

COURS RDFIA deep Image

<https://cord.isir.upmc.fr/teaching-rdfia/>

Matthieu Cord
Sorbonne University

Course Outline

1. Computer Vision and ML basics: visual (local) feature detection and description, Bag of Word Image representation, Linear classification (SVM)
2. Introduction to Neural Networks (NNs)
3. Machine Learning basics (2): Risk, Classification, Datasets, benchmarks and evaluation
4. Neural Nets for Image Classification
5. Vision Transformers
6. Transfer learning and domain adaptation
7. Segmentation and Detection
8. Generative models with GANs
9. Generative models with diffusion
10. Large VL models: CLIP, StableDiffusion, Flamingo
11. Control (to be checked) -- Explainable AI, Fairness
- 12/13 Bayesian deep learning
- 14 Robustness



Info about practicals

Course 1
Visual Representation of images [Bag of Features and Bag of Words](#)

Course 2
[Supervised Learning: Neural Net architectures](#)

Course 3
[Supervised Learning: theory and practices](#)
[Supervised Learning: SVM algorithm](#)

Weakly updated

Course 4
[Supervised Learning: Dataset evaluation and Extra on BoW](#)
[Neural Nets for Image Classification](#)

Course 5
[Large scale convolutional neural nets](#)

Course 6
[VERY Large scale convolutional neural nets and Beyond ImageNet](#)

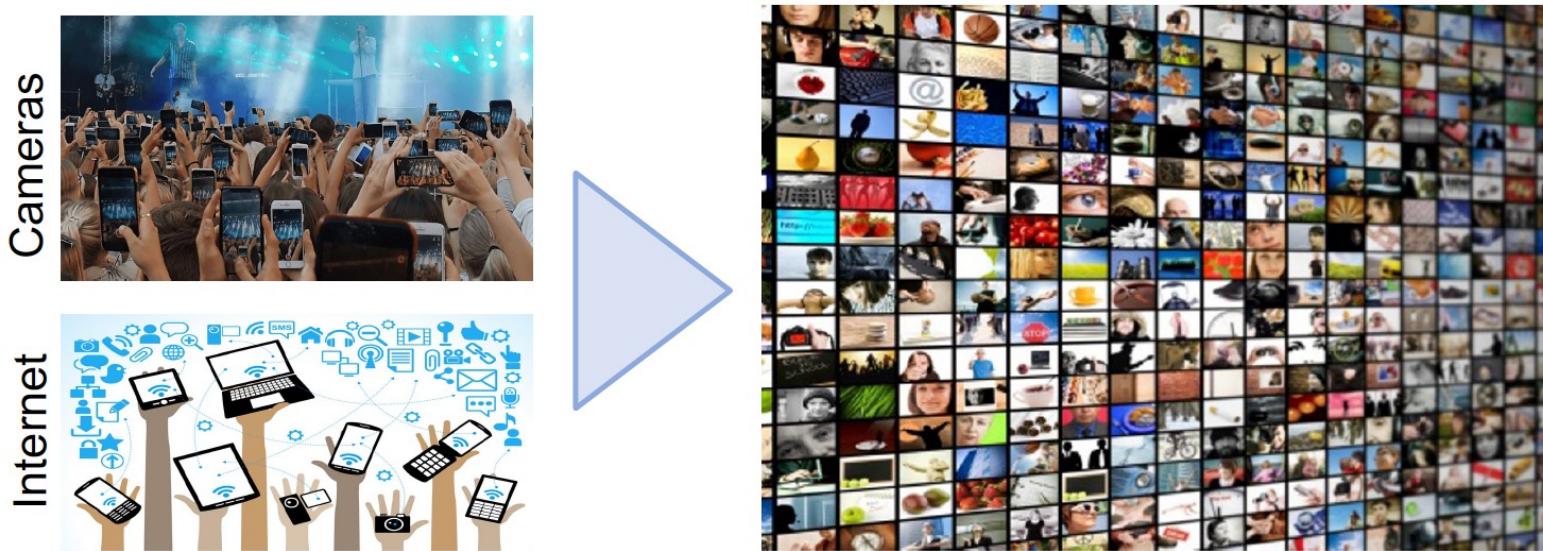
Course 7 [Transformers for Images](#)

Course 8
[Visual Transfer Learning: transfer and domain adaptation](#)

Course 9
[Generative models for Vision – GAN \(1\)](#)

Course 10
[GAN \(2\)++](#)

Evaluations: Control (30%) + Practicals (3 reports; 70%)
can be modified by 10% between the 2 evaluations



- **Facts:** Exponential increase in quantity of images/videos taken across the world
 - YouTube: 500h of video / min
 - Facebook: 300M photos / day

COMPUTER VISION:

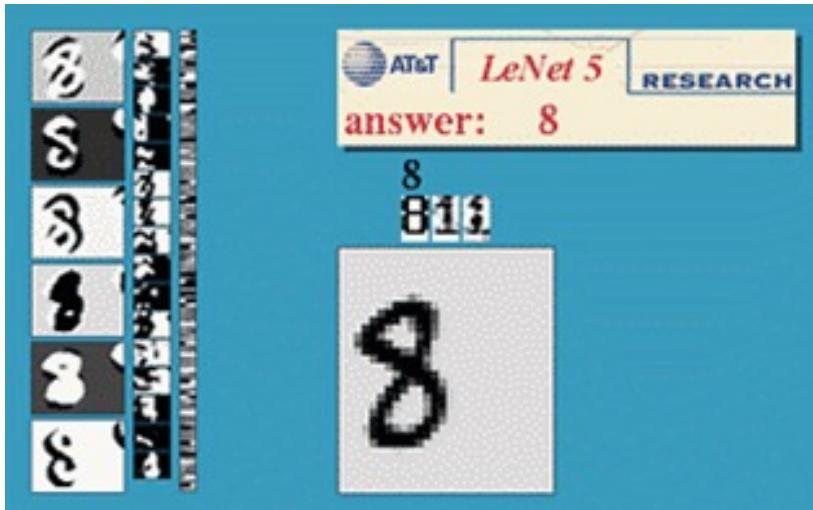
(Processing, analyzing and) **understanding visual data**

=>**WHERE ARE WE NOW?**

Source (many slides): Cornell CV course

Deployed: Optical character recognition (OCR)

- If you have a scanner, it probably came with OCR software



Digit recognition, AT&T labs
<http://www.research.att.com/~yann/>

LYCH428

LYCH428

LYCH428

License plate readers

http://en.wikipedia.org/wiki/Automatic_number_plate_recognition



Automatic check processing

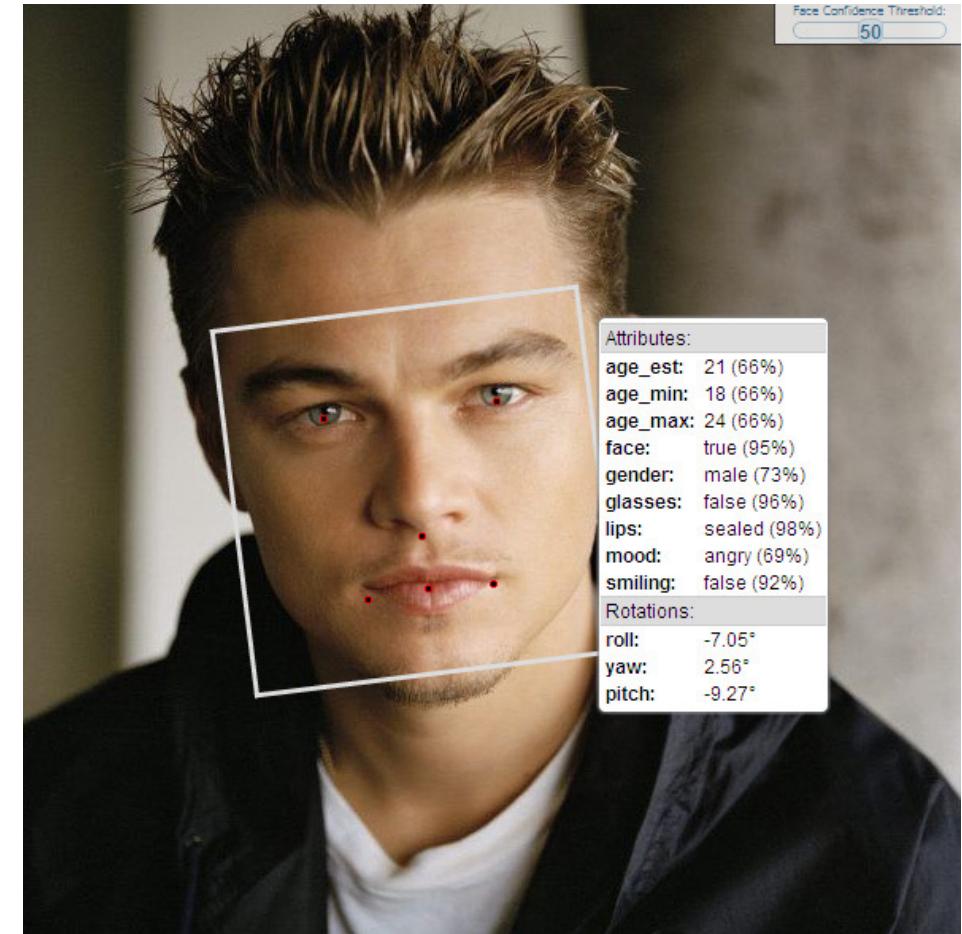
Source: S. Seitz

Deployed: Face detection



- Cameras now detect faces
 - Canon, Sony, Fuji, ...

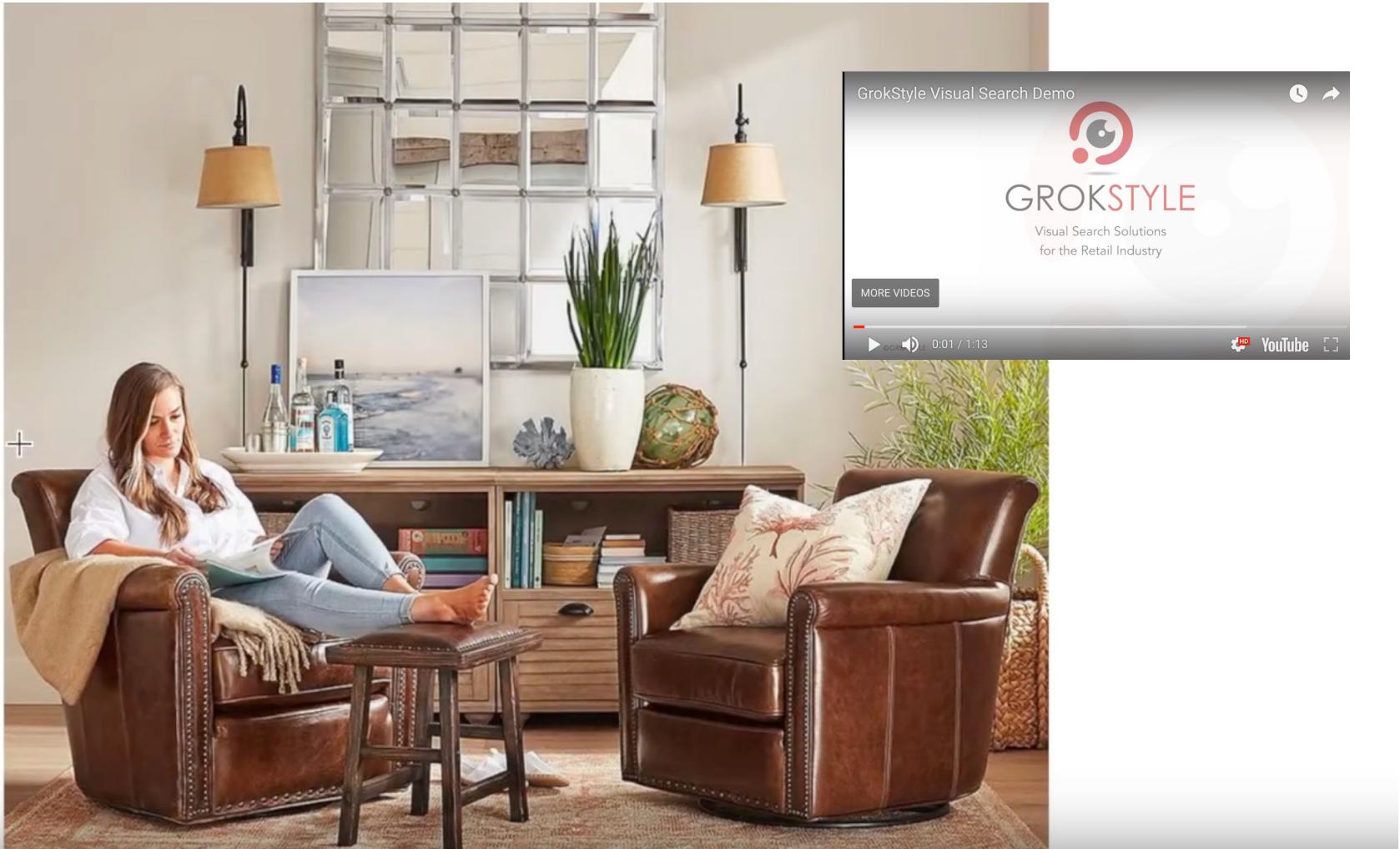
Deployed&Significant progress: Face Recognition



