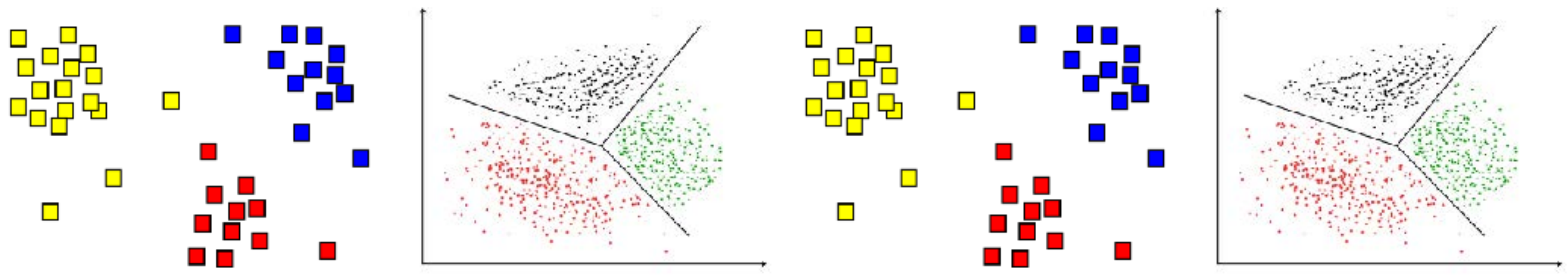
Please get your **iClickers** —

Quiz 5: **6** questions

CPSC 425: Computer Vision



Lecture 18: Visual Classification 1, Bag of Words

Menu for Today

Topics:

— **Visual Classification**

— **Bag of Words**, K-means

Readings:

— **Today's** Lecture: Szeliski 11.4, 12.3-12.4, 9.3, 5.1-5.2

Reminders:

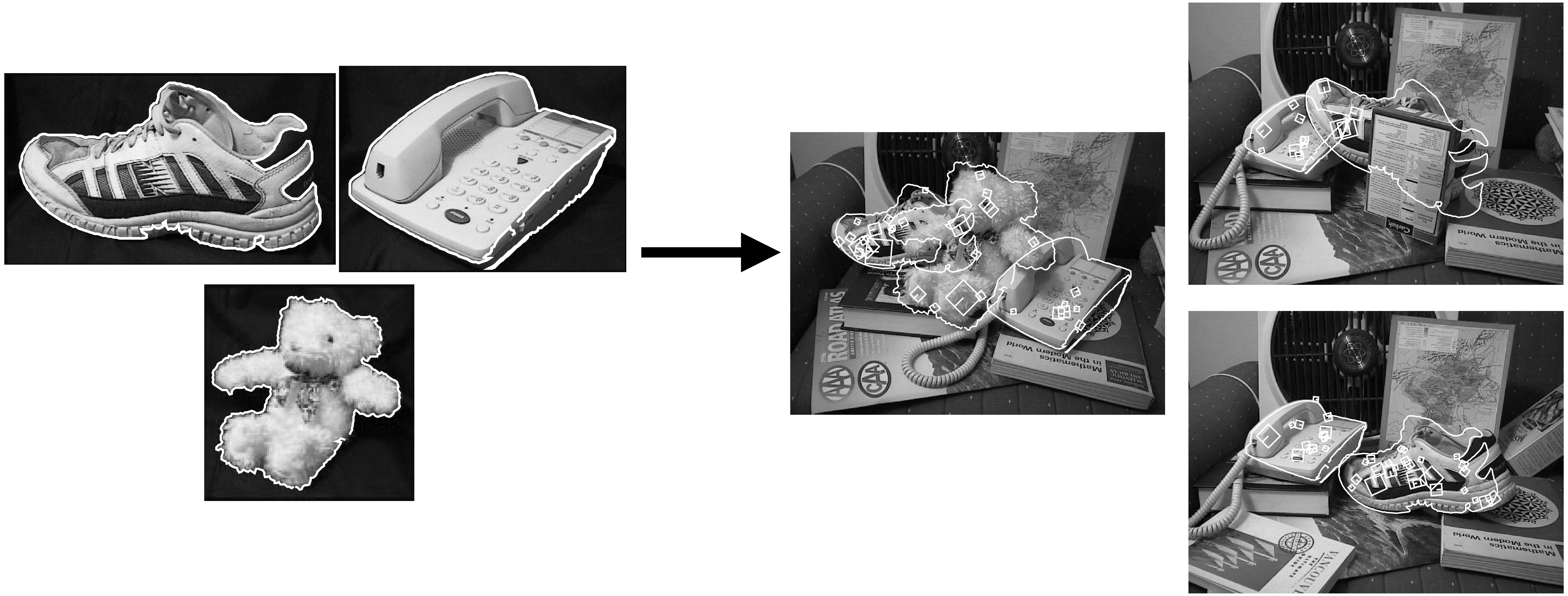
— **Assignment 4**: due **TOMORROW**

Learning **Goals**

Understanding the visual classification “**pipeline**”

Object Recognition

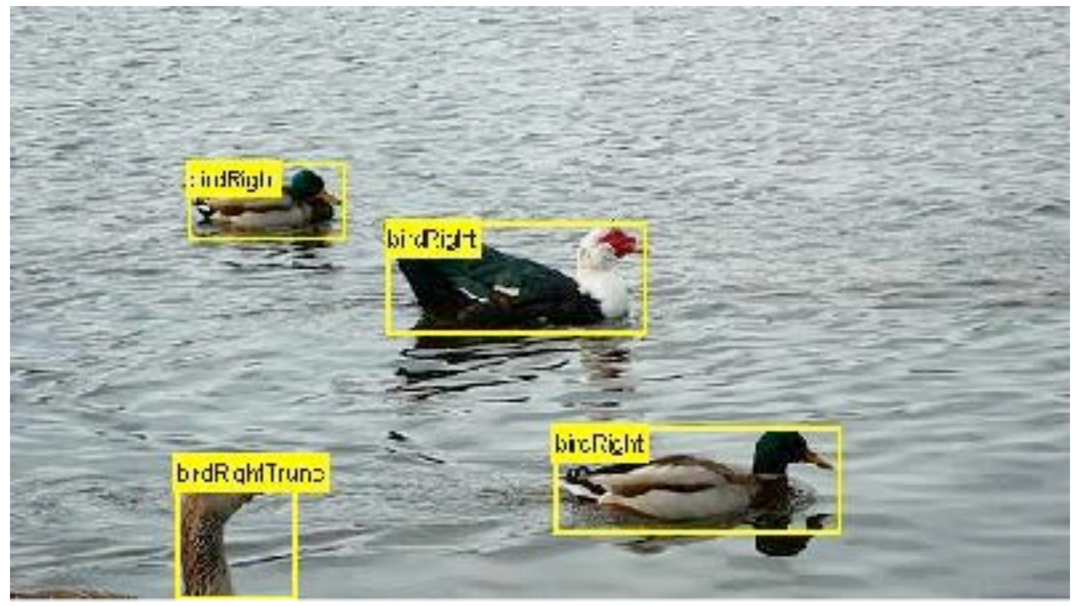
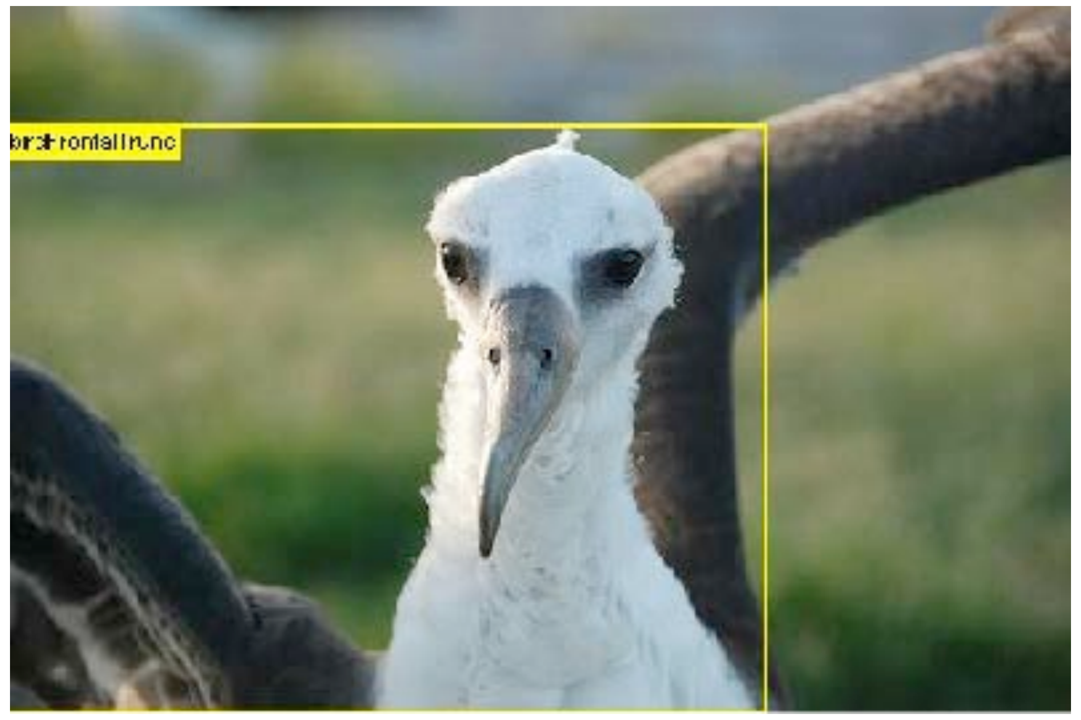
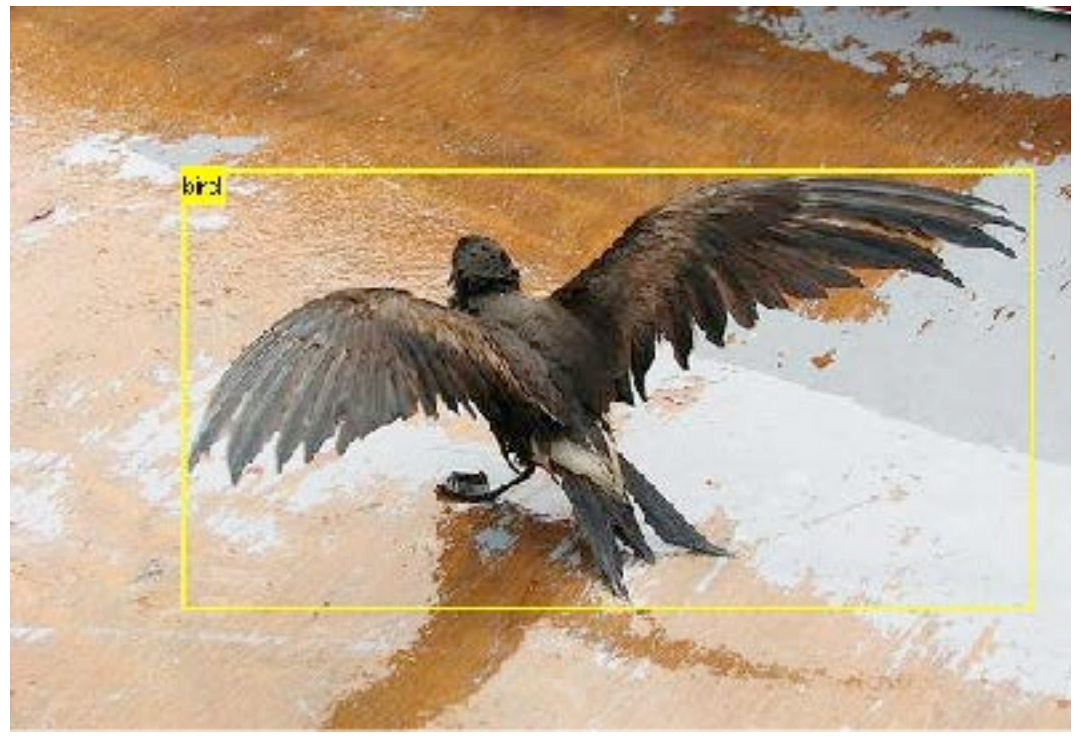
- Object recognition with SIFT features [Lowe 1999]



What is present? Where? What orientation?

Object Recognition

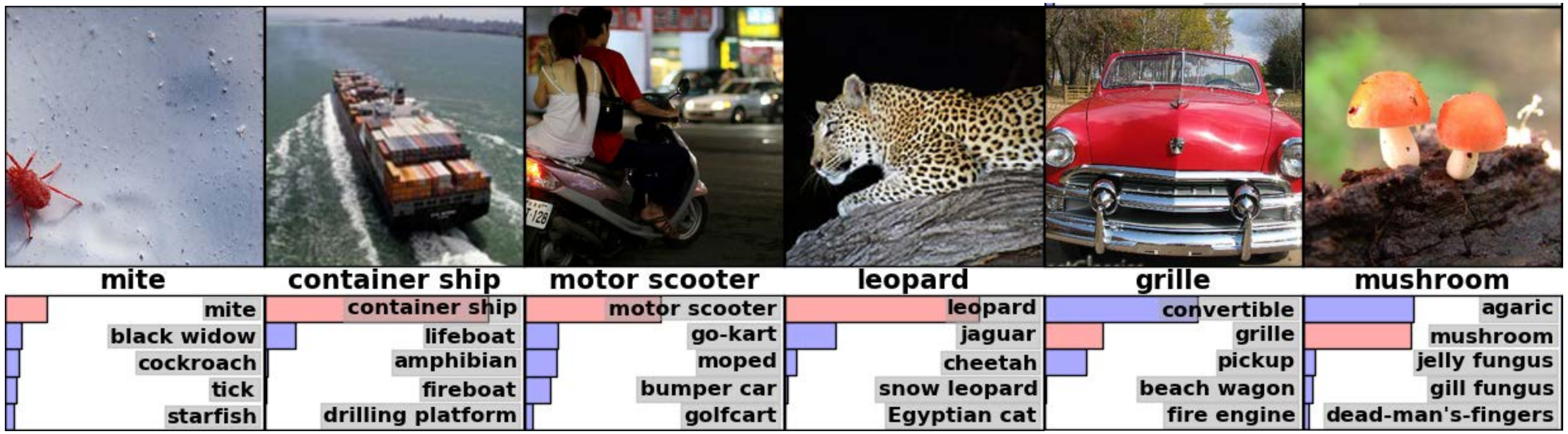
- PASCAL Visual Object Classes Challenges [2005-2012]

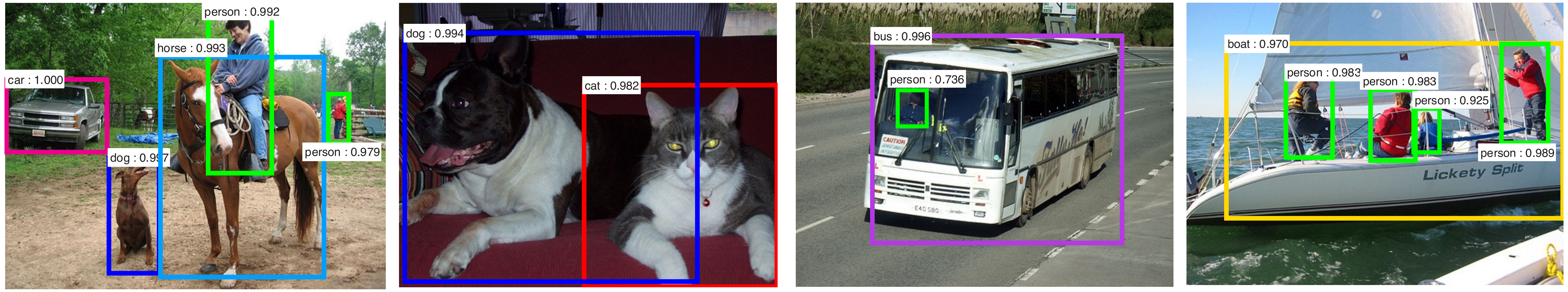

What is present? Where? What orientation?

Classification and Detection

- Classification: Label per image, e.g., ImageNet



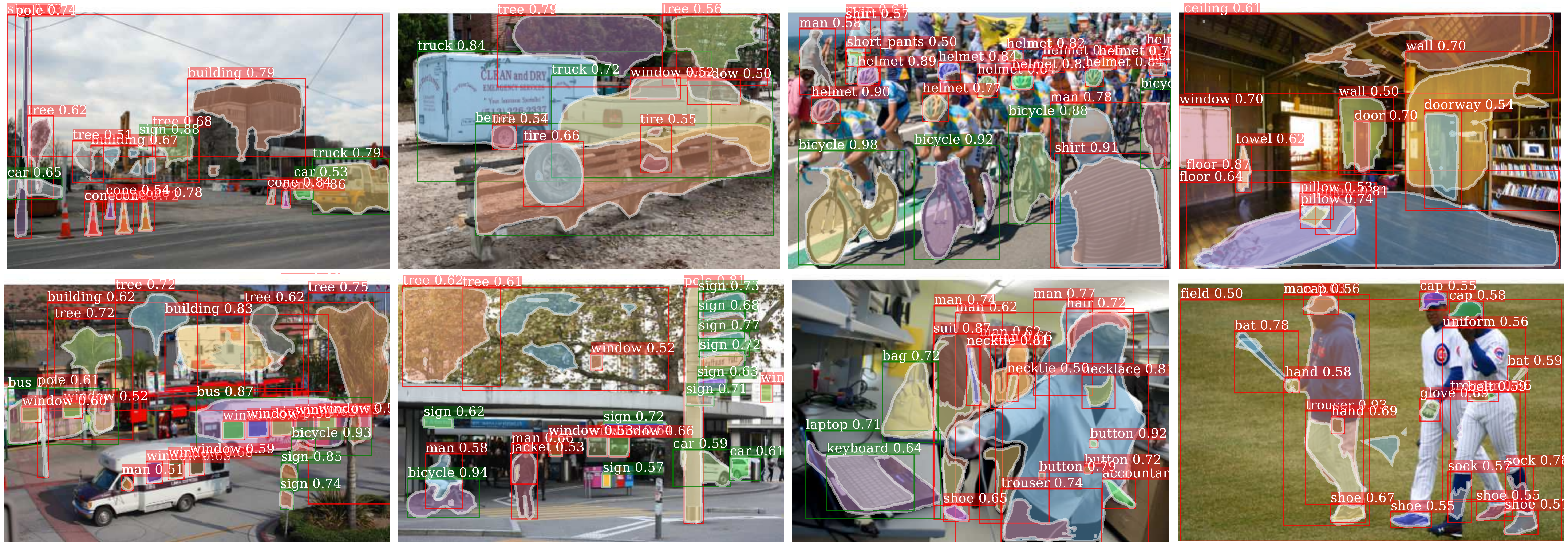
- Detection: Label per region, e.g., PASCAL VOC



[Krizhevsky et al 2011][Ren et al 2016]

Segmentation

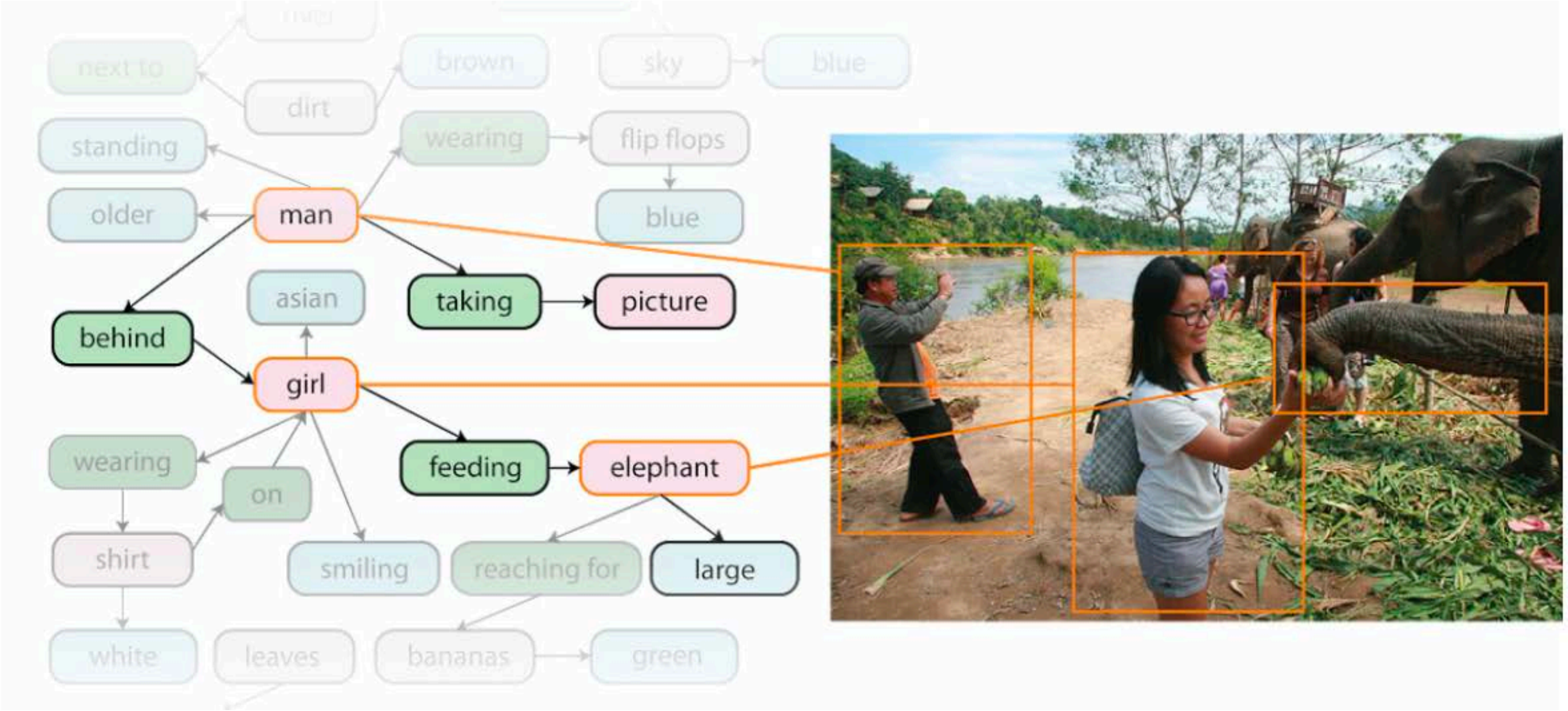
- Segmentation: Label per pixel, e.g., MS COCO



[Hu et al 2017]

Structured Image Understanding

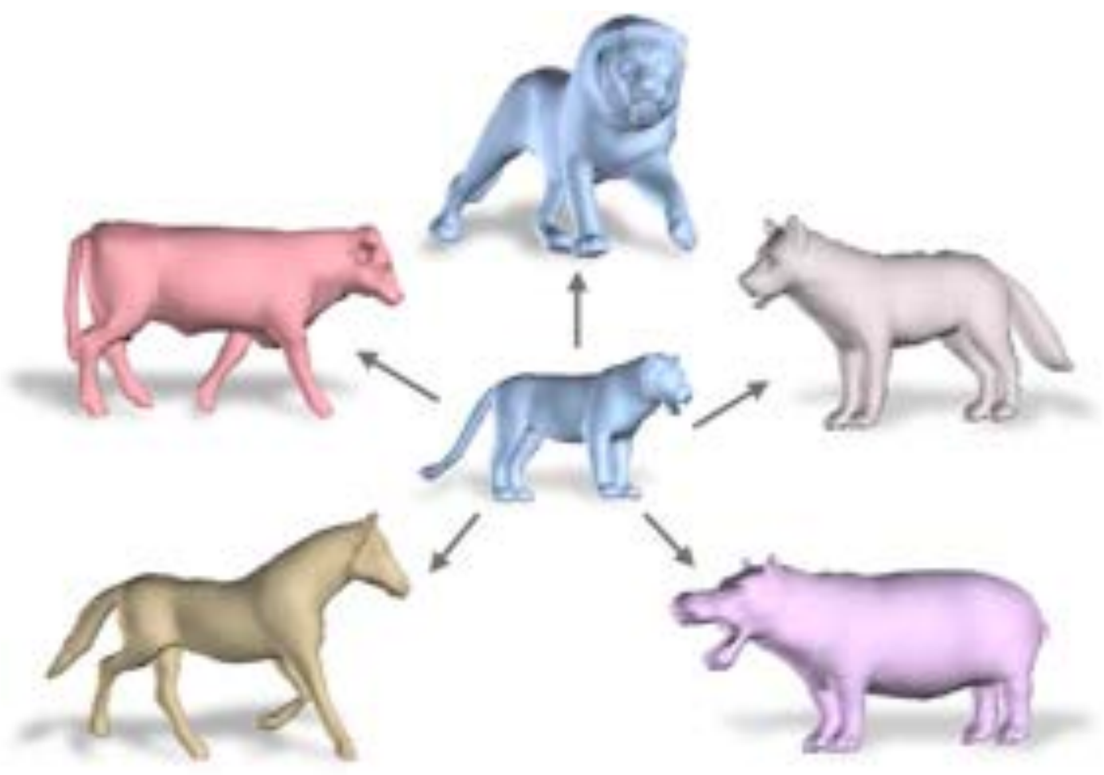
- "Girl feeding large elephant"
- "A man taking a picture behind girl"

