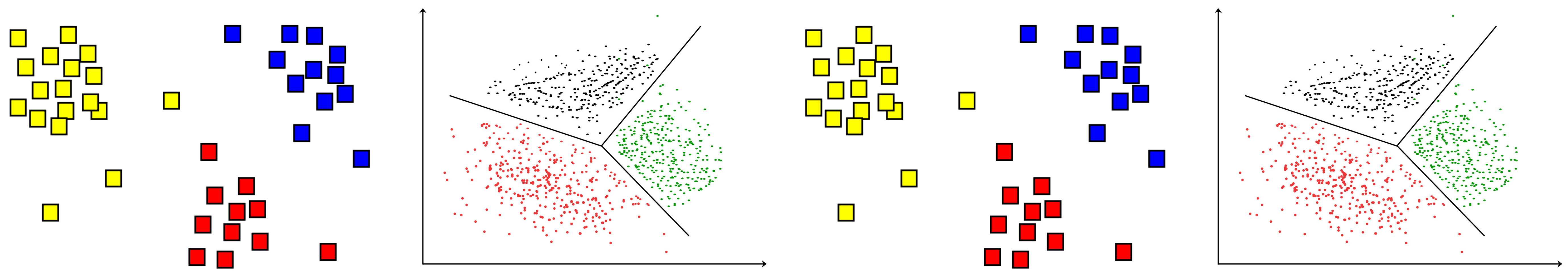
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 19: Visual Classification 1, Bag of Words

Menu for Today

Topics:

— Visual **Classification**

— **Bag of Words** Representations

Readings:

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

Reminders:

— **Assignment 2** graded and posted

— **Assignment 3** will be graded by the end of the week

— **Assignment 4** is due **today**

— **Assignment 5**: Scene Recognition with Bag of Words is now available

Today's "**fun**" Example:

Audio-Visual Scene Analysis with
Self-Supervised Multisensory Features

Andrew Owens Alexei A. Efros
UC Berkeley

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Today's "**fun**" Example:

Audio-Visual Scene Analysis with Self-Supervised Multisensory Features

Andrew Owens Alexei A. Efros

UC Berkeley

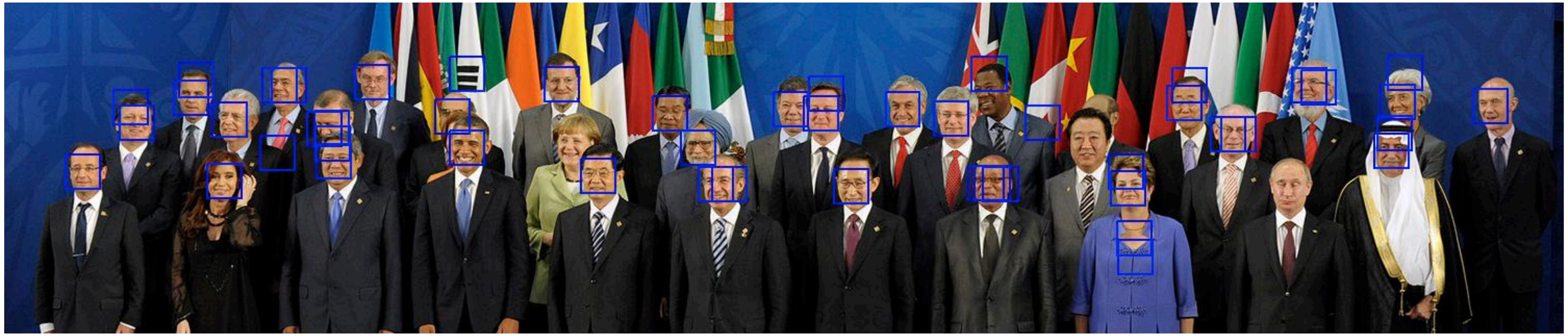
BAIR

BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Object **Recognition / Detection**

