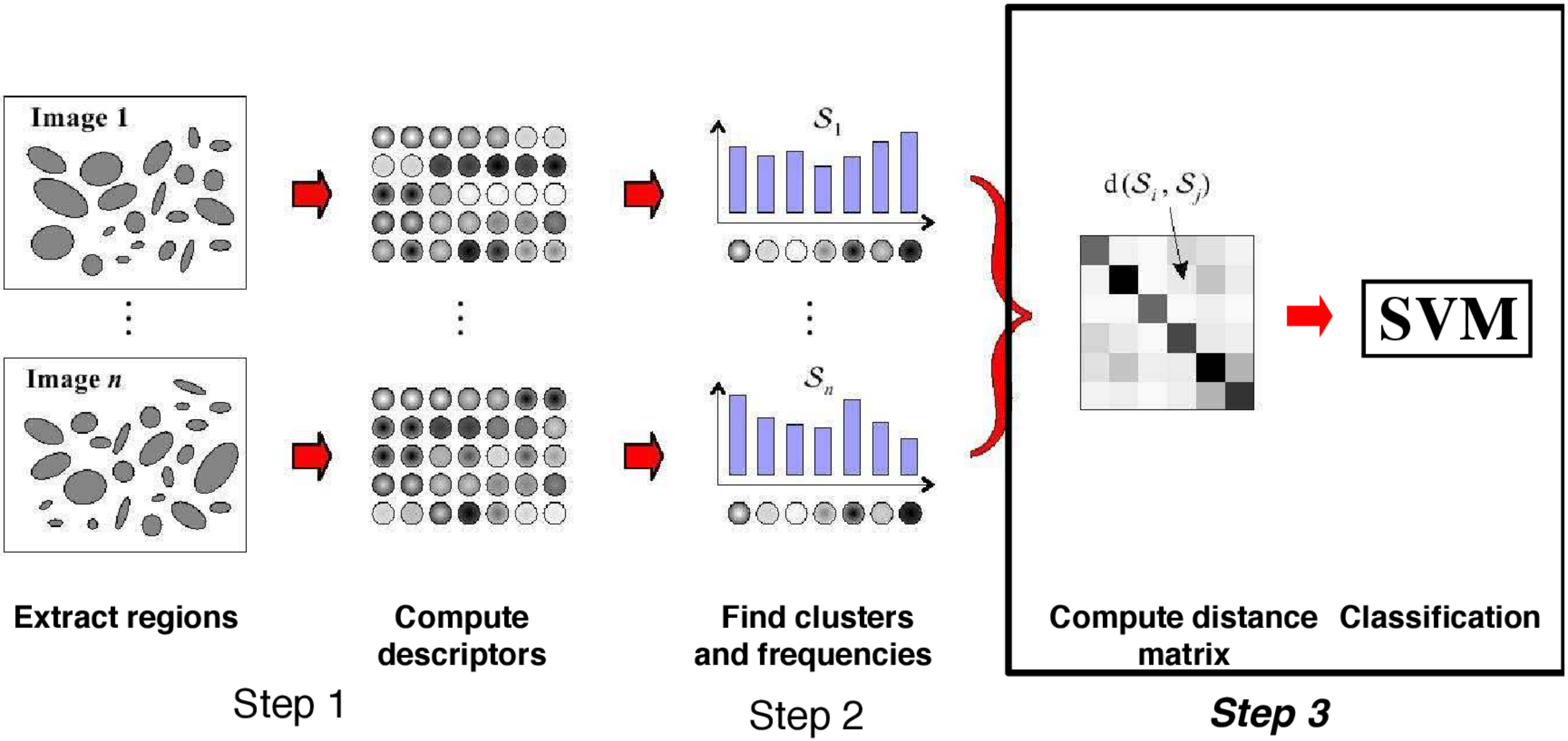
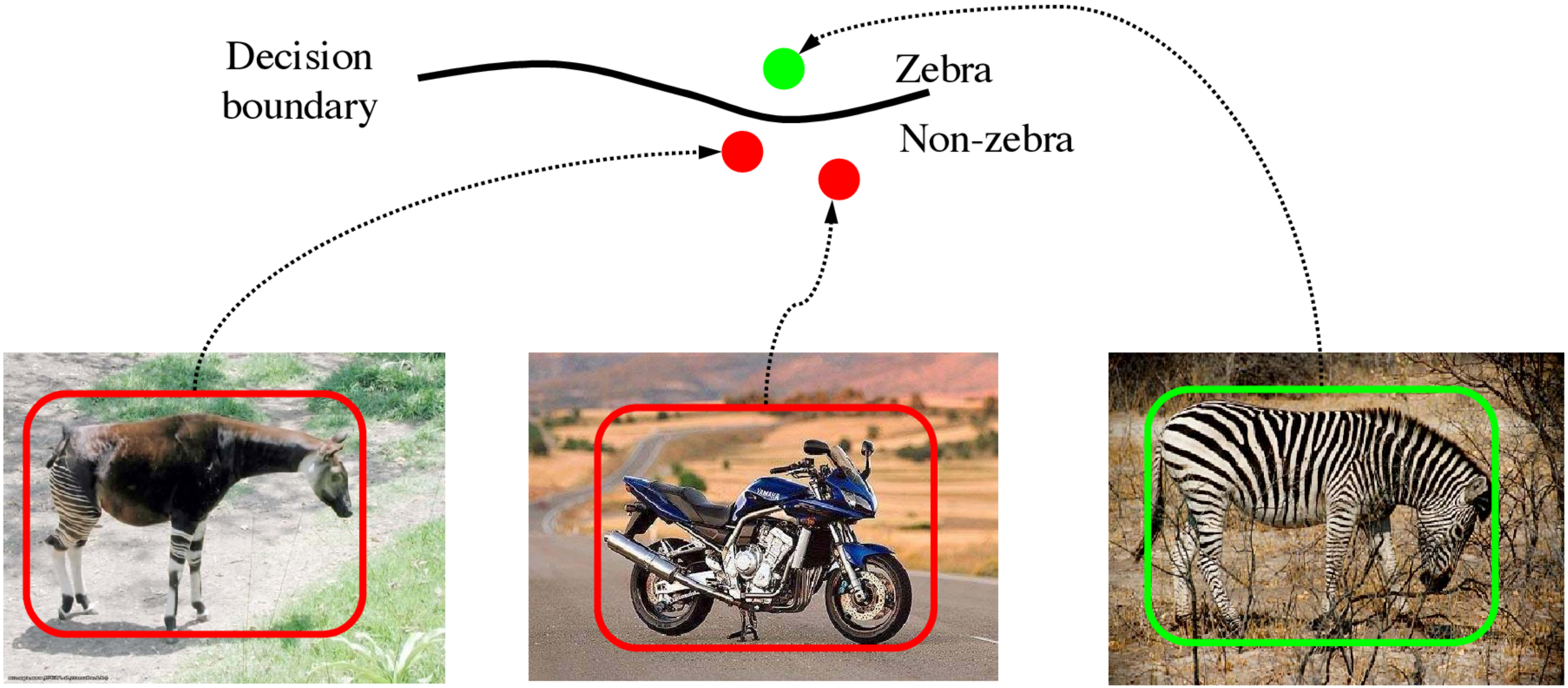
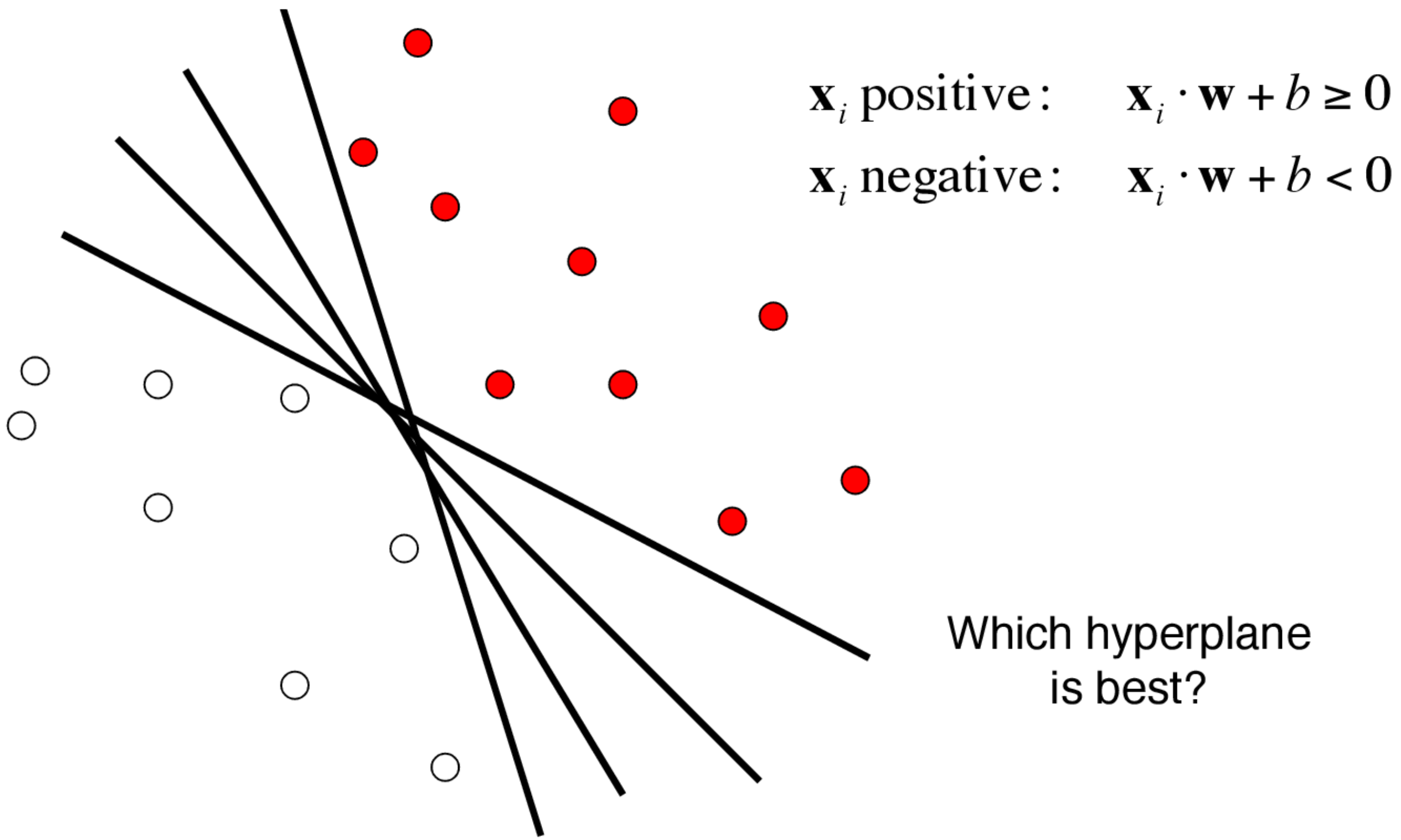


Image Classification

Goal: Learn a decision rule (classifier) to assign V to an object/class.



Linear Classification

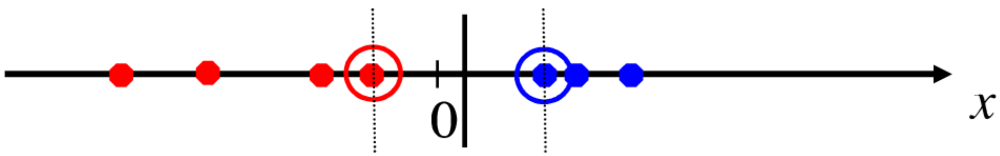


For instance: support vector machines (SVM), logistic regression, linear discriminant analysis (LDA), naive Bayes classification, ...