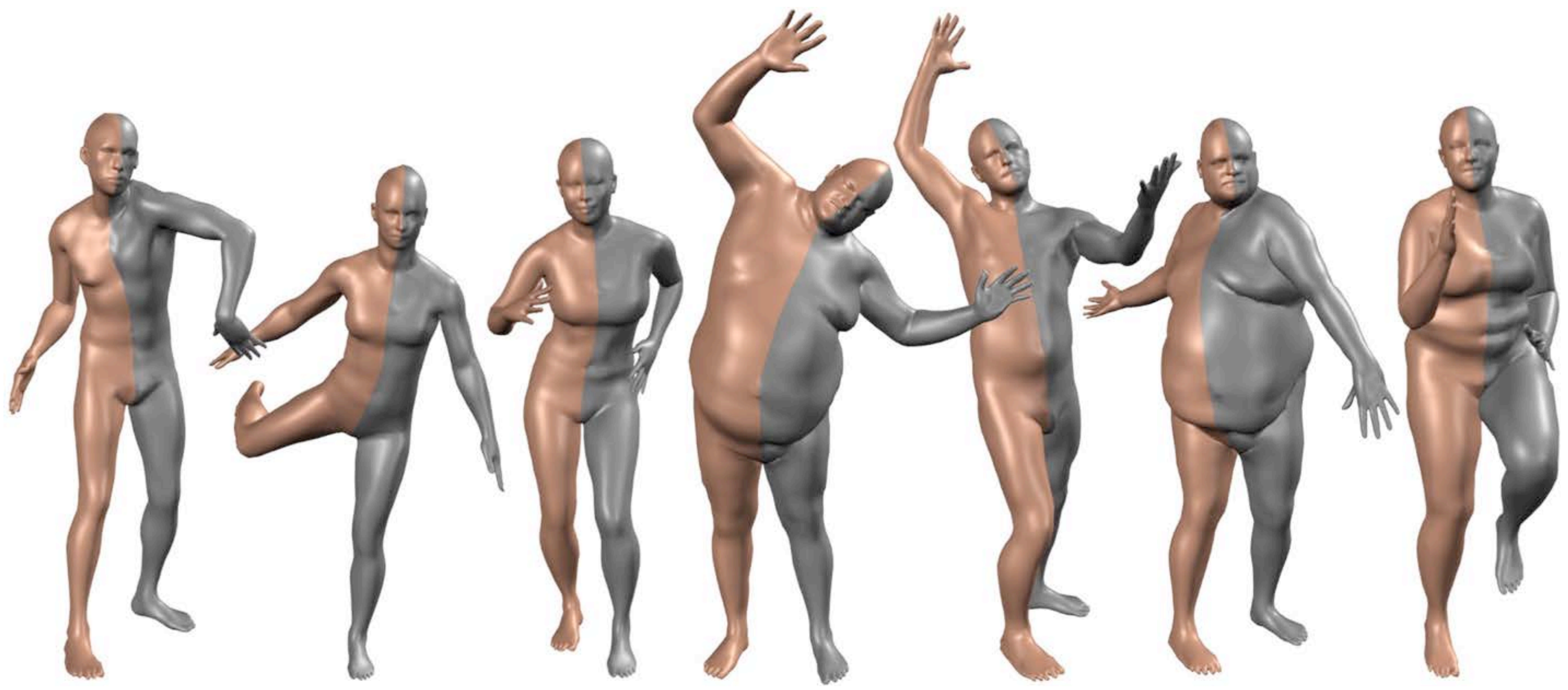


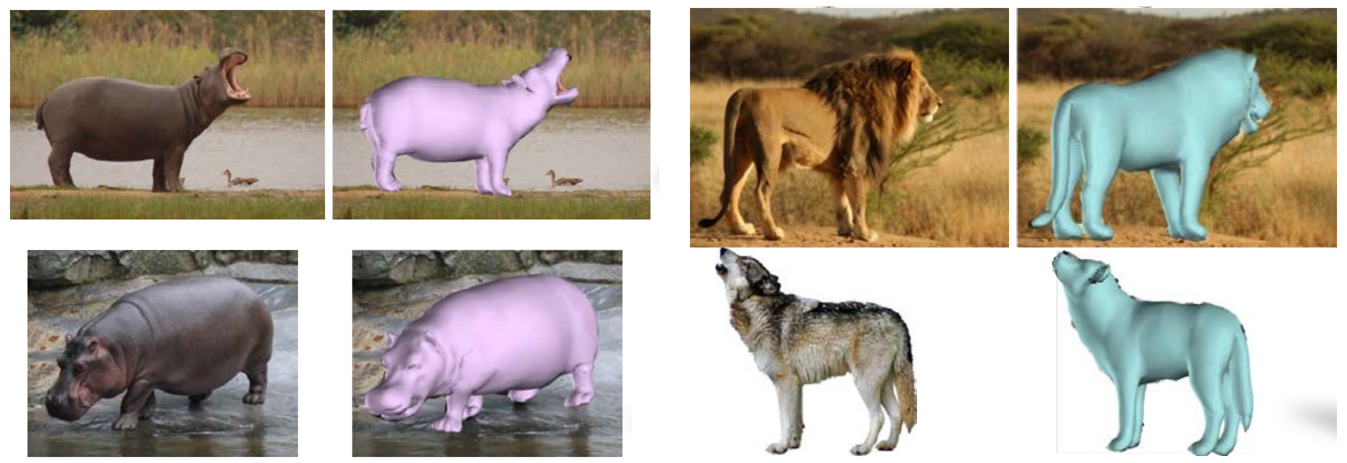
visualgenome.org [Krishna et al 2017]

Shape + Tracking

- Other vision applications might need shape modelling (possibly deformable) and/or tracking in video







[Zuffi et al 2017] [SMPL Loper et al 2015]

We'll focus on single image classification today

Classification: Instance vs Category

Instance of Aeroplane (Wright Flyer)

Category of Aeroplanes [Caltech 101]

Classification: Instance vs Category

Instance of a cat

Category of domestic cats

Taxonomy of Cats

- ↳ Mammals (Class Mammalia)
  - ↳ Therians (Subclass Theria)
    - ↳ Placental Mammals (Infraclass Placentalia)
      - ↳ Ungulates, Carnivorans, and Allies (Superorder Laurasiatheria)
        - ↳ Carnivorans (Order Carnivora)
          - ↳ Felines (Family Felidae)
            - ↳ Small Cats (Subfamily Felinae)
              - ↳ Genus *Felis*
                - ↳ Chinese Mountain Cat (*Felis bieti*)
                - ↳ Domestic Cat (*Felis catus*)
                - ↳ Jungle Cat (*Felis chaus*)
                - ↳ African Wildcat (*Felis lybica*)
                - ↳ Sand Cat (*Felis margarita*)
                - ↳ Black-footed Cat (*Felis nigripes*)
                - ↳ European Wildcat (*Felis silvestris*)

Bengal Tiger
[Omveer Choudhary]

Ocelot
[Jitze Couperus]

European Wildcat
[the wasp factory]

[inaturalist.org]

Taxonomy of Boats

vehicle

craft

watercraft

sailing vessel

sailboat

trimaran

[Deng et al 2009]

WordNet

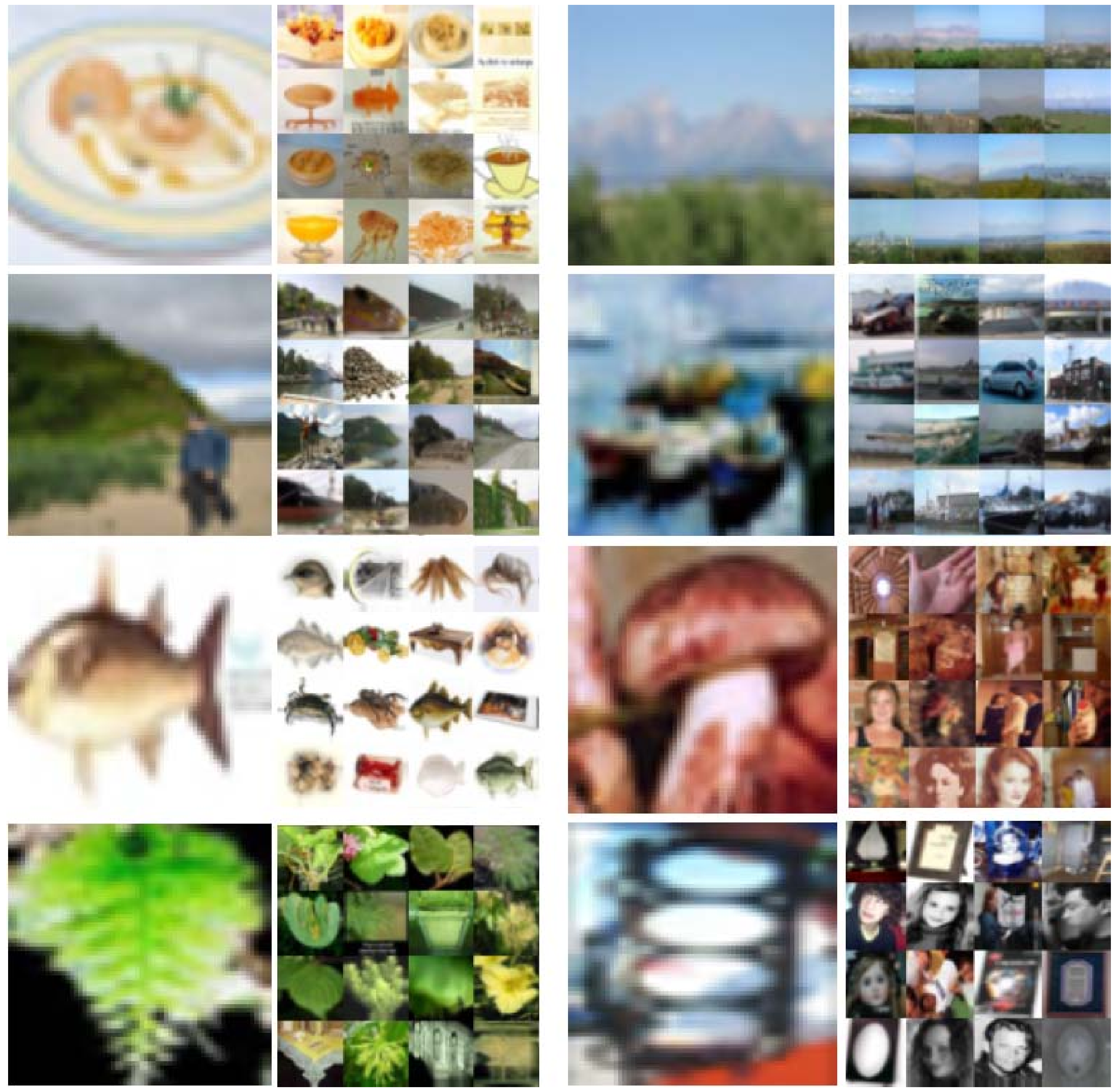
- We can use language to organise visual categories
- This is the approach taken in ImageNet [Deng et al 2009], which uses the WordNet lexical database [wordnet.princeton.edu]
- As in language, visual categories have complex relationships
- e.g., a "sail" is part of a "sailboat" which is a "watercraft"

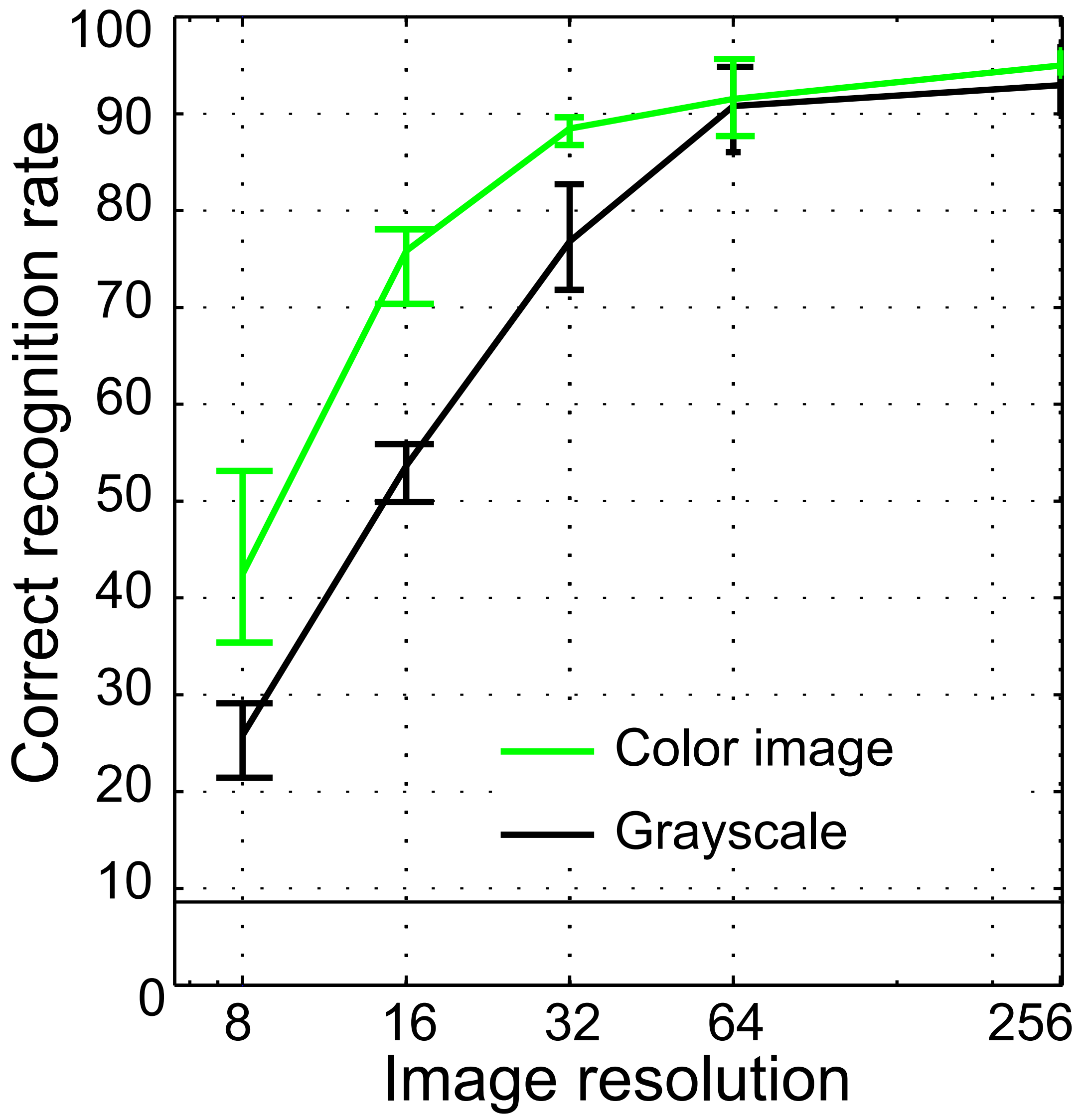
- S: (n) **sailboat**, sailing boat (a small sailing vessel; usually with a single mast)
    - ***direct hyponym*** / *full hyponym*
        - S: (n) catboat (a sailboat with a single mast set far forward)
        - S: (n) sharpie (a shallow-draft sailboat with a sharp prow, flat bottom, and triangular sail; formerly used along the northern Atlantic coast of the United States)
        - S: (n) trimaran (a fast sailboat with 3 parallel hulls)
    - *part meronym*
    - *direct hypernym* / *inherited hypernym* / *sister term*
        - S: (n) sailing vessel, sailing ship (a vessel that is powered by the wind; often having several masts)

If we call a "sailboat" a watercraft, is this wrong? What if we call it a "sail"?

Tiny Image Dataset

- Precursor to ImageNet and CIFAR10/100
- 80 million images collected via image search circa 2008 using 75,062 noun synsets from WordNet (labels are noisy)
- Very small images (32x32xRGB) used to minimise storage
- Note human performance is still quite good at this scale!



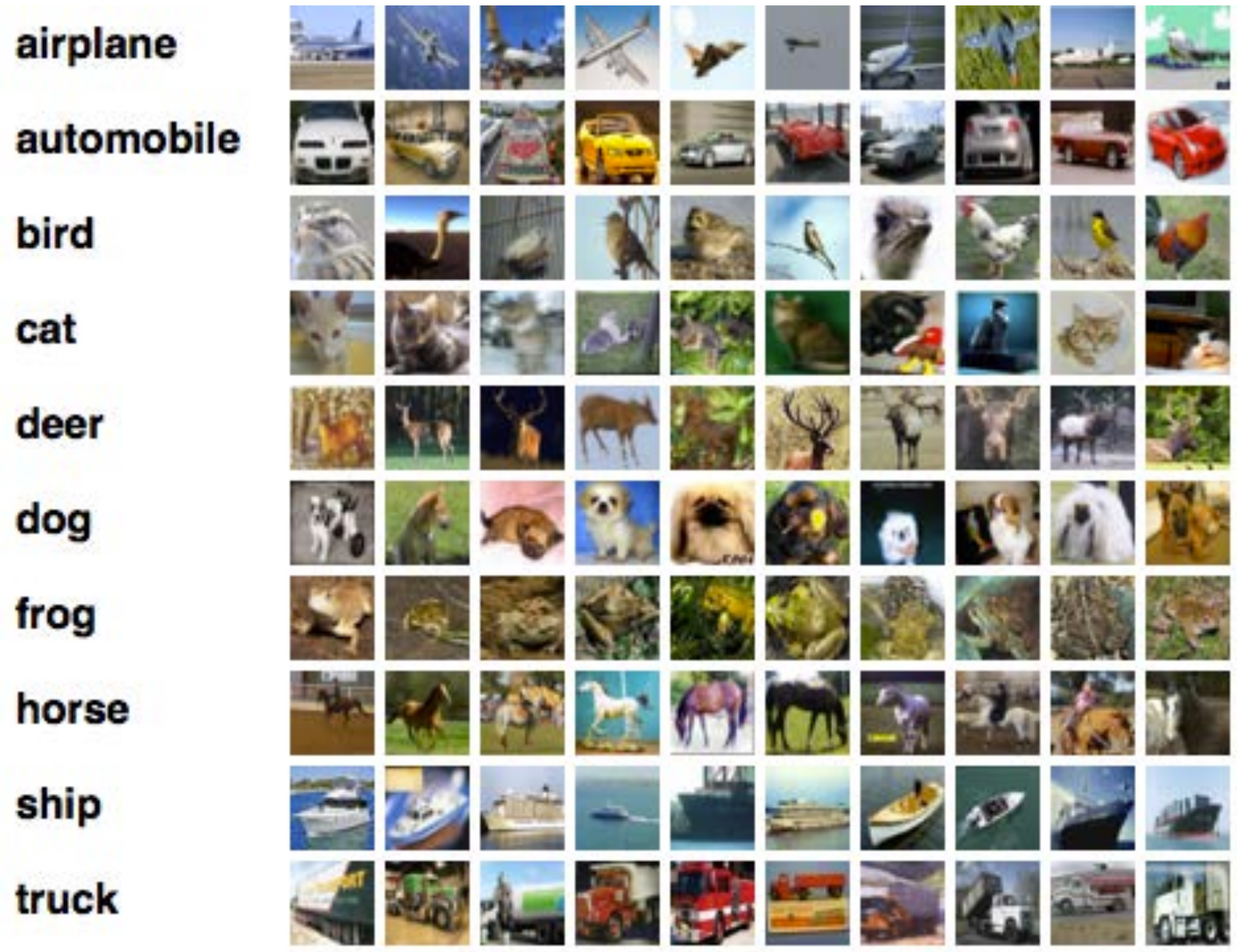


a) Scene recognition

[Torralba Freeman Fergus 2008]

CIFAR10 Dataset

- Hand labelled set of 10 categories from Tiny Images dataset
- 60,000 32x32 images in 10 classes (50k train, 10k test)



Good test set for visual recognition problems

Classification

Problem:

Assign new observations into one of a fixed set of categories (classes)

Key Idea(s):

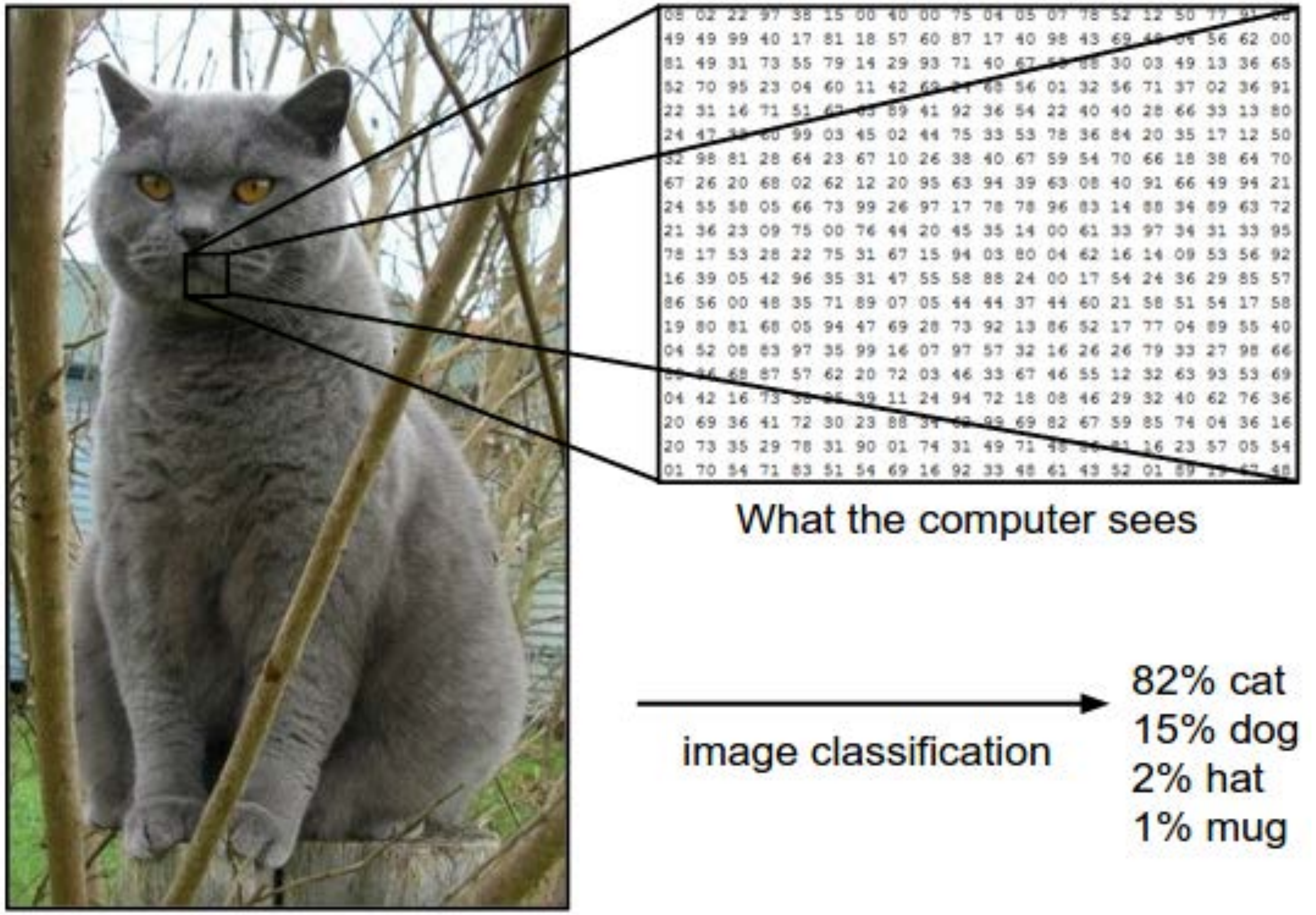
Build a model of data in a given category based on observations of instances in that category

Classification

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

→ cat

Classification



Classification

A **classifier** is a procedure that accepts as input a set of features and outputs a class **label**

Classifiers can be binary (face vs. not-face) or multi-class (cat, dog, horse, ...).

We build a classifier using a **training set** of labelled examples $\{(\mathbf{x}_i, y_i)\}$, where each $\mathbf{x}_i$ is a feature vector and each y_i is a class label.

Given a previously unseen observation, we use the classifier to predict its class label.

Classification

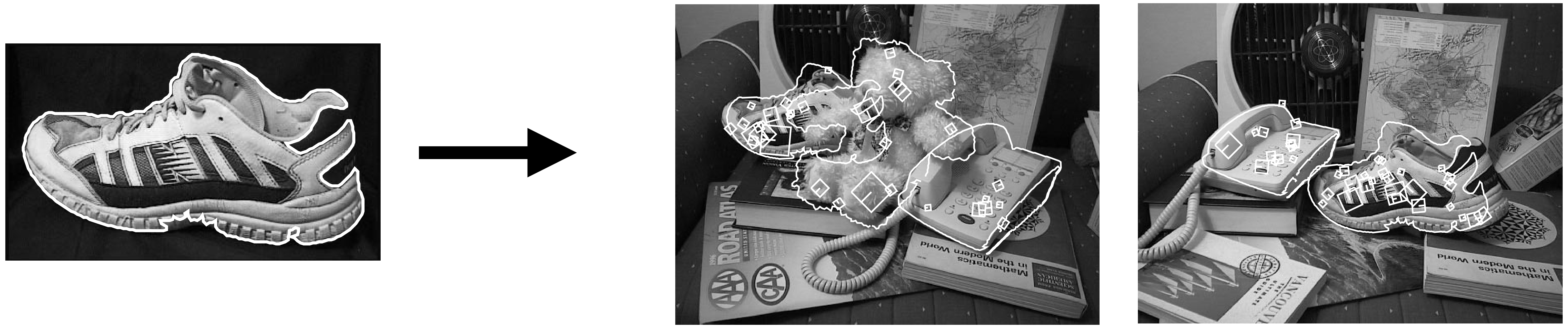
- Collect a database of images with labels
- Use ML to train an image classifier
- Evaluate the classifier on test images

Label

Feature vector computed from the image

Instance Recognition using Local Features

- Feature-based object instance recognition is similar to image registration (2D) or camera pose estimation (3D):

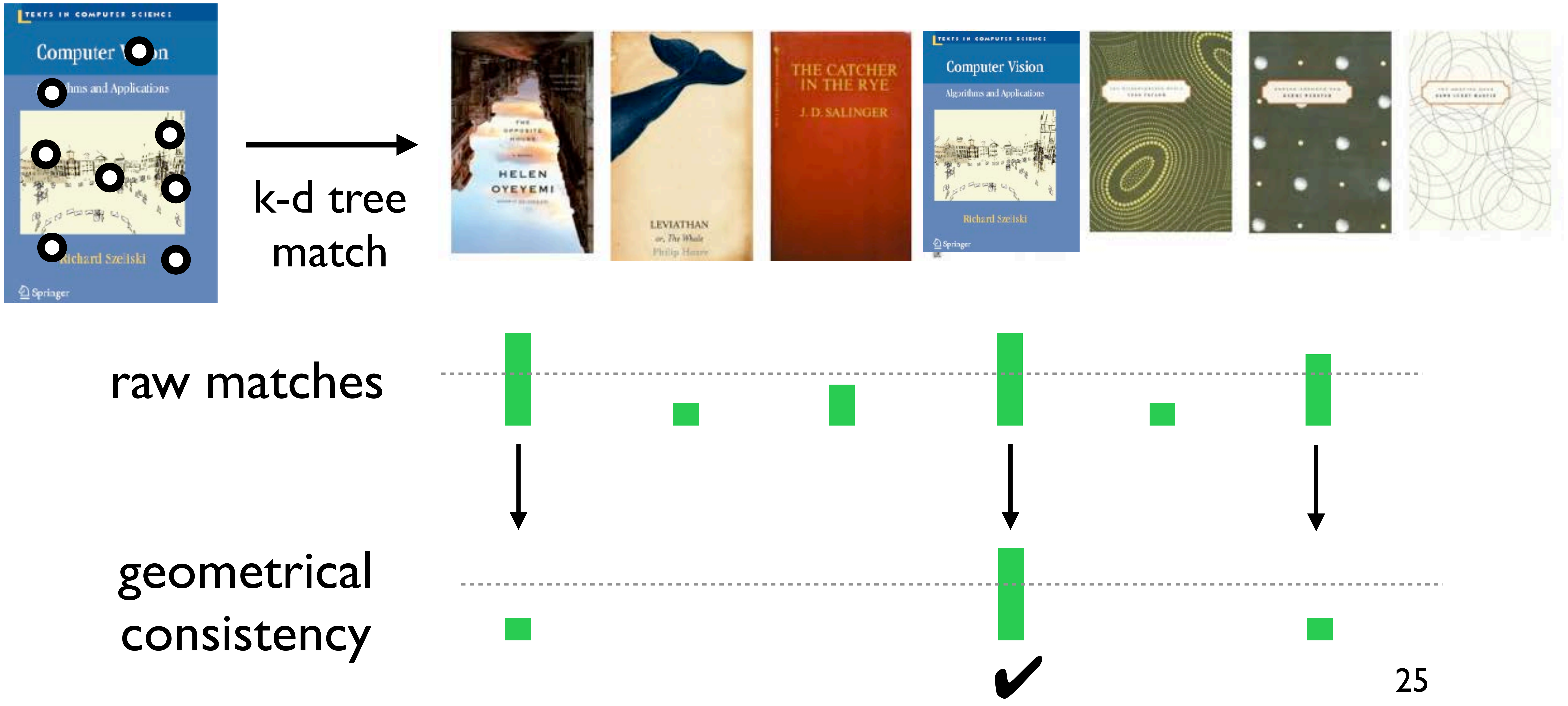


1. Detect Local Features (e.g., SIFT) in all images
2. Match Features using Nearest Neighbours
3. Find geometrically consistent matches using RANSAC (with Affine/Homography or Fundamental matrix)