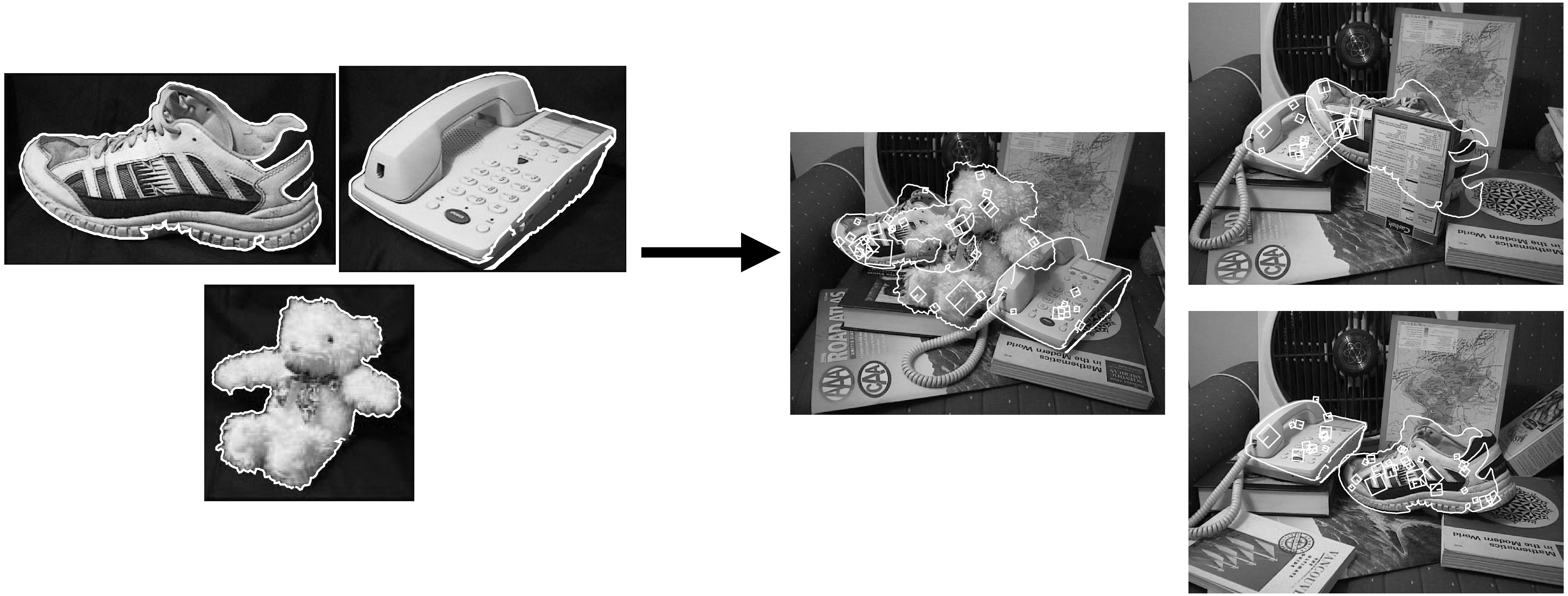
Template matching …

$*$  $\longrightarrow$ Non-max suppress + threshold $\longrightarrow$



Object **Recognition / Detection**

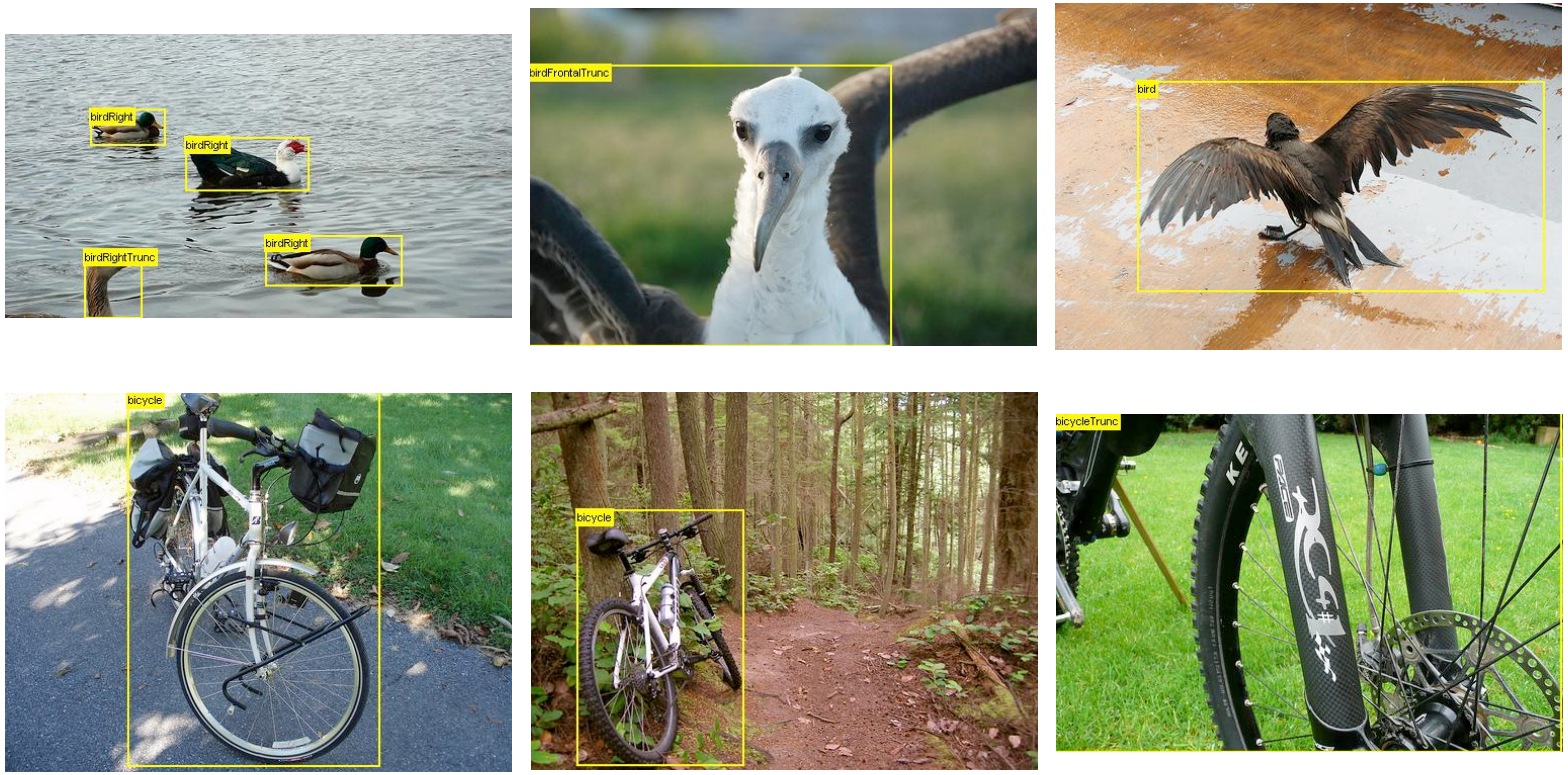
Object recognition with SIFT features and RANSAC [Lowe 1999]