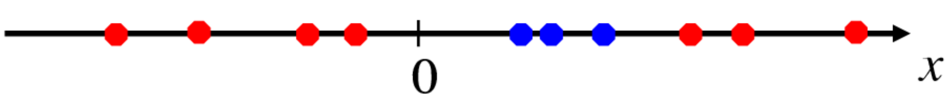
(see “Introduction to Machine Learning”)

Nonlinear Classification

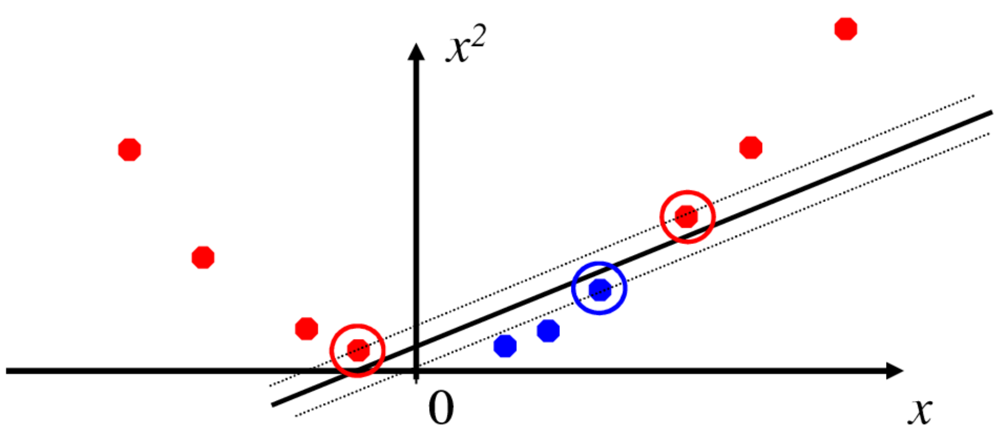
Datasets that are linearly separable work out great:



But what if the data set is just too hard?

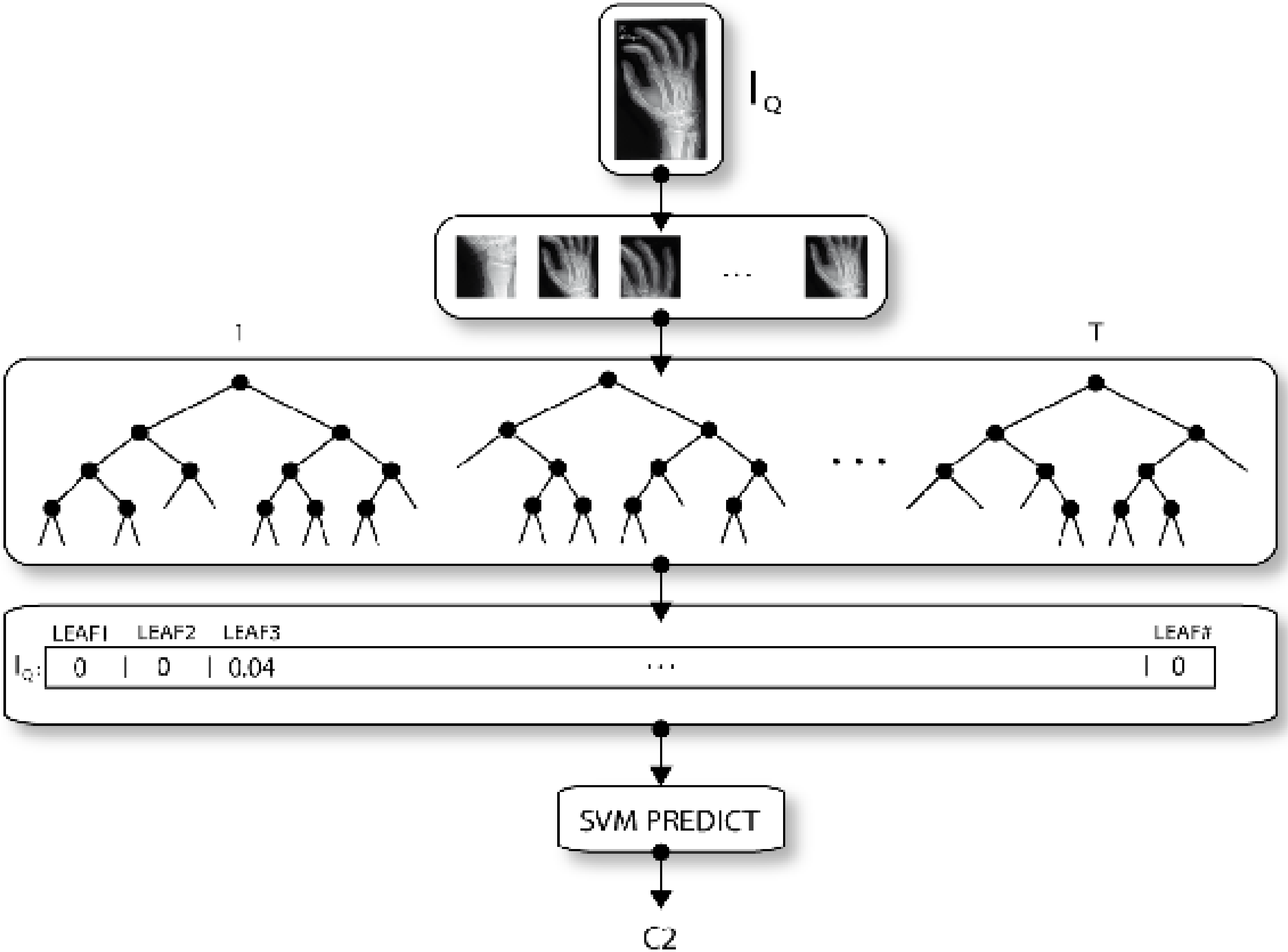


Map the data to a higher dimensional space where it is linearly separable:



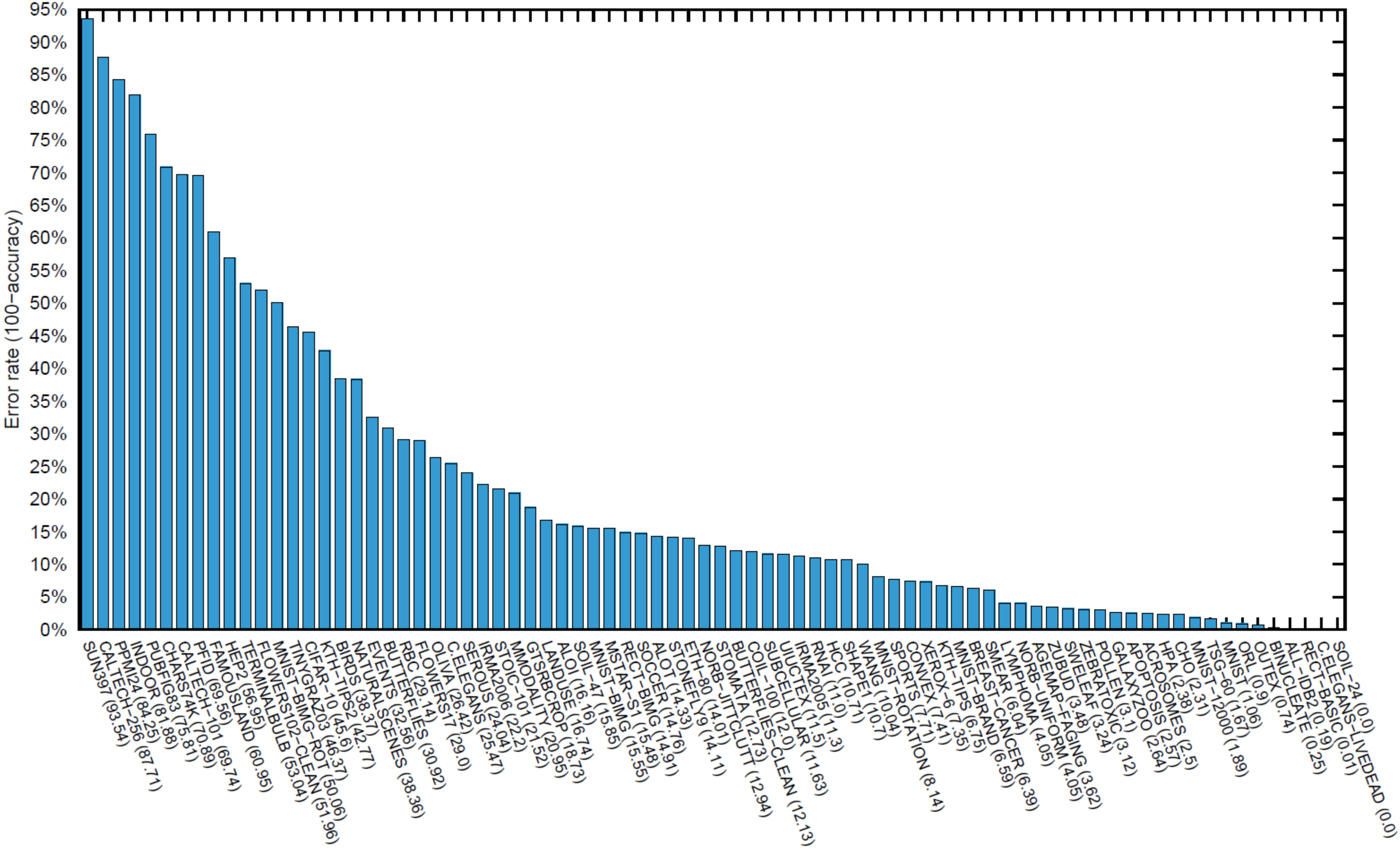
(see “Introduction to Machine Learning”)

Extra-Trees for Feature Learning : prediction



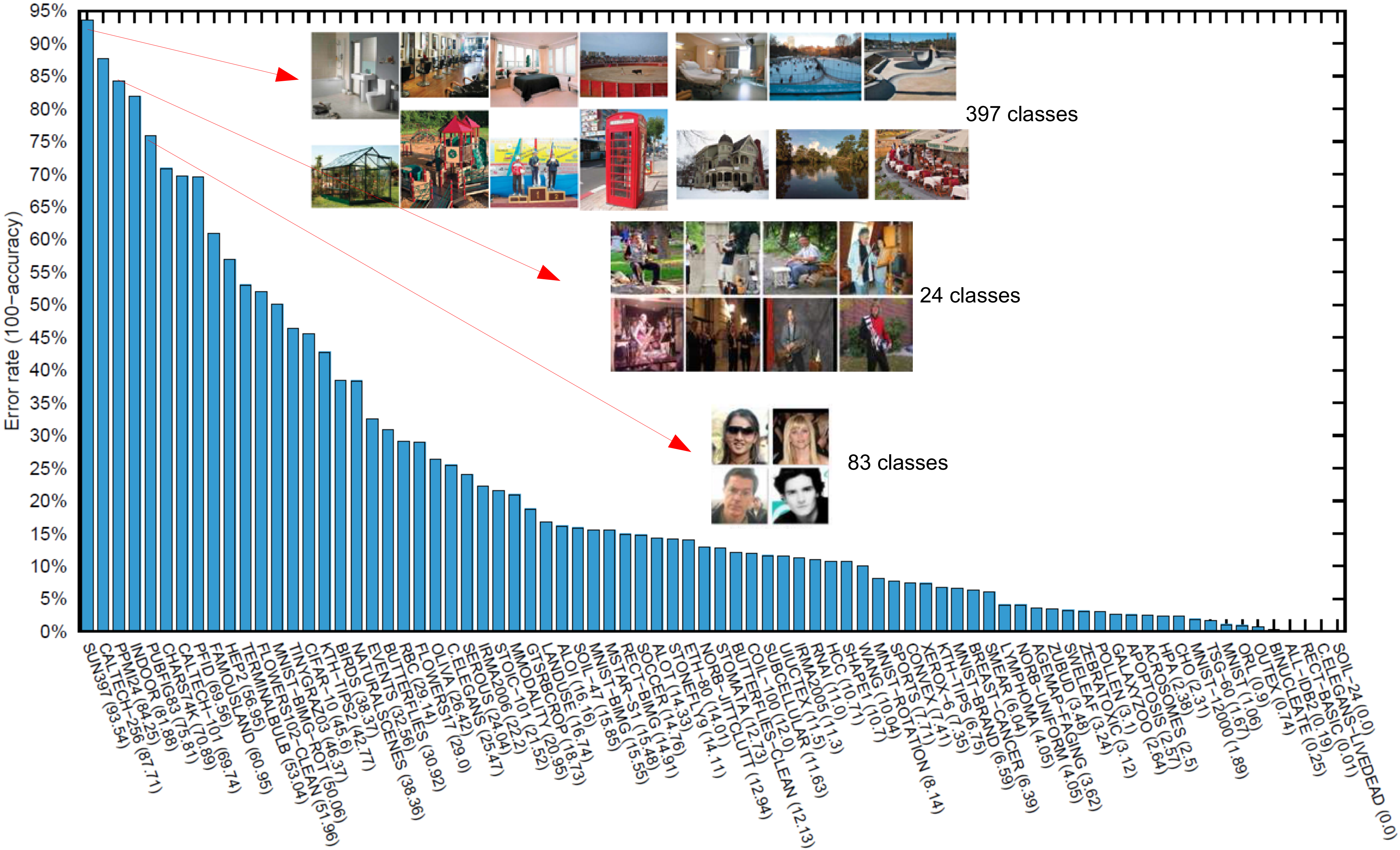
Parameters :
Nsw = nb subwindows

Overall results (error rates)