- The final stage is to verify the match, e.g., require that # consistent matches > threshold

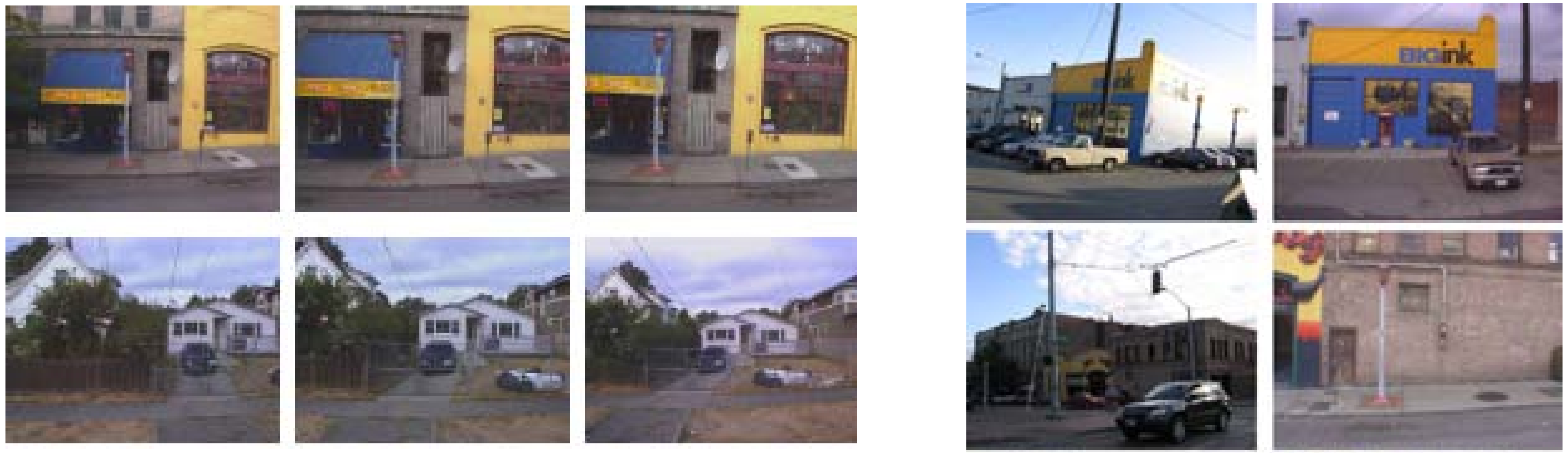
Scaling Local Feature Recognition

- To avoid performing all pairwise comparisons $O(n^2)$:
- Match query descriptors to entire database using k-d tree
- Select subset with max # raw matches and check geometry

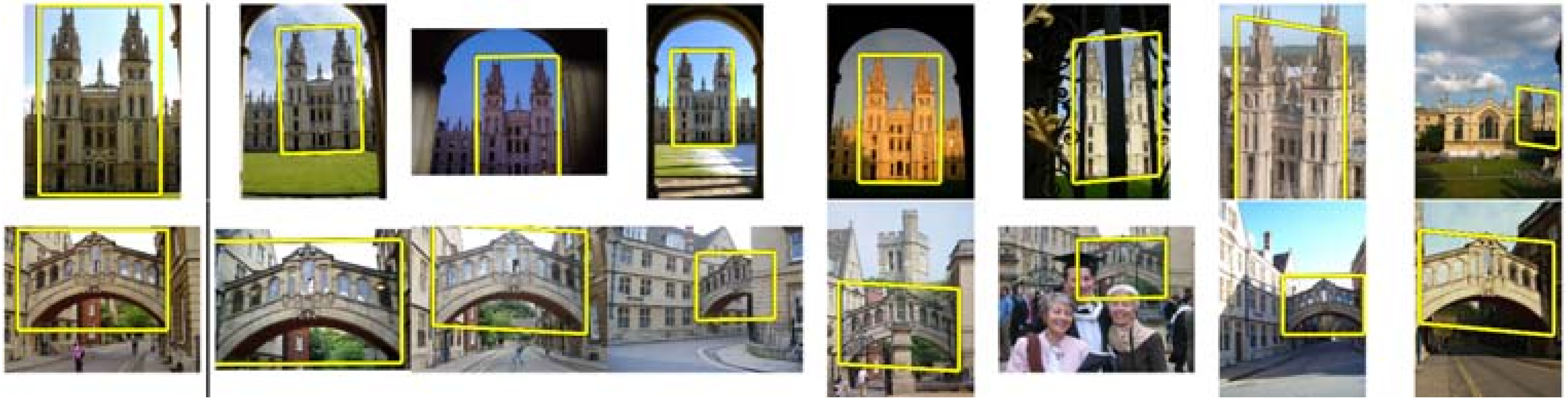


Application: Location Recognition

- Find photo in streetside imagery



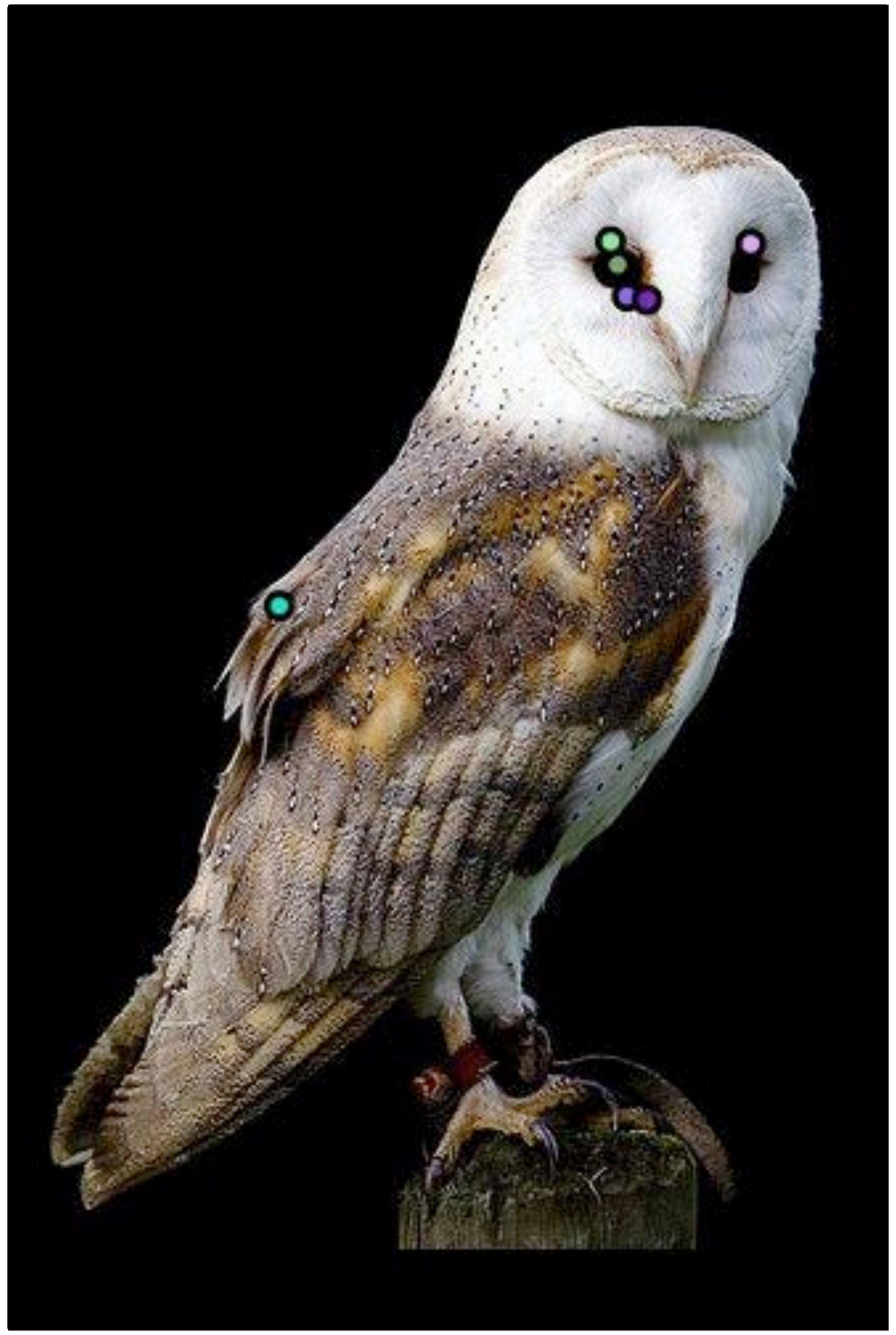
[Schindler Brown Szeliski 2007]



[Philbin et al 2007]

Local Feature Recognition Failures

- Features + RANSAC fails with large appearance variation, e.g., most object categories and some instance problems



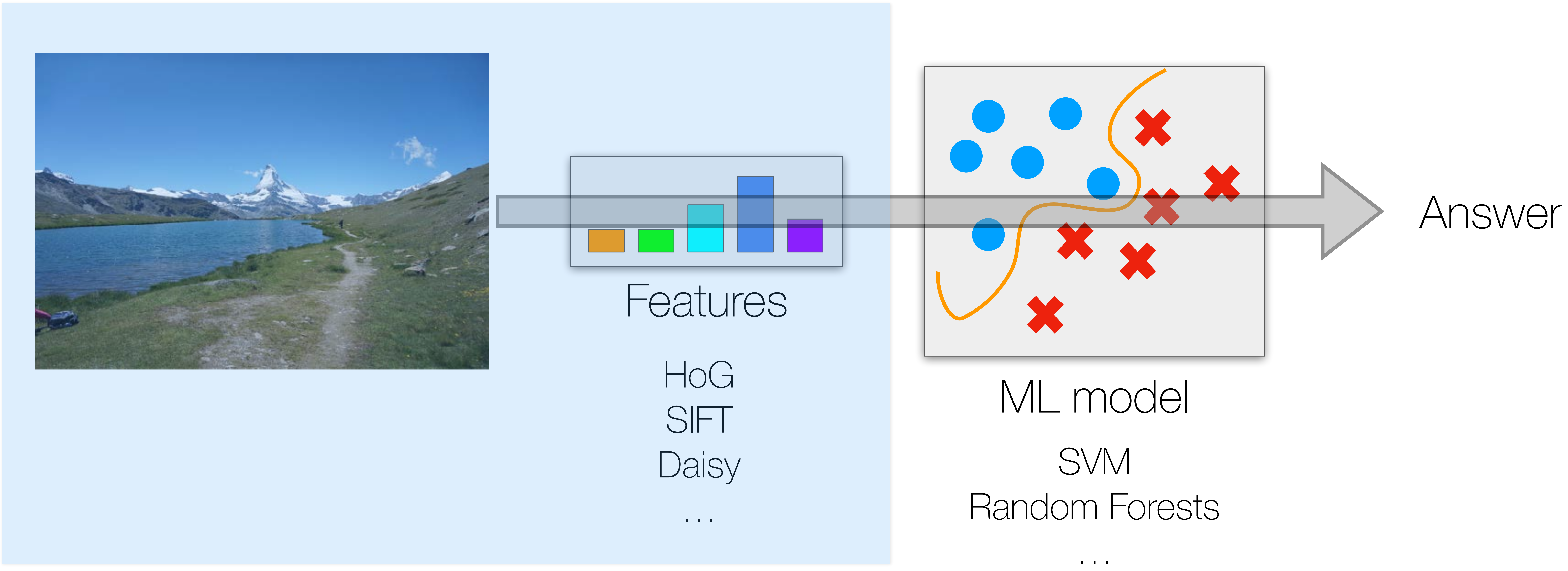
Few correct matches

Local Feature Recognition Failures

- Features + RANSAC fails with large appearance variation, e.g., most object categories and some instance problems

No correct matches

Traditional Image Classification Pipeline



How do we then represent images?

Visual **Words**

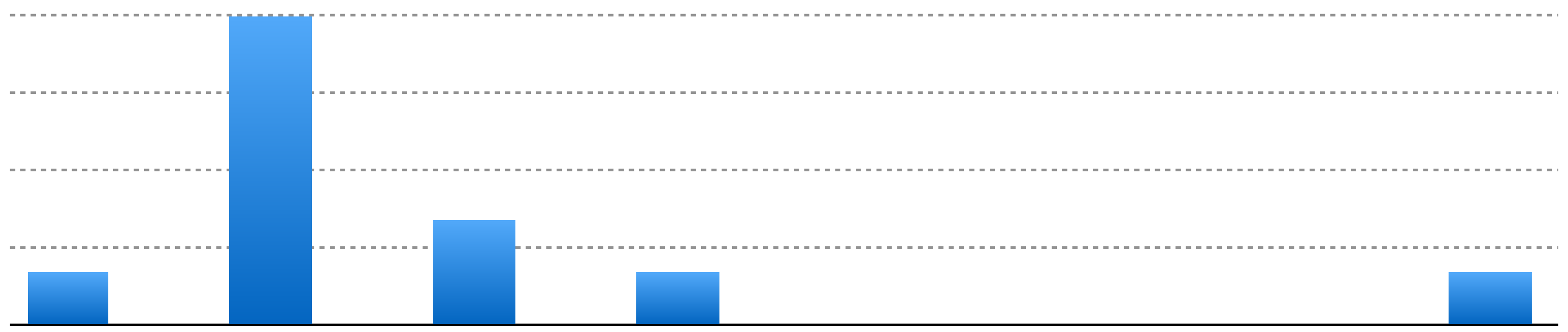
Many algorithms for image classification accumulate evidence on the basis of **visual words**.

To classify a text document (e.g. as an article on sports, entertainment, business, politics) we might find patterns in the occurrences of certain words.

Vector Space Model

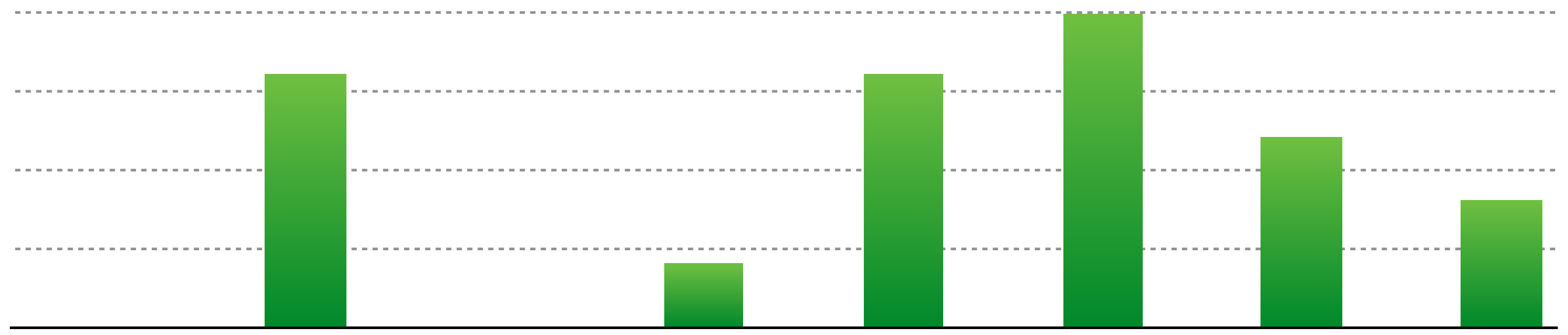
G. Salton. ‘Mathematics and Information Retrieval’ Journal of Documentation, 1979


The Newsp
DARPA Selects Carnegie Me



1	6	2	1	0	0	0	1
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor


Tartan Tim
Bio-Inspired Robotic Device



0	4	0	1	4	5	3	2
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor

http://www.fodey.com/generators/newspaper/snippet.asp

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Vector Space Model

A document (datapoint) is a vector of counts over each word (feature)

$$\boldsymbol{v}_d = [n(w_{1,d}) \quad n(w_{2,d}) \quad \cdots \quad n(w_{T,d})]$$

$n(\cdot)$ counts the number of occurrences

just a histogram over words

What is the similarity between two documents?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Vector Space Model

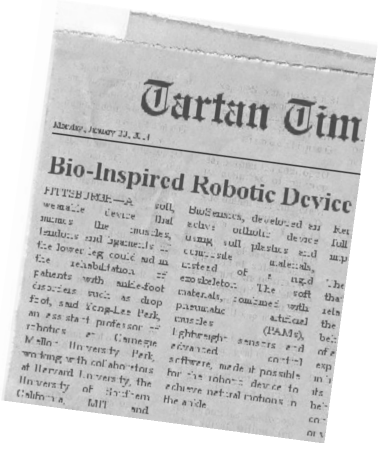
A document (datapoint) is a vector of counts over each word (feature)

$$\boldsymbol{v}_d = [n(w_{1,d}) \quad n(w_{2,d}) \quad \cdots \quad n(w_{T,d})]$$

$n(\cdot)$ counts the number of occurrences

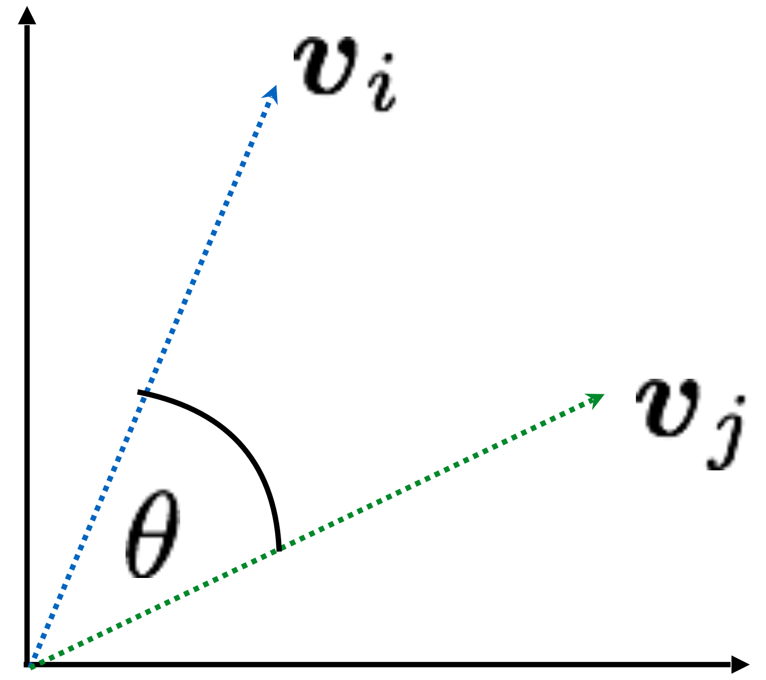
just a histogram over words

What is the similarity between two documents?



Use any distance you want but the cosine distance is fast and well designed for high-dimensional vector spaces:

$$d(\boldsymbol{v}_i, \boldsymbol{v}_j) = \cos\theta = \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{||\boldsymbol{v}_i|| \, ||\boldsymbol{v}_j||}$$



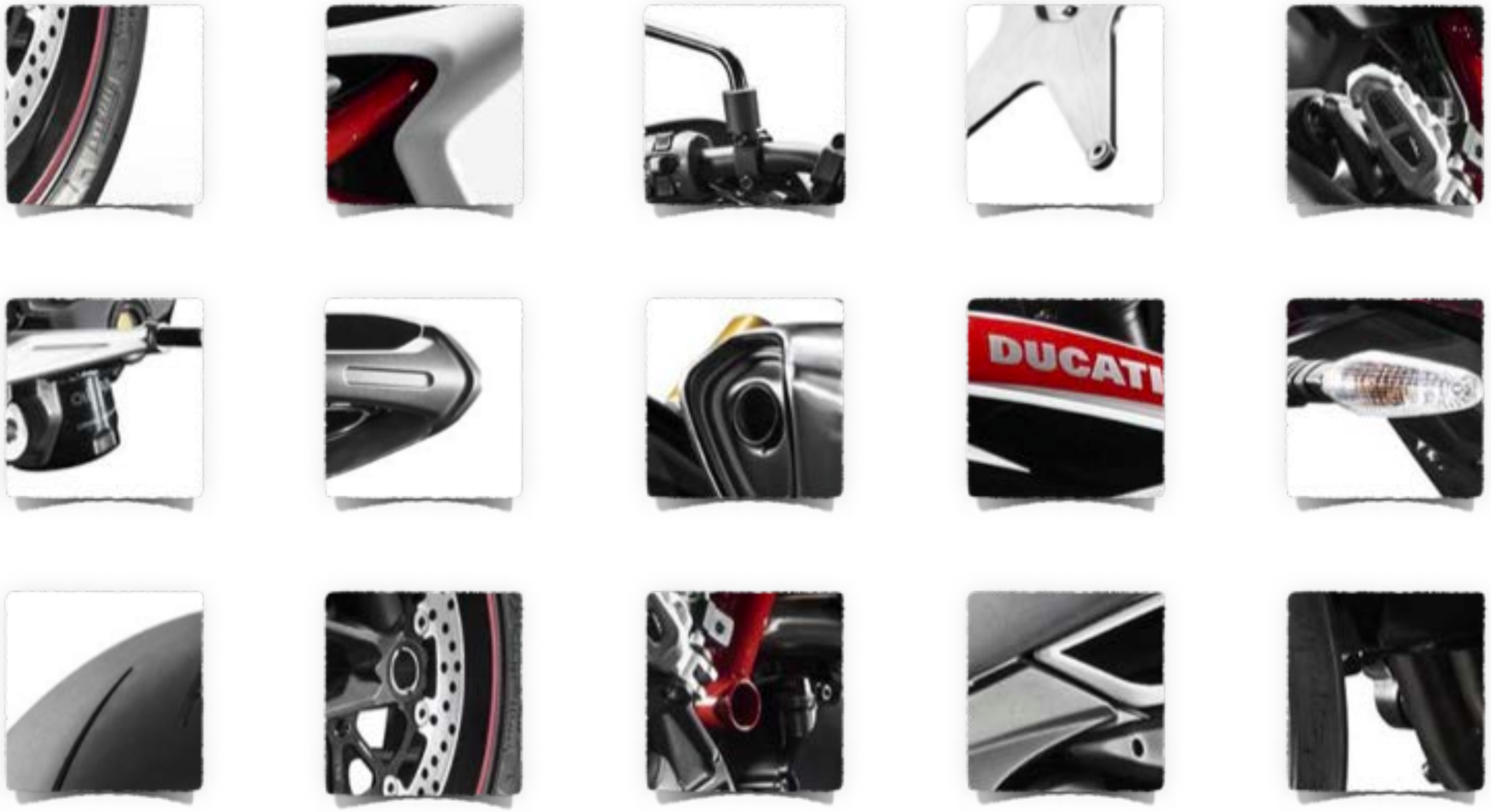
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Visual **Words**

In images, the equivalent of a **word** is a **local image patch**. The local image patch is described using a descriptor such as SIFT.

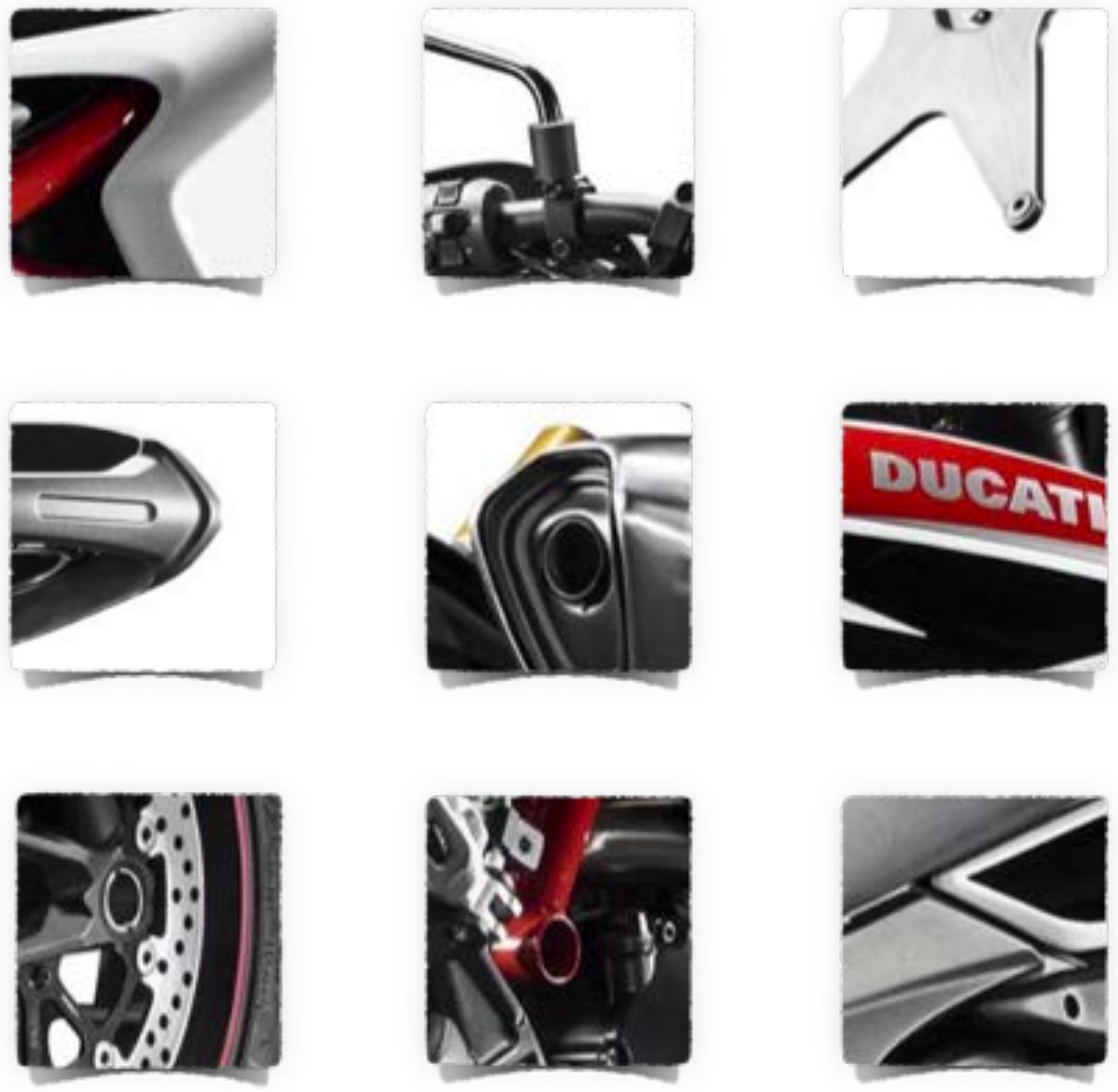
We construct a **vocabulary** or **codebook** of local descriptors, containing representative local descriptors.