What is present? Where? What orientation?

Object **Recognition / Detection**

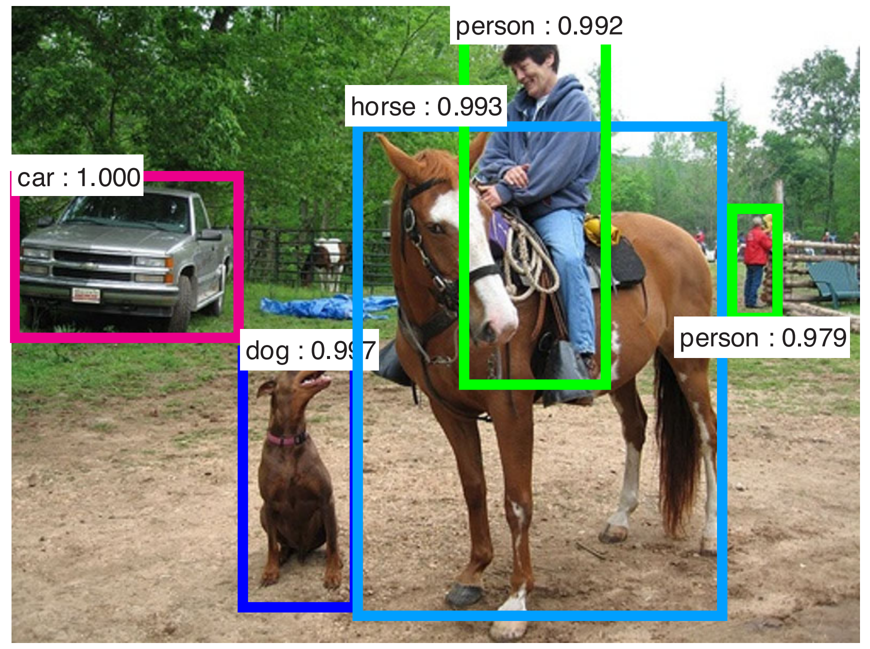
PASCAL Visual Object Classes Challenges [2005-2012]