Significant progress: Recognizing objects



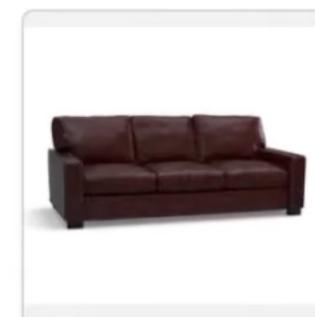
Ex: Recognition-based product search



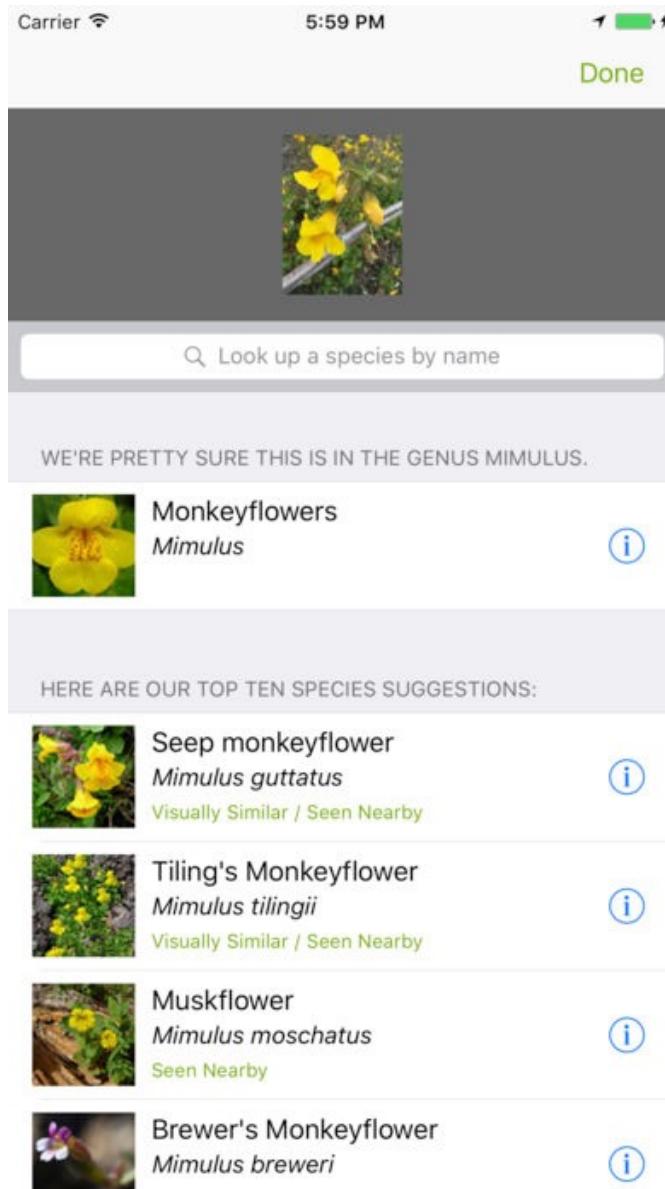
Recognition-based product search



Recognition-based product search



Significant progress: Species recognition



iNaturalist dataset

Challenges:

- fine-grained recognition
- Detecting rare concepts

Challenges: Fully autonomous driving



Challenges: Medical Imaging, Health

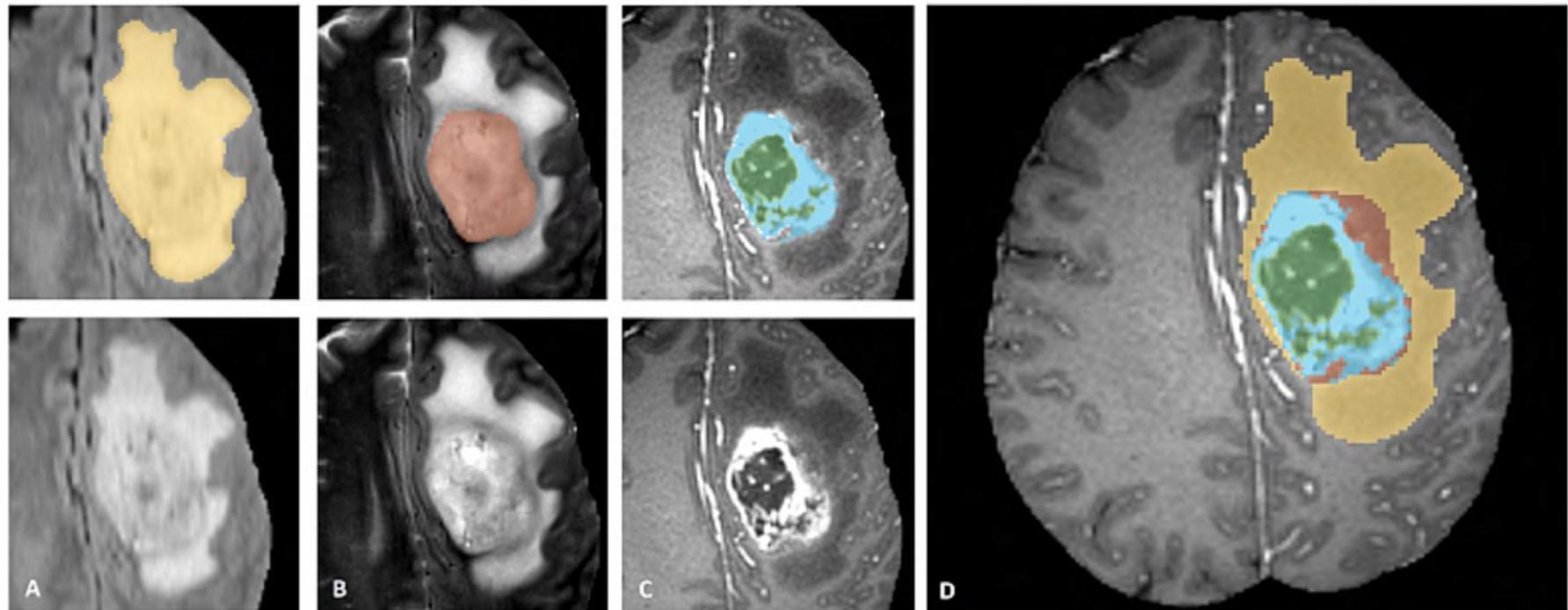


Fig.1: Glioma sub-regions. Shown are image patches with the tumor sub-regions that are annotated in the different modalities (top left) and the final labels for the whole dataset (right). The image patches show from left to right: the whole tumor (yellow) visible in T2-FLAIR (Fig.A), the tumor core (red) visible in T2 (Fig.B), the enhancing tumor structures (light blue) visible in T1Gd, surrounding the cystic/necrotic components of the core (green) (Fig. C). The segmentations are combined to generate the final labels of the tumor sub-regions (Fig.D): edema (yellow), non-enhancing solid core (red), necrotic/cystic core (green), enhancing core (blue). (Figure taken from the [BraTS IEEE TMI paper](#).)

Challenges: Medical Imaging, Health

Building system to detect Covid in chest x rays

What should a metric measure?

Accuracy = $P(\text{pred. label} == \text{true label})$

Accuracy of candidate system = 95%

Is this good? Did it actually help / work?

Artificial intelligence / Machine learning

Hundreds of AI tools have been built to catch covid. None of them helped.

Some have been used in hospitals, despite not being properly tested. But the pandemic could help make medical AI better.



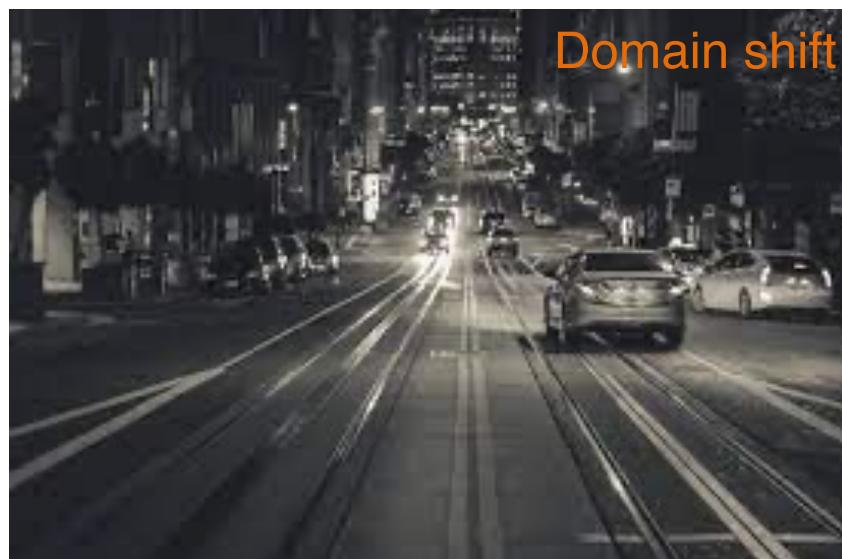
Why?

by Will Douglas Heaven

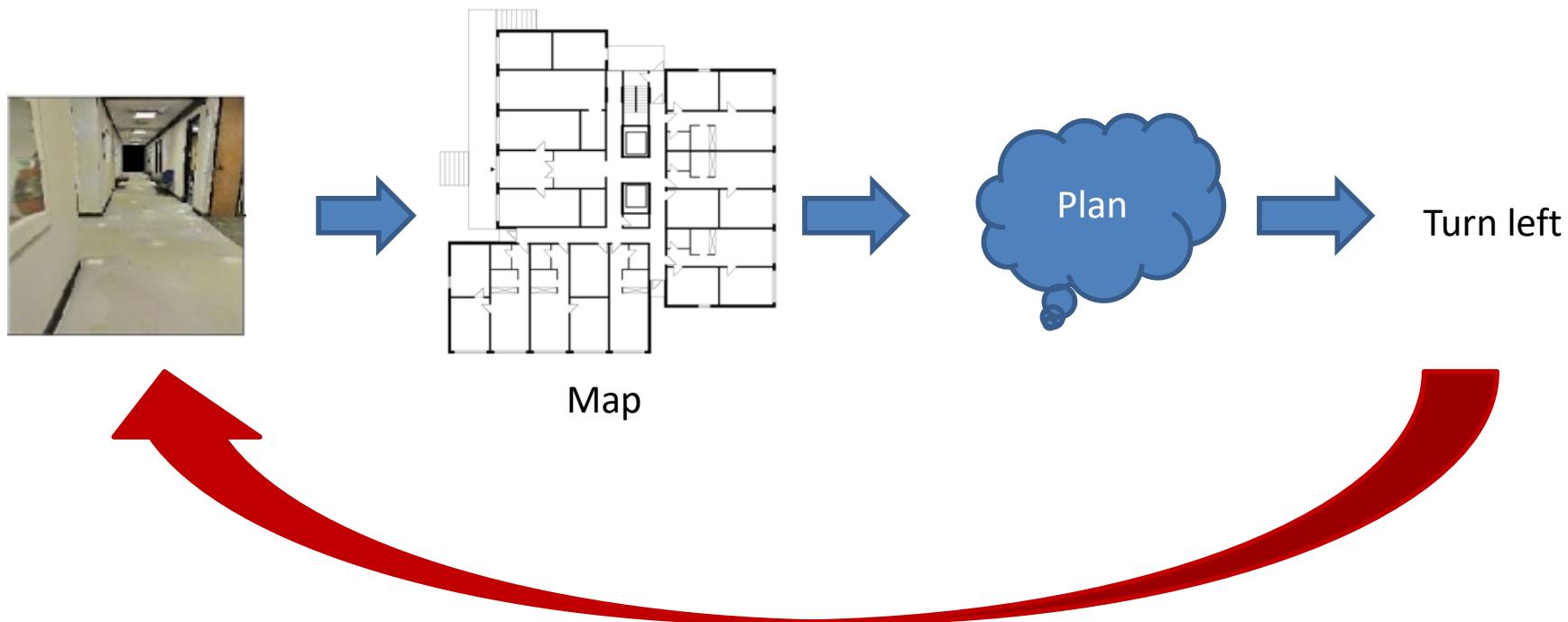
July 30, 2021

Typical issues that plague deployment

- Images seen during deployment are very different: **domain shift**
- Meaning of classes etc. change: **concept drift**
- Unforeseen circumstances, e.g., new classes: **open world**