What **Objects** do These Parts Belong To?



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Some local feature are very informative



An object as

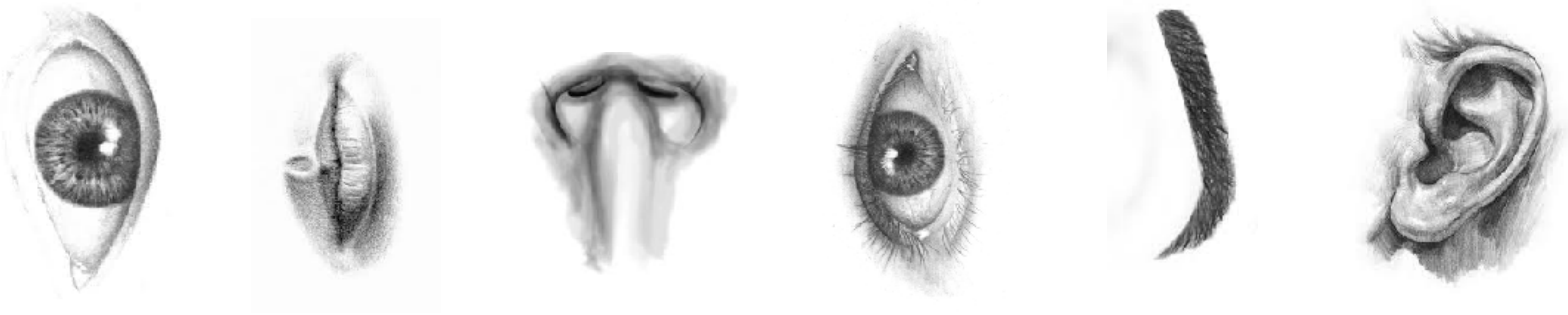
a collection of local features

(bag-of-features)

- deals well with occlusion
- scale invariant
- rotation invariant

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

(**not so**) Crazy Assumption



spatial information of local features
can be ignored for object recognition (i.e., verification)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Standard **Bag-of-Words** Pipeline (for image classification)

Dictionary Learning:
Learn Visual Words using clustering

Encode:
build Bags-of-Words (BOW) vectors
for each image

Classify:
Train and test data using BOWs

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Standard **Bag-of-Words** Pipeline (for image classification)

Dictionary Learning:
Learn Visual Words using clustering

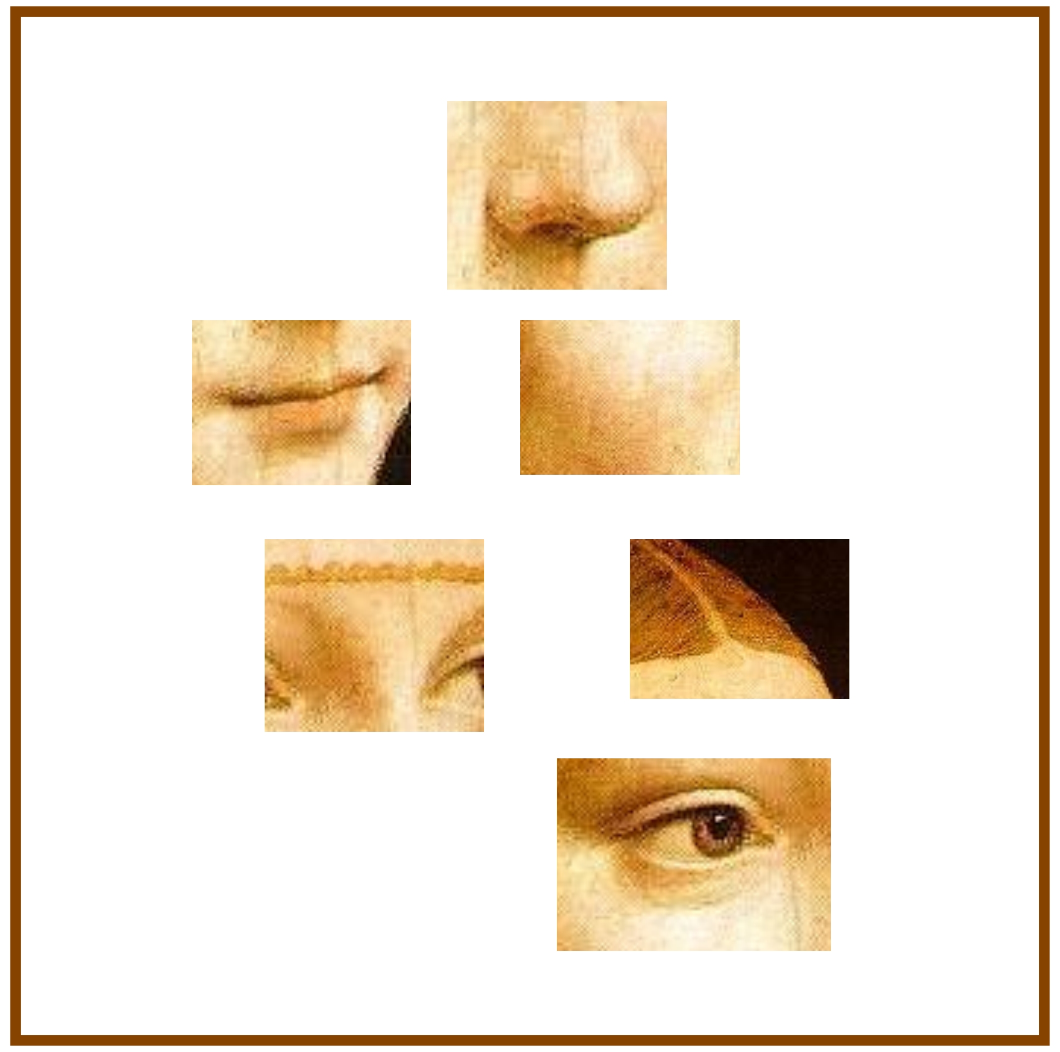
Encode:
build Bags-of-Words (BOW) vectors
for each image

Classify:
Train and test data using BOWs

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

1. **Dictionary Learning**: Learn Visual Words using Clustering

1. **Extract features** (e.g., SIFT) from images



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

1. **Dictionary Learning**: Learn Visual Words using Clustering

2. **Learn visual dictionary** (e.g., K-means clustering)

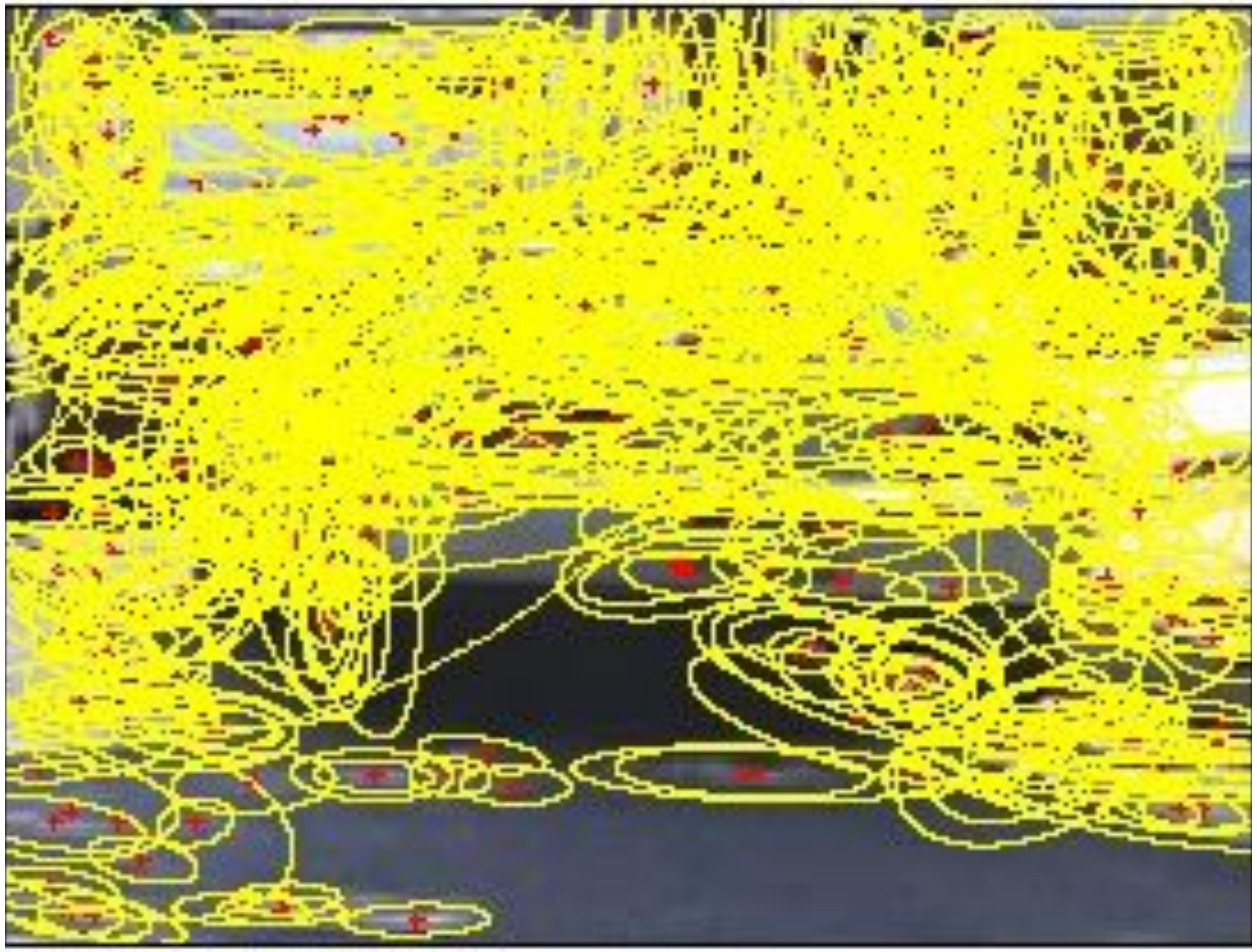
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

What **Features** Should We Extract?

- Regular grid
  Vogel & Schiele, 2003
  Fei-Fei & Perona, 2005

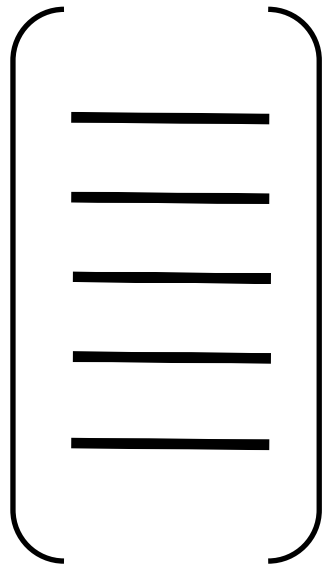
- Interest point detector
  Csurka et al. 2004
  Fei-Fei & Perona, 2005
  Sivic et al. 2005

- Other methods
  Random sampling (Vidal-Naquet & Ullman, 2002)
  Segmentation-based patches (Barnard et al. 2003)



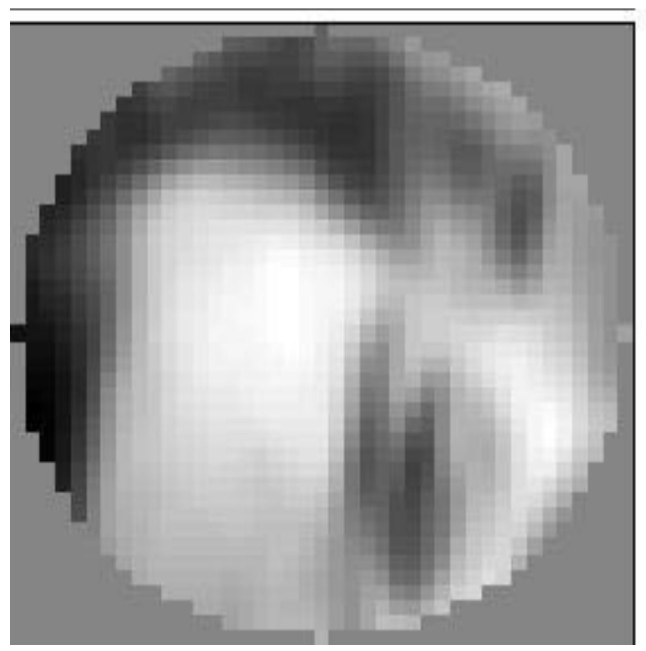
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Extracting **SIFT** Patches

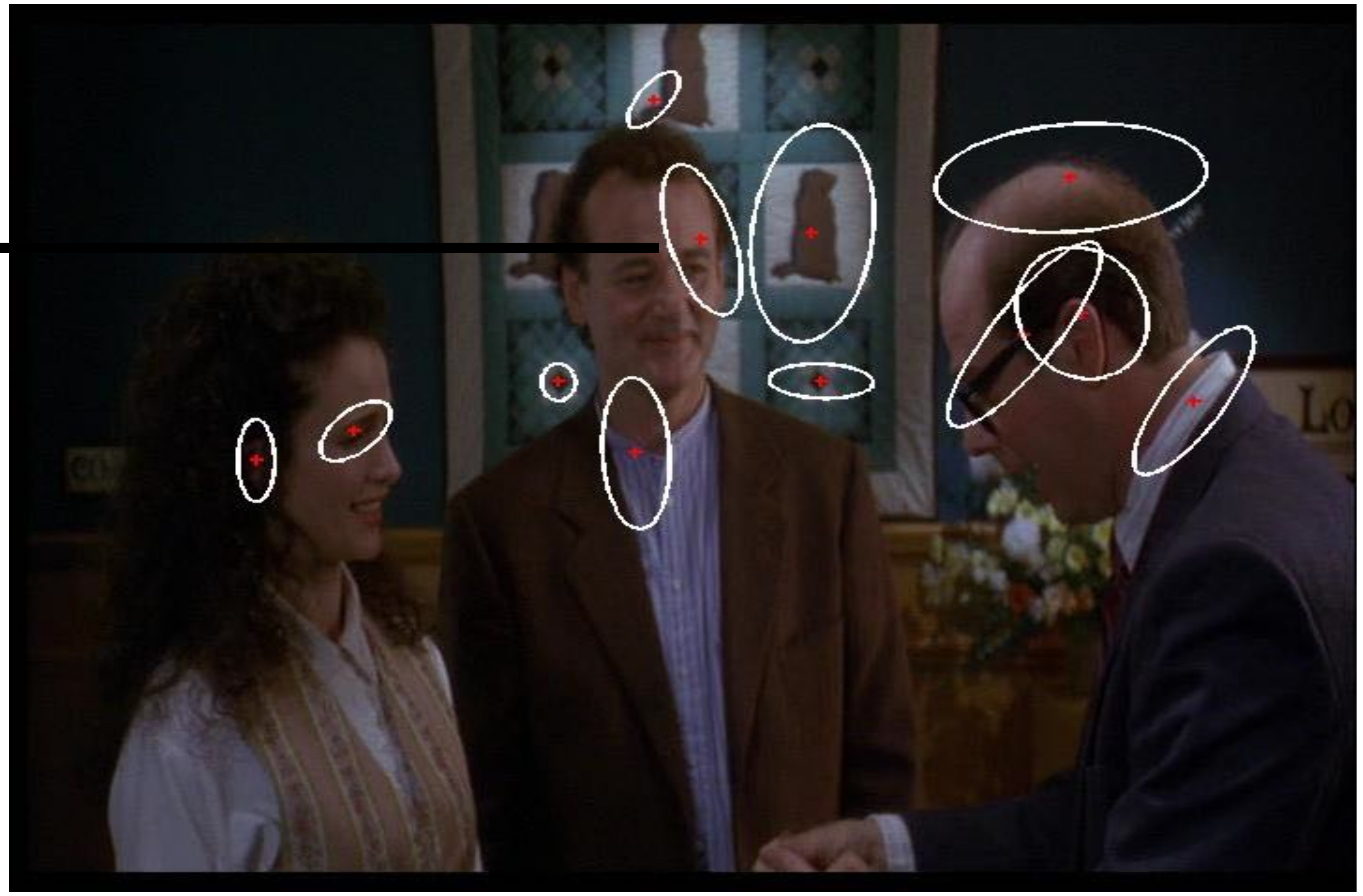


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

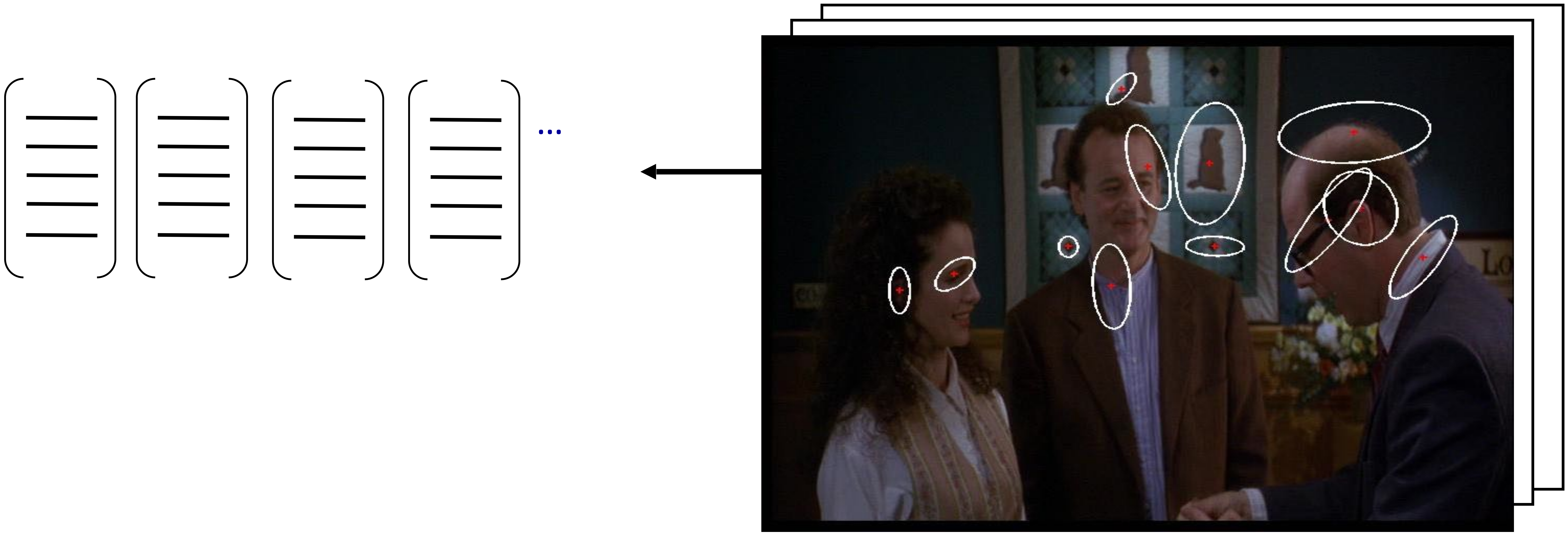
[Mikojaczyk and Schmid '02]

[Mata, Chum, Urban & Pajdla, '02]

[Sivic & Zisserman, '03]

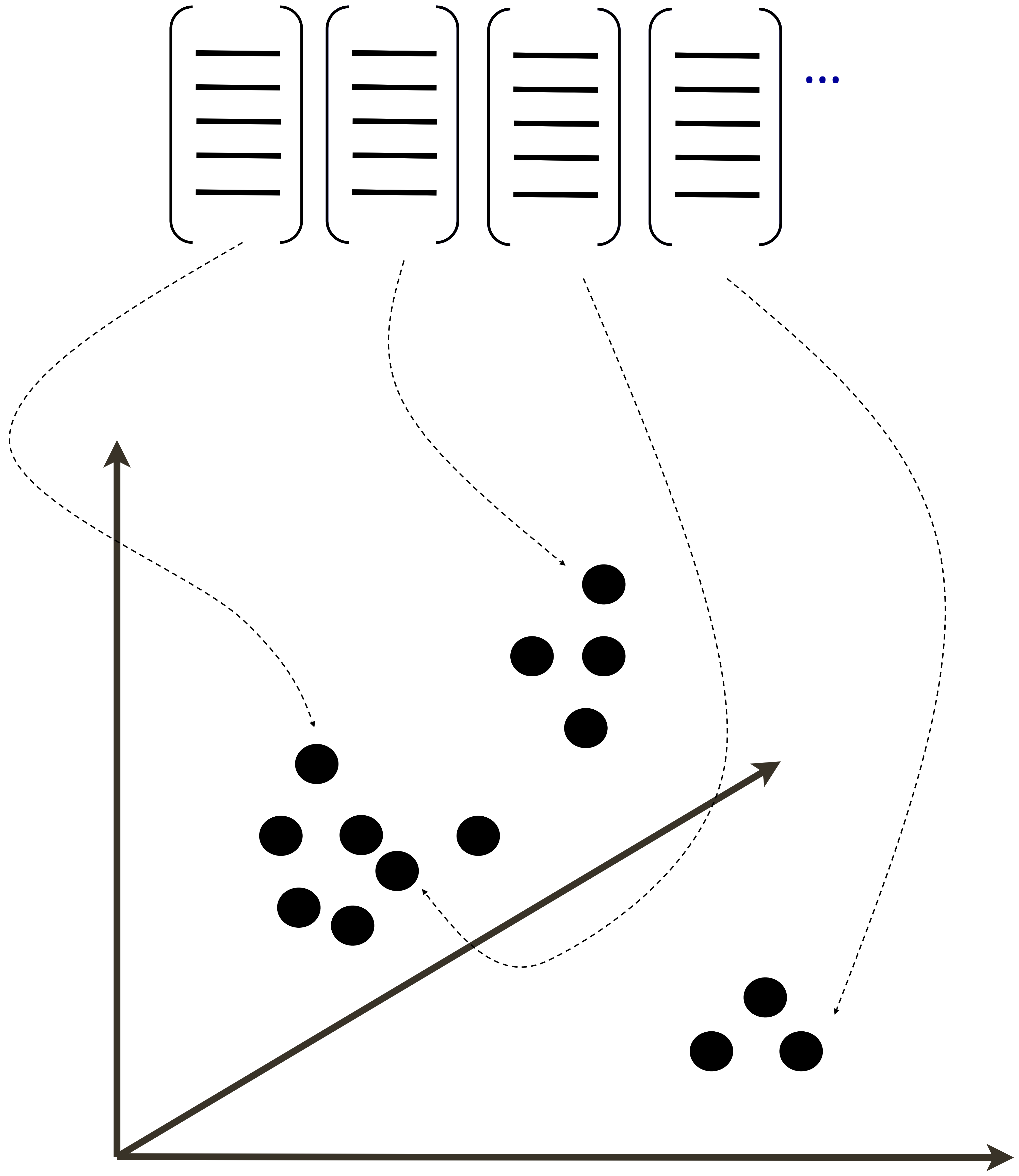
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Extracting **SIFT** Patches



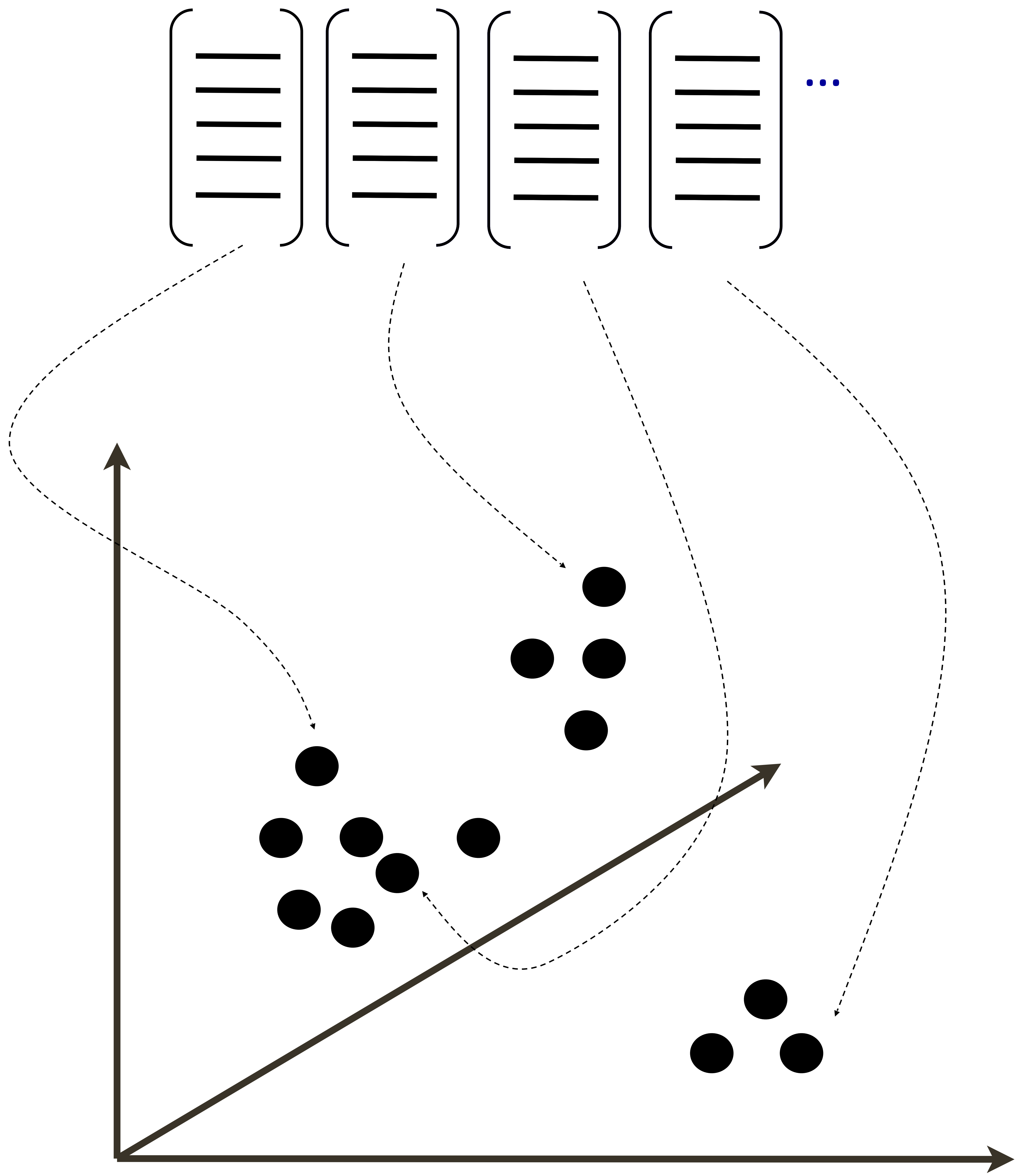
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