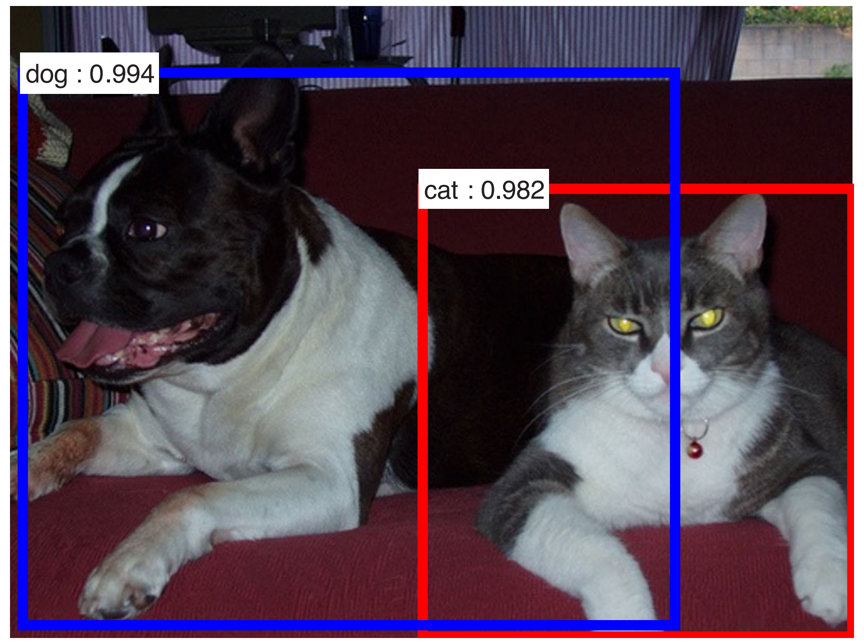


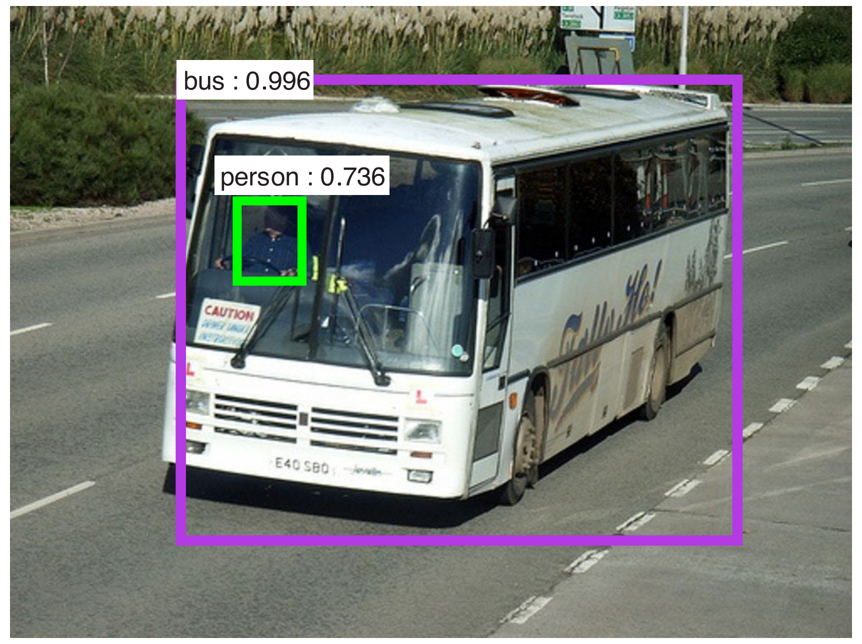
What is present? Where? What orientation?