Challenges: Integrating Vision and Action, Robotics



Challenges: Understanding complex situations / Reasoning



Challenges: Visual Reasoning

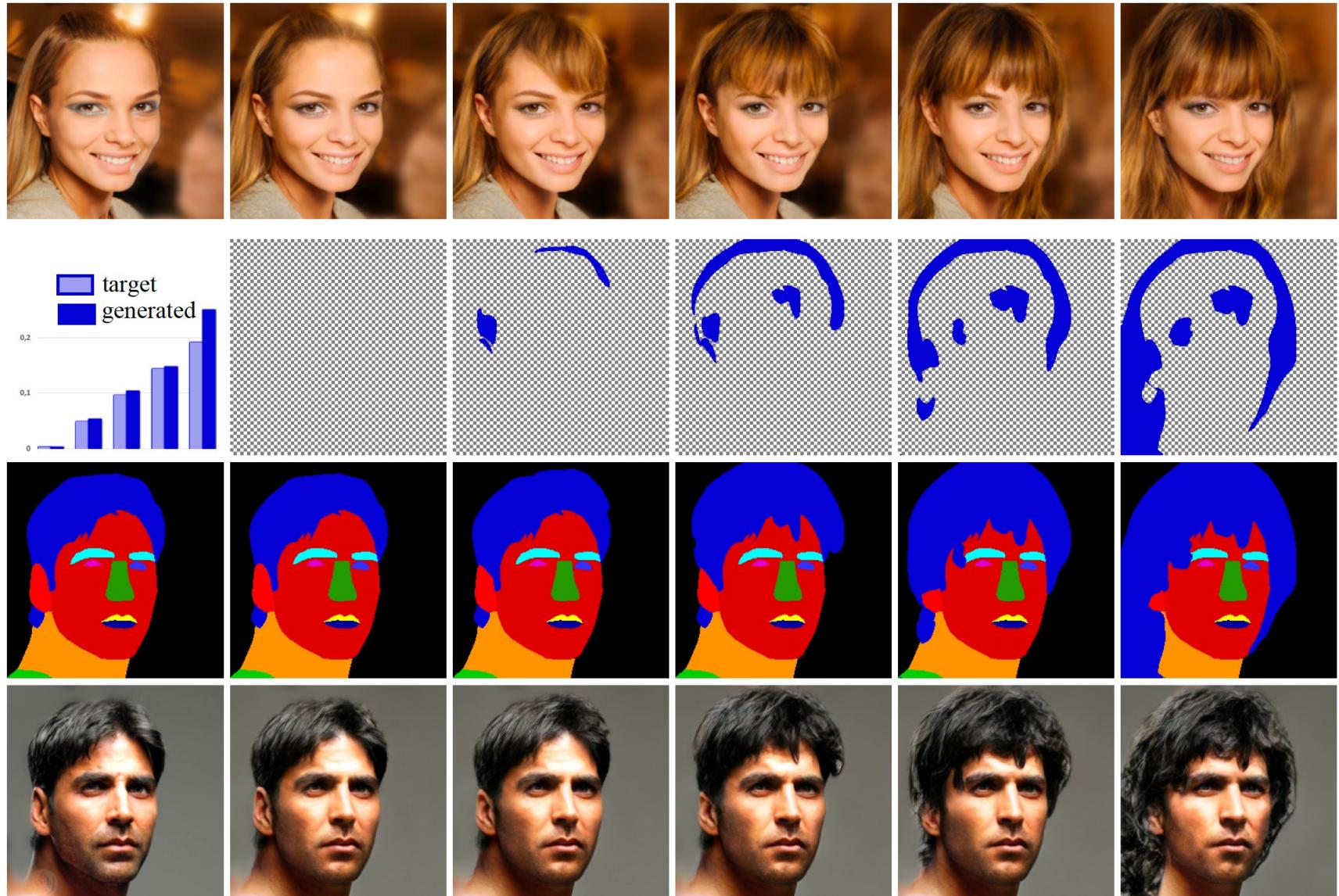
VQA task: Why is this funny?



The picture above is funny.

Andrej Karpathy

Challenges: Generative models for images-edition, manipulation (with GANs)



Challenges: Image Generation in 2023 (Diffusion Models) from Text

Sprouts in the shape of text 'Imagen' coming out of a fairytale book



A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.



A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.



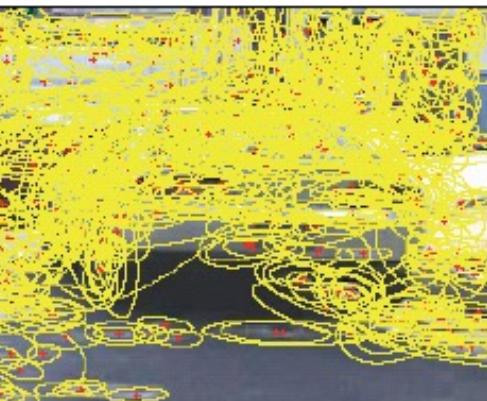
Course Outline

1. Computer Vision and Machine Learning basics

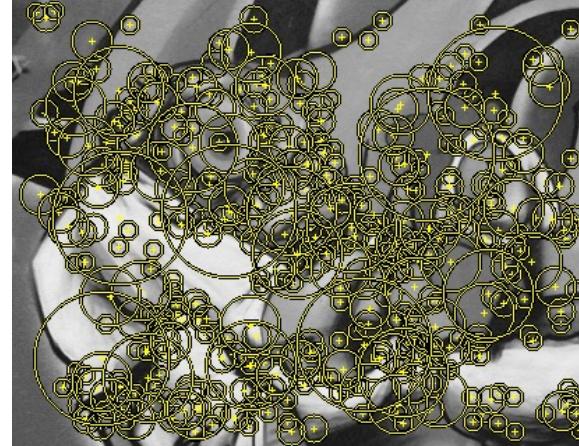
Visual (local) feature detection

Local feature detection and description

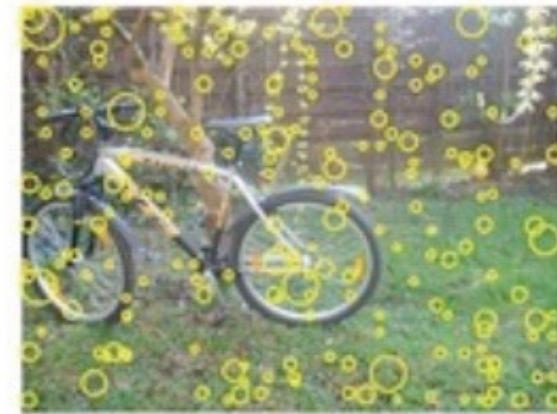
Points/Regions of Interest detection



Sparse, at
interest points



Dense, uniformly

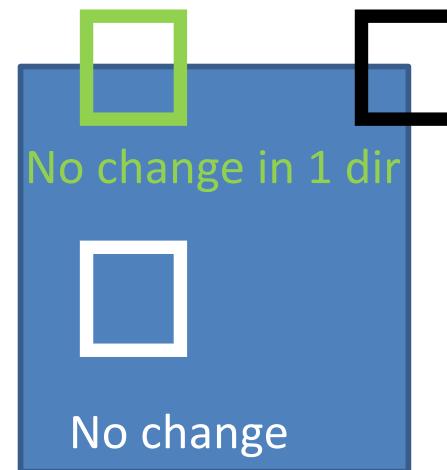


Randomly

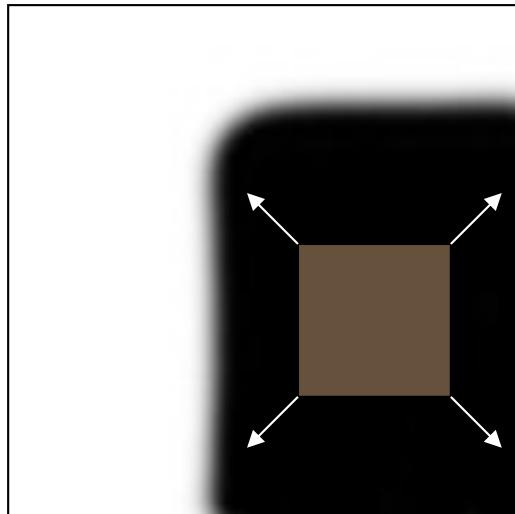
One example: Corner detection (Harris corner detector)

Corner detection

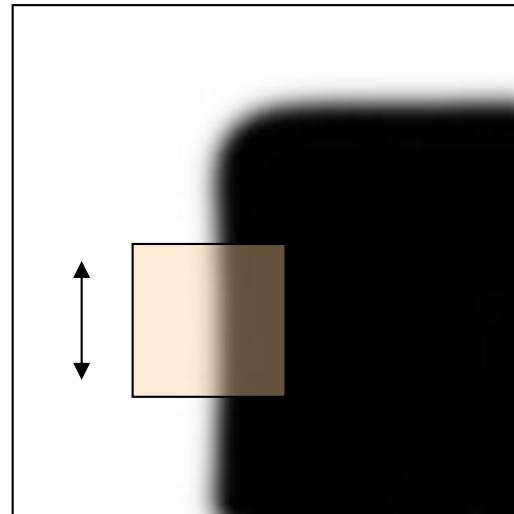
- Corner point: singular point highly informative, rare, ...
- Basic idea for Algo: For each pixel (x,y) from image I , *translating* a centered window: Iff (x,y) is a corner, it should cause large differences in patch appearance (whatever the translation)



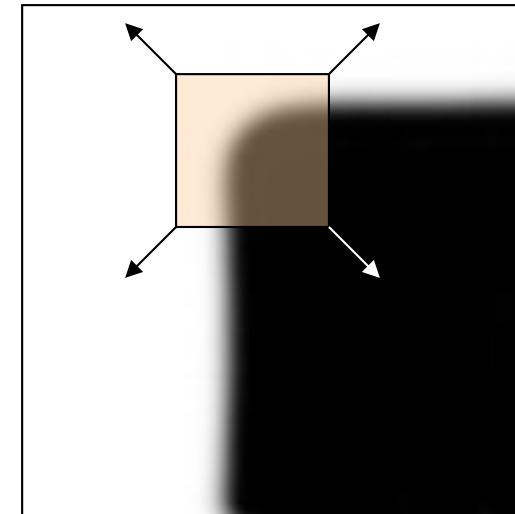
Corner Detection: Basic Idea



“flat” region:
no change in
all directions



“edge”:
no change
along the edge
direction



“corner”:
significant
change in all
directions

Corner detection op == For all pix, shift a window in *any direction*, keep the ones that give a *large change* in intensity

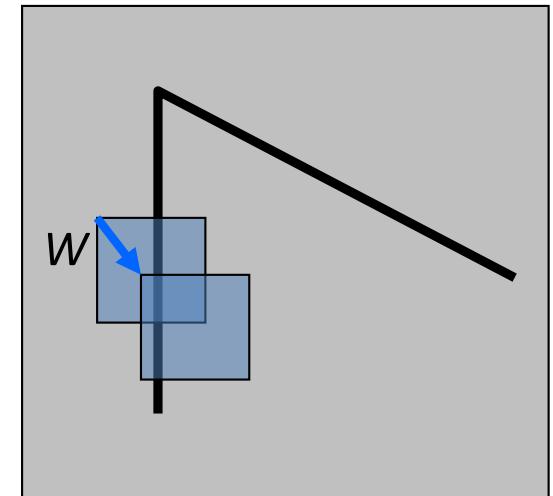
Harris corner detection: algo1

Consider a pix (x,y) , a small window W , a shifting vector (u,v) :