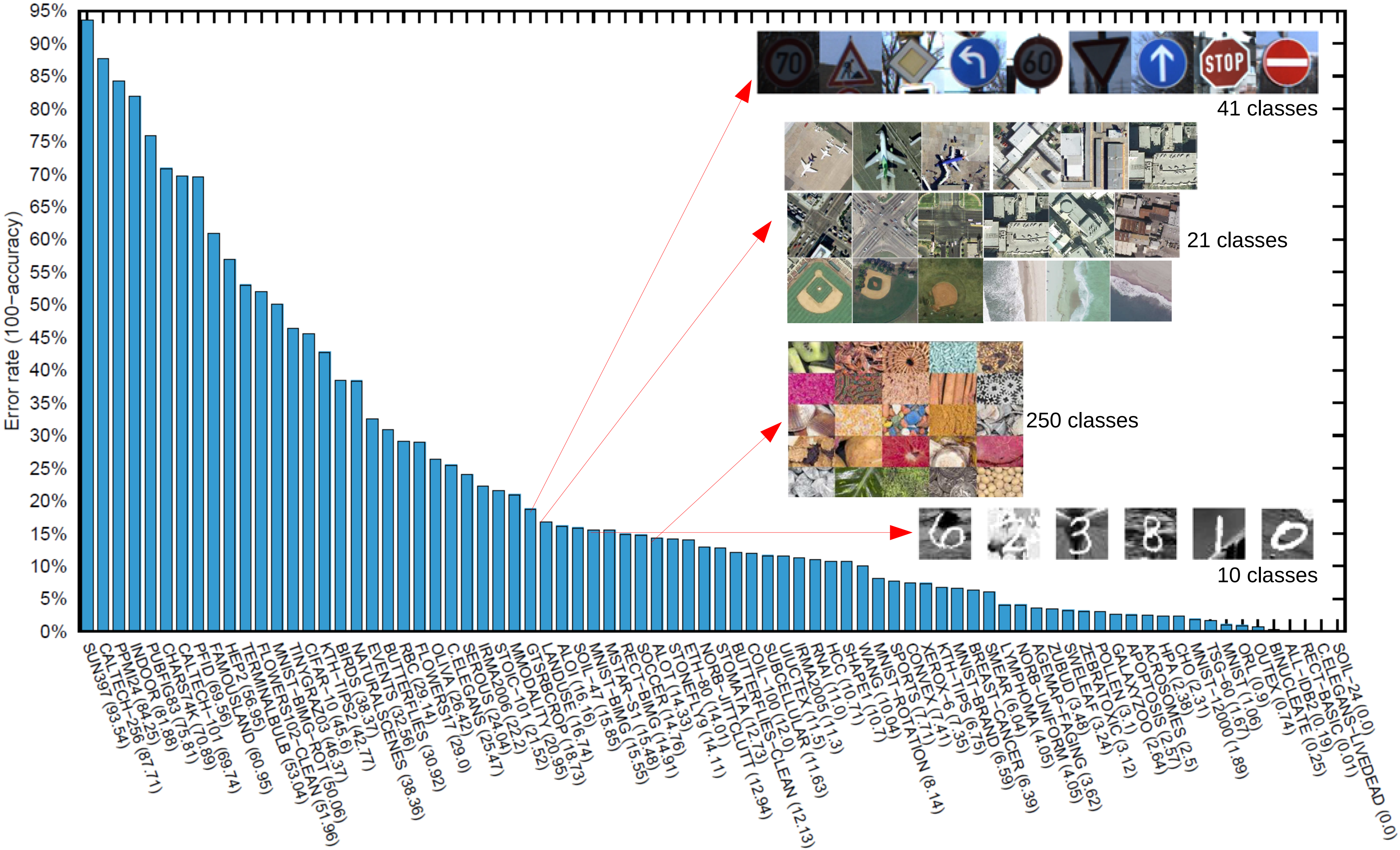
Marée, Geurts, Wehenkel, 2016

Overall results (error rates)



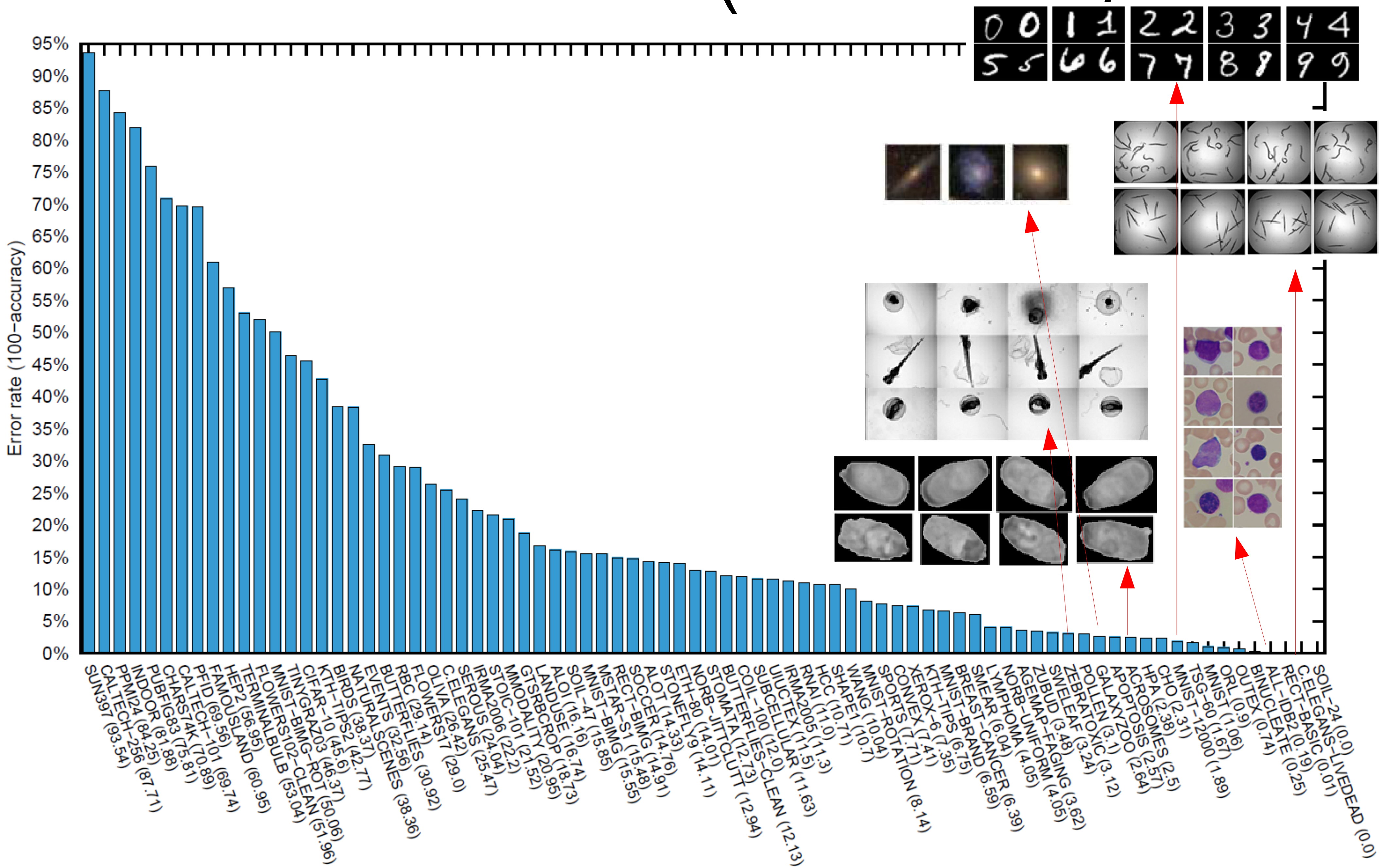
Marée, Geurts, Wehenkel, 2016

Overall results (error rates)



Marée, Geurts, Wehenkel, 2016

Overall results (error rates)



Marée, Geurts, Wehenkel, 2016

Deep Transfer learning

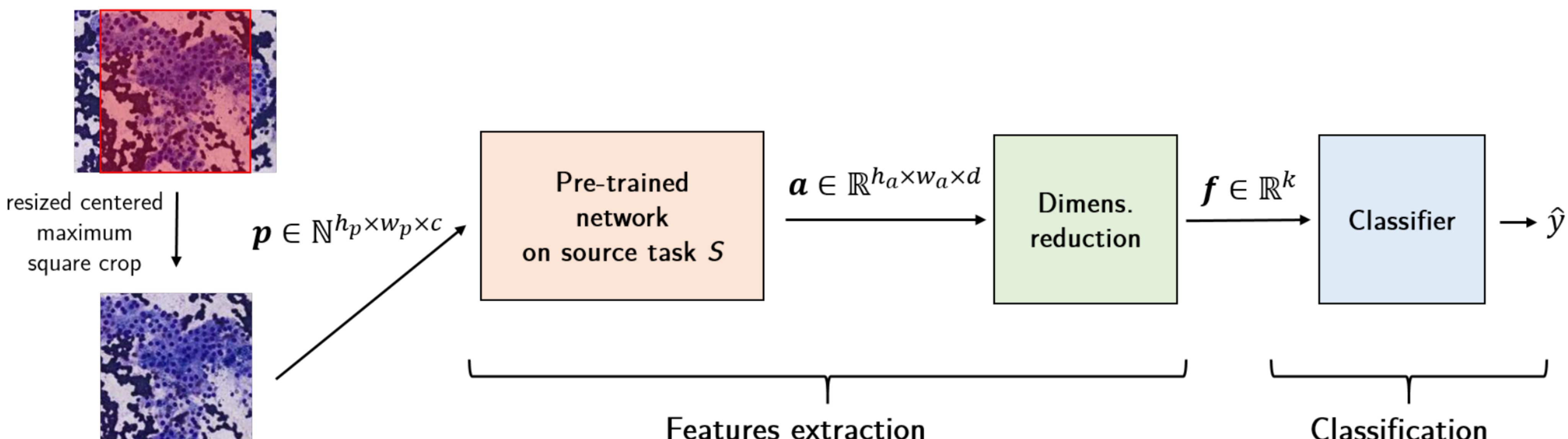
ImageNet Dataset

IMAGENET

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang et al. "Imagenet large scale visual recognition challenge." International Journal of Computer Vision 115, no. 3 (2015): 211-252. [web]