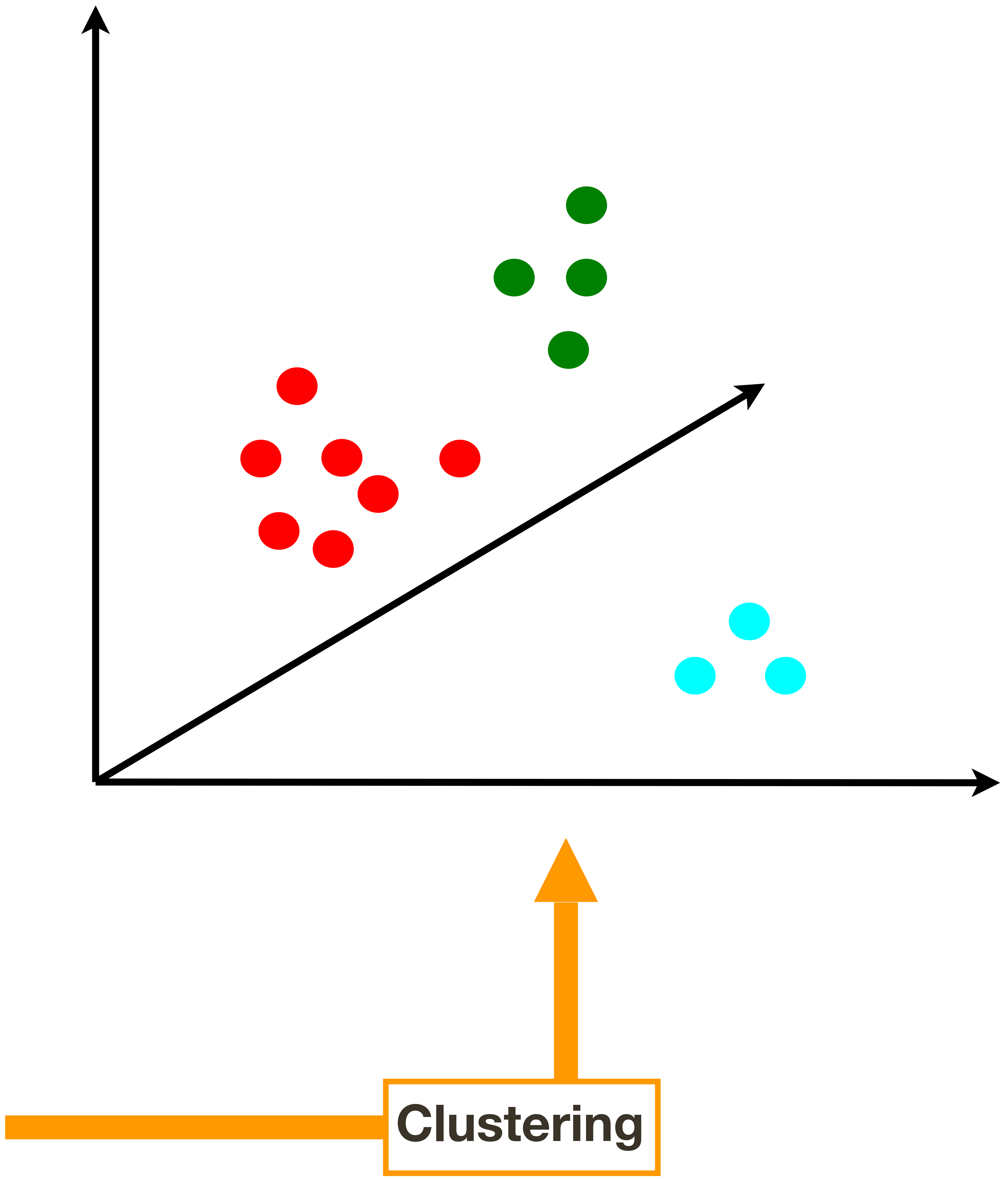
Creating **Dictionary**



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

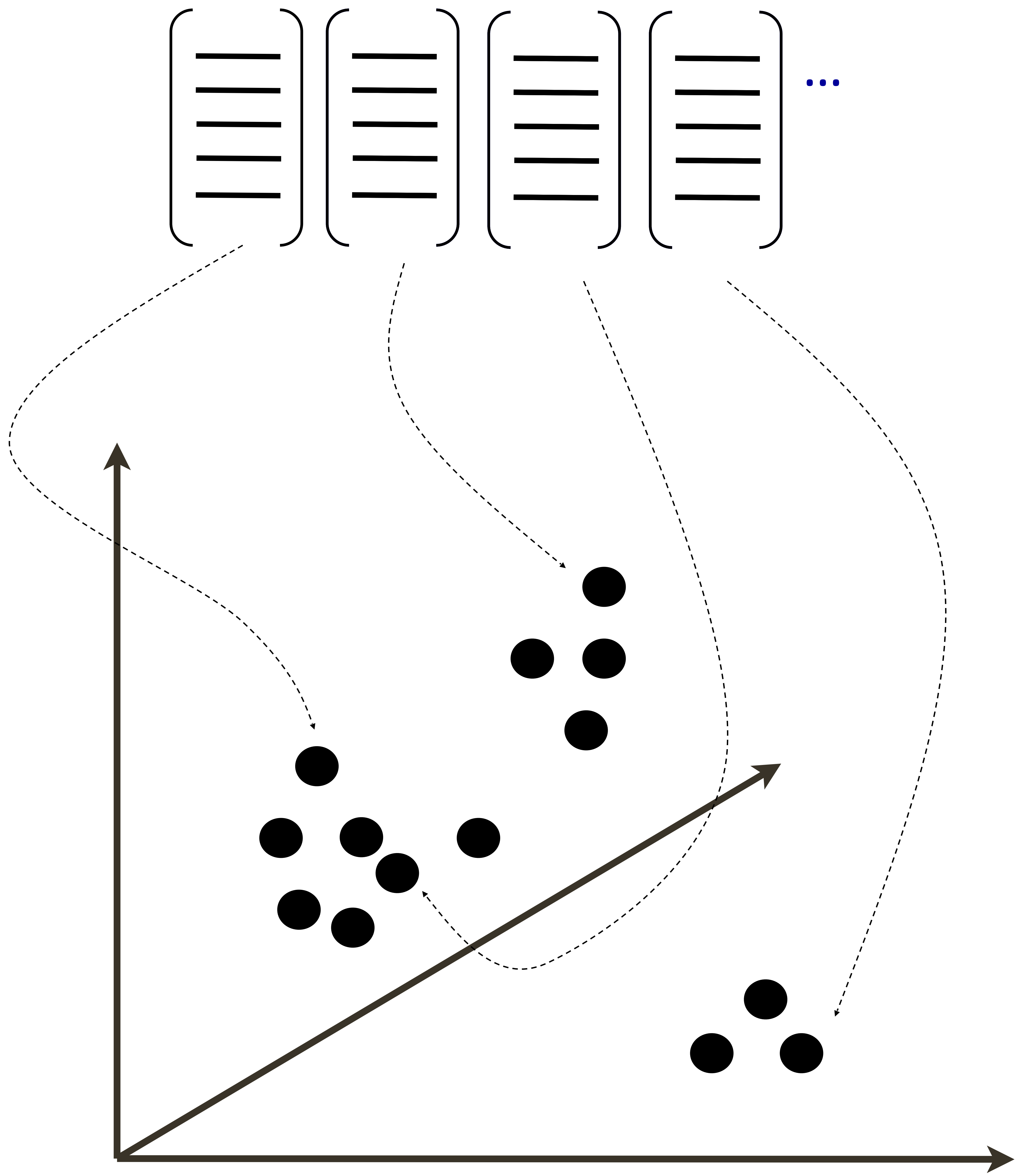
Creating **Dictionary**

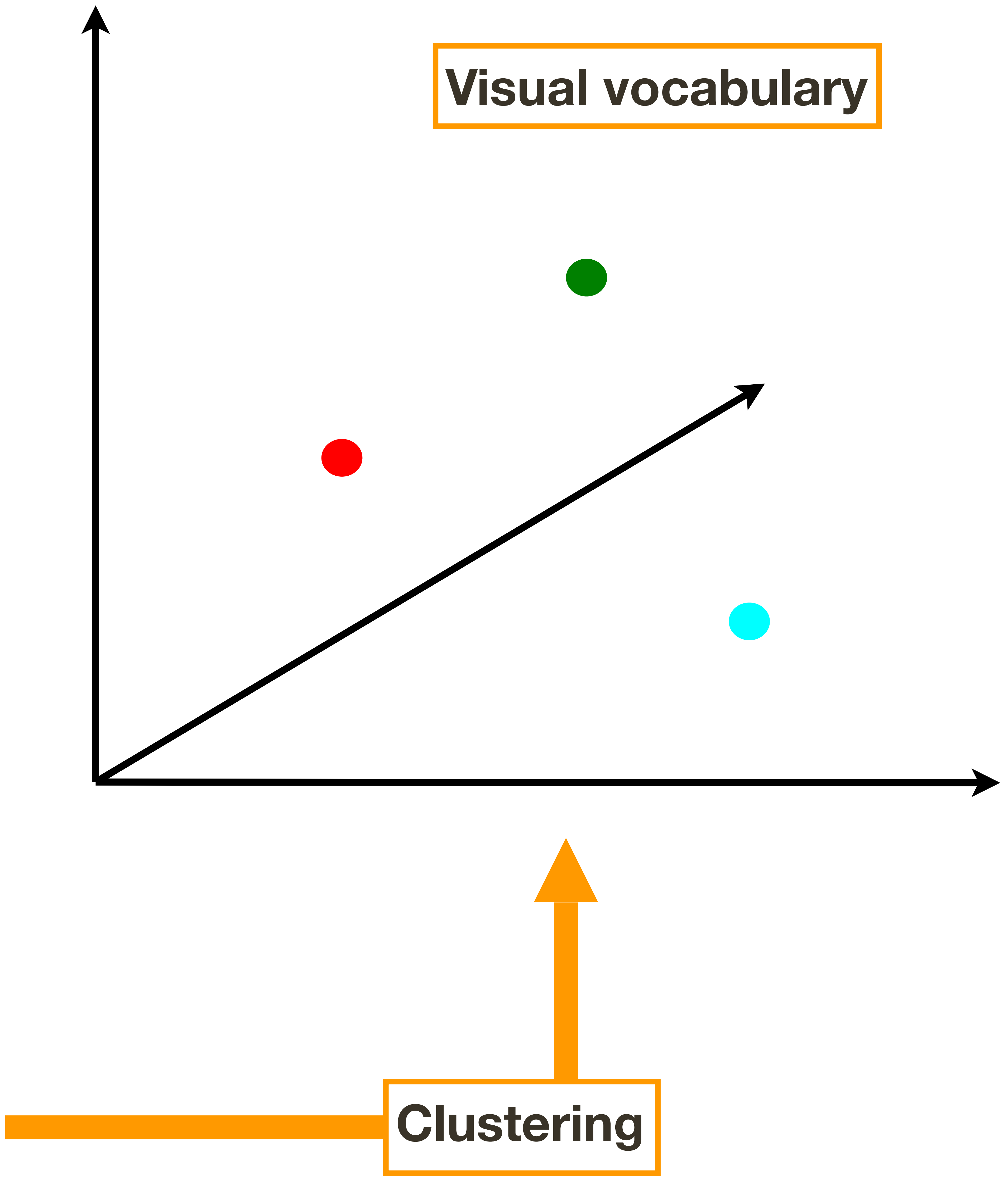




Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Creating **Dictionary**





Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

K-means clustering

K-Means Clustering

Assume we know how many clusters there are in the data - denote by K

Each cluster is represented by a cluster center, or mean

Our objective is to minimize the representation error (or quantization error) in letting each data point be represented by some cluster center

Minimize

$$\sum_{i \in clusters} \left\{ \sum_{j \in i^{th} \ cluster} ||x_j - \mu_i||^2 \right\}$$

K-Means Clustering

K-means clustering alternates between two steps:

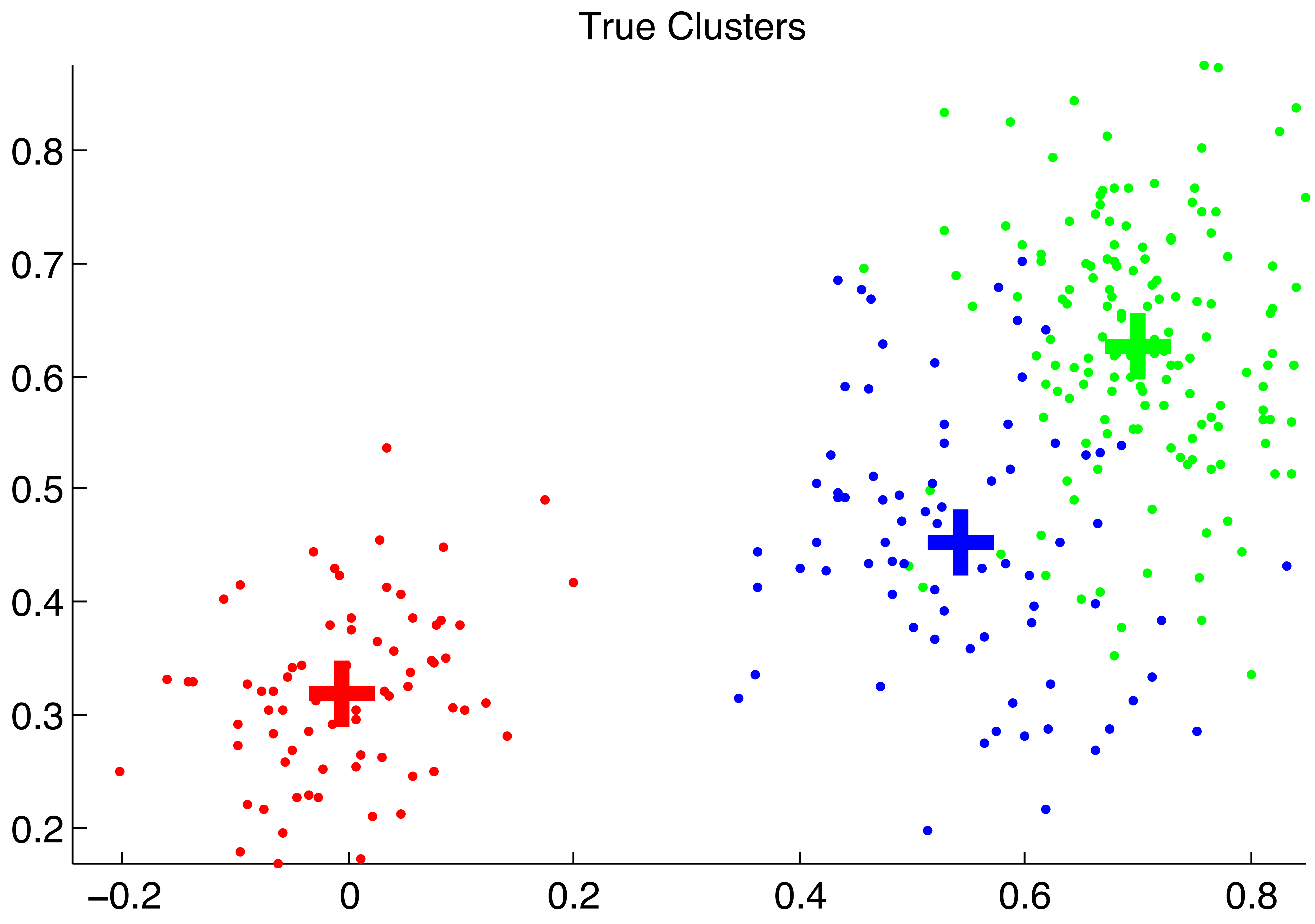
1. Assume the cluster centers are known (fixed). Assign each point to the closest cluster center.

2. Assume the assignment of points to clusters is known (fixed). Compute the best center for each cluster, as the mean of the points assigned to the cluster.

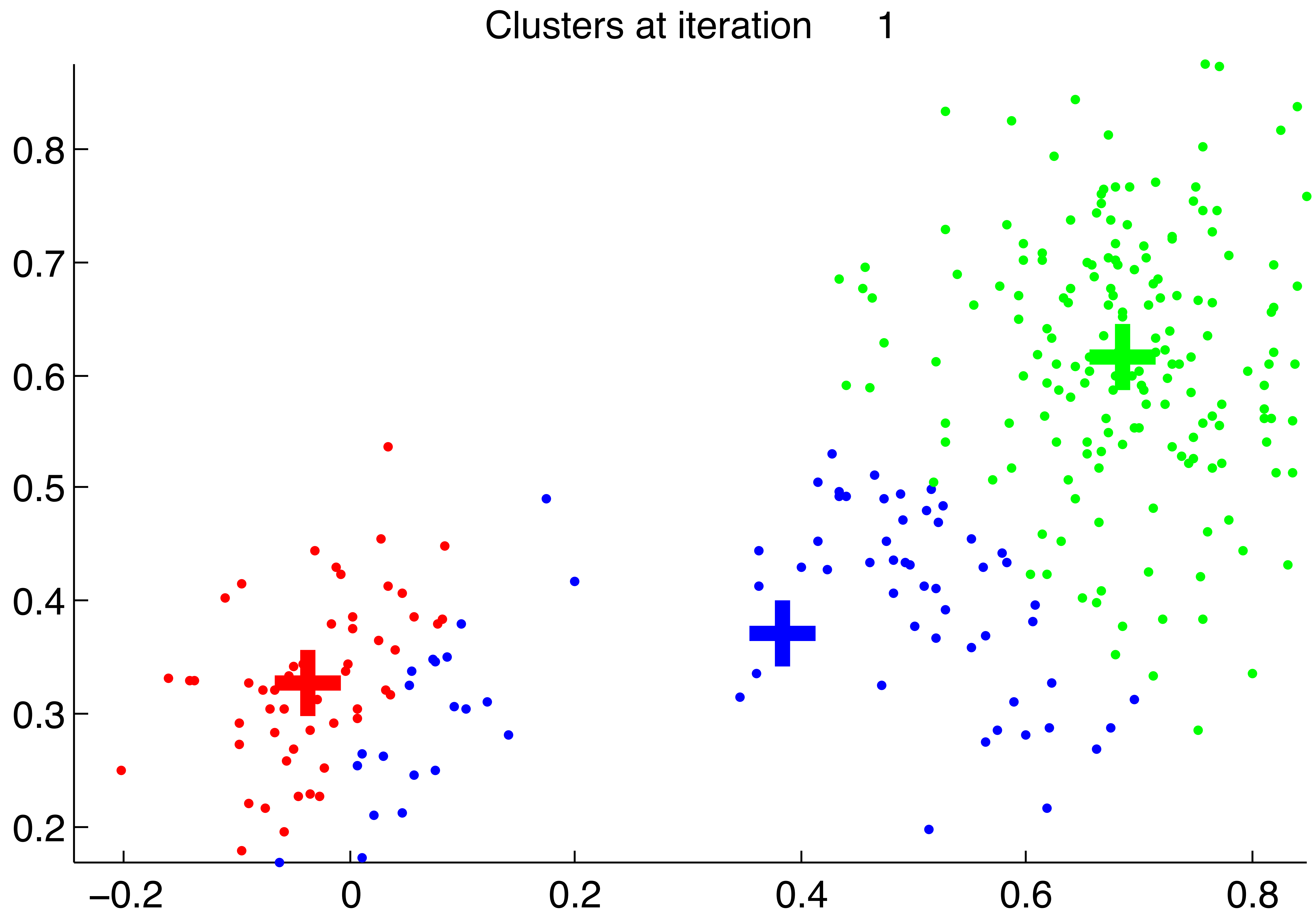
The algorithm is initialized by choosing K random cluster centers

K-means converges to a local minimum of the objective function
— Results are initialization dependent

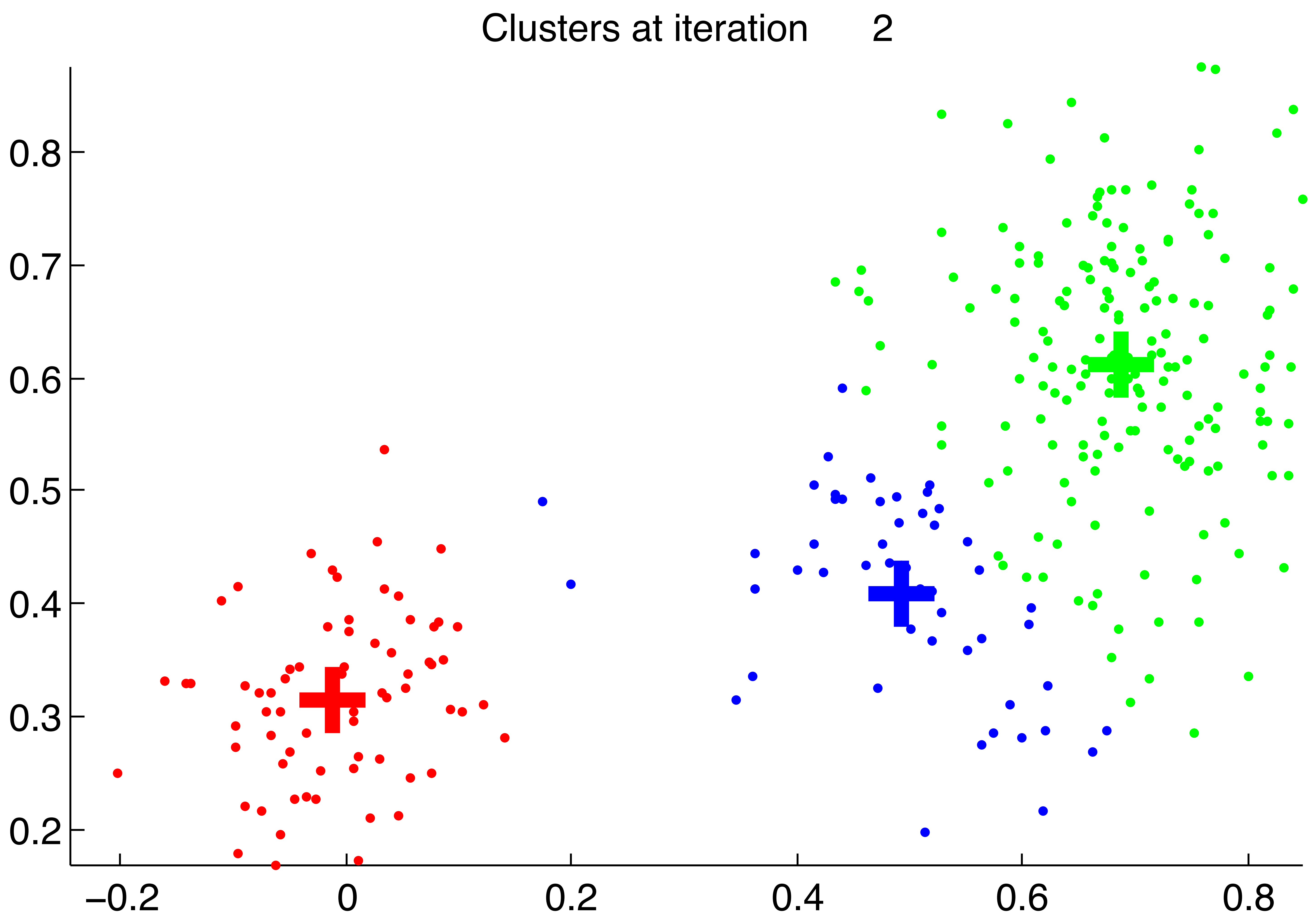
K-Means Clustering Example



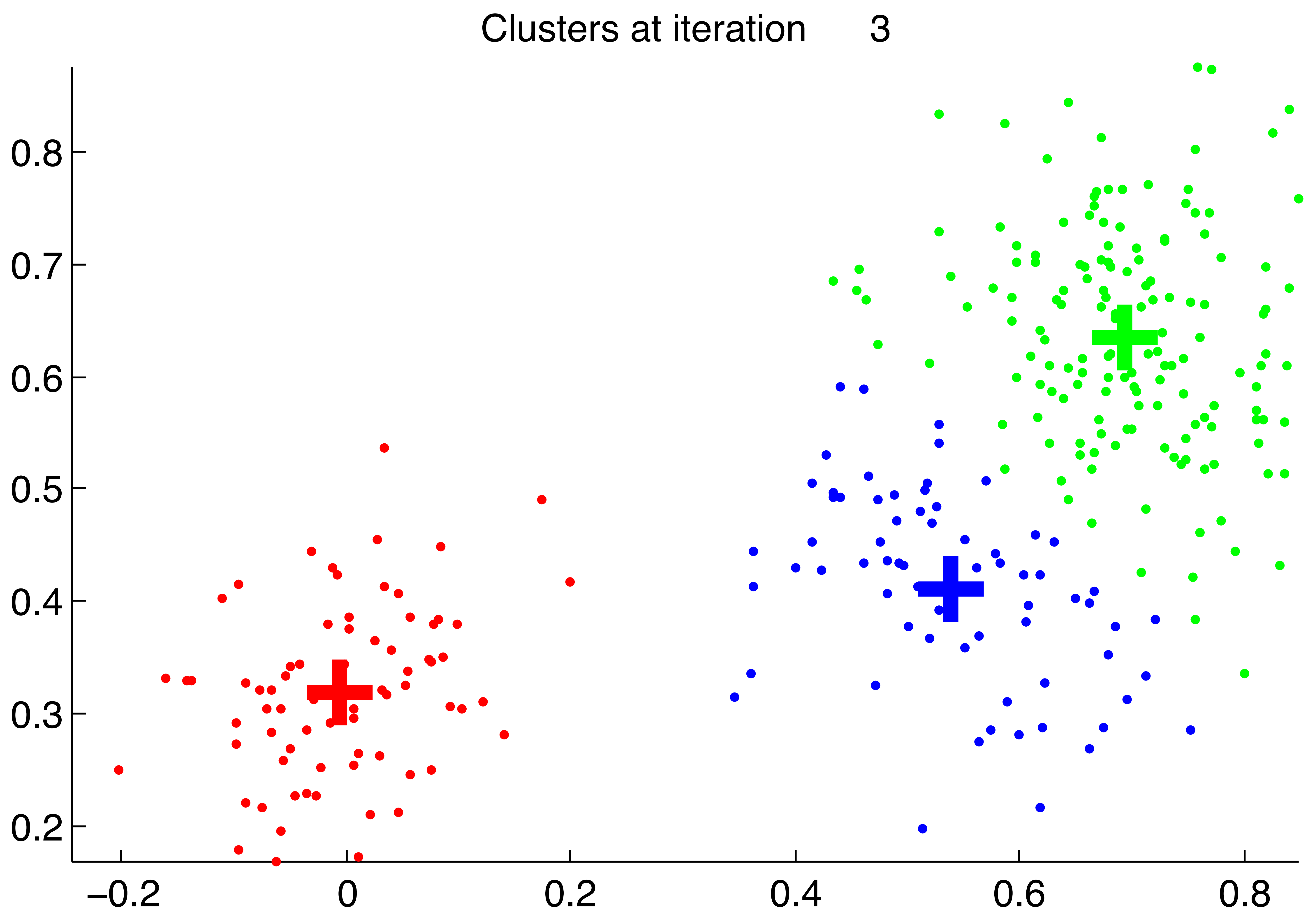
K-Means Clustering Example



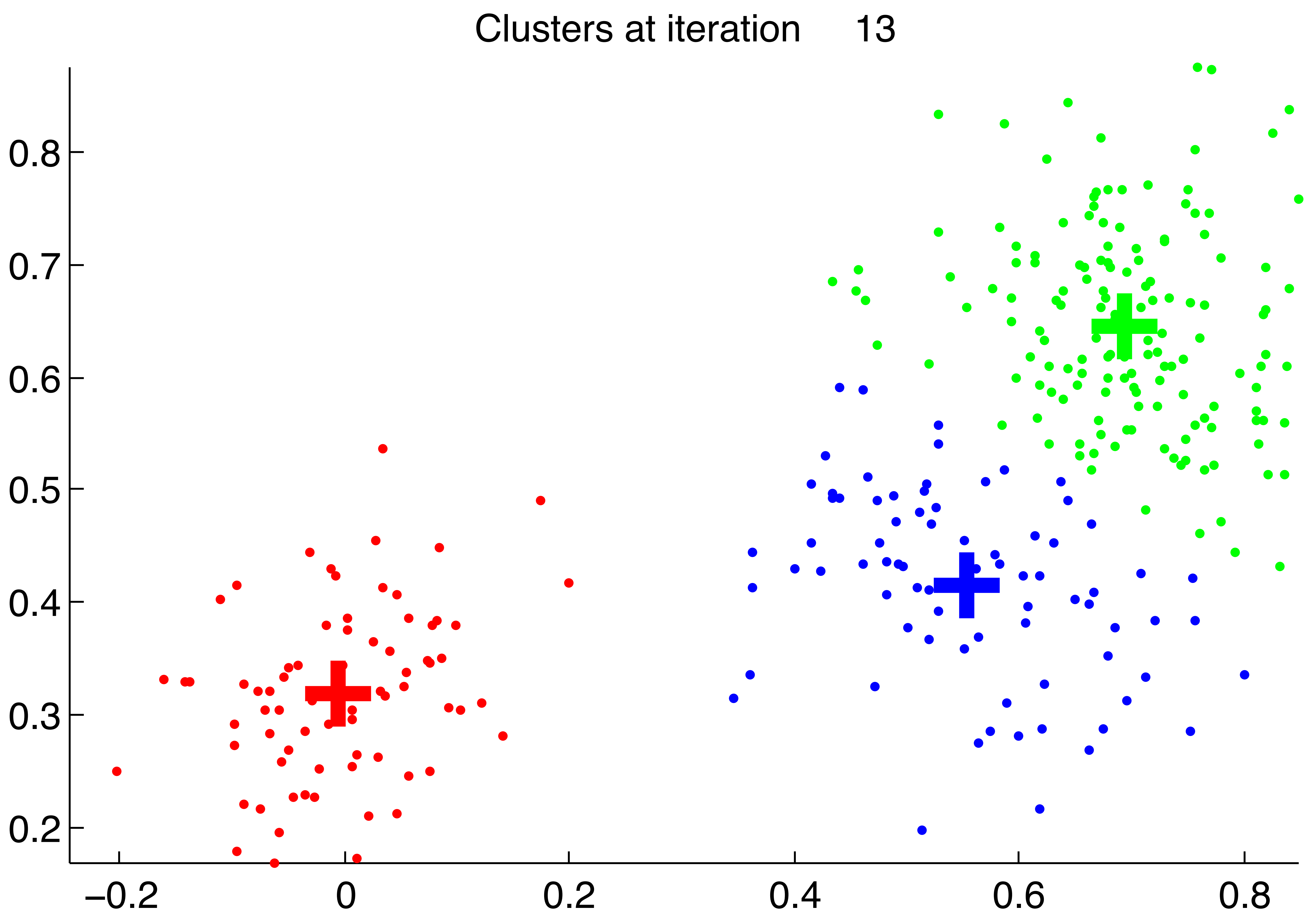
K-Means Clustering Example



K-Means Clustering Example



K-Means Clustering Example



Expectation Maximization

Description [edit]