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD “error” $E(u,v)$:

$$E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2$$

- To select (x,y) as corner, $E(u,v)$ has to be *as high as possible for all shifting dir (u,v) !*



ALGO 1: very computationally expensive

Harris detector example

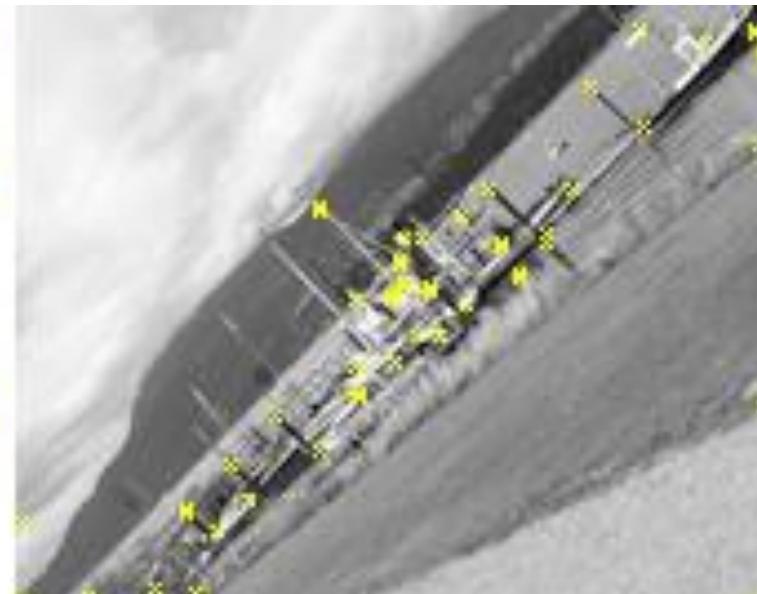
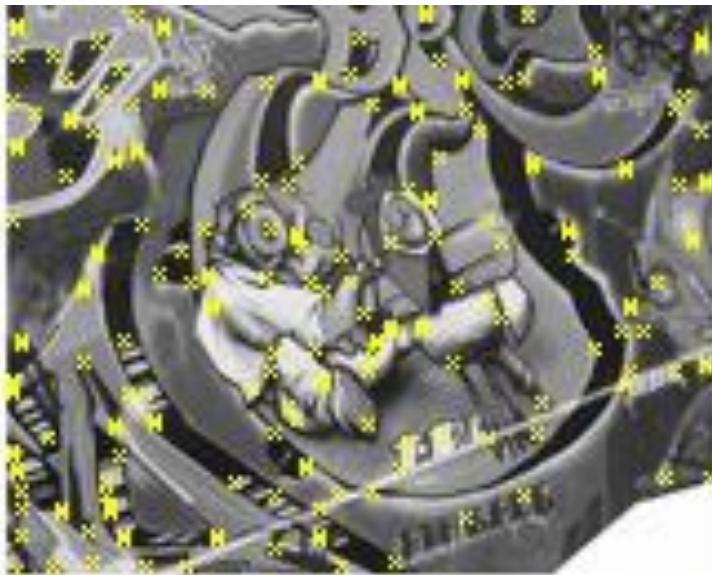


Harris features (in red)



Local feature detection

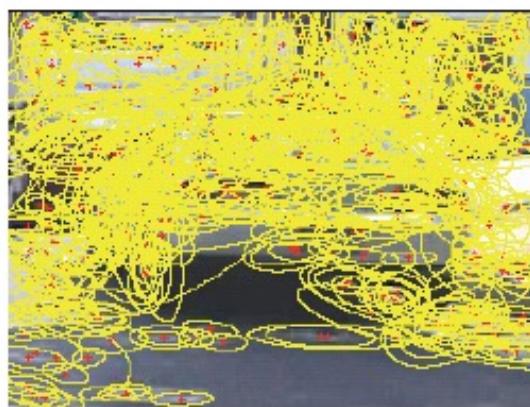
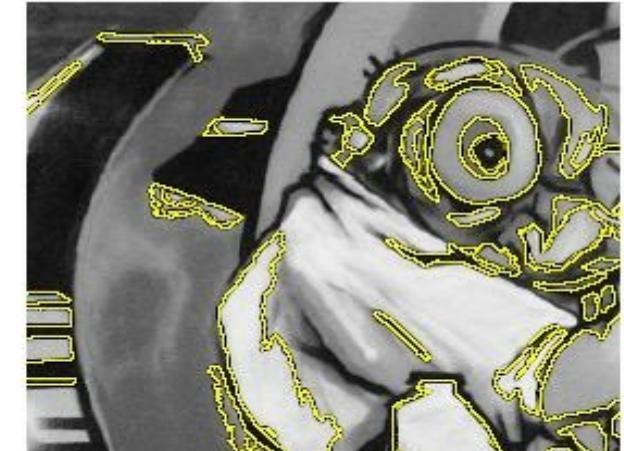
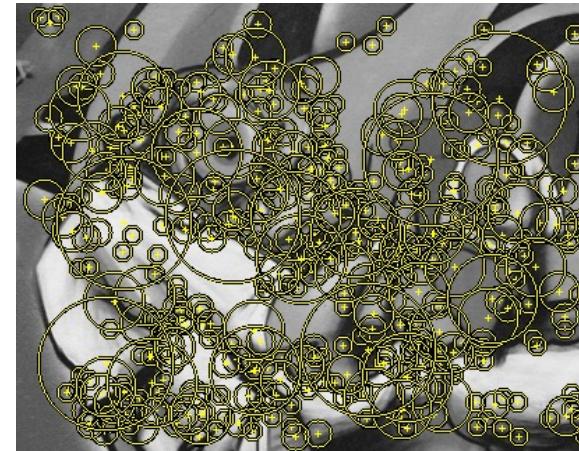
Looking for repeatability



Local feature detection

One example: Corner detection (Harris corner detector)

Many other Points/Regions of Interest detectors



**Sparse, at
interest points**

Dense, uniformly

Randomly

Course Outline

1. Computer Vision and Machine Learning basics

Visual (local) feature detection

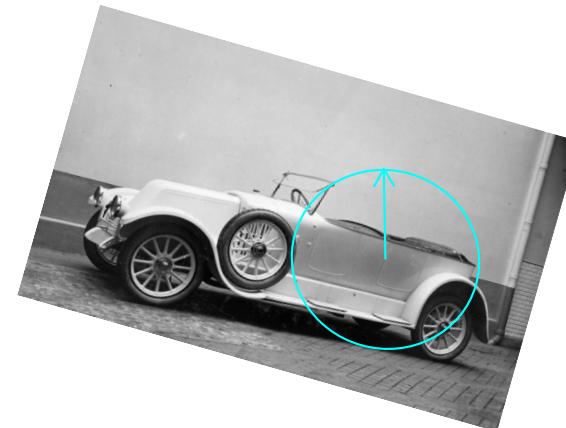
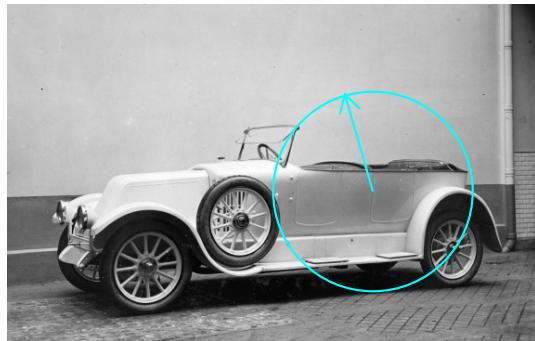
Visual (local) feature description

Local feature **description**

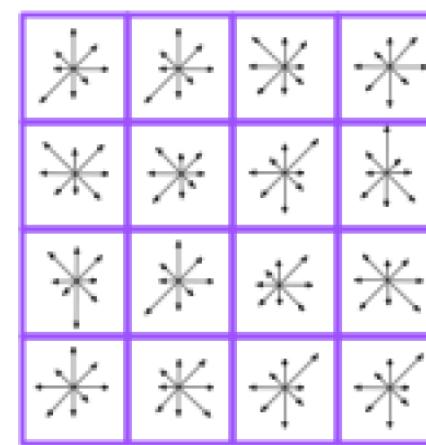
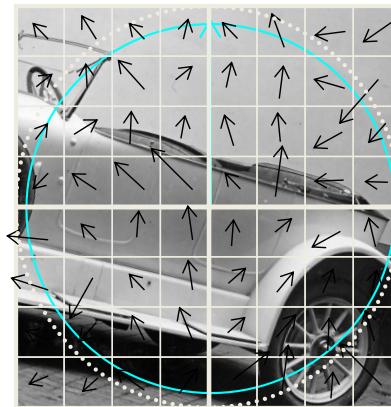
Many Points/Regions of Interest descriptors

One example: SIFT descriptor

Local description (always looking for invariance)



SIFT descriptors/features



*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*

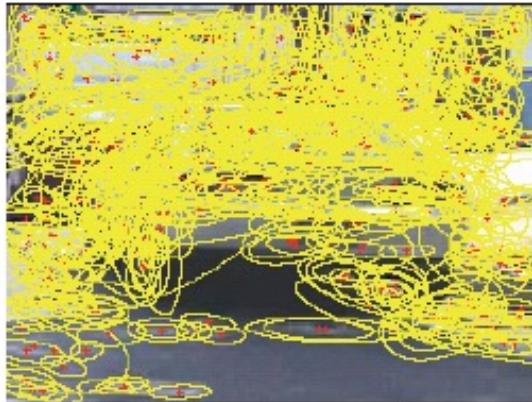
10
17
35
77
35
8
44
3
27
3
0
...

Feature descriptors

- Expected properties?
 - Similar patches => close descriptors
 - Invariance (robustness) to geom. transformation : rotation, scale, view point, luminance, semantics ? ...



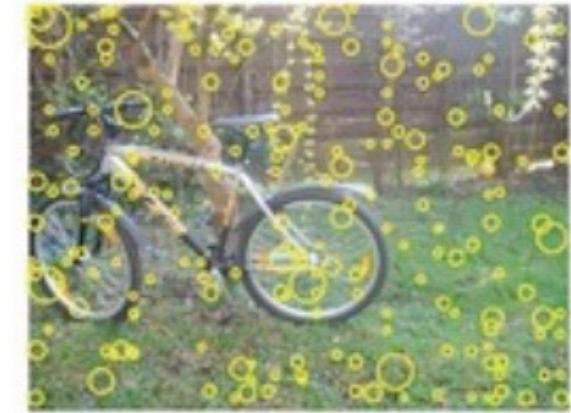
BoF: (First) Image representation



Sparse, at
interest points



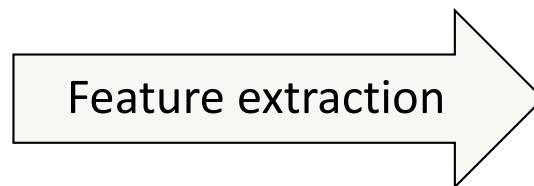
Dense, uniformly



Randomly



Multiple interest
operators



A bag of features
BoF

Bag of Feature (BoF) Model

Image



(features)



Image representation

- BoF (Bag of features)
 - Local signatures: not a scalable representation
 - Not a *semantic* representation
- The missing bits: **the visual word**
- From BoF to Bag of (Visual) words



Course Outline

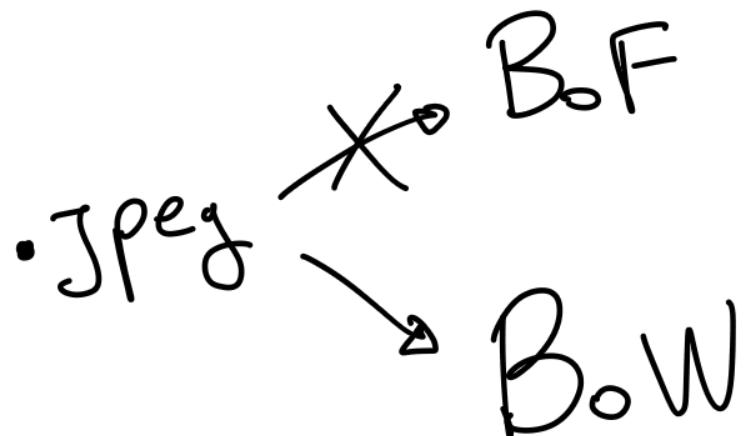
1. Computer Vision and Machine Learning basics