3

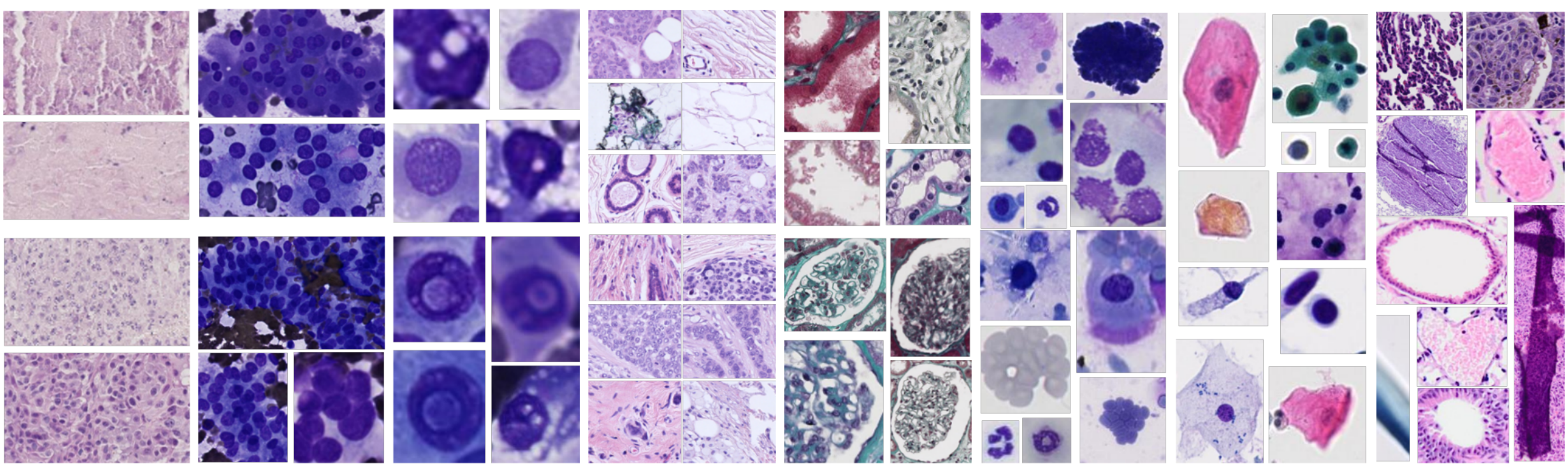
$\mathcal{N}$	**Last layer**
	# feat.
Mobile	1024
DenseNet	1920
IncResV2	1536
ResNet	2048
IncV3	2048
VGG19	512
VGG16	512



Deep features / transfer learning

Dataset	Domain	Cls	Train		Validation		Test		Total	
			Images	Slides	Images	Slides	Images	Slides	Images	Slides
Necrosis (N)	Histo	2	695	9	96	1	91	3	882	13
ProliferativePattern (P)	Cyto	2	1179	19	167	4	511	13	1857	36
CellInclusion (C)	Cyto	2	1644	21	173	2	1821	22	3638	45
MouseLba (M)	Cyto	8	1722	9	716	4	1846	7	4284	20
HumanLba (H)	Cyto	9	4051	50	346	5	1023	9	5420	64
Lung (L)	Histo	10	4881	669	562	73	888	140	6331	882
Breast (B)	Histo	2	14055	22	4206	8	4771	4	23032	34
Glomeruli (G) [25]	Histo	2	12157	91	2448	12	14608	102	29213	205

Table 1. Sizes and splits of the datasets.

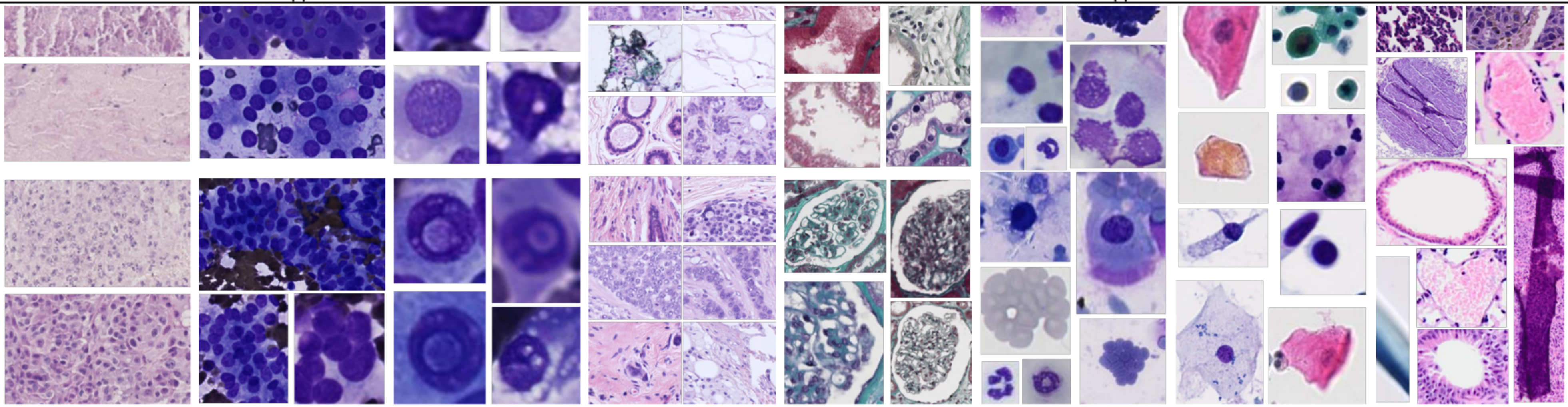


(a) Necrosis (b) ProliferativePattern (c) CellInclusion (d) Breast (e) Glomeruli (f) MouseLba (g) HumanLba (h) Lung

(Mormont et al., 2018)

Deep features / transfer learning

	Datasets							
Strategy	**C**	**P**	**G**	**N**	**B**	**M**	**L**	**H**
Baseline (ET-FL)	0.9250	0.8268	0.9551	0.9805	0.9345	0.7568	0.8547	0.6960
Last layer	0.9822	0.8893	0.9938	0.9982	0.9603	0.7996	0.9133	0.7820
Feat. select.	0.9676	0.8861	0.9843	0.9994	0.9597	0.7438	0.8941	0.7703
Merg. networks	0.9897	0.8984	0.9948	0.9864	0.9549	0.8169	0.9155	0.7928
Merg. layers	0.9808	0.8906	0.9944	0.9964	0.9639	0.7941	0.9268	0.7977
Inner ResNet	0.9748	0.8959	0.9949	0.9964	0.9664	0.8131	0.9291	0.8113
Inner DenseNet	0.9862	0.8984	0.9962	0.9917	0.9699	0.8012	0.9268	0.7967
Inner IncResV2	0.9873	0.8948	0.9962	0.9982	0.9720	0.8137	0.9234	0.7713
Fine-tuning	0.9926	0.8797	0.9977	0.9970	0.9873	0.8727	0.9405	0.8641
Metric	Roc AUC					Accuracy (multi-class)		



(a) Necrosis (b) ProliferativePattern (c) CellInclusion (d) Breast (e) Glomeruli (f) MouseLba (g) HumanLba (h) Lung

(Mormont et al., 2018)

1. Feature extraction

2. Object Recognition / Image classification
   Challenges
   Bag-of-features
   End-to-end learning
   Dataset quality control

3. Intelligent robotics / AI in Biomedecine

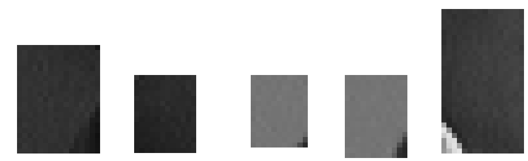
Benchmark dataset quality issues

Int J Comput Vis (2008) 79: 225–230
DOI 10.1007/s11263-008-0143-7

SHORT PAPER

Evaluation of Face Datasets as Tools for Assessing the Performance of Face Recognition Methods

Lior Shamir



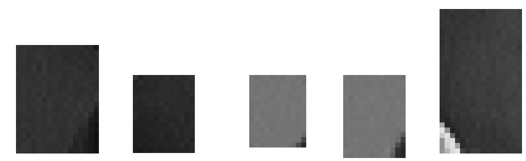
Benchmark dataset quality issues

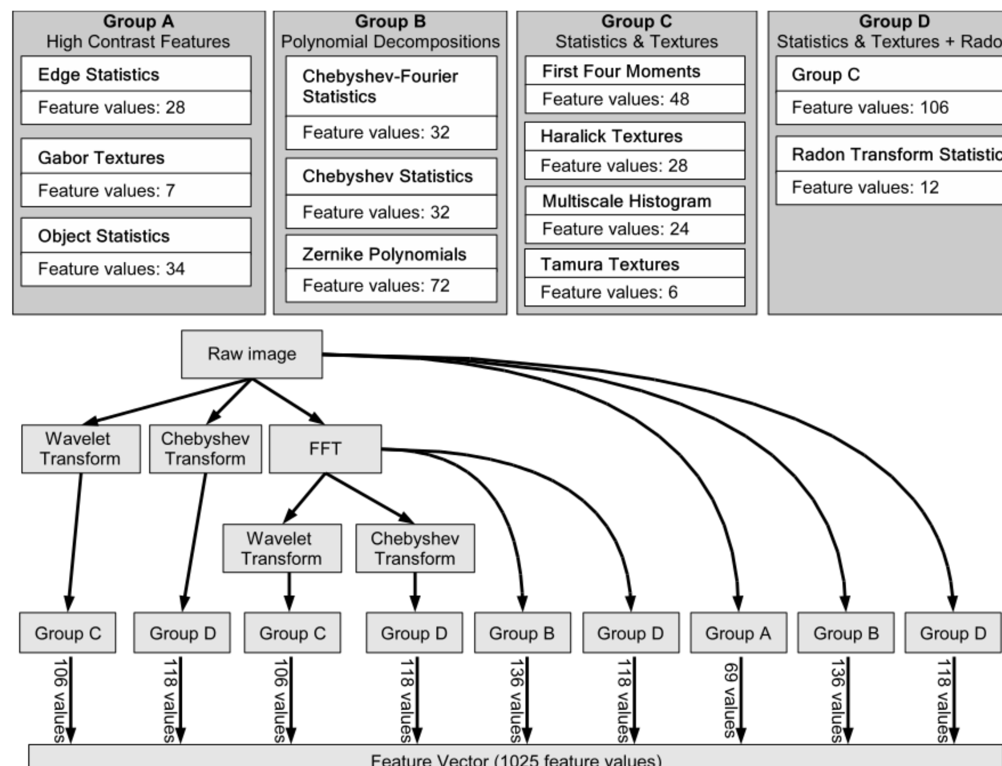
Int J Comput Vis (2008) 79: 225–230
DOI 10.1007/s11263-008-0143-7

SHORT PAPER

Evaluation of Face Datasets as Tools for Assessing the Performance of Face Recognition Methods

Lior Shamir





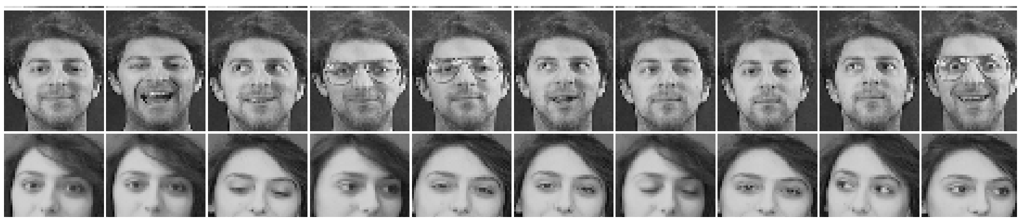
Benchmark dataset issues : hidden artefacts