Object **Classification** and **Detection**

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



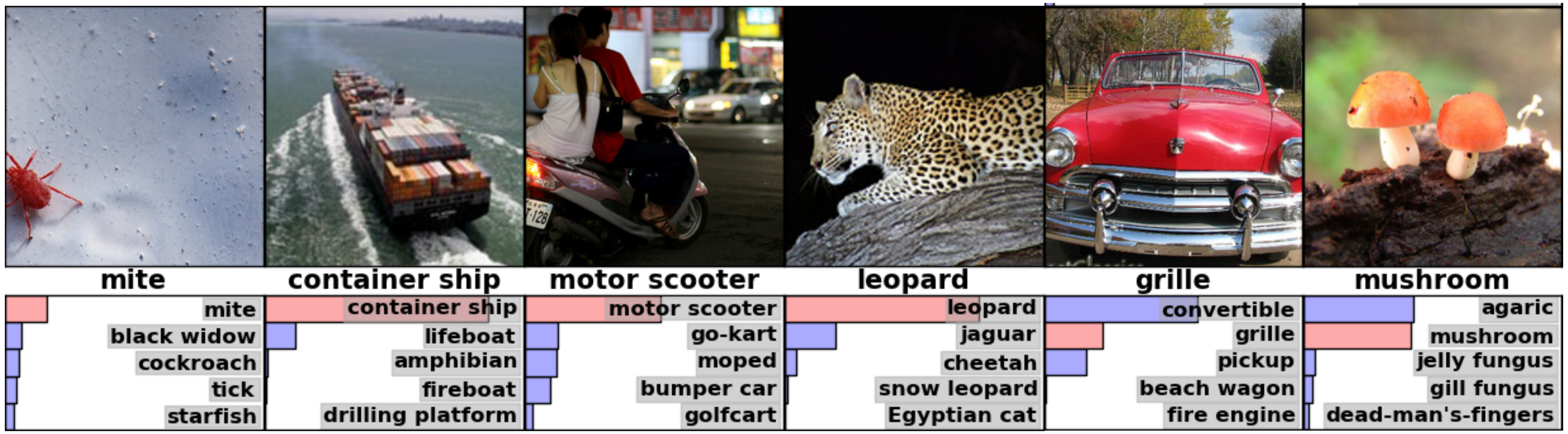




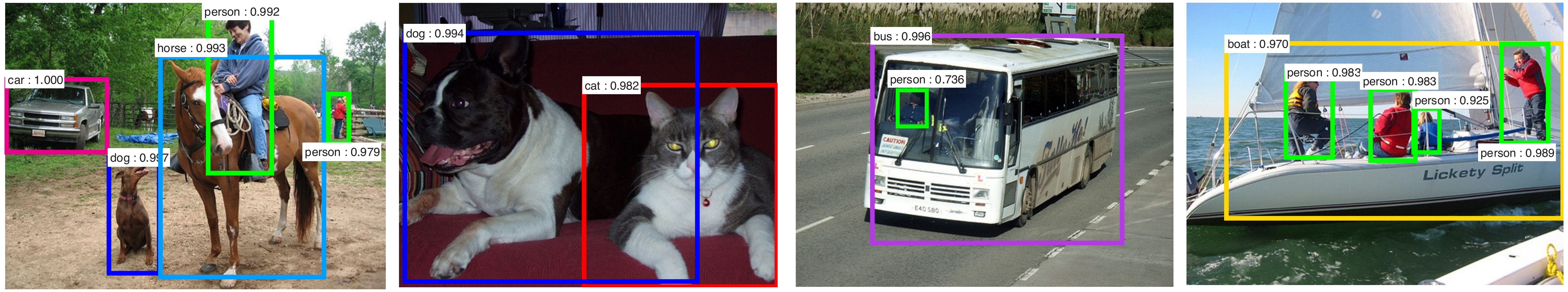
[Krizhevsky et al 2011][Ren et al 2016]

Object **Classification** and **Detection**

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



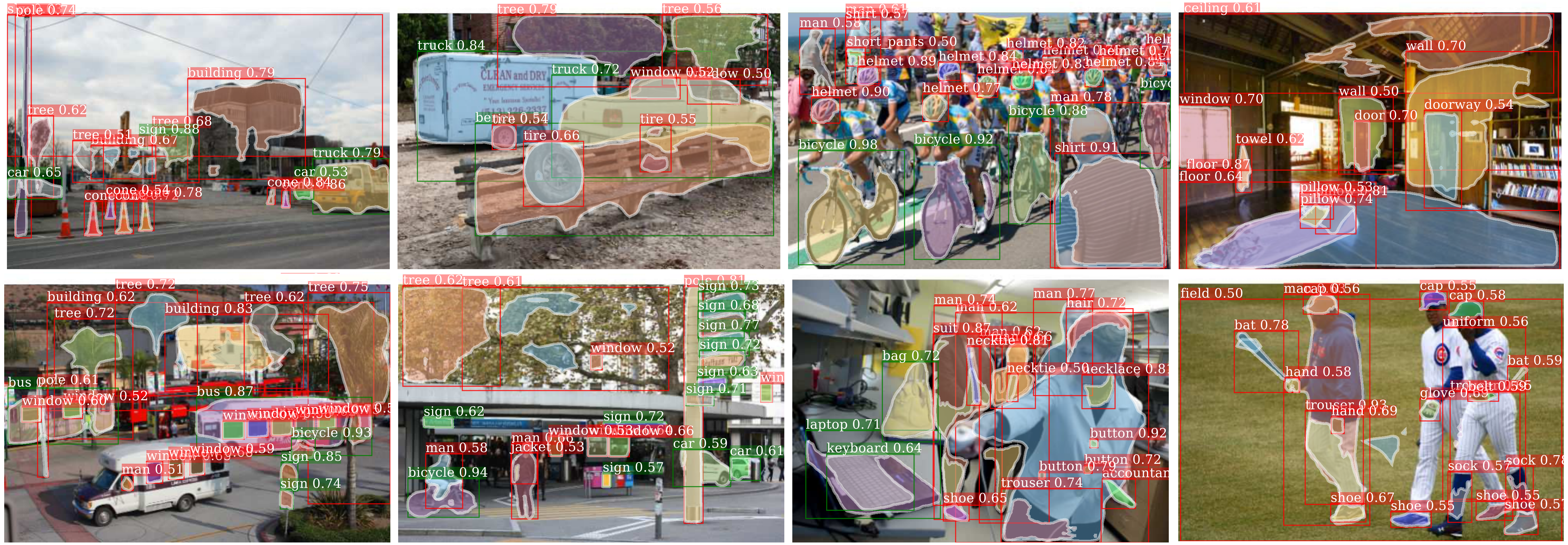
Detection: Label per region, e.g., PASCAL VOC