Visual (local) feature detection

Visual (local) feature description

Bag of Word Image representation

1. Introduction to Bag of Words
2. Visual Dictionary
3. Image signature
4. Whole recognition pipeline

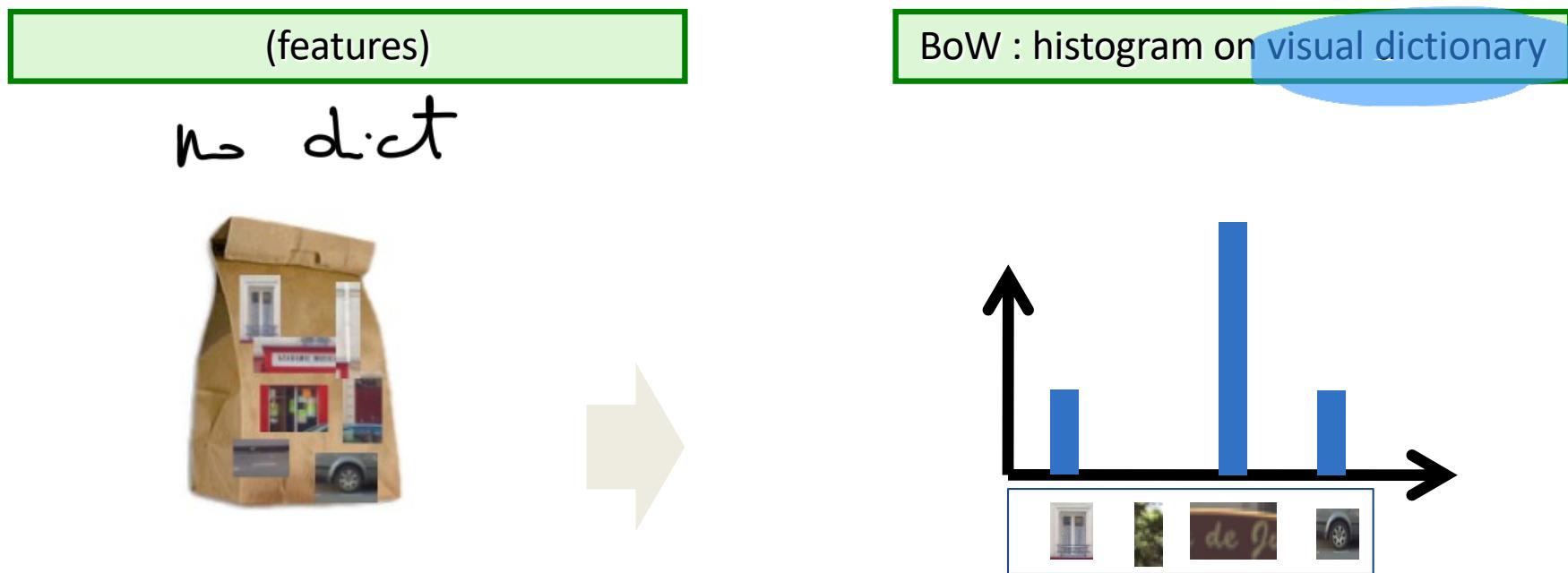


Bag of Words (BoW) model: basic explication with textual representation and color indexing



Comparing 2 docs using visual/color/word occurrences

Bag of Visual Words (BoW)



Questions:

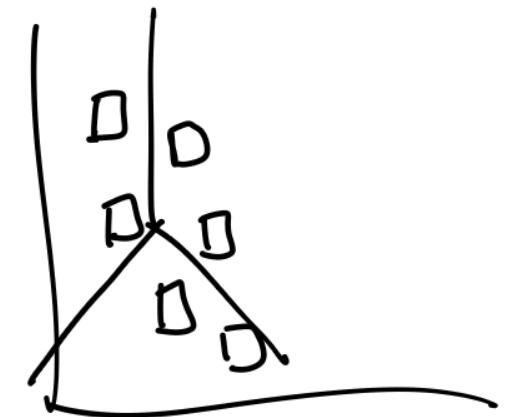
1. Which dictionary ?
2. How to project the BoF onto the dico
3. How to compute the histogram?

Course Outline

1. Computer Vision Introduction:

Visual (local) feature detection and description,
Bag of Word Image representation

1. Introduction to Bag of Words
2. **Visual Dictionary**
3. Image signature
4. Whole recognition pipeline



Visual space clustering

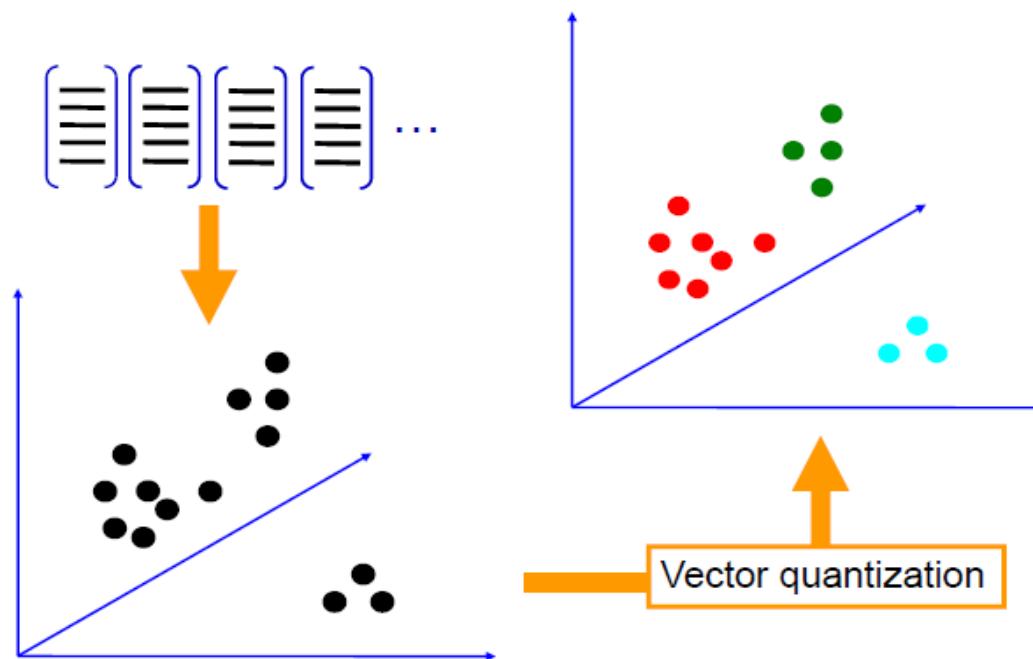
1. Extraction of local features (pattern/visual words) in images
 - Training dataset in classification
 - Image dataset in retrieval
2. Clustering of feature space



Training set but no labels => UNSUPERVISED Learning

Visual space clustering

- Many algorithms for clustering :
 - K-Means
 - Vectorial Quantization
 - Gaussian Mixture Models
 - ...



Clustering with K clusters

Input: set of n points $\{x_j\}_n$ in R^d

Goal: find a set of K ($K << n$) points $w=\{w_k\}_K$
that gives an approximation of the n input points,
ie. minimizing mean square error $C(w)$:

$$C(w) = \sum_{i=1}^n \min_k \|x_i - w_k\|^2$$

At k fixed, complexity is $O(n^{(Kd+1)} \log(n))$

A lot of strategies to approximate the global optimization problem

Clustering with K clusters

$$C(w) = \sum_{i=1}^n \min_k \|x_i - w_k\|^2$$

K-means Algorithm:

Init K centers (c_k) by sampling K points w_k in R^d

1. (Re)assign each point x_i to the cluster s_i with the center w_{s_i} so that $\text{dist}(x_i, w_{s_i})$ is less than dist from x_i to any other clusters
2. Move all w_k inside each cluster as the new barycenter from all the points assigned to the cluster k (equ. to minimize the corresponding mean square error)
3. Go to step 1 if some points changed clusters during the last iteration

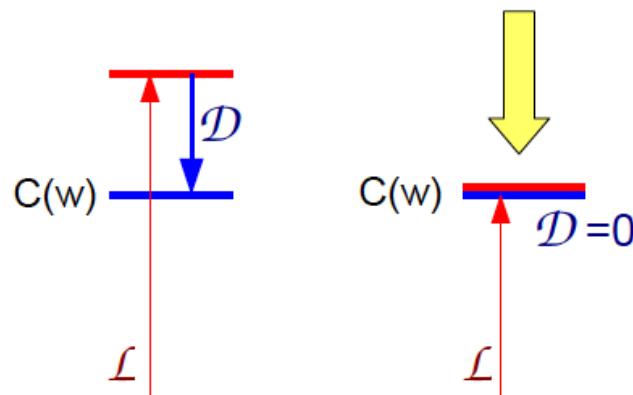
Output: the set of the final K cluster centers $\{c_k = w_k\}$

K-means : why it is successful ?

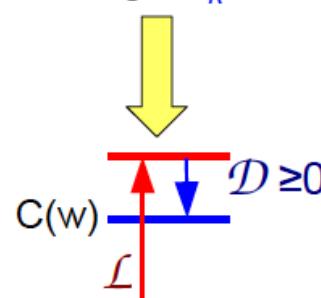
Consider an arbitrary cluster assignment s_i .

$$C(w) = \sum_{i=1}^n \min_k \|x_i - w_k\|^2 = \underbrace{\sum_{i=1}^n \|x_i - w_{s_i}\|^2}_{\mathcal{L}(s,w)} - \underbrace{\sum_{i=1}^n \|x_i - w_{s_i}\|^2 - \min_k \|x_i - w_k\|^2}_{\mathcal{D}(s,w) \geq 0}$$

1. Change s_i to minimize \mathcal{D} leaving $C(w)$ unchanged.

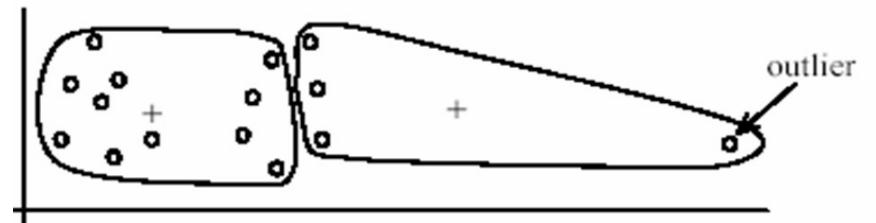


2. Change w_k to minimize \mathcal{L} . Meanwhile \mathcal{D} can only increase.



Clustering

- K-means :
 - Pros
 - Simplicity
 - Convergence (local min)
 - Cons
 - Memory-intensive
 - Depending on K
 - Sensitive to initialization
 - Sensitive to artifacts
 - Limited to spherical clusters
 - Concentration of clusters to areas with high densities of points
(Alternatives : radial based methods)
- K-Means deeply used in practice



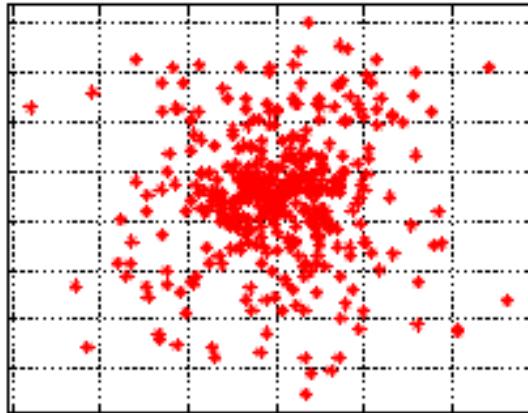
(A): Undesirable clusters



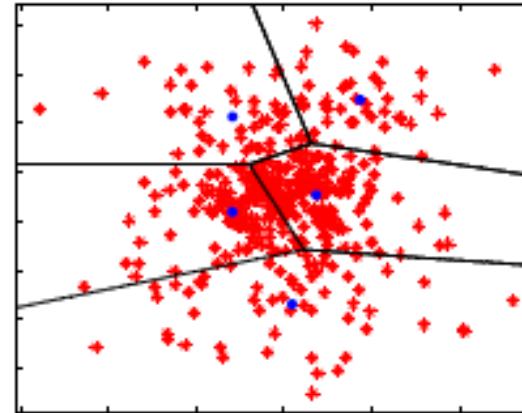
(B): Ideal clusters

Clustering

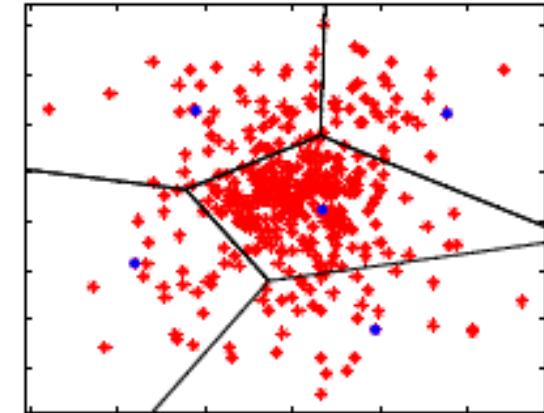
- Uniform / K-means / radius-based :