Int J Comput Vis (2008) 79: 225–230
DOI 10.1007/s11263-008-0143-7

SHORT PAPER

Evaluation of Face Datasets as Tools for Assessing the Performance of Face Recognition Methods

Lior Shamir



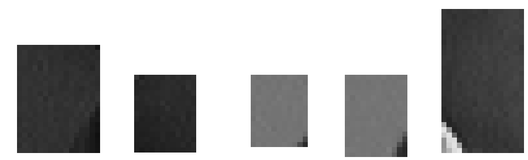


Table 1 Classification accuracy of the face datasets using a small non-facial area

Dataset	Subjects	Images per subject	Original image size	Non-facial area	Random accuracy	Non-facial accuracy
ORL	40	10	92 × 112	20 × 20 (bottom right)	0.025	0.788
JAFFE	10	22	256 × 256	25 × 200 (top left)	0.1	0.94
Indian Face Dataset (Females)	22	11	160 × 120	42 × 80 (top left)	0.045	0.73
Indian Face Dataset (Males)	39	11	160 × 120	42 × 80 (top left)	0.0256	0.58
Essex	100	20	196 × 196	42 × 100 (top left)	0.01	0.97
Yale B	10	576	640 × 480	100 × 300 (top left)	0.1	0.99
Color FERET	994	5	512 × 768	100 × 100 (top left)	$\sim$ 0.001	0.135

Benchmark dataset issues : hidden artefacts

Journal of Microscopy

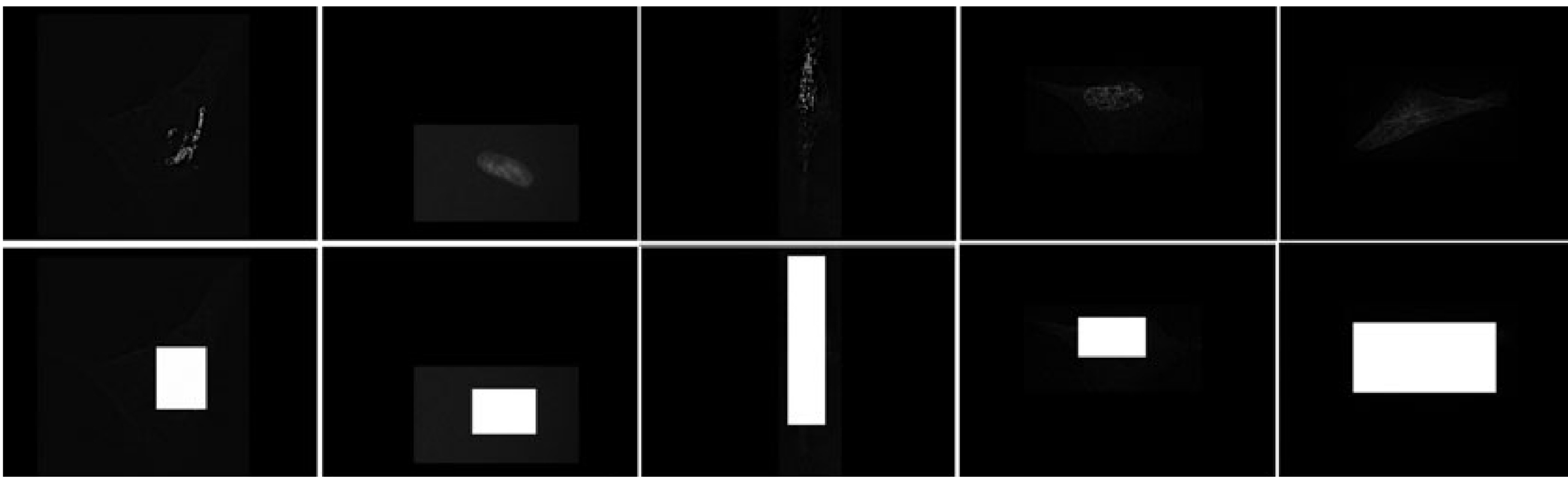
Journal of Microscopy, Vol. 243, Pt 3 2011, pp. 284–292 doi: 10.1111/j.1365-2818.2011.03502.x

Received 11 January 2011; accepted 17 March 2011

Assessing the efficacy of low-level image content descriptors for computer-based fluorescence microscopy image analysis

L. SHAMIR

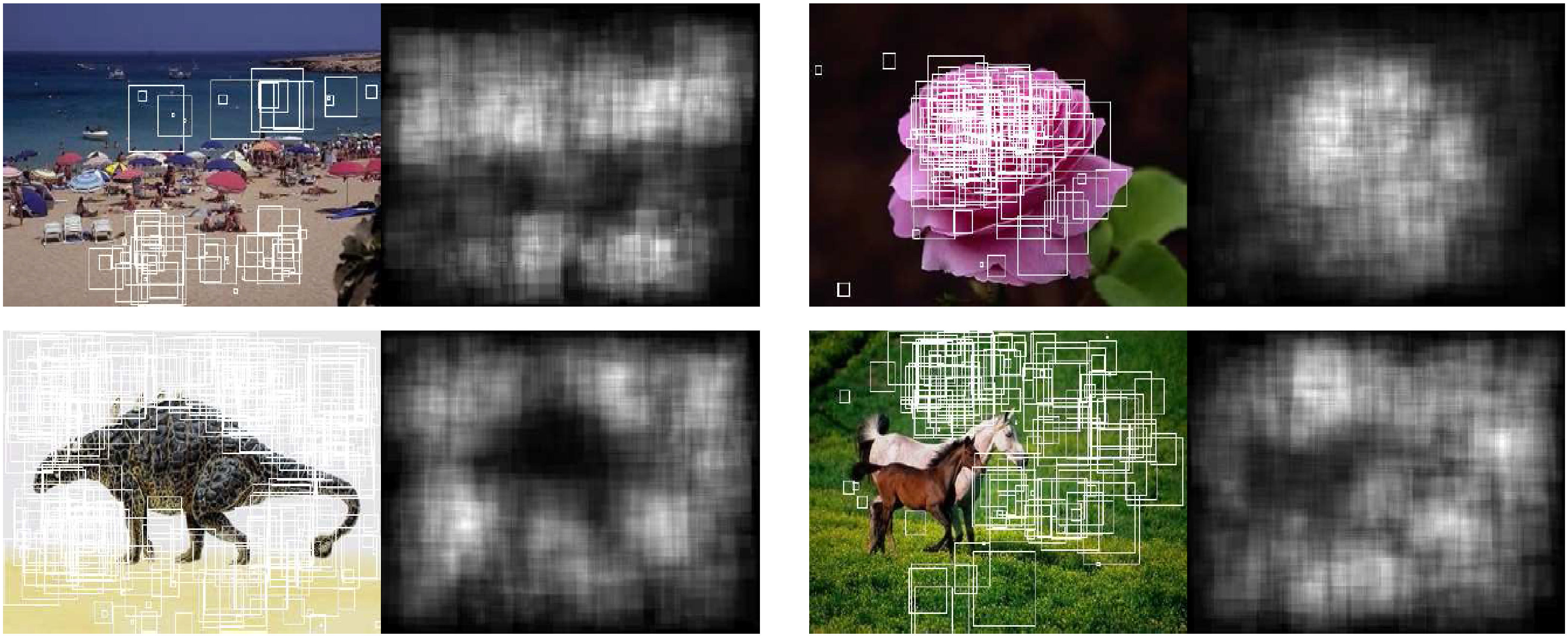
Department of Computer Science, Lawrence Technological University, Southfield, Michigan, U.S.A.



→ 88 % recognition rate using images without protein patterns !

Benchmark dataset issues : hidden artefacts

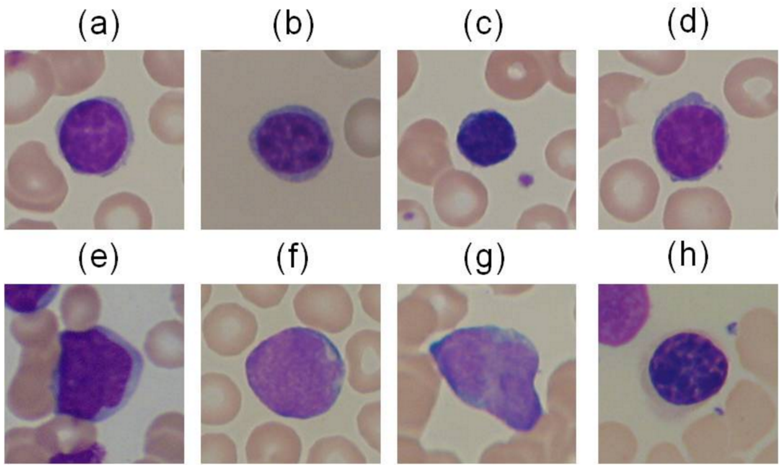
WANG dataset (PAMI, 2001) : 10 categories (beach, dinosaur, flower, horse, food, city, ...)



→ 44 % recognition rate using only 50x50 background data… OK ?

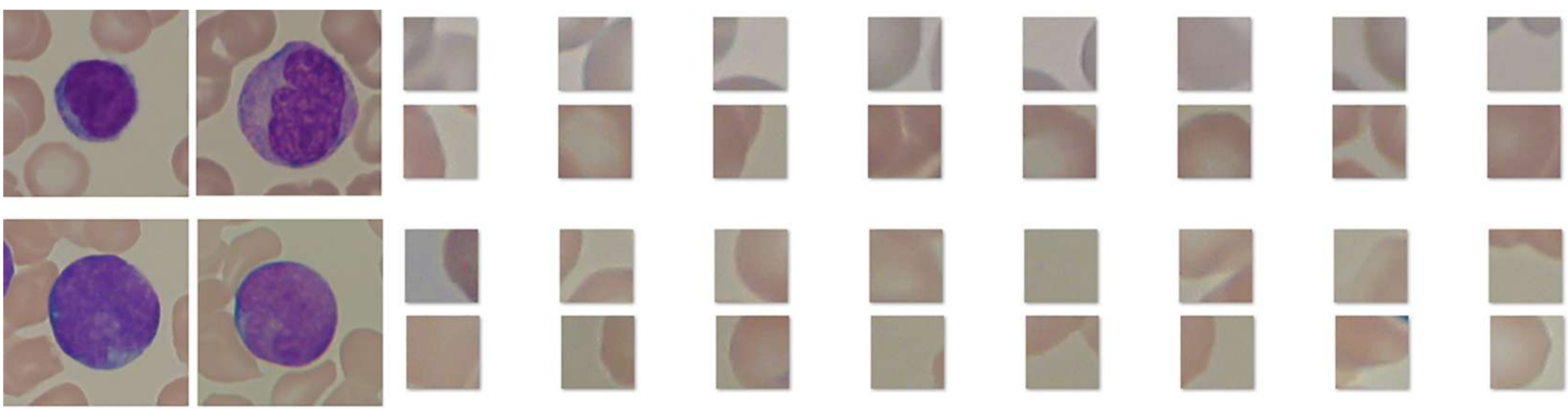
NO ! Two classes (dinosaurs & horses) are almost perfectly recognized using background only !

Benchmark dataset issues : hidden artefacts



Examples of the images contained in ALL-IDB2: healthy cells from non-ALL patients (a-d), probable lymphoblasts from ALL patients (e-h).

ALL-IDB: the acute lymphoblastic leukemia image database for image processing, Proc. IEEE Int. Conf. on Image Processing (ICIP 2011).



→ 90 % recognition rate using only 50x50 background regions !