The symbols [edit]

Given the statistical model which generates a set $\mathbf{X}$ of observed data, a set of unobserved latent data or missing values $\mathbf{Z}$, and a vector of unknown parameters $\boldsymbol{\theta}$, along with a likelihood function $L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$, the maximum likelihood estimate (MLE) of the unknown parameters is determined by maximizing the marginal likelihood of the observed data

$$L(\boldsymbol{\theta}; \mathbf{X}) = p(\mathbf{X} \mid \boldsymbol{\theta}) = \int p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})\, d\mathbf{Z} = \int p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta}) p(\mathbf{Z} \mid \boldsymbol{\theta})\, d\mathbf{Z}$$

However, this quantity is often intractable since $\mathbf{Z}$ is unobserved and the distribution of $\mathbf{Z}$ is unknown before attaining $\boldsymbol{\theta}$.

The EM algorithm [edit]

The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying these two steps:

Expectation step (E step): Define $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$ as the expected value of the log likelihood function of $\boldsymbol{\theta}$, with respect to the current conditional distribution of $\mathbf{Z}$ given $\mathbf{X}$ and the current estimates of the parameters $\boldsymbol{\theta}^{(t)}$:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\mathbf{Z} \sim p(\cdot \mid \mathbf{X}, \boldsymbol{\theta}^{(t)})}[\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})]$$

Maximization step (M step): Find the parameters that maximize this quantity:

$$\boldsymbol{\theta}^{(t-1)} = \underset{\boldsymbol{\theta}}{\arg\max}\ Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$$

More succinctly, we can write it as one equation:

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\arg\max}\, \mathrm{E}_{\mathbf{Z} \sim p(\cdot \mid \mathbf{X}, \boldsymbol{\theta}^{(t)})}[\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})]$$

From Wikipedia — I will not ask you this

Expectation Maximization

A simpler version

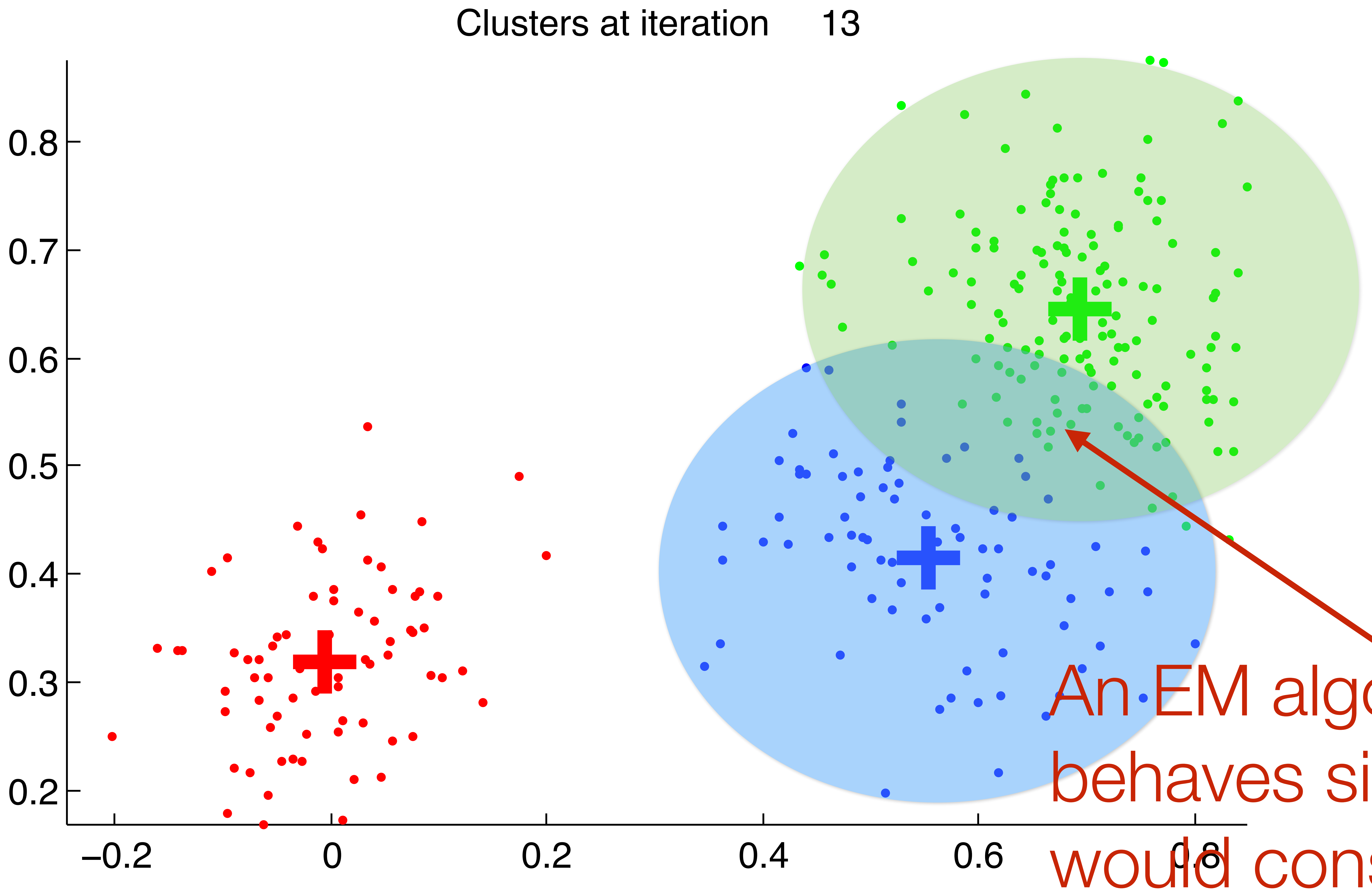
The K-Means centers

Given a model repeat

1. Create an "**expectation**" of the (log-)likelihood with the current hypothesis
2. Update the hypothesis to one that **maximizes** the expectation above

Not exactly the *hard* assignments of K-Means

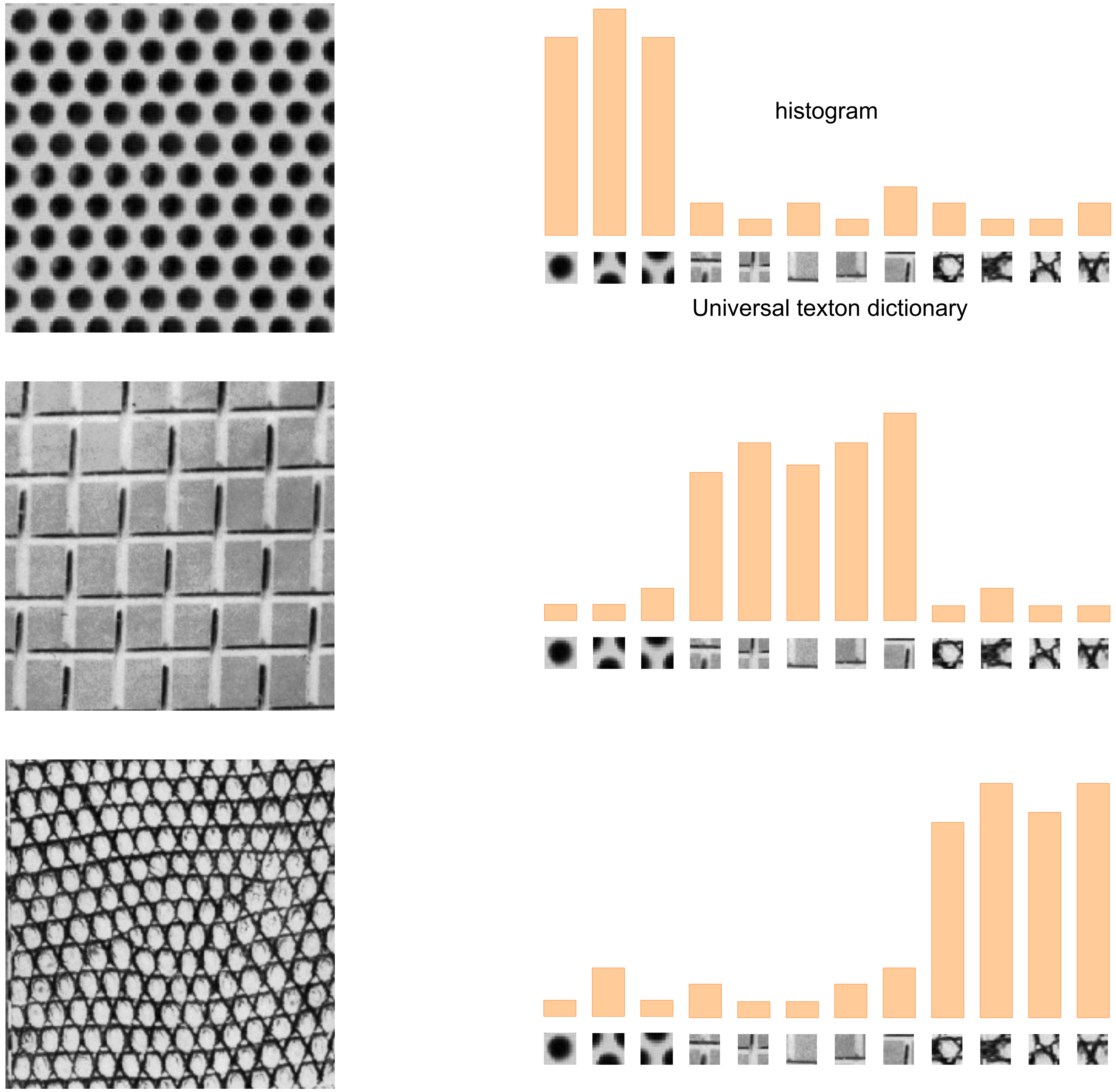
K-Means Clustering Example



An EM algorithm that behaves similarly would consider this as Gaussian Mixture

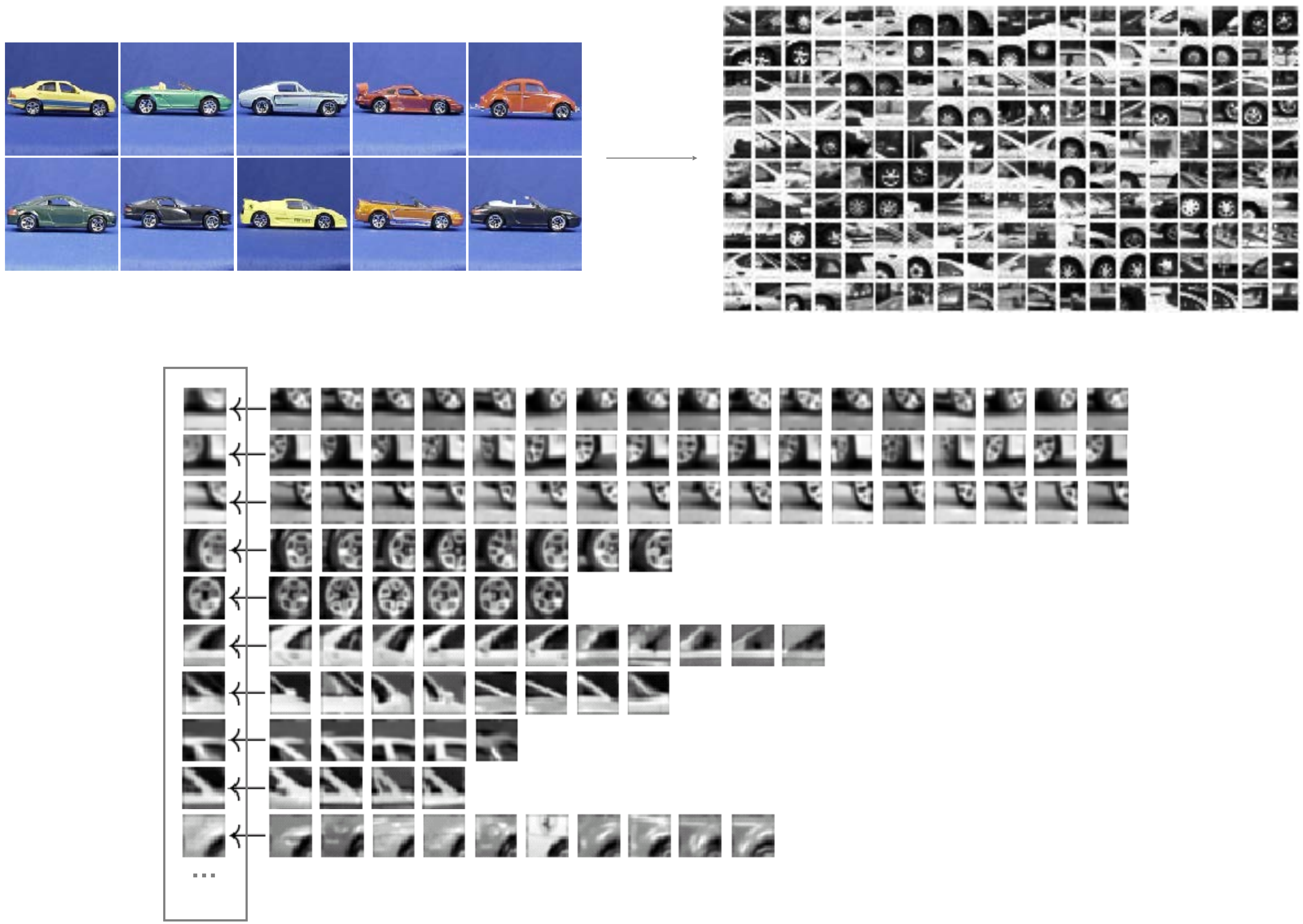
Recall: Texture Representation

Now we know how to create this



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Example **Visual Dictionary**



Source: B. Leibe

Example **Visual Dictionary**

Appearance codebook

Source: B. Leibe

Standard **Bag-of-Words** Pipeline (for image classification)

Dictionary Learning:
Learn Visual Words using clustering

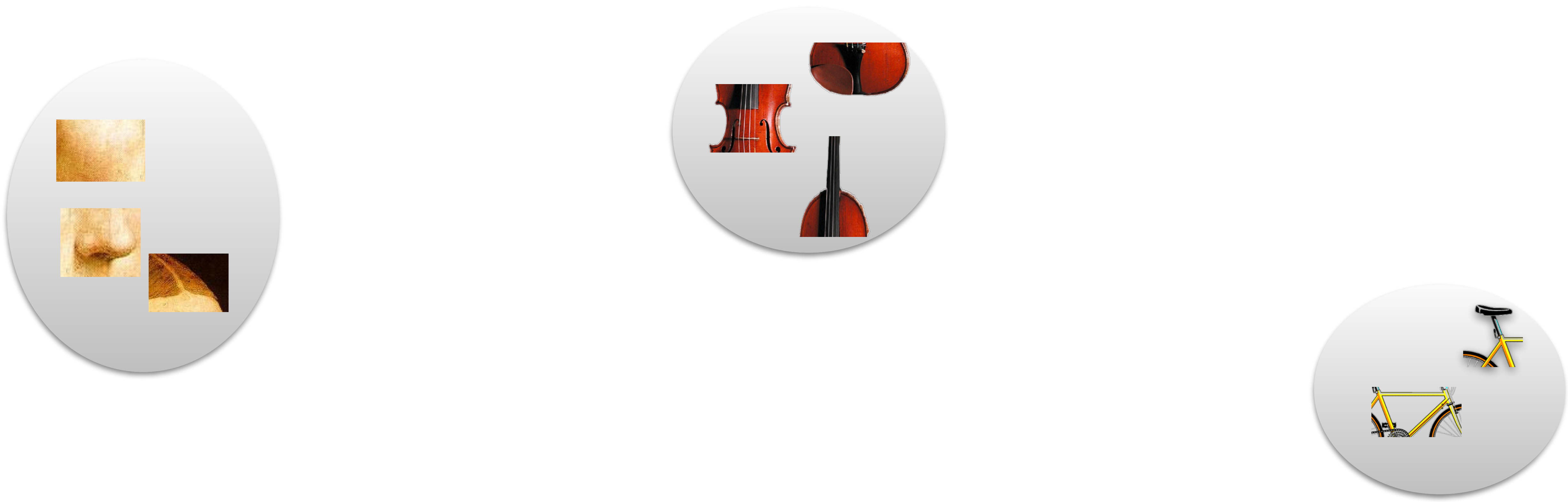
Encode:
build Bags-of-Words (BOW) vectors
for each image

Classify:
Train and test data using BOWs

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

2. **Encode:** build Bag-of-Words (BOW) vectors for each image

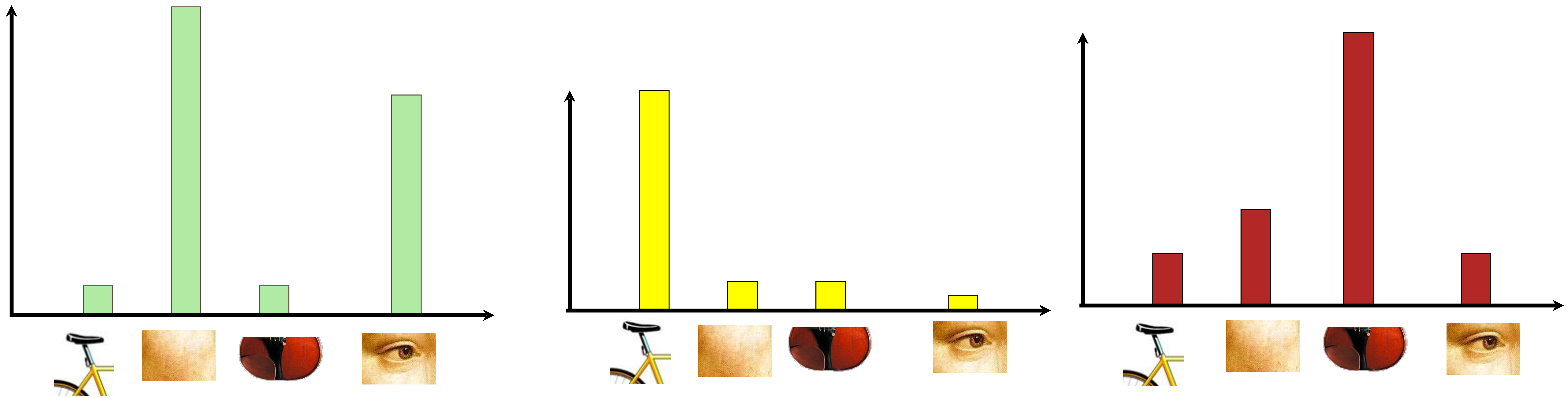
1. **Quantization**: image features gets associated to a visual word (nearest cluster center)



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

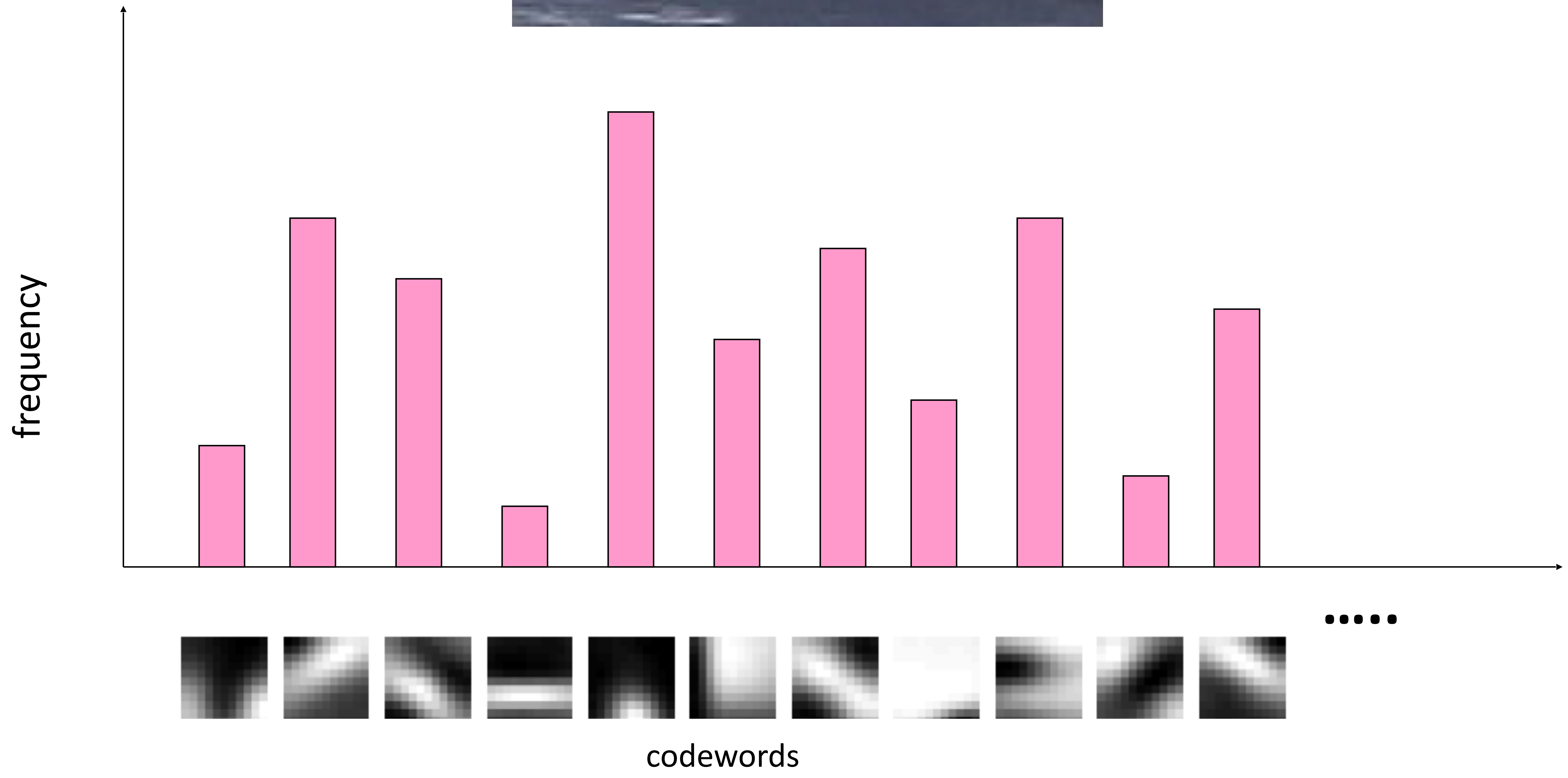
2. Encode: build Bag-of-Words (BOW) vectors for each image

2. **Histogram**: count the number of visual word occurrences



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

2. **Encode:** build Bag-of-Words (BOW) vectors for each image



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Standard **Bag-of-Words** Pipeline (for image classification)