(a) Histogram



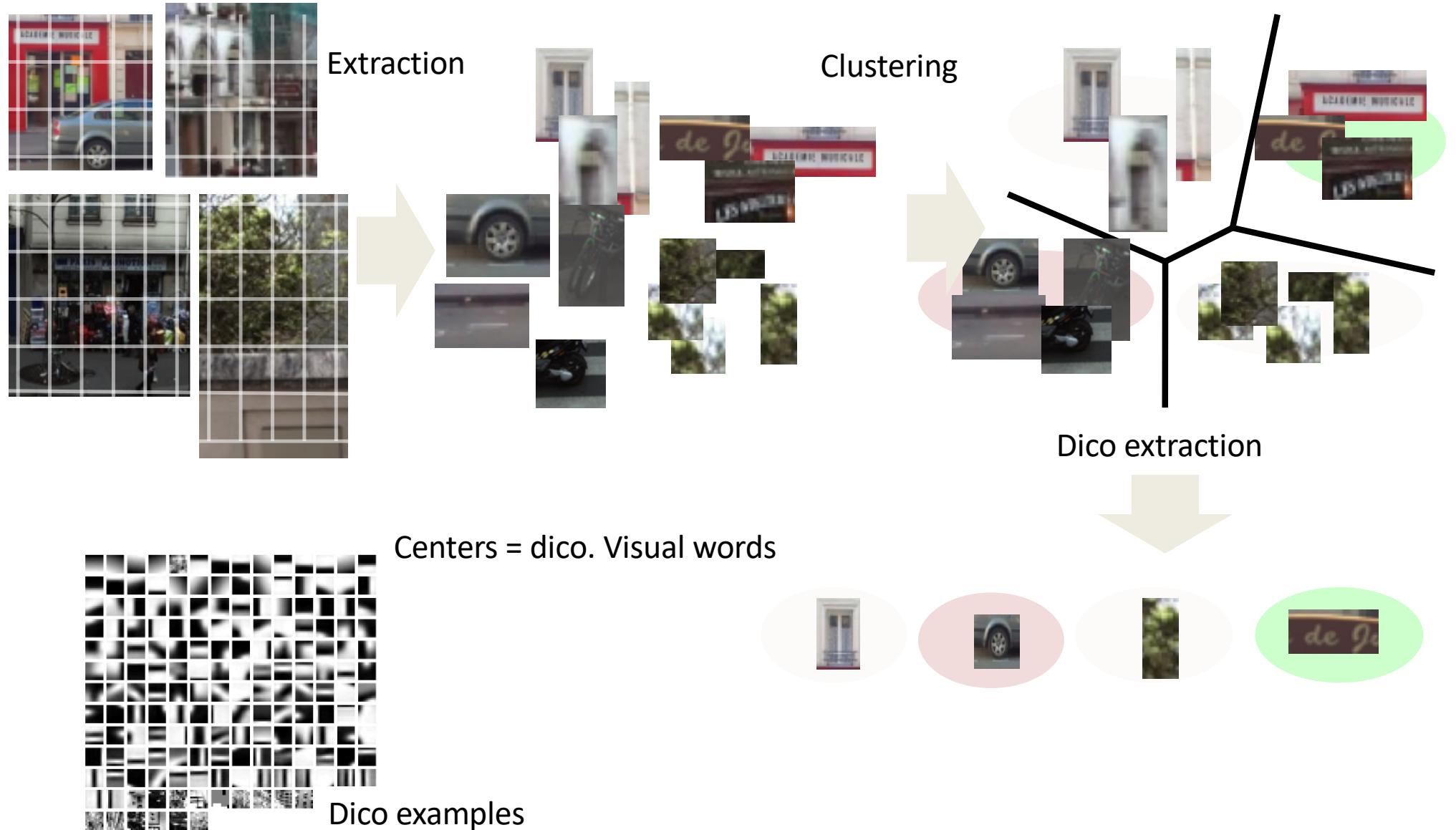
(b) K -means



(c) Radius-based

- *Radius-based clustering assigns all features within a fixed radius of similarity r to one cluster.*

Dictionary = K Visual words



Course Outline

1. Computer Vision Introduction:

Visual (local) feature detection and description,
Bag of Word Image representation

1. Introduction to Bag of Words
2. Visual Dictionary
- 3. Image signature**
4. Whole recognition pipeline

Bag-of-Words (BoW) image signature

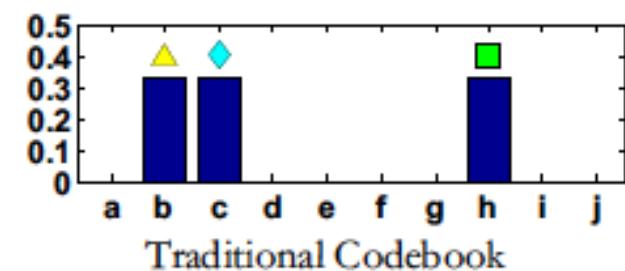
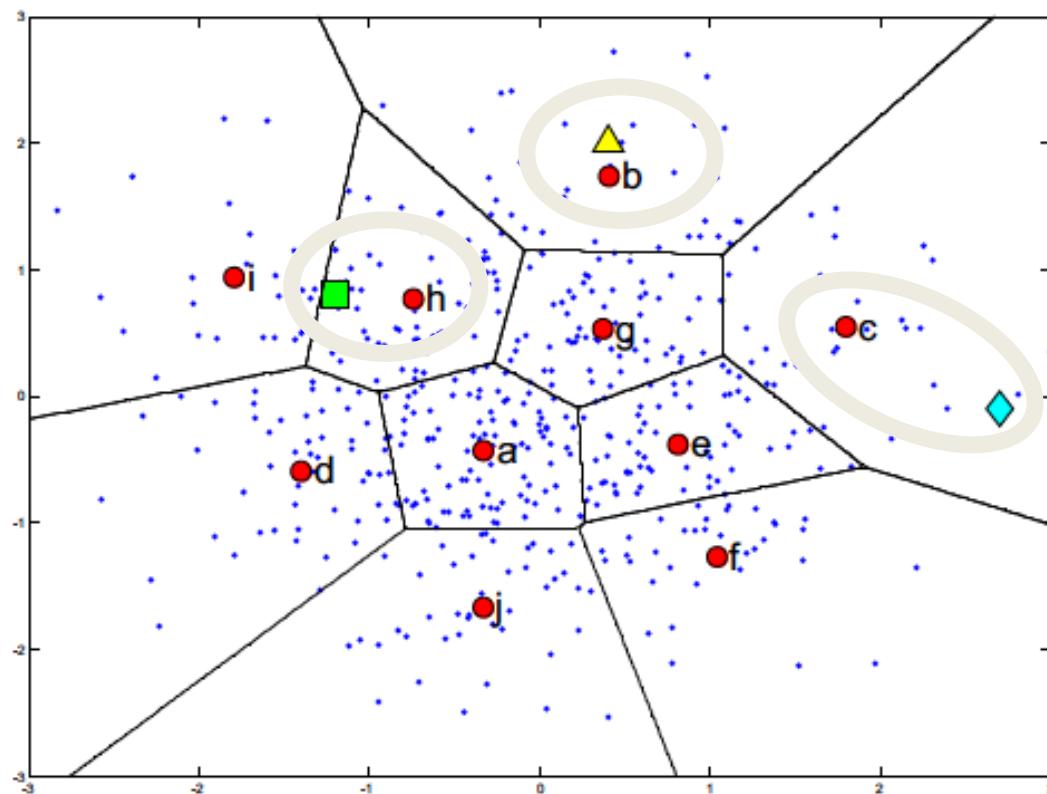
- For each image:
 - For each local feature: find the closest visual word
 - Increase the corresponding bin in histogram of visual dico



- **Image signature (global Index):**
 - **Vector (histogram of M bins)**
 - $M = \text{dimension}$ $K = \text{dico size}$
 - **Each term represents a Likelihood to get this visual word**

Bag-of-Words (BoW) image signature

- Original BoW strategy: **hard assignment/coding**
 - Find the closest cluster for each feature
 - Assign a fix weight (e.g. 1)



Bag-of-Words (BoW) image signature

Sum pooling : initial BoW strategy (just counting occurrences of words in the document)

assign the feature to the closest cluster

Classical BoW = **hard coding + sum pooling**

1. Find the closest cluster for each feature
2. Assign a fix weight (e.g. 1) to this cluster

represent each image as a vector of frequency of the clusters

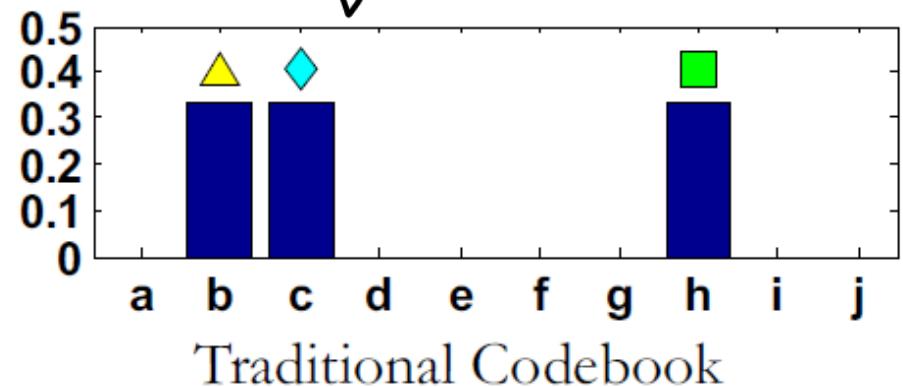
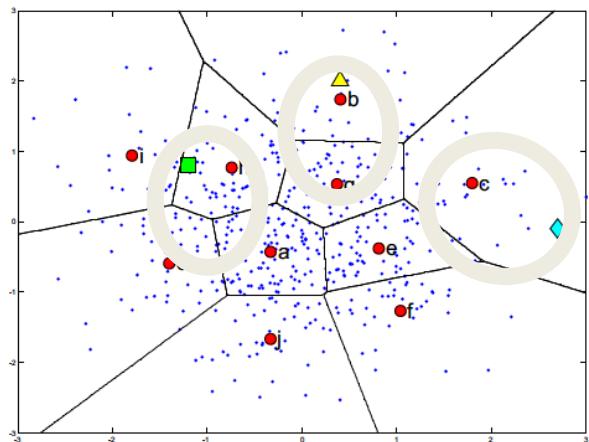
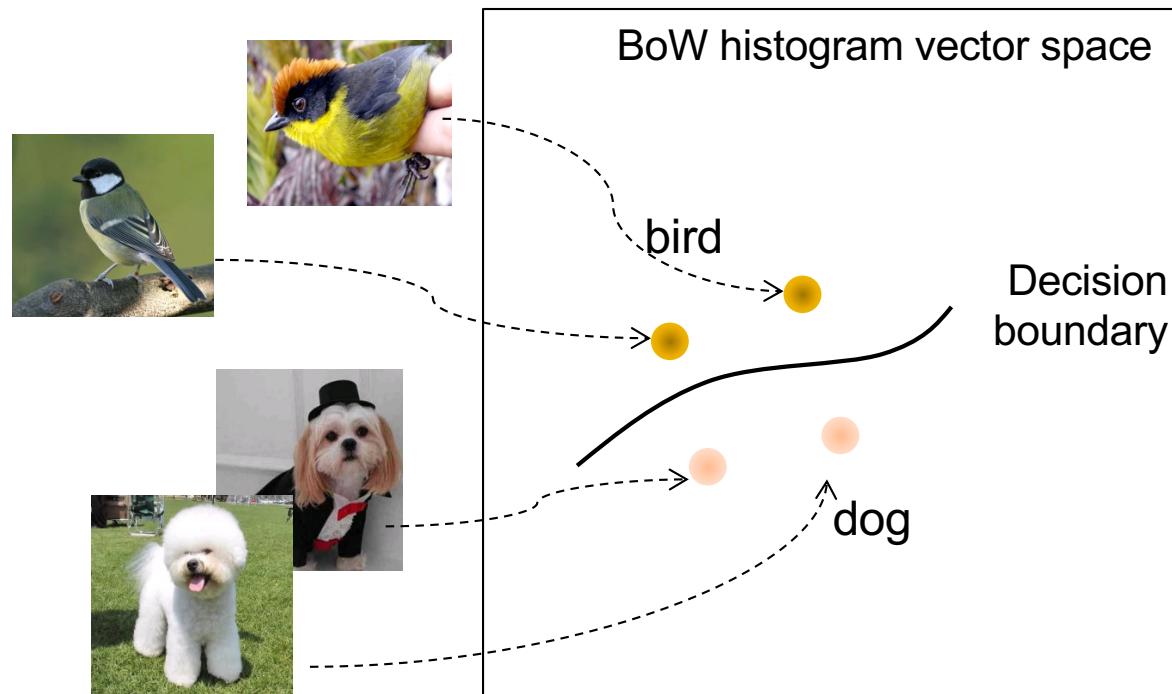


Image classification based on BoW



Learn a classification model to determine the decision boundary

Classification model to determine the decision boundary

SVM classifiers

SVM

Notations:

- Image/Patterns $\mathbf{x} \in \mathbf{X}$
- Φ : function transforming the patterns into feature vectors $\Phi(\mathbf{x})$
- $\langle \cdot, \cdot \rangle$ dot product in the feature space endowed by $\Phi(\cdot)$
- Classes $y = \pm 1$

Early kernel classifiers derived from the perceptron [Rosenblatt58]:

- taking the sign of a linear discriminant function:

$$f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b$$

- Classifiers called Φ -machines

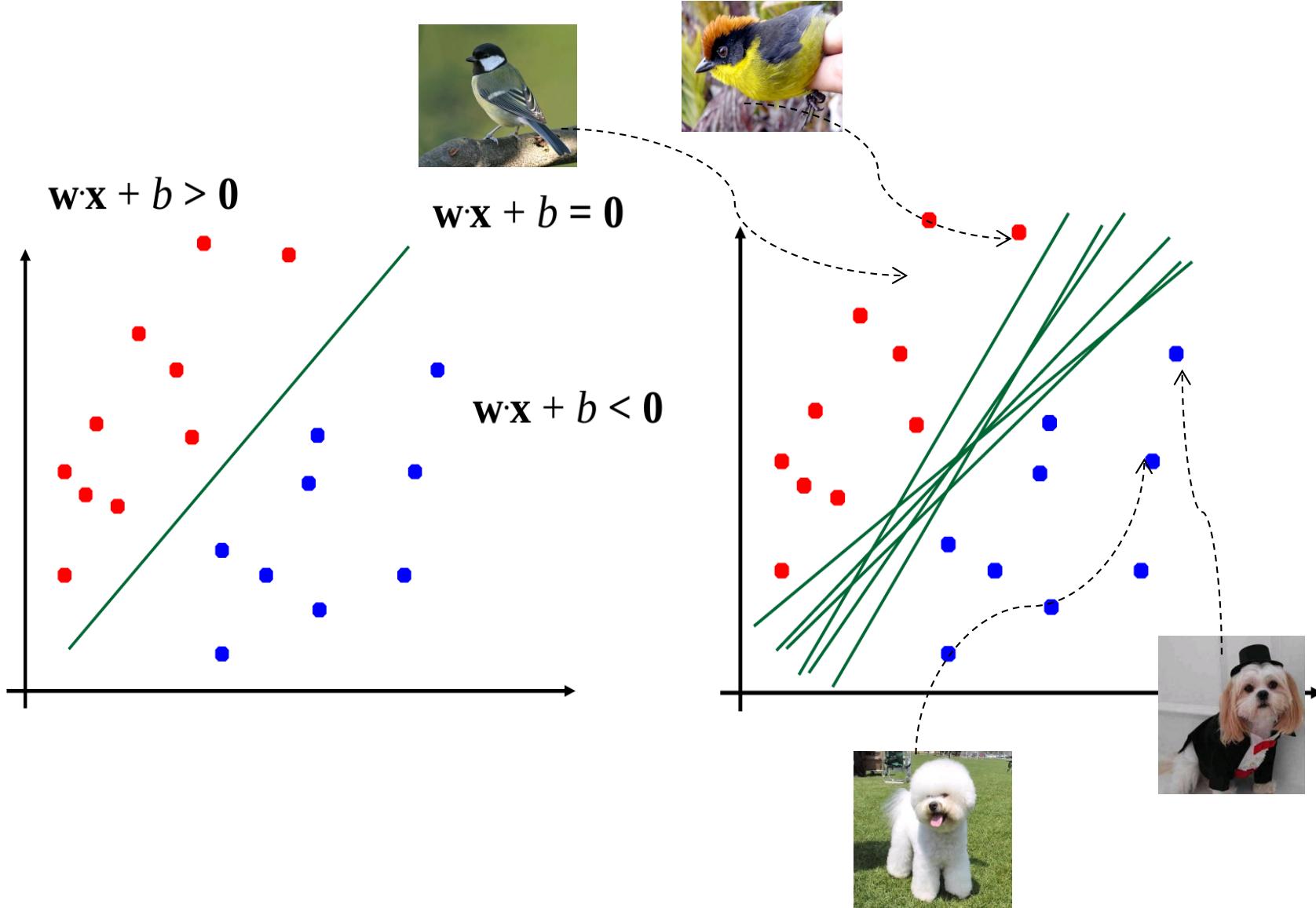
SVM

- Question: how to find/estimate f ?
 - Feature function Φ usually hand-chosen for each problem
 - Several Φ for image processing like BoW
 - w and b : parameters to be determined

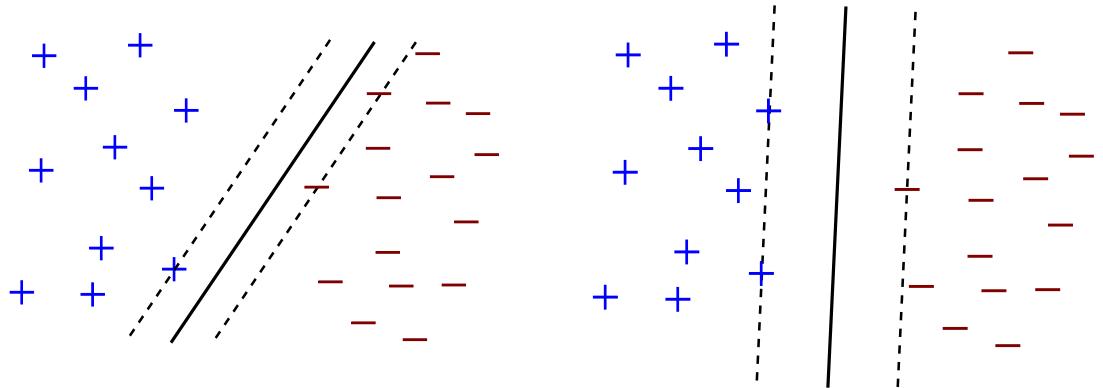
$$f(x) = \langle w, \Phi(x) \rangle + b$$

- Learning algorithm on a set of training examples:
 $\mathcal{A} = (x_1, y_1) \cdots (x_n, y_n)$

Which hyperplane ? w? b?



SVM



SVM optimization: maximizing the margin between + and -

Def.: Margin = distance between the hyperplanes $f(x) = 1$ and $f(x) = -1$
(dashed lines in Figure).

better generalization

Intuitively, a classifier with a larger margin is more robust to fluctuations

Hard Margin

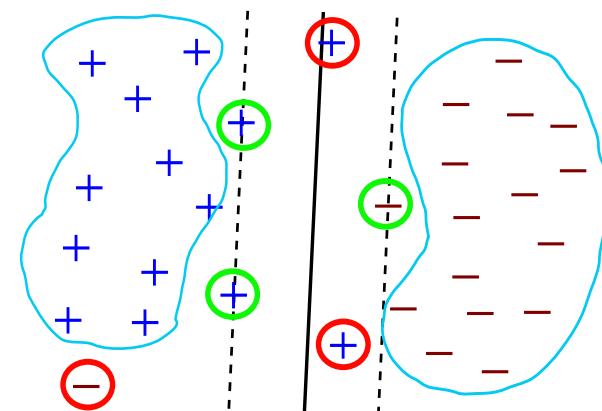
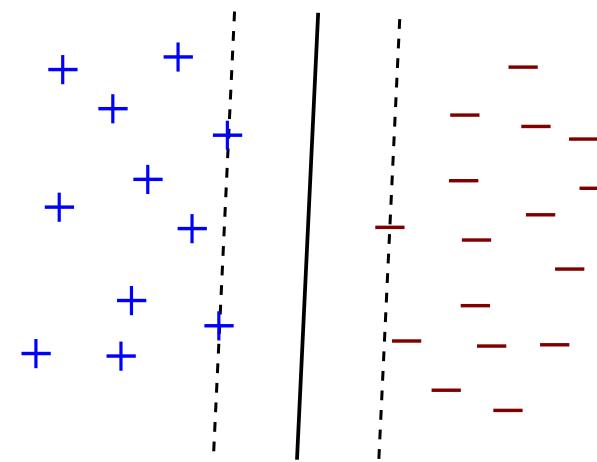
Final expression for the Hard Margin SVM optimization:

$$\min_{w,b} P(w,b) = \frac{1}{2} \|w\|^2 \quad \text{with} \quad \forall i \quad y_i f(x_i) \geq 1$$

Minimizing the norm of $w \Leftrightarrow$ maximizing the margin

SVM

- Hard Margin: OK if data are linearly separated
- Otherwise: noisy data (in red) disrupt the optim.
- Solution: Soft SVM



SVM: Soft Margin

Introducing the slack variables ξ_i , one usually gets rid of the inconvenient max of the loss and rewrite the problem as

$$\min_{w,b} P(w,b) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad \text{with} \quad \begin{cases} \forall i \quad y_i f(x_i) \geq 1 - \xi_i \\ \forall i \quad \xi_i \geq 0 \end{cases}$$

it's ok if you go over margin] relax the hard margin

For very large values of the hyper-parameter C, **Hard Margin** case:

- Minimization of $\|w\|$ (ie margin maximization) under the constraint that all training examples are correctly classified with a loss equal to zero.