Summary

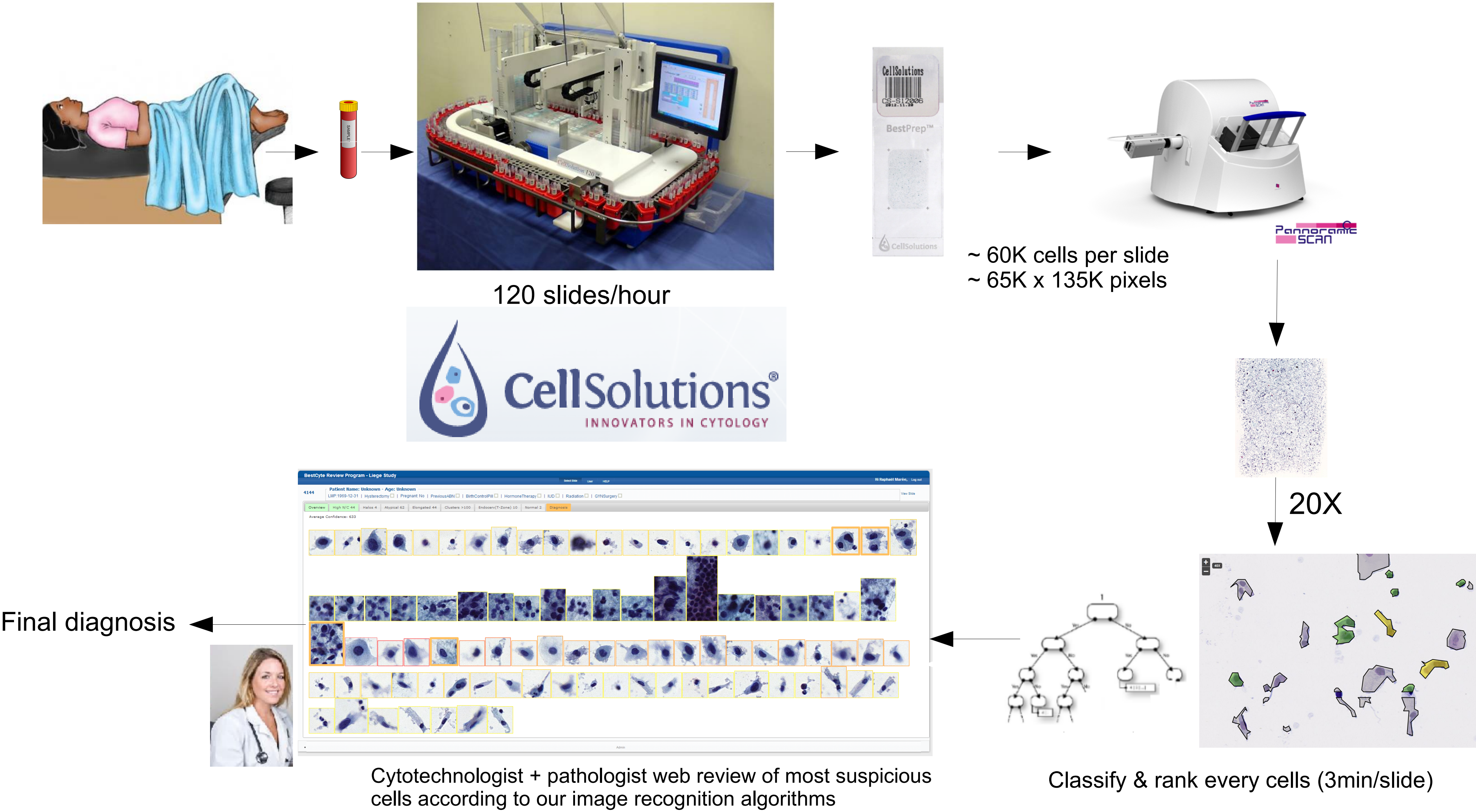
- Many features have been designed to ease vision tasks
- Many learning approaches have been designed
- Dataset collection should be controlled
- Several (controlled) vision tasks can be solved with end-to-end learning / deep transfer learning but it requires tuning and accuracy is still not high enough

1. Feature extraction

2. Object Recognition / Image classification
   Challenges
   Bag-of-features
   End-to-end learning
   Quality control

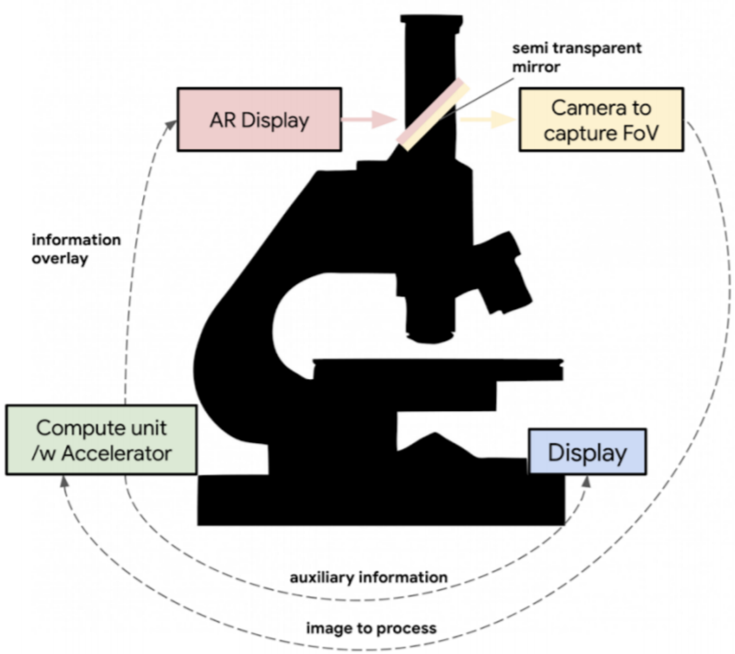
3. Intelligent robotics / AI in Biomedecine

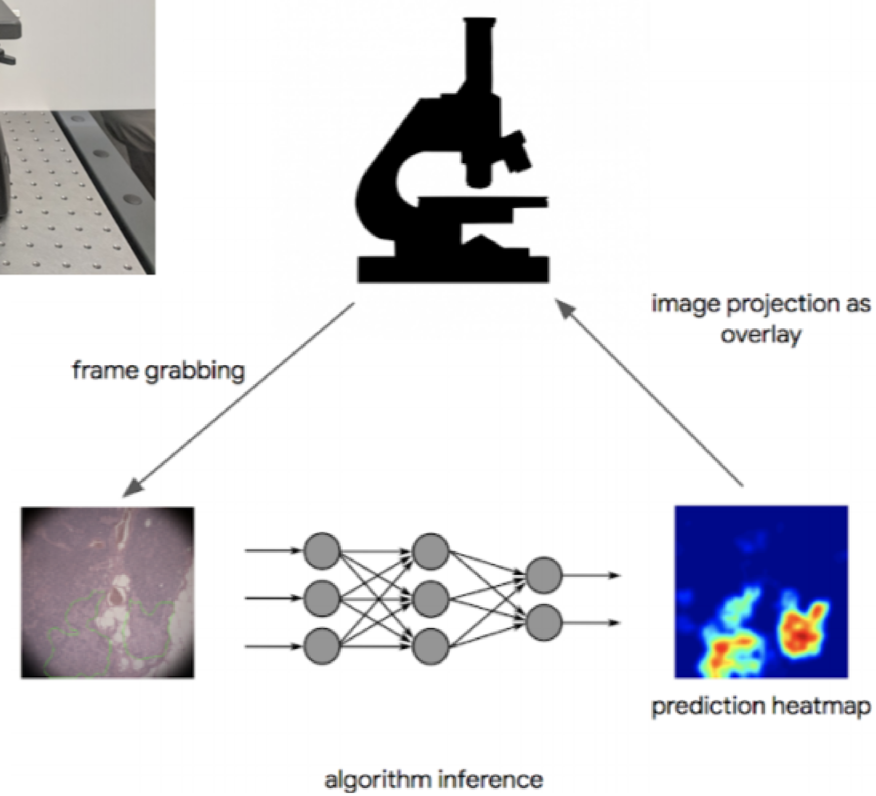
Cervical Cancer screening : hybrid workflow



Evaluation of CellSolutions BestPrep Automated Thin-Layer Liquid-Based Cytology Papanicolaou Slide Preparation and BestCyte Cell Sorter Imaging System, Delga et al. , Acta Cytologica, 2014;58(5):469-77

An Augmented Reality Microscope for Realtime Automated Detection of Cancer





(Chen et al., Google, 2018)

CARE : Content-Aware Image Restoration: Pushing the Limits of Fluorescence Microscopy

Problem : Imaging spatial/temporal trade-offs : too much laser power or exposure time is usually detrimental to the sample

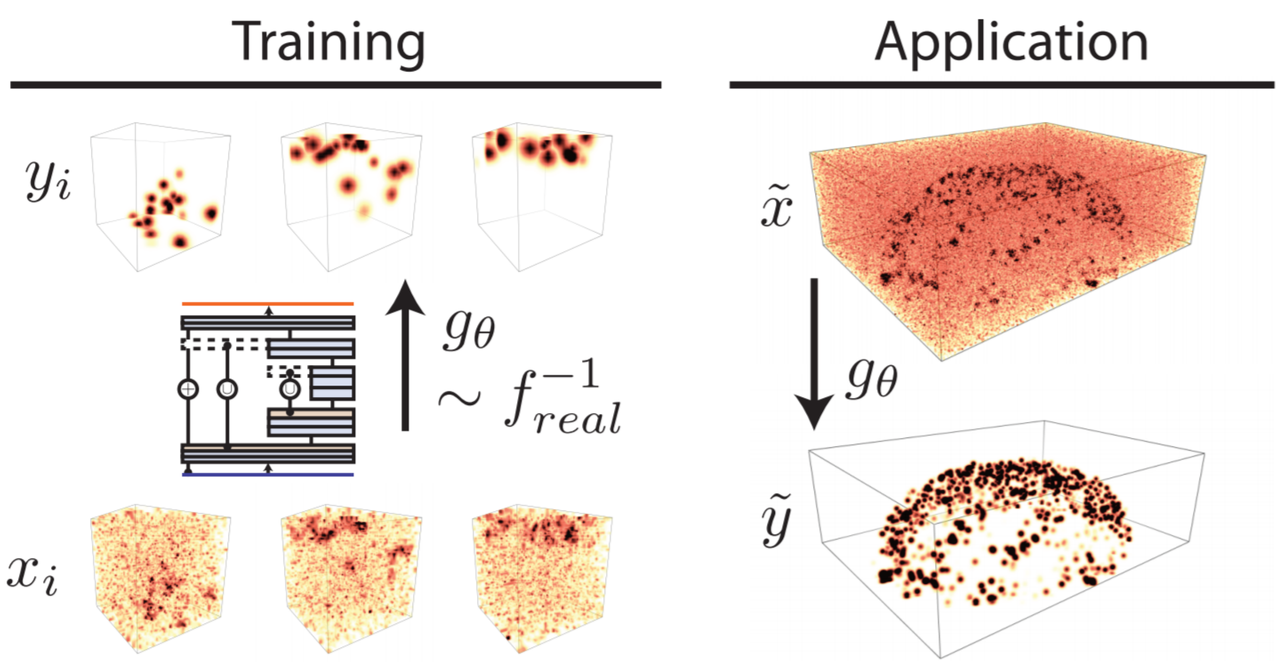
Solution :
Acquisition of well-registered pairs of images (fixed samples):

- *A low-SNR image at a laser power compatible with live imaging*
- *A high-SNR image serving as ground-truth*