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

Segmentation

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

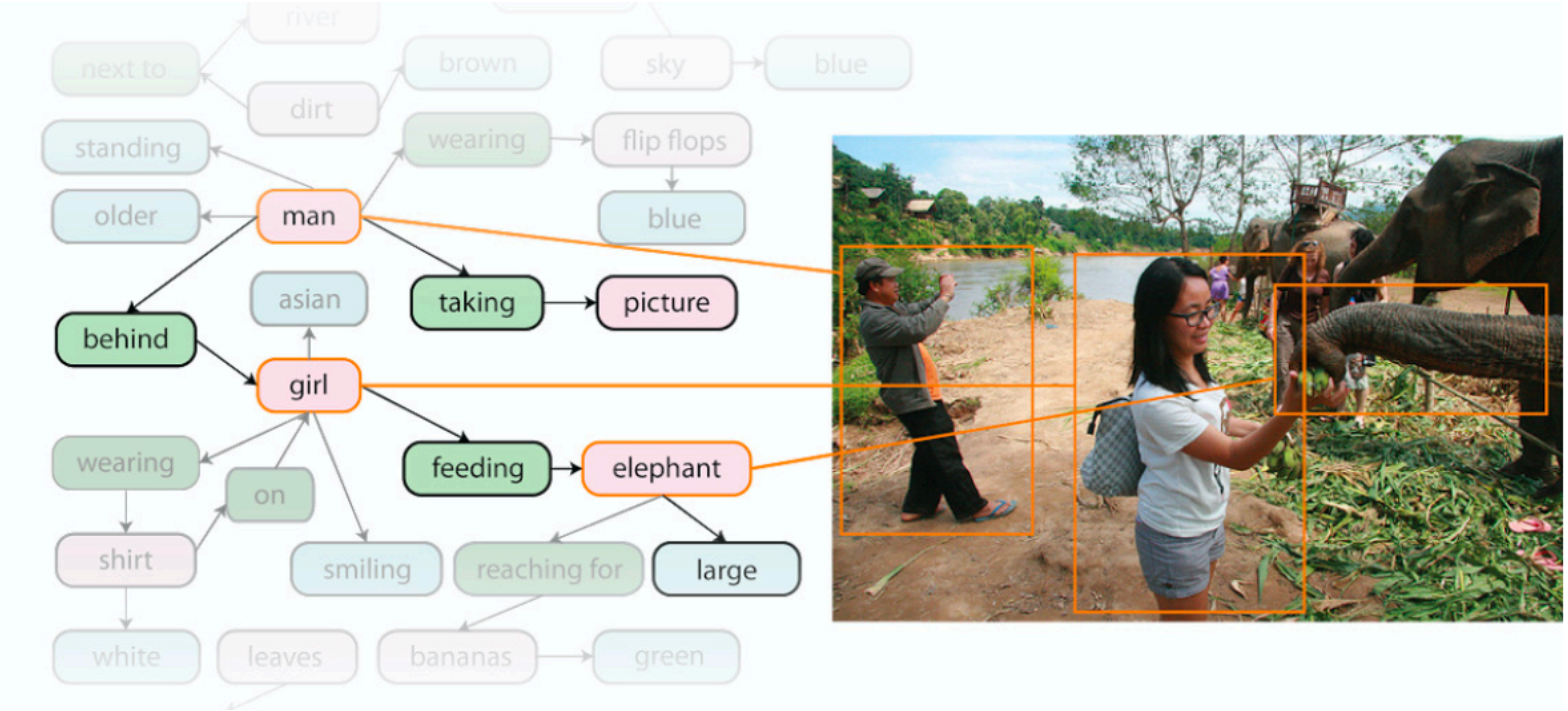


[Hu et al 2017]

Structured **Image Understanding**

"Girl feeding large elephant"

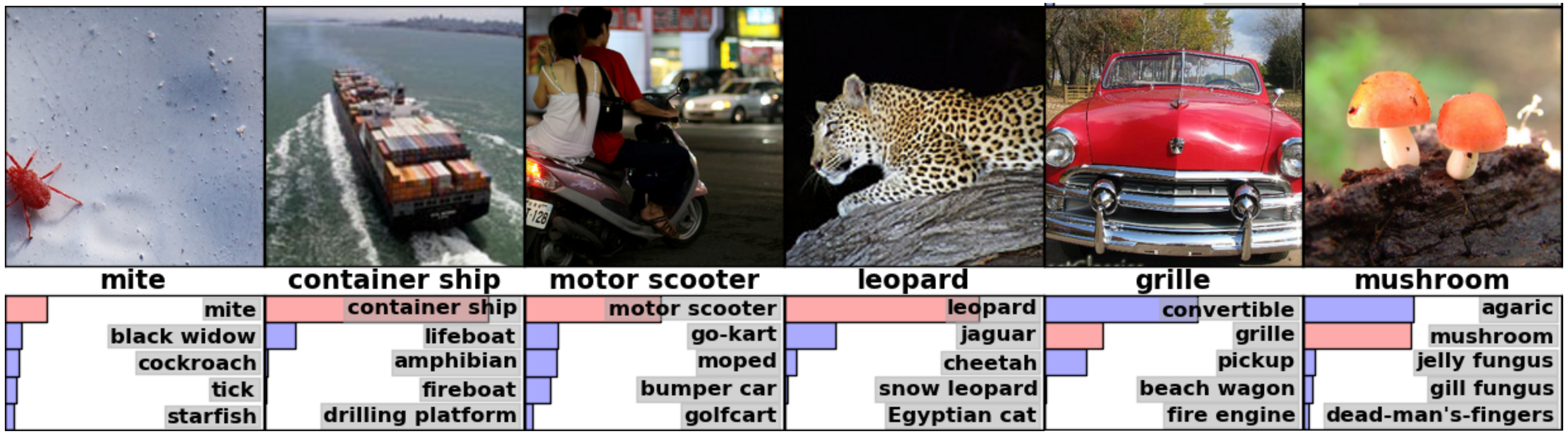
"A man taking a picture behind girl"



visualgenome.org [Krishna et al 2017]

Object **Classification**

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



[Krizhevskv et al 2011][Ren et al 2016]

Classification: **Instance** vs. **Category**

Instance of Aeroplane (Wright Flyer)

Category of Aeroplane

[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]

Classification

Problem:

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

Key Idea(s):

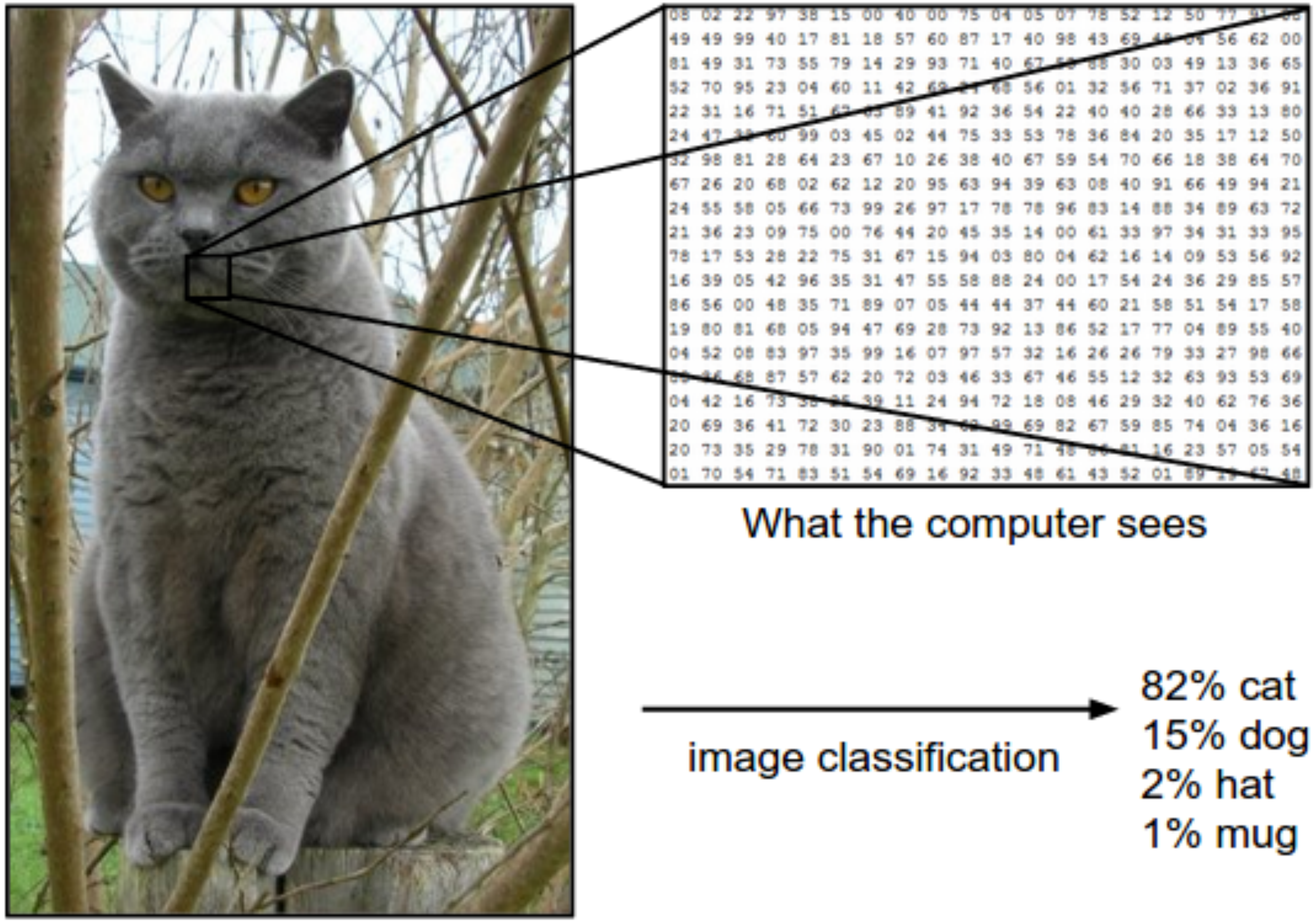
Build a model of data in a given category based on observations of instances in that category

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** (probability over class labels)

Classification

A **classifier** is a procedure that accepts as input a set of features and outputs a class **label** (probability over class labels)

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

Binary: $[0]/[1]$

Multi-class: $[1, 0, 0, 0, ...]$ (one-hot)

Classification

A **classifier** is a procedure that accepts as input a set of features and outputs a class **label** (probability over class labels)

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.

Binary: $[0]/[1]$

Multi-class: $[1, 0, 0, 0, ...]$ (one-hot)

Classification

A **classifier** is a procedure that accepts as input a set of features and outputs a class **label** (probability over class labels)

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.

Binary: $[0]/[1]$ **Multi-class**: $[1, 0, 0, 0, ...]$ (one-hot)

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

Example 1: A Toy Classification Problem

Categorize images of fish
— “Atlantic salmon” vs “Pacific salmon”

Use **features** such as length, width, lightness, fin shape & number, mouth position, etc.

Given a previously unobserved image of a salmon, use the learned classifier to guess whether it is an Atlantic or Pacific salmon

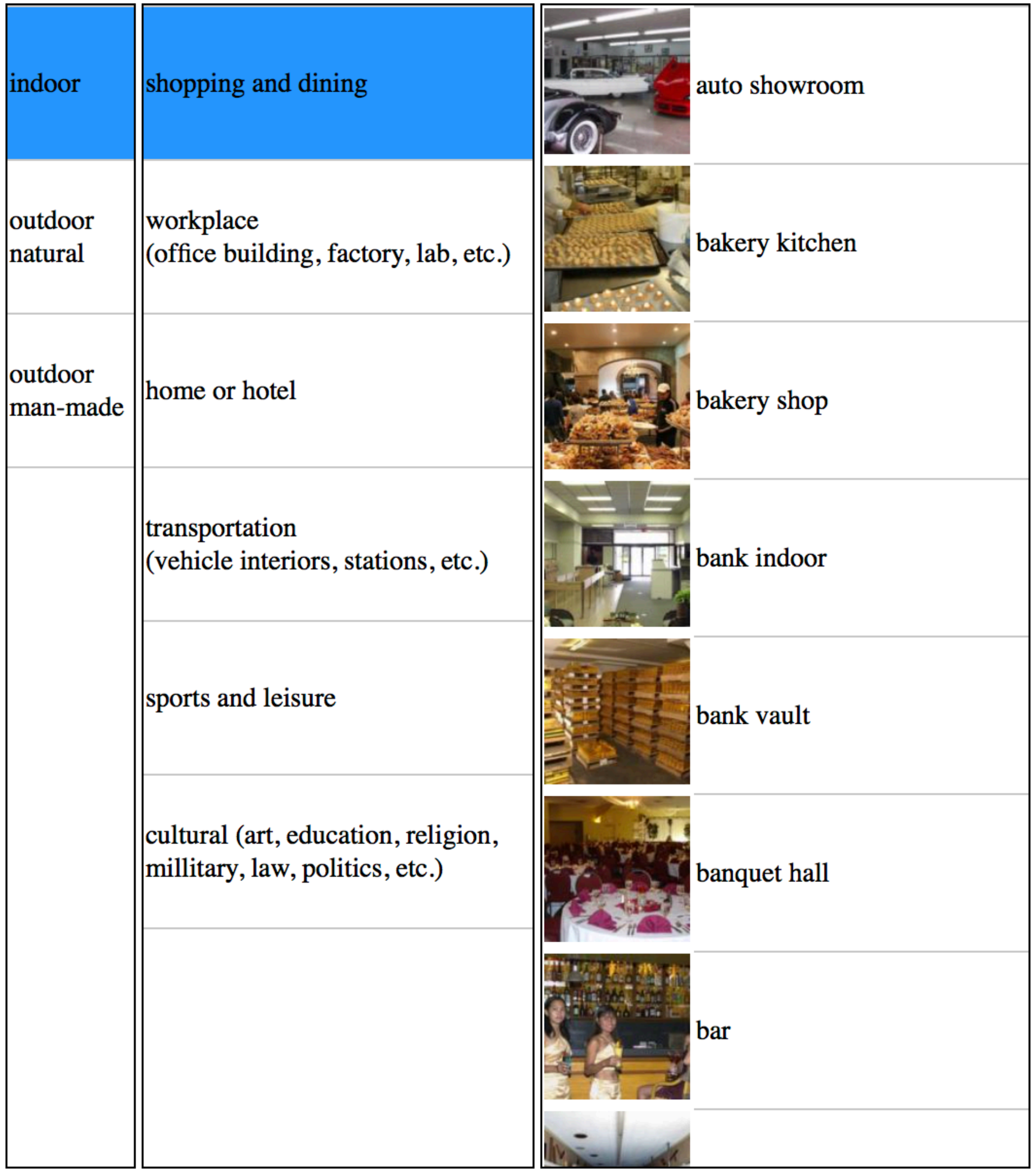
Figure credit: Duda & Hart

Example 2: Real Classification Problem

SUN Dataset

- 131K images

- 908 **scene** categories