Dictionary Learning:
Learn Visual Words using clustering

Encode:
build Bags-of-Words (BOW) vectors
for each image

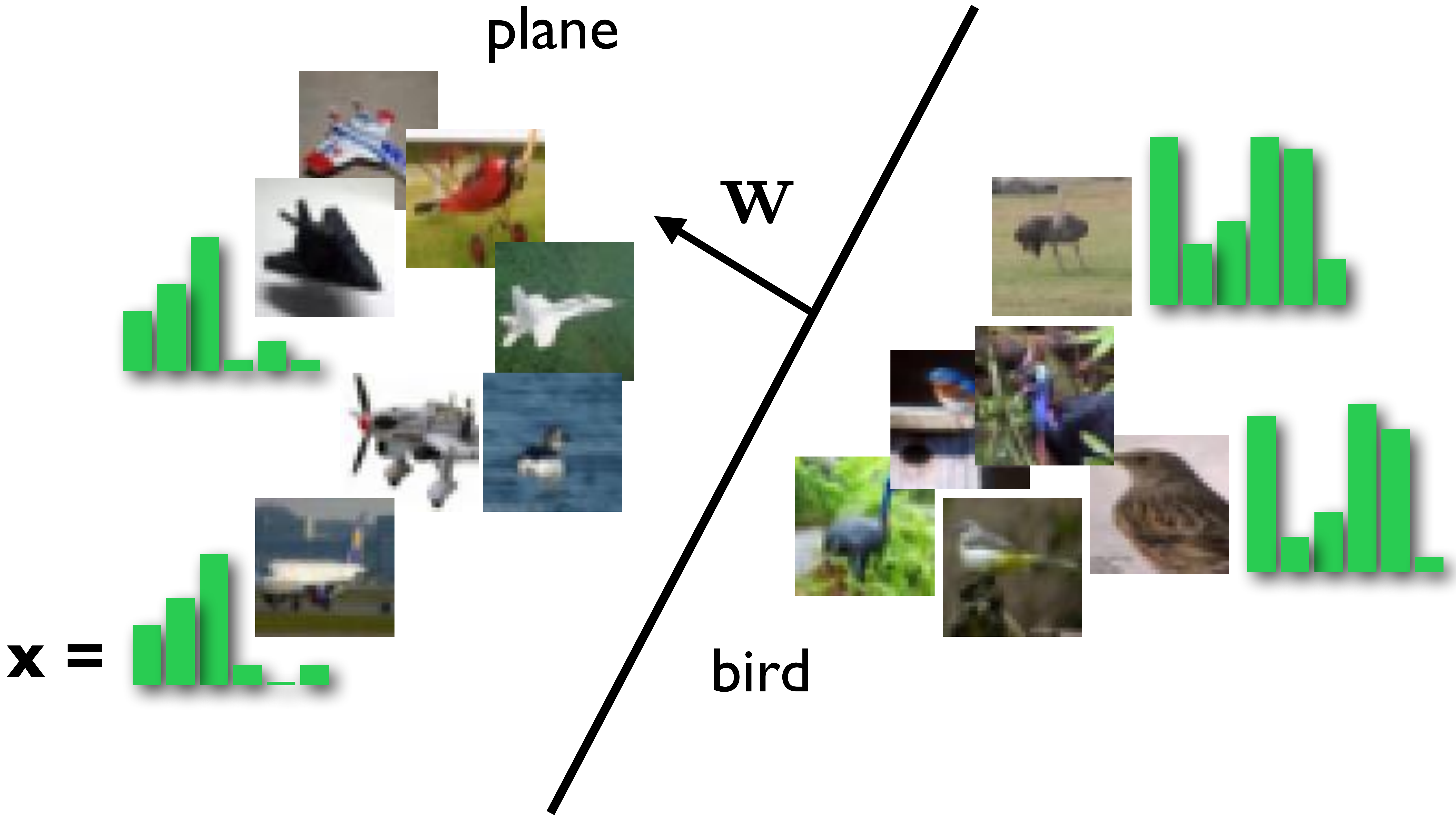
Classify:
Train and test data using BOWs

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Classify Visual Word Histograms

e.g., bird vs plane classifier as linear classifier in space of histograms

Histograms of visual word frequencies = vector **x**, linear classifier **w**



Support Vector Machines (SVM)

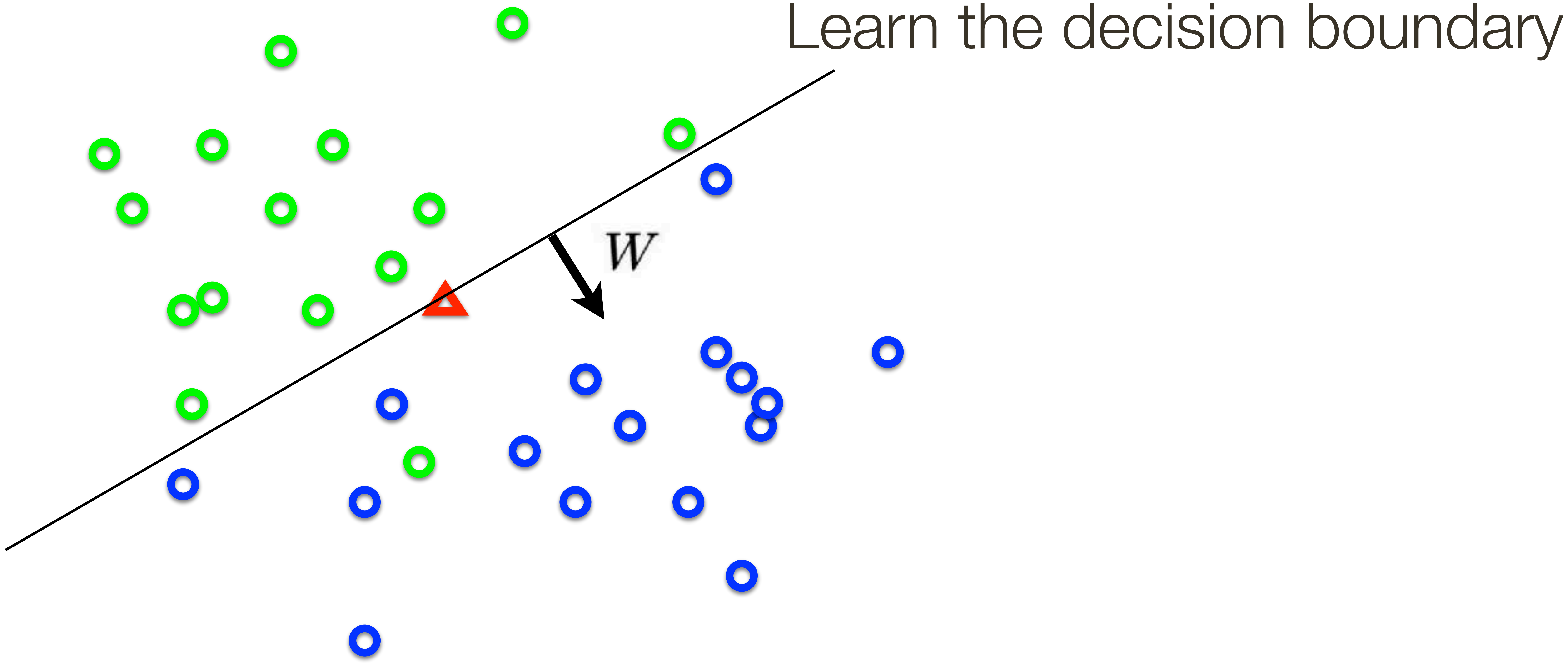


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

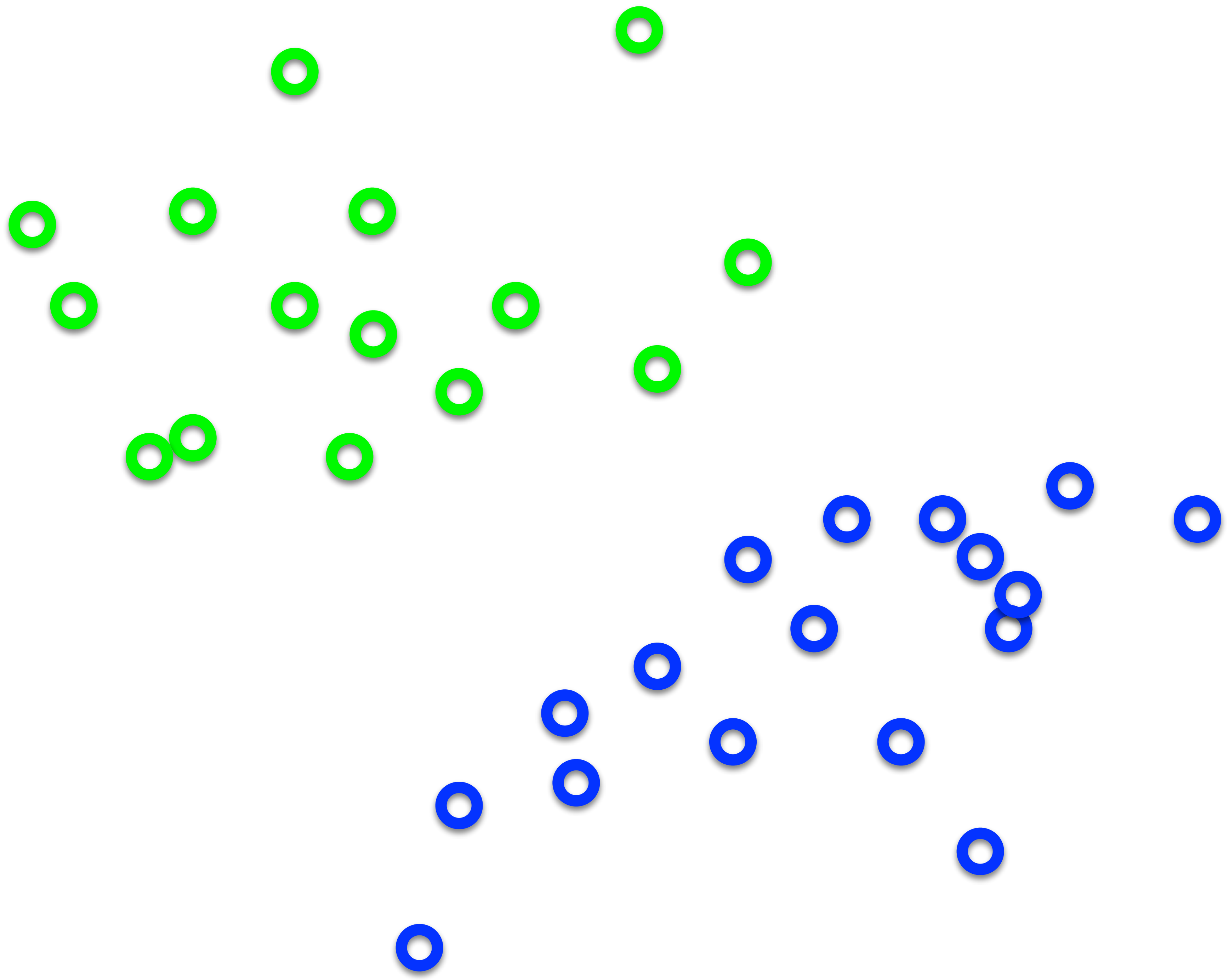


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

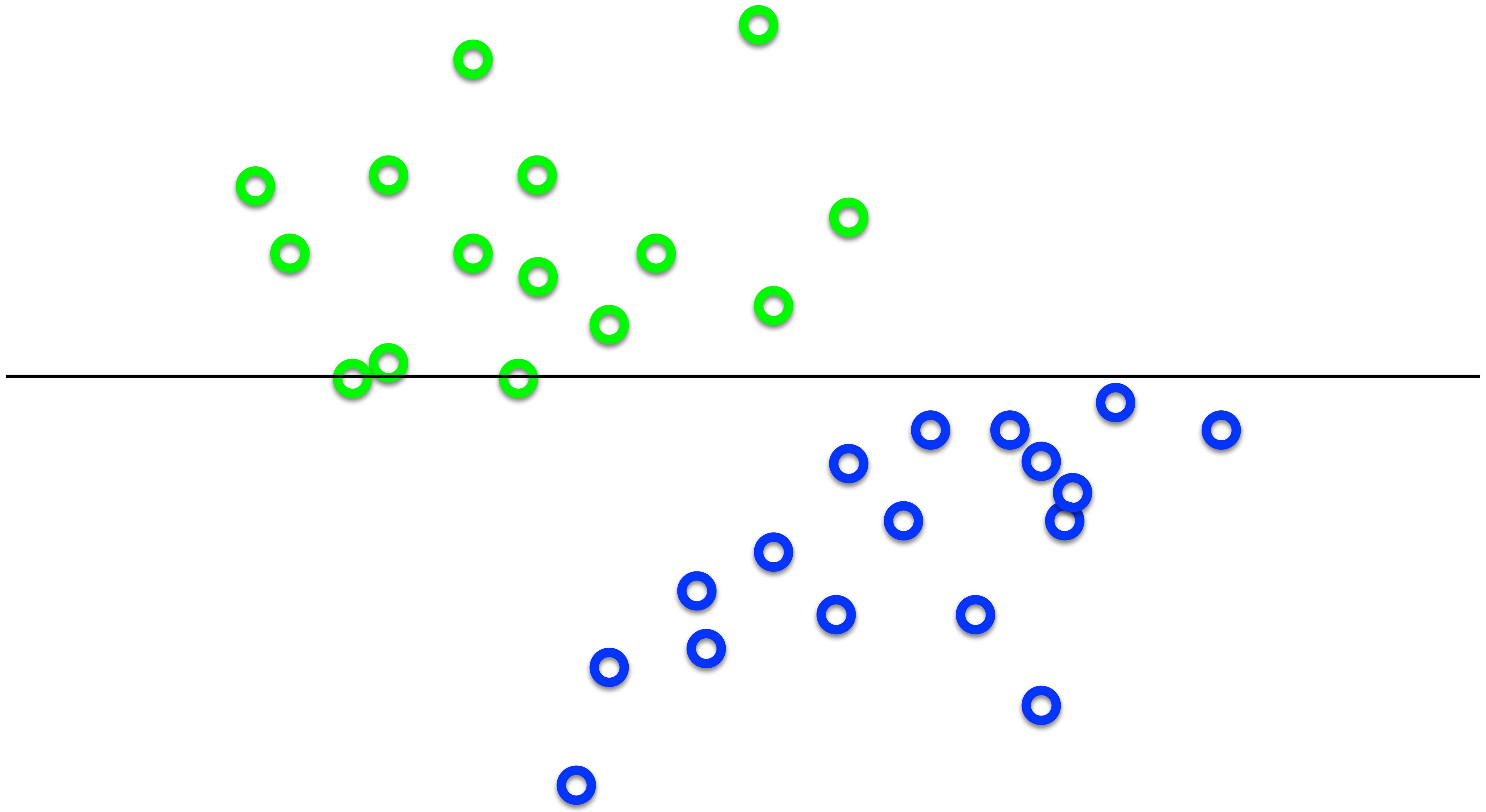


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

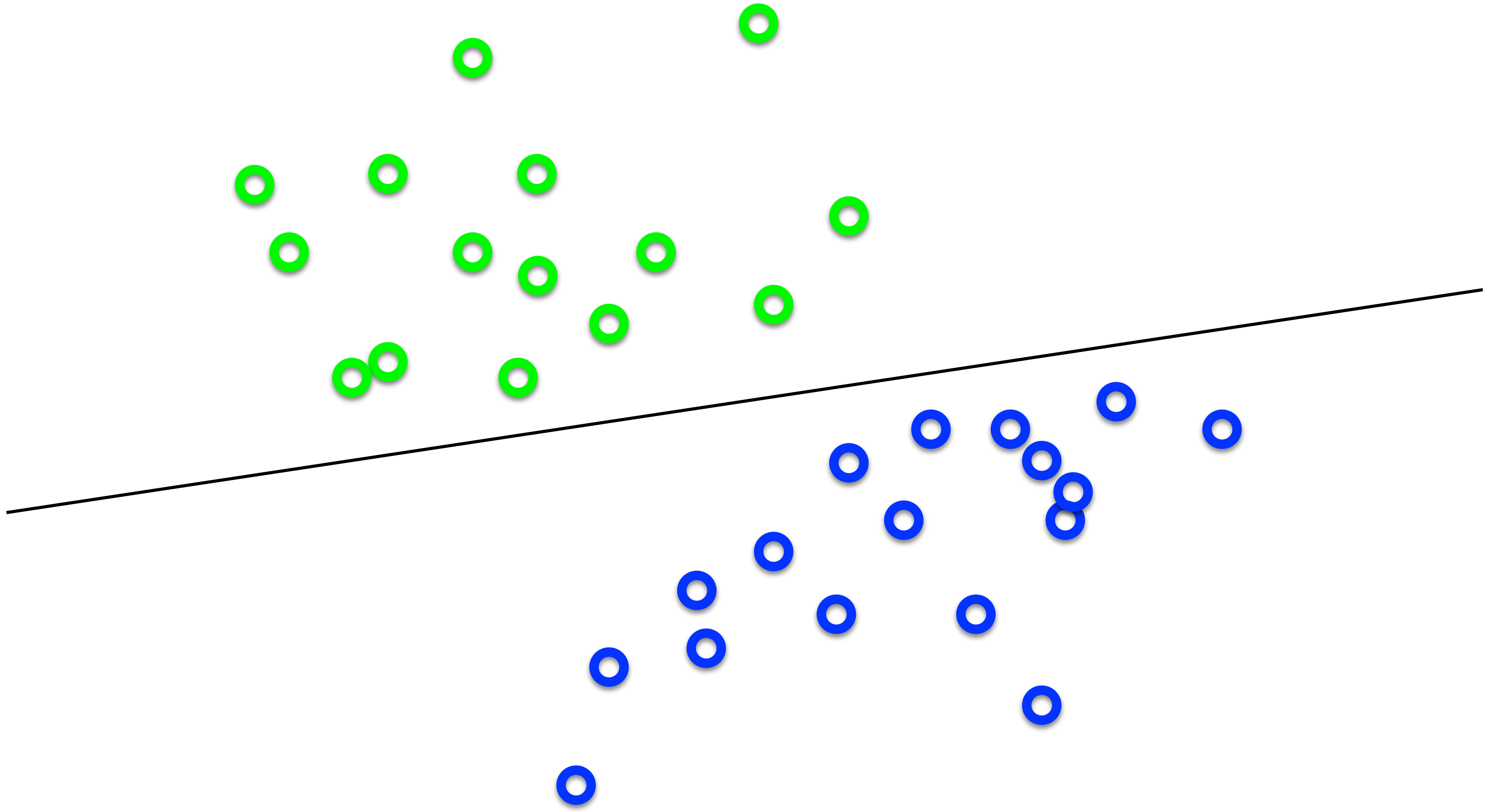


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

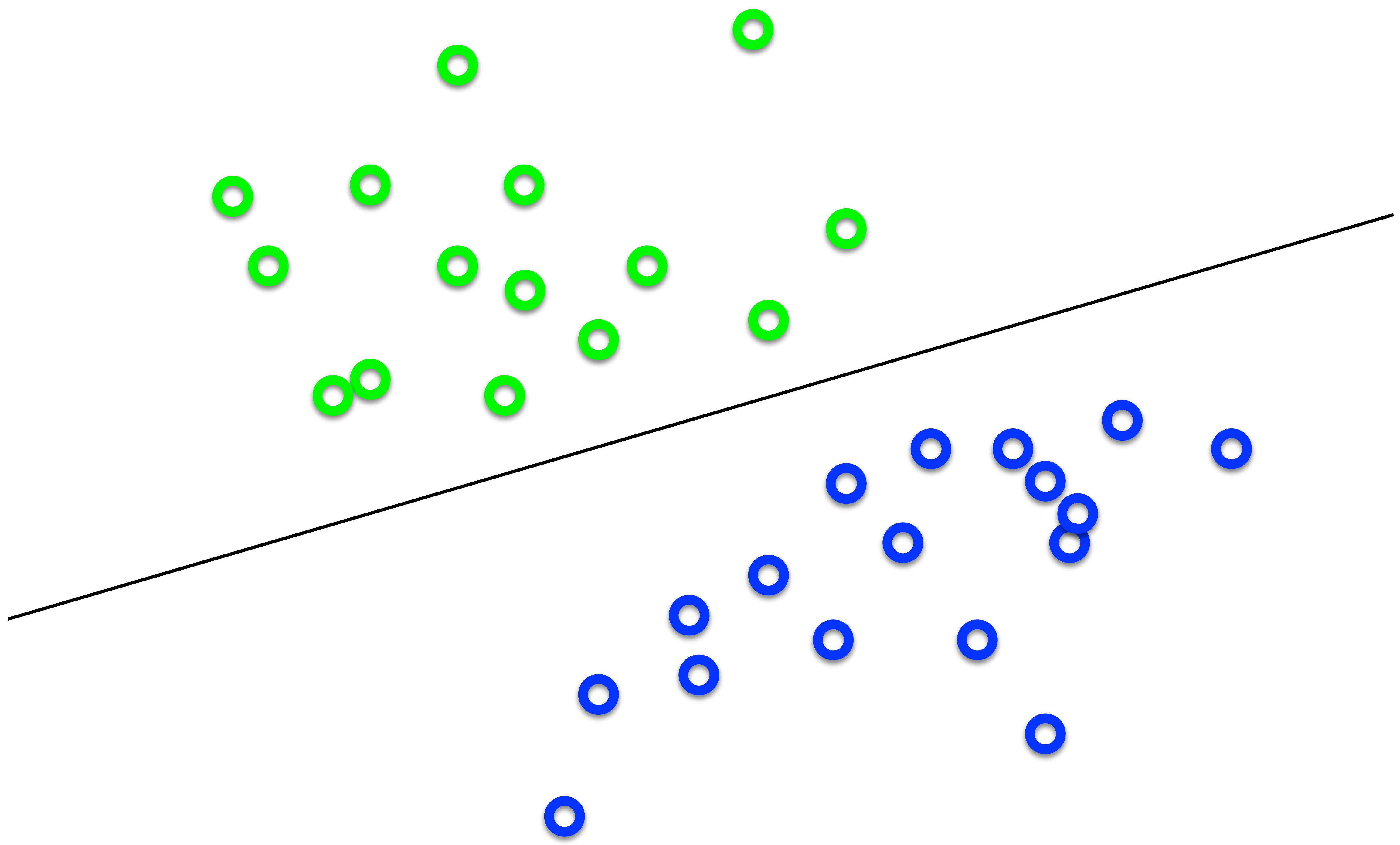


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

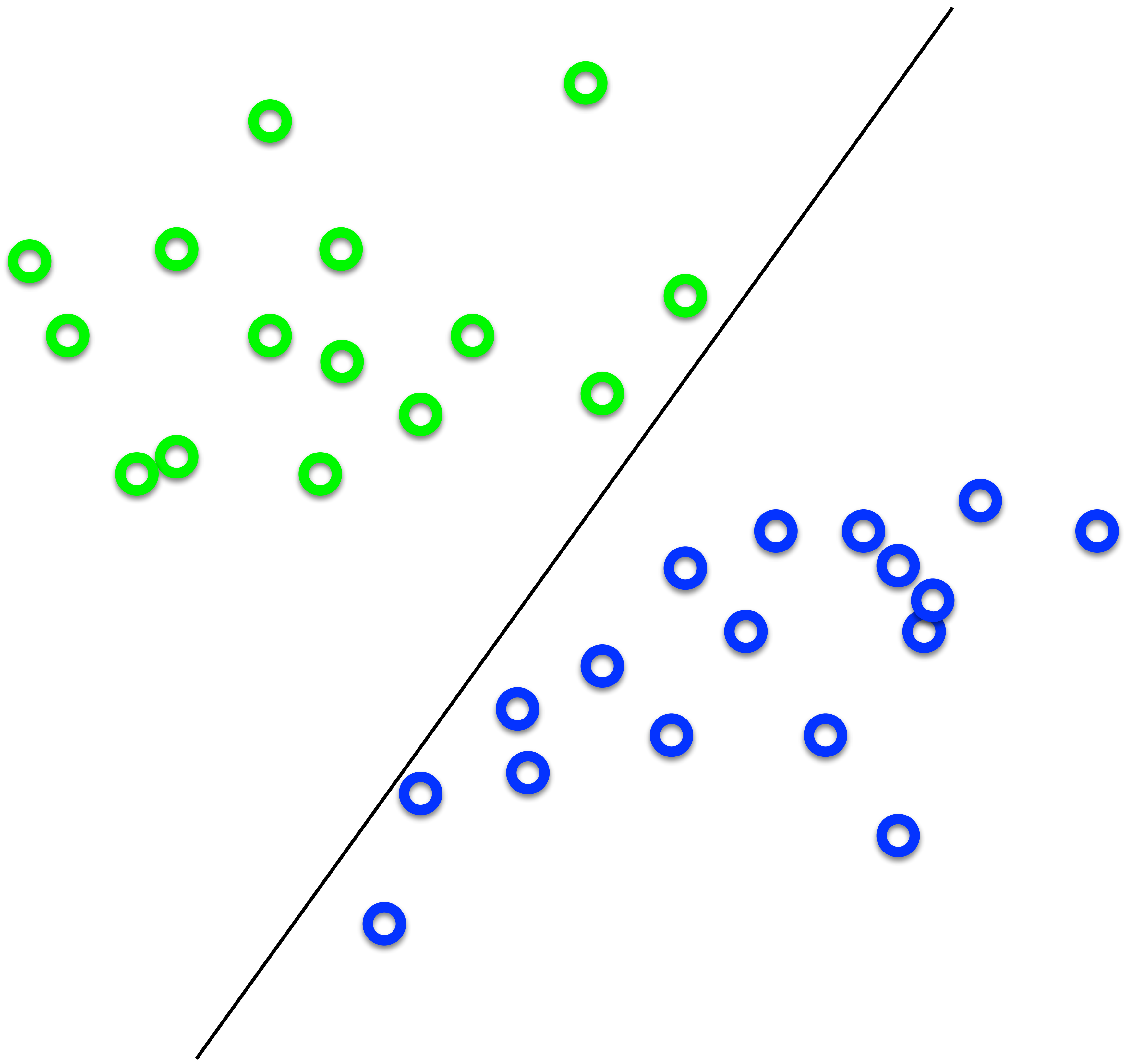
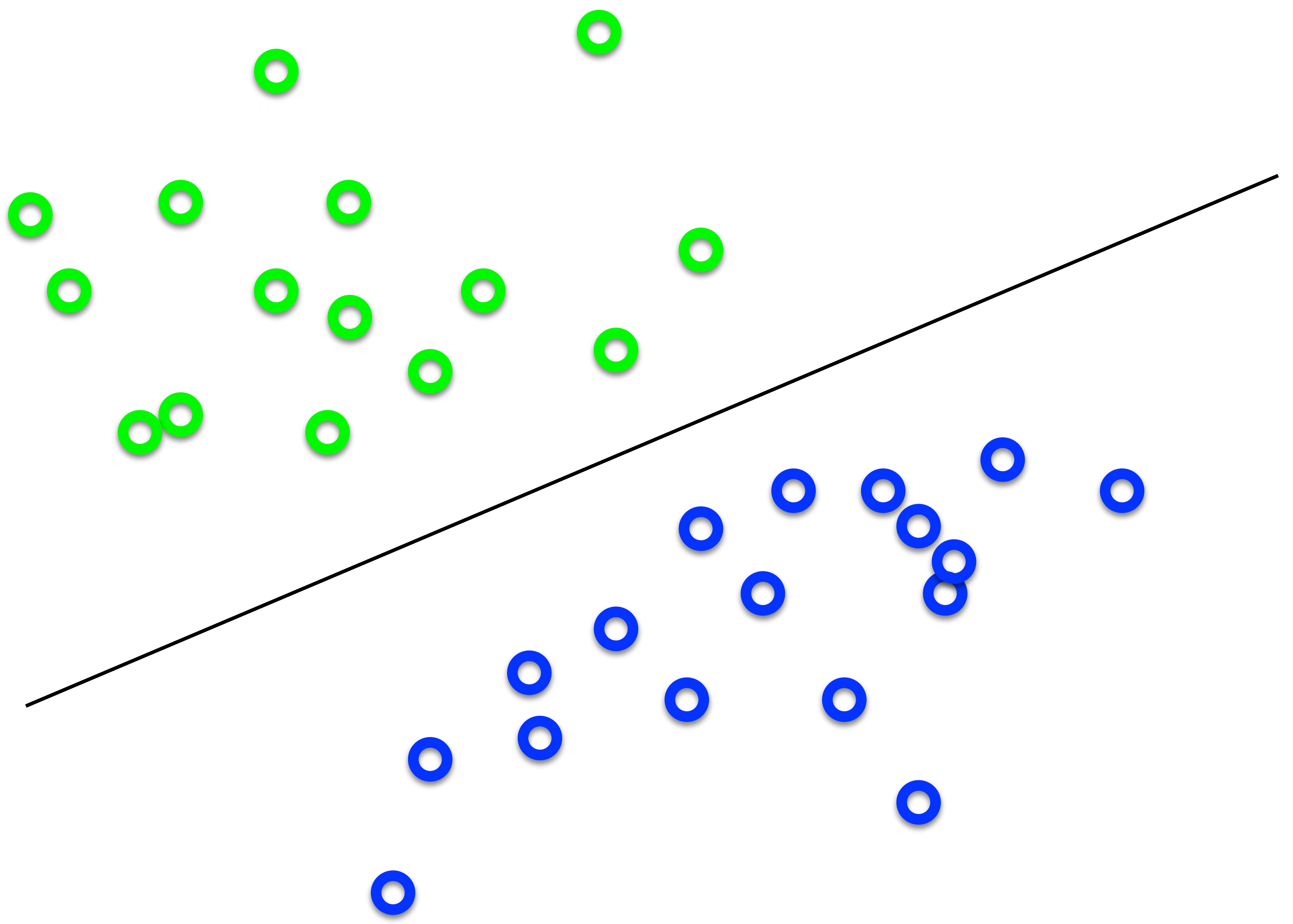