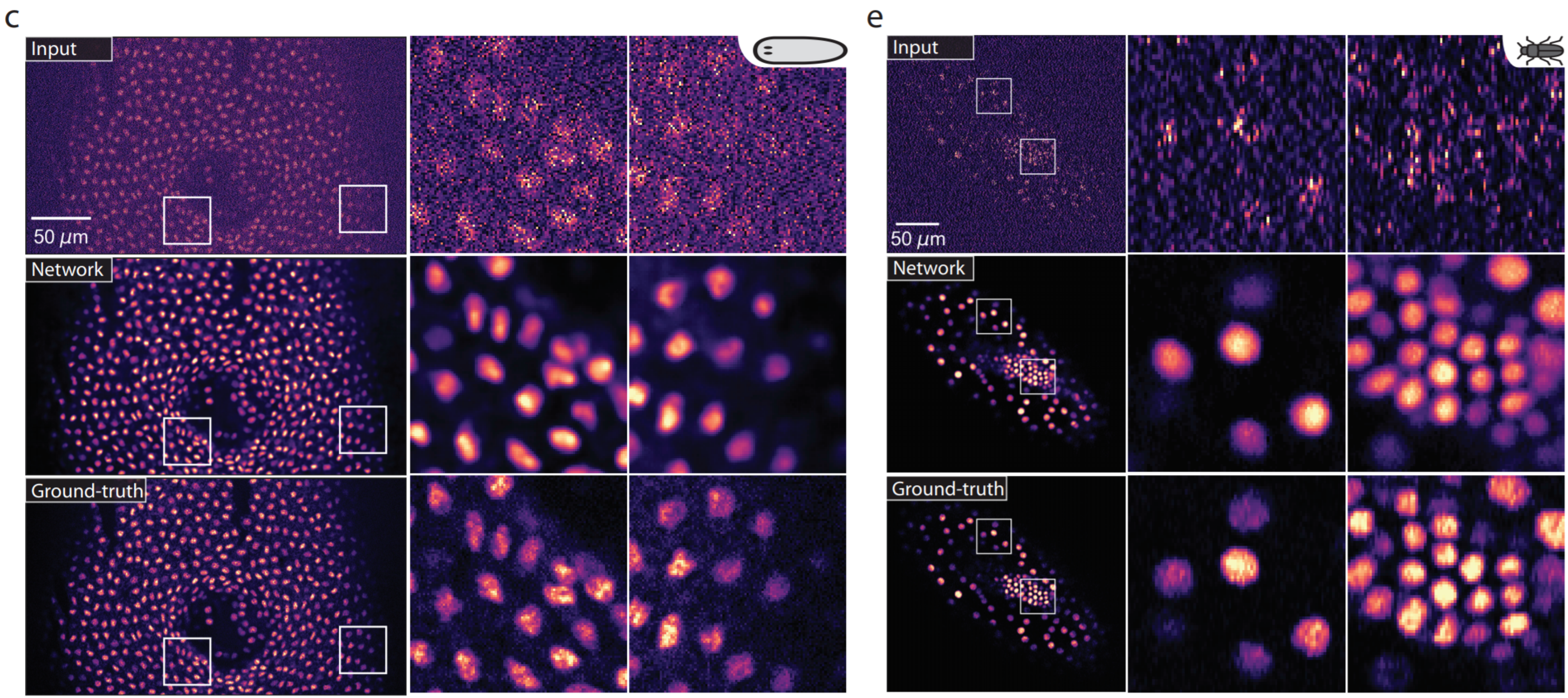
→ *Train CARE networks (residual version of a U-Net type topology), and apply the trained networks to remove noise in previously unseen live data.*

.

(Weigert et al., 2017)



CARE : Content-Aware Image Restoration: Pushing the Limits of Fluorescence Microscopy



Restoration 1024×1024×100 < 20 seconds (single GPU)

(Weigert et al., 2017)

Intelligent high content imaging

High content imaging at high resolution (possibly multispectral) can quickly generate an overwhelming amount of data and require a prohibitive acquisition time.

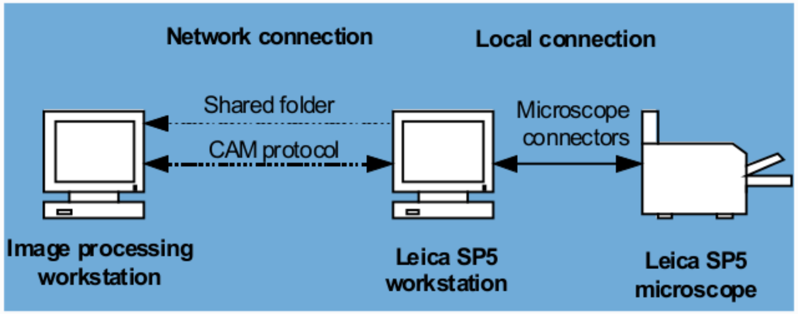


Figure 1. Hardware setup.

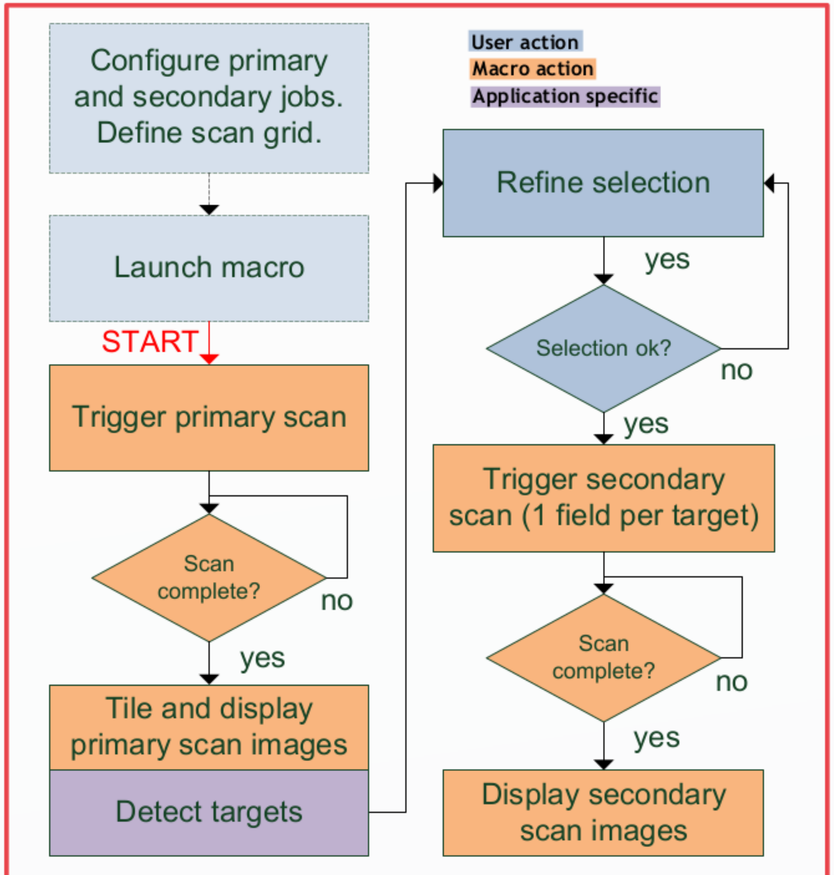


Figure 2a.

2-step acquisition

(Tosi et al., 2018)

Intelligent high content imaging

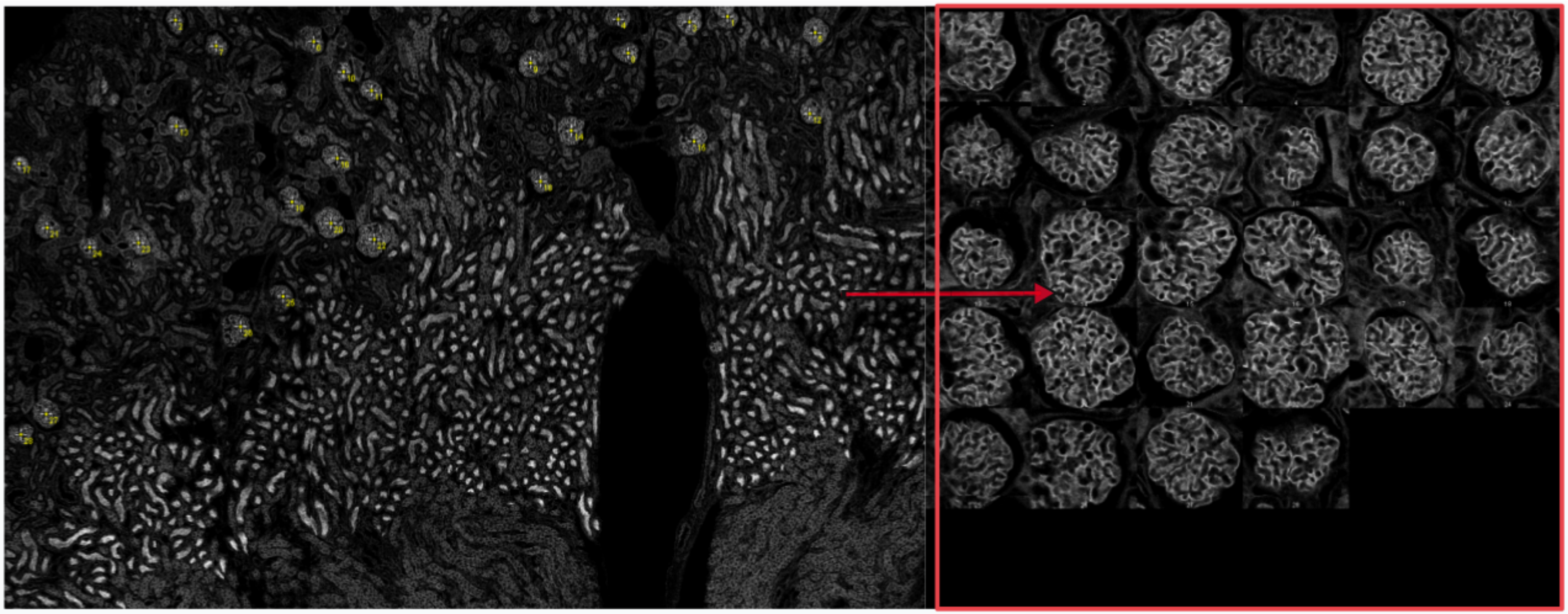


Figure 3. Primary scan of the GFP channel of a stained mouse kidney slice (left), imaging conditions: 2x3 images grid, 20x lens, zoom=1. The detected targets are marked with a yellow cross. Preview montage of the selected targets (right): the user can select or deselect the targets by clicking them on the montage.