Smaller values of C relax this constraint: Soft Margin case

- SVMs that produces markedly better results on **noisy problems**.

small C \rightarrow small $C \sum \xi_i \rightarrow$ accept errors \rightarrow regularization

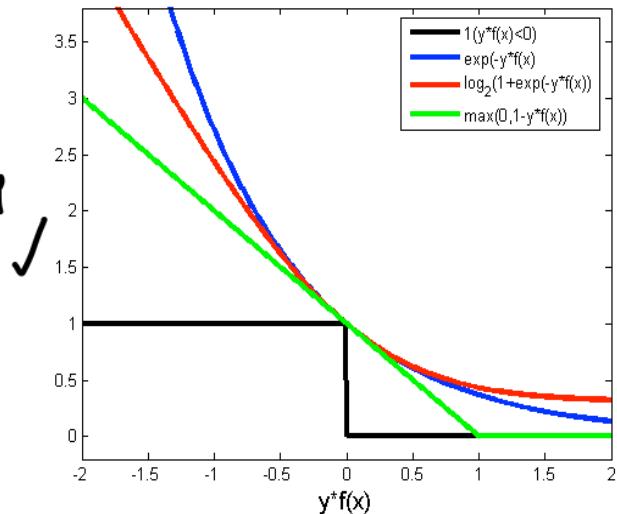
SVM learning scheme

$$\ell(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)) \quad \begin{matrix} y_i f(x_i) > 1 \\ \ell = 0 \end{matrix} \checkmark$$

$$\min_{w, b} P(w, b) = \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^n \ell(y_i, f(x_i))$$

REG

Equivalently, minimizing the following objective function in feature space with the hinge loss function:



ℓ as little as possible

$y_i f(x_i) \geq 1$,

but depending on C

avg to avoid dependence on n

$$\min_{w, b} P(w, b) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \ell(y_i f(x_i))$$

Regularization

Margin
Maximization

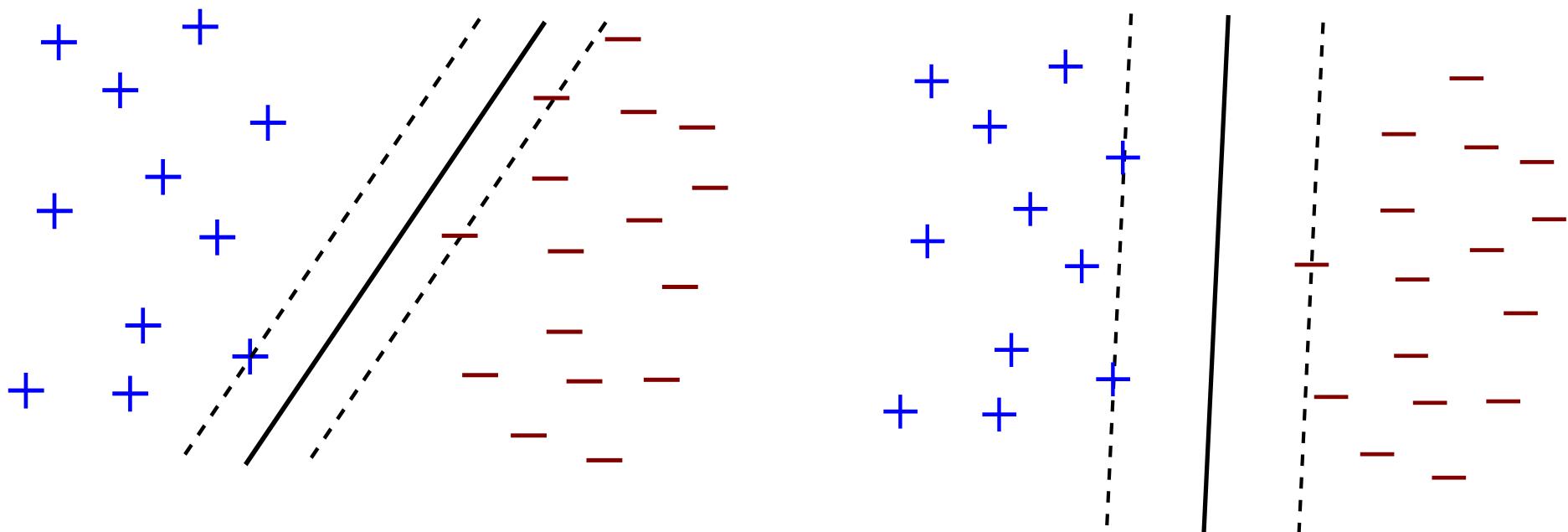
Constraint
satisfaction

Data
fitting

Solving equation: SVM

Support Vector Machines (SVM) defined by three incremental steps:

1. [Vapnik63]: linear classifier / separates the training examples with the **widest margin** => Optimal Hyperplane



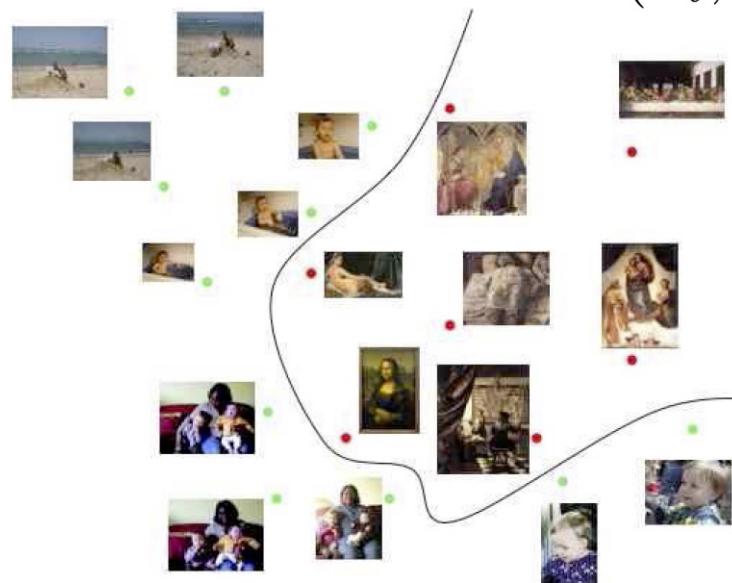
Solving equation: SVM

img \rightarrow BoW dict \rightarrow vector of length $\text{len}(\text{dict})$ with lengths $\xrightarrow{\text{SVM class}}$

Support Vector Machines (SVM) defined by three incremental steps:

1. [Vapnik63]: linear classifier / separates the training examples with the widest margin => Optimal Hyperplane
2. [Guyon93] Optimal Hyperplane built in the feature space induced by a kernel function

$$k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

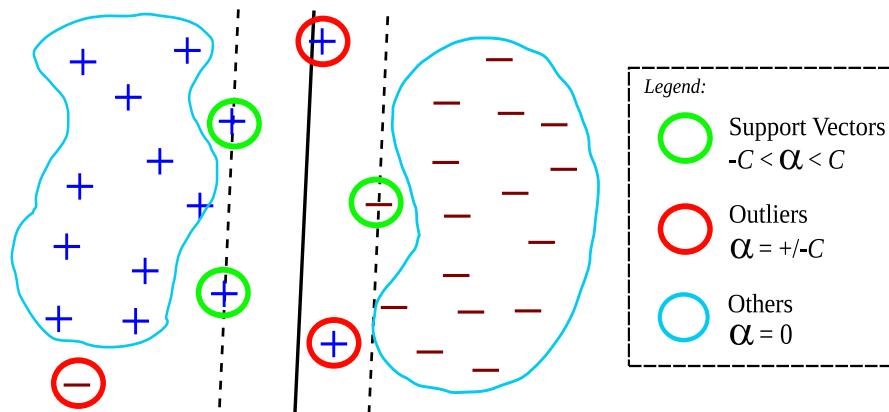


Generalized to
more complex
boundaries
(non-linear)

Solving equation: SVM

Support Vector Machines (SVM) defined by three incremental steps:

1. [Vapnik63]: linear classifier / separates the training examples with the widest margin =>Optimal Hyperplane
2. [Guyon93] Optimal Hyperplane built in the feature space induced by a kernel function
3. [Cortes95] soft version: noisy problems addressed by allowing some examples to violate the margin constraint



Classification pipeline

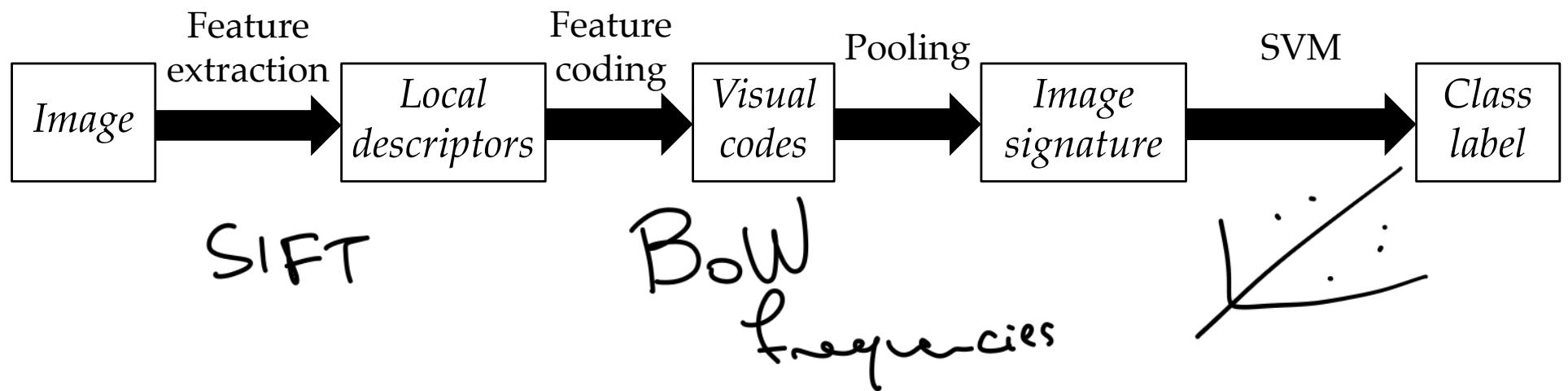
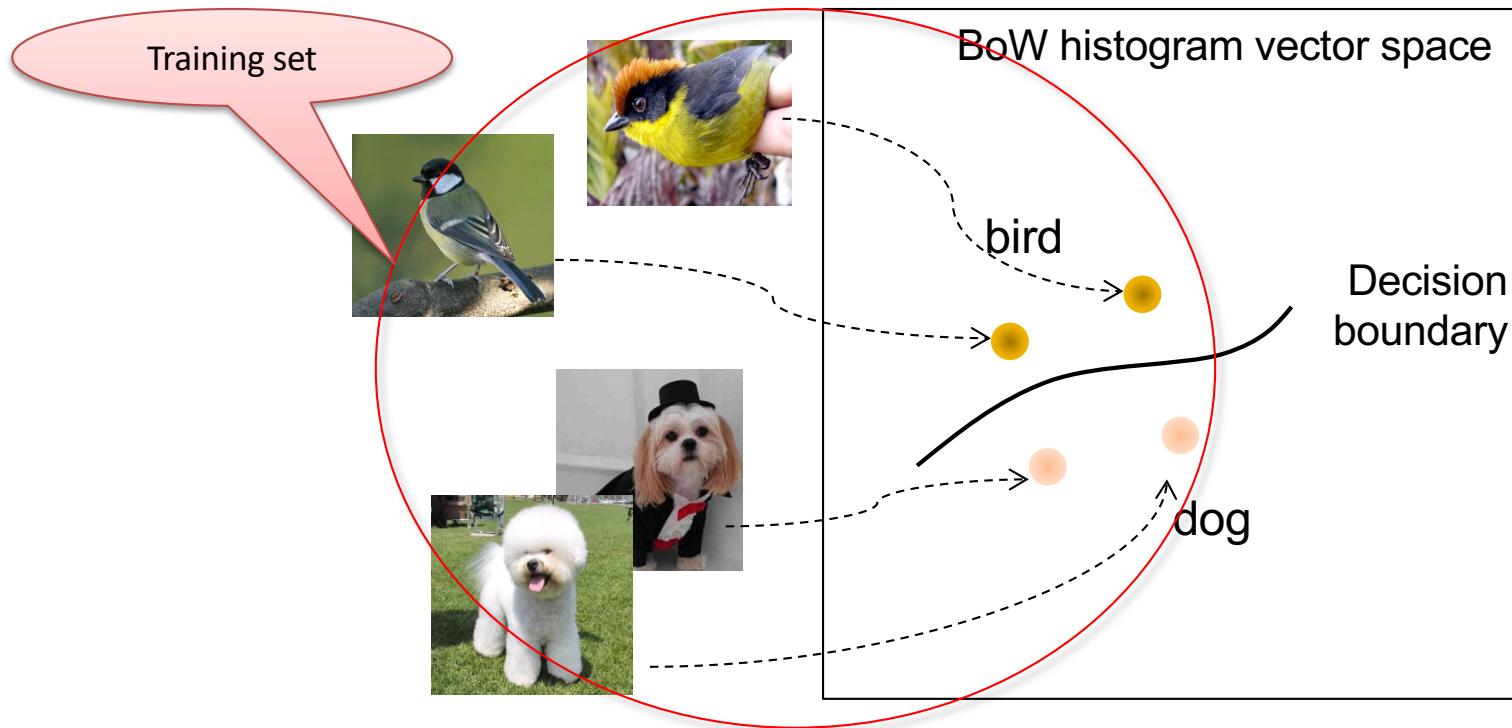


Image classification based on BoW



Learn a classification model to determine the decision boundary