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

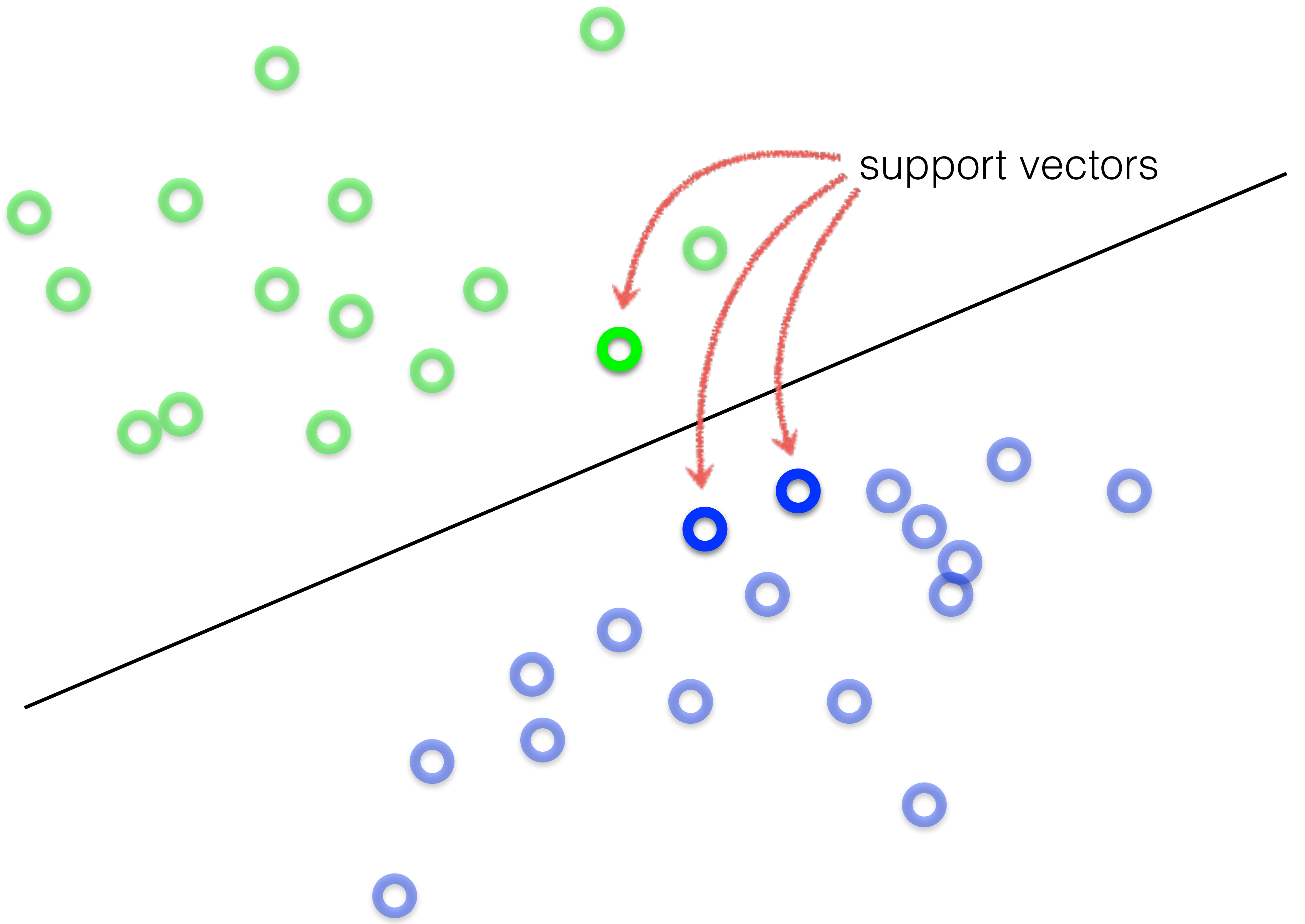


Intuitively, the line that is the farthest from all interior points

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

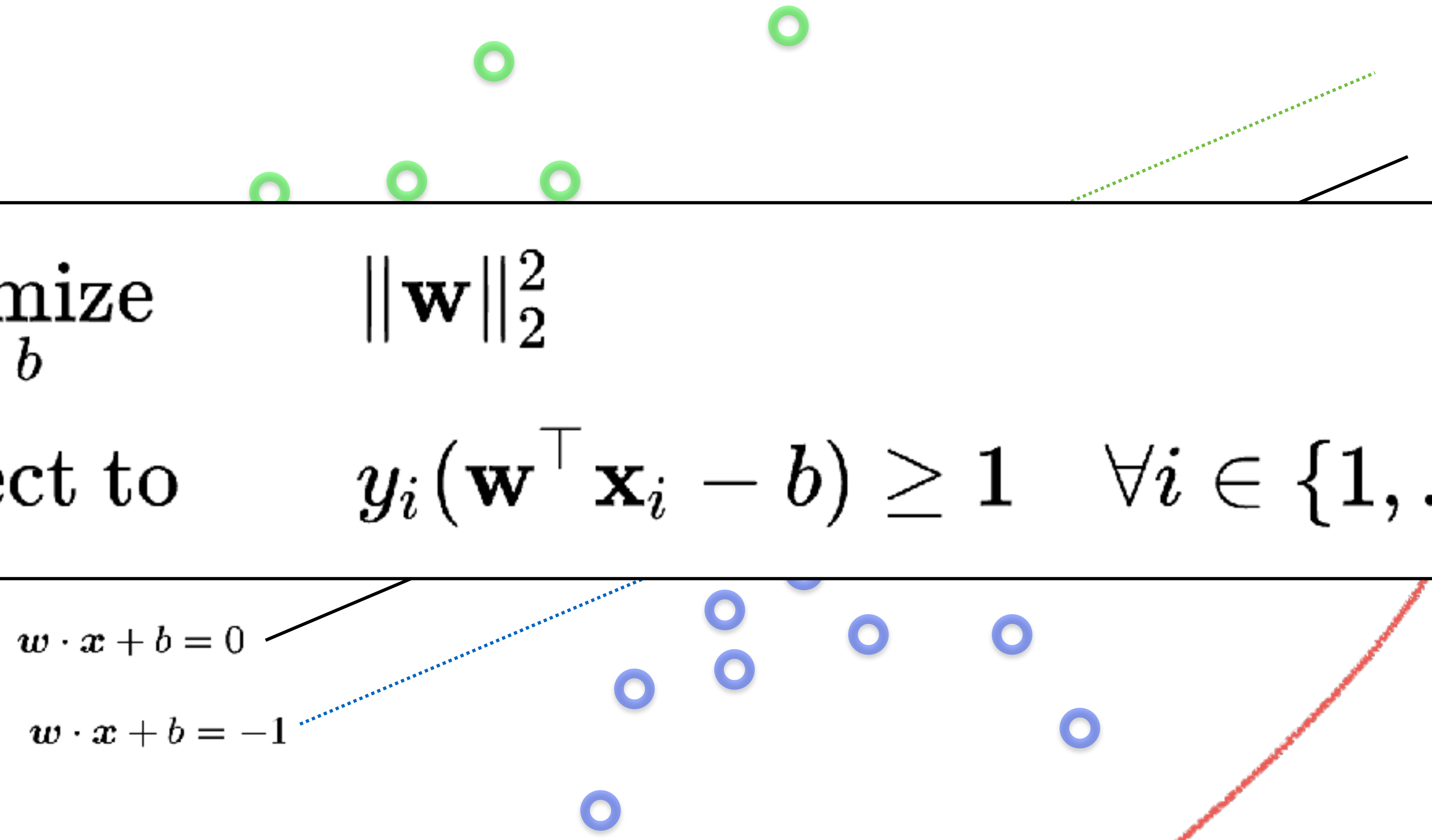
What's the best **w** ?



Want a hyperplane that is far away from 'inner points'

Support Vector Machines (SVM)

Find hyperplane **w** such that …

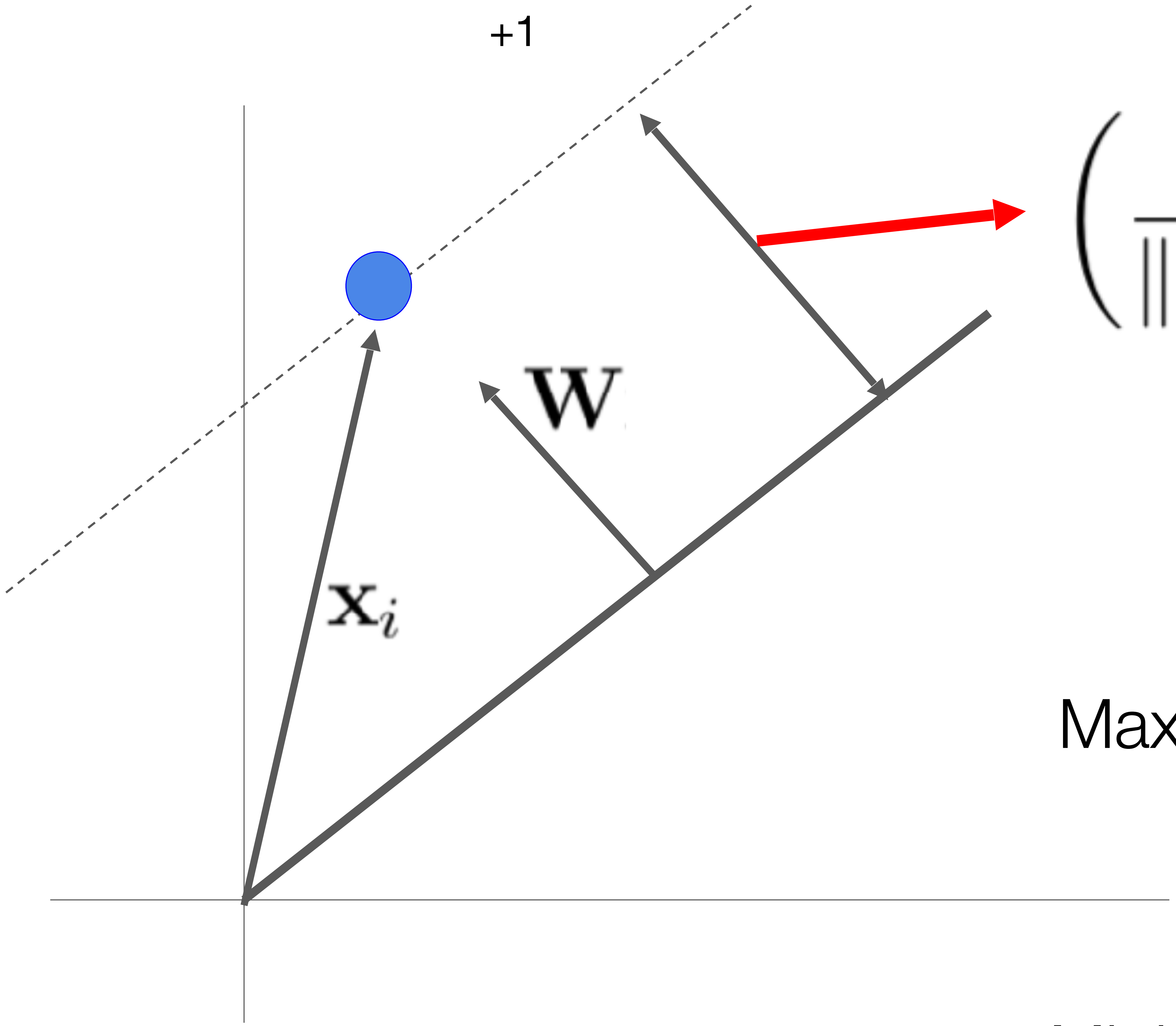


$$\begin{array}{ll} \underset{\mathbf{w},\, b}{\text{minimize}} & \|\mathbf{w}\|_2^2 \\ \text{subject to} & y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 \quad \forall i \in \{1, \ldots, n\} \end{array}$$

the gap between parallel hyperplanes $\frac{2}{\|w\|}$ is maximized

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Distance to the **border**



Becomes 1 because it's the thing at the border (+1)

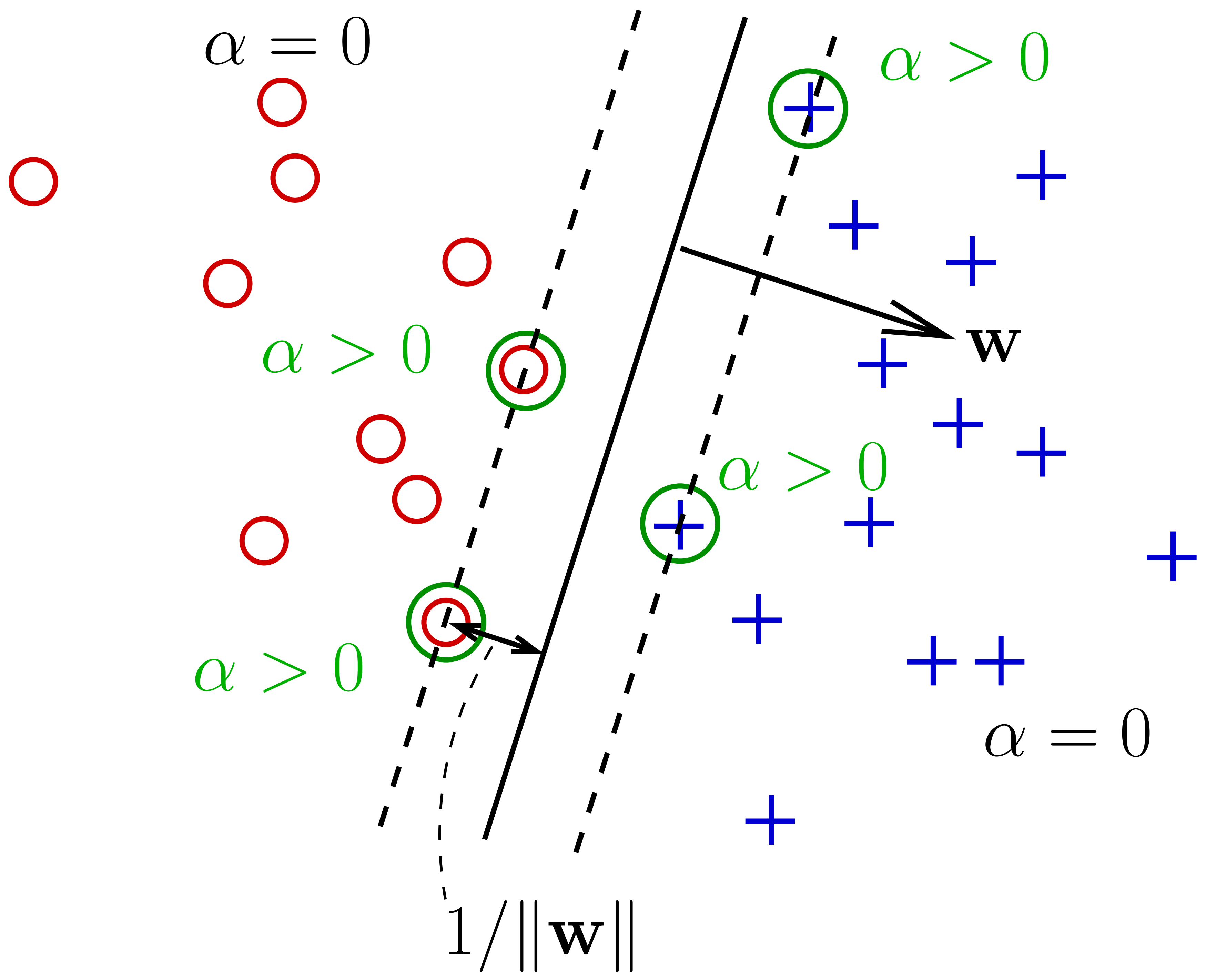
$$\left(\frac{\mathbf{W}}{\|\mathbf{W}\|_2}\right)^\top \mathbf{x}_i = \frac{1}{\|\mathbf{W}\|_2}$$

Maximize $\dfrac{1}{\|\mathbf{W}\|_2^2}$

Minimize $\|\mathbf{W}\|_2^2$

Support Vectors

- The active constraints are due to the data that define the classification boundary, these are called **support vectors**



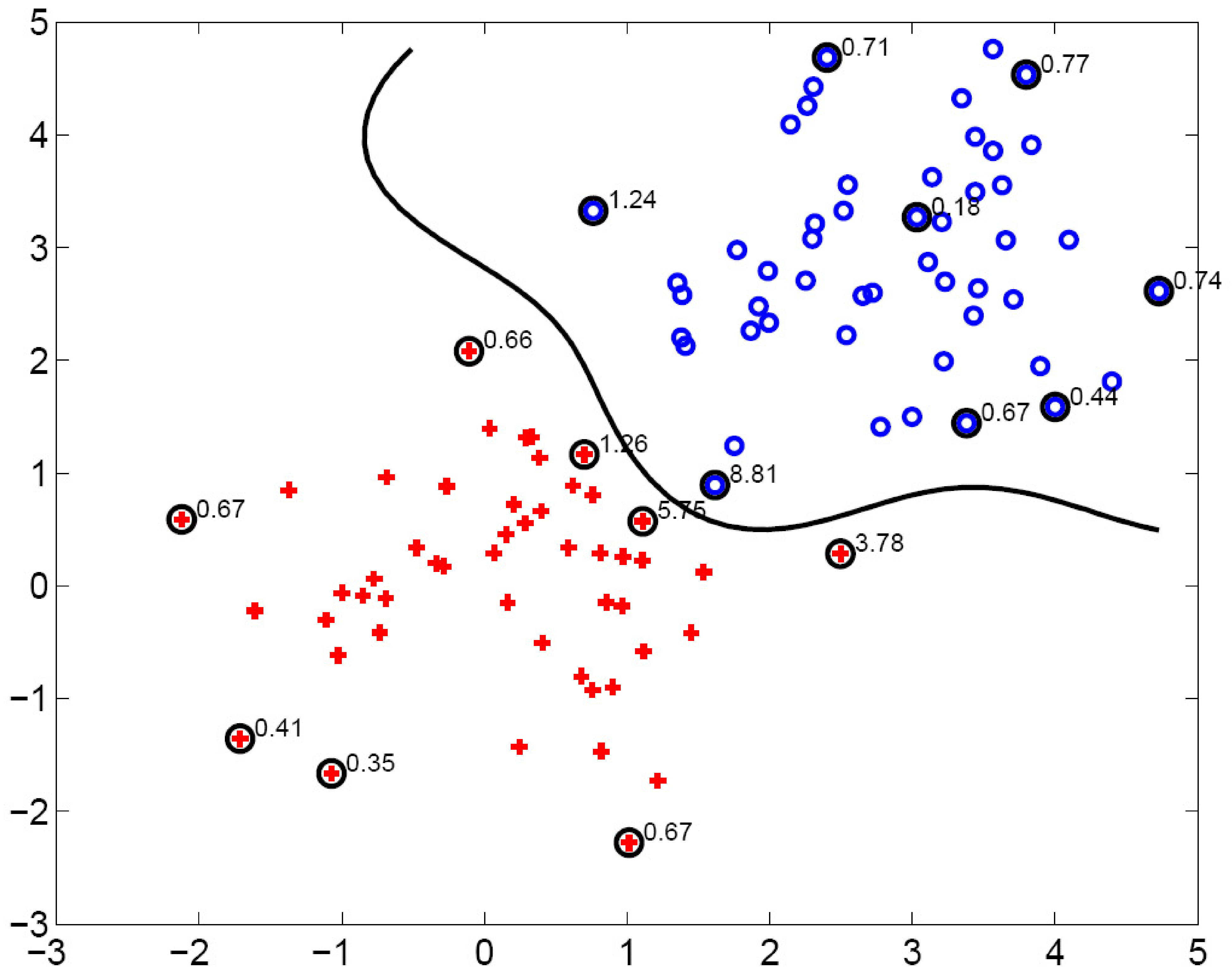
Final classifier can be written in terms of the support vectors:

$$\hat{y} = \text{sign}\left(\hat{w}_0 + \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}\right)$$

Non-Linear SVM

- Replace inner product with kernel

$$\mathbf{x}_i^T \mathbf{x} \rightarrow \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) \rightarrow k(\mathbf{x}_i, \mathbf{x})$$



- Data are (ideally) linearly separable in $\phi(x)$
- But we don't need to know $\phi(x)$, we just specify $k(x,y)$
- Points with $\alpha>0$ (circled) are support vectors
- Other data can be removed without affecting classifier

Bag-of-Words Representation

Algorithm:

Initialize an empty K -bin histogram, where K is the number of codewords
Extract local descriptors (e.g. SIFT) from the image
For each local descriptor **x**

Map (Quantize) **x** to its closest codeword → **c**(**x**)

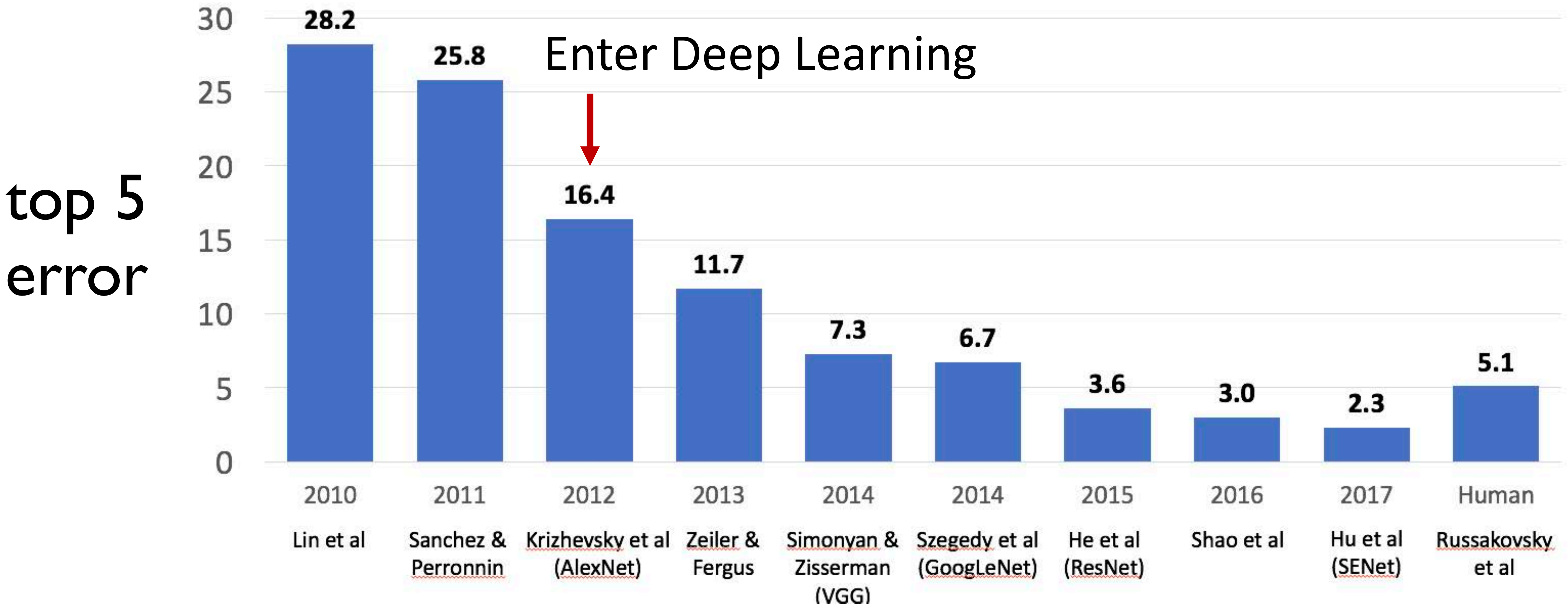
Increment the histogram bin for **c**(**x**)

Return histogram

We can then classify the histogram using a trained classifier, e.g. a support vector machine or k-Nearest Neighbor classifier

Alexnet

- Won the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 by a large margin
- Some ingredients: Deep neural net (Alexnet), Large dataset (Imagenet), Lots of compute (2 GPU weeks), non-saturating activation functions (ReLU)



[J. Johnson]

Summary

Factors that make image classification hard
— intra-class variation, viewpoint, illumination, clutter, occlusion...

A codebook of **visual words** contains representative local patch descriptors
— can be constructed by clustering local descriptors (e.g. SIFT) in training images

The **bag of words** model accumulates a histogram of occurrences of each visual word

An supervised classifier, such as a **Support Vector Machine (SVM)** is then used to classify the word histograms