(Tosi et al., 2018)

Intelligent high content imaging

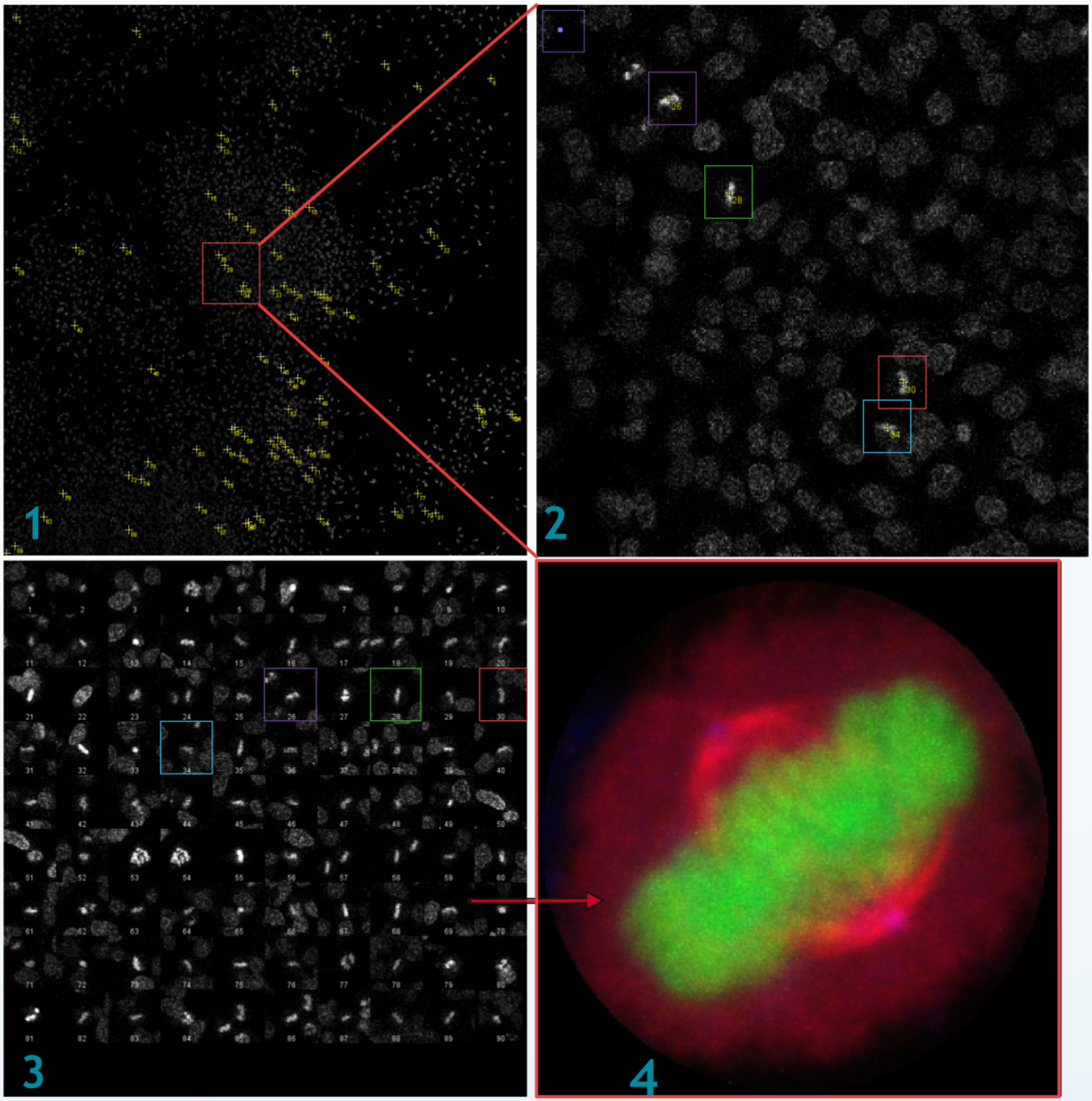
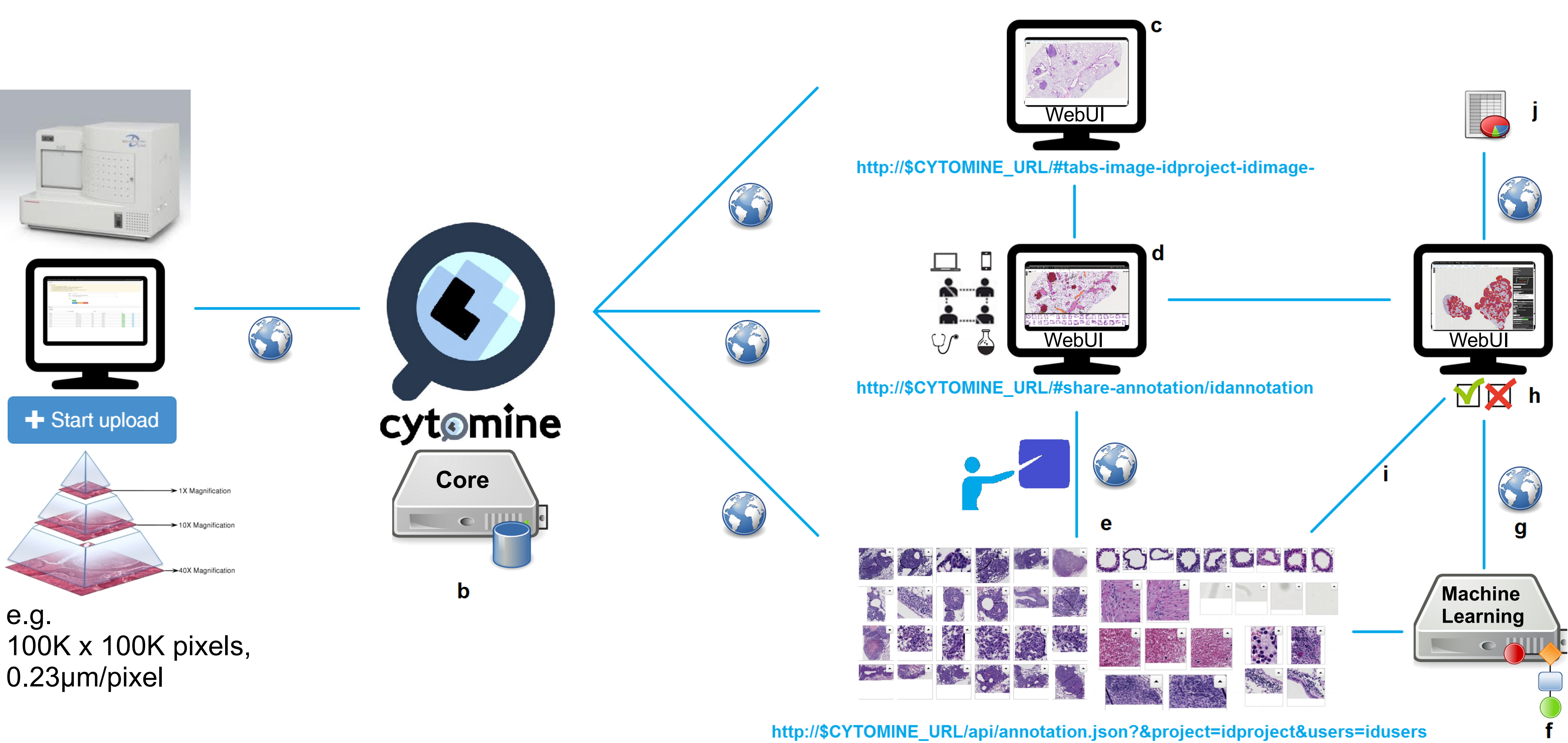


Figure 4. Primary scan of the DAPI channel of cultured cells (1), imaging conditions: 10 x 10 images grid, 63x lens, zoom=1, 256x256 pixels, 3 z planes MIP. Detected mitotic cells are marked by a yellow cross. Zoomed in area (2). Preview of the detected targets and their surrounding (3). Maximum intensity projection of a high content stack of a detected mitotic cell acquired during the secondary scan (4).

(Tosi et al., 2018)

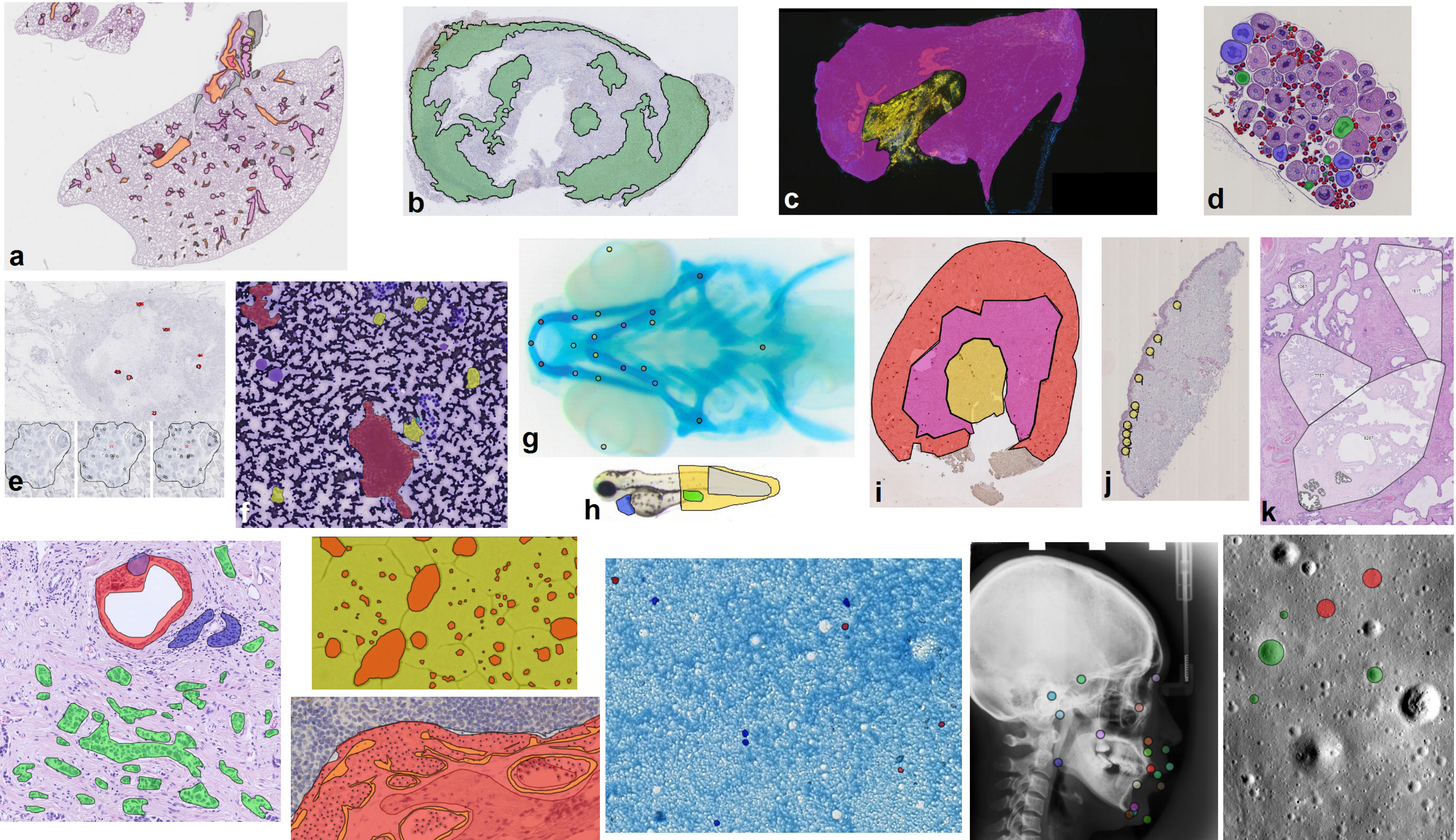
cytomine enables collaboration through the web



(Marée et al. Bioinformatics, 2016)

cytomine is versatile and scalable

Applications in research and education : > 5000 users, > 50 000 images, > 1M annotations



LIÈGE université
UNIVERSITÉ DE NAMUR
UPMC SORBONNE UNIVERSITÉS
Institut Pasteur
Medizinische Universität Graz
Telemis
iSTAR Medical
Next generation ophthalmic implants
UNIVERSITÉ DE STRASBOURG
OHSU

References

Marc Van Droogenbroeck, Olivier Barnich. Design of Statistical Measures for the Assessment of Image Segmentation Schemes, CAIP, 2005

Sezgin, Sankur. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging 13(1), 146 – 165 (January 2004).

Salahat, Qaseimeh. Recent Advances in Features Extraction and Description Algorithms: A Comprehensive Survey

T. Tuytelaars, K. Mikolajczyk, et al., “Local invariant feature detectors: a survey,” Foundations and trends in computer graphics and vision, vol. 3, no. 3, pp. 177–280, 2008.

Y. Li, S. Wang, Q. Tian, and X. Ding, “A survey of recent advances in visual feature detection,” Neurocomputing, vol. 149, pp. 736–751, 2015.

Scott Krig. Computer Vision Metrics, Springer 2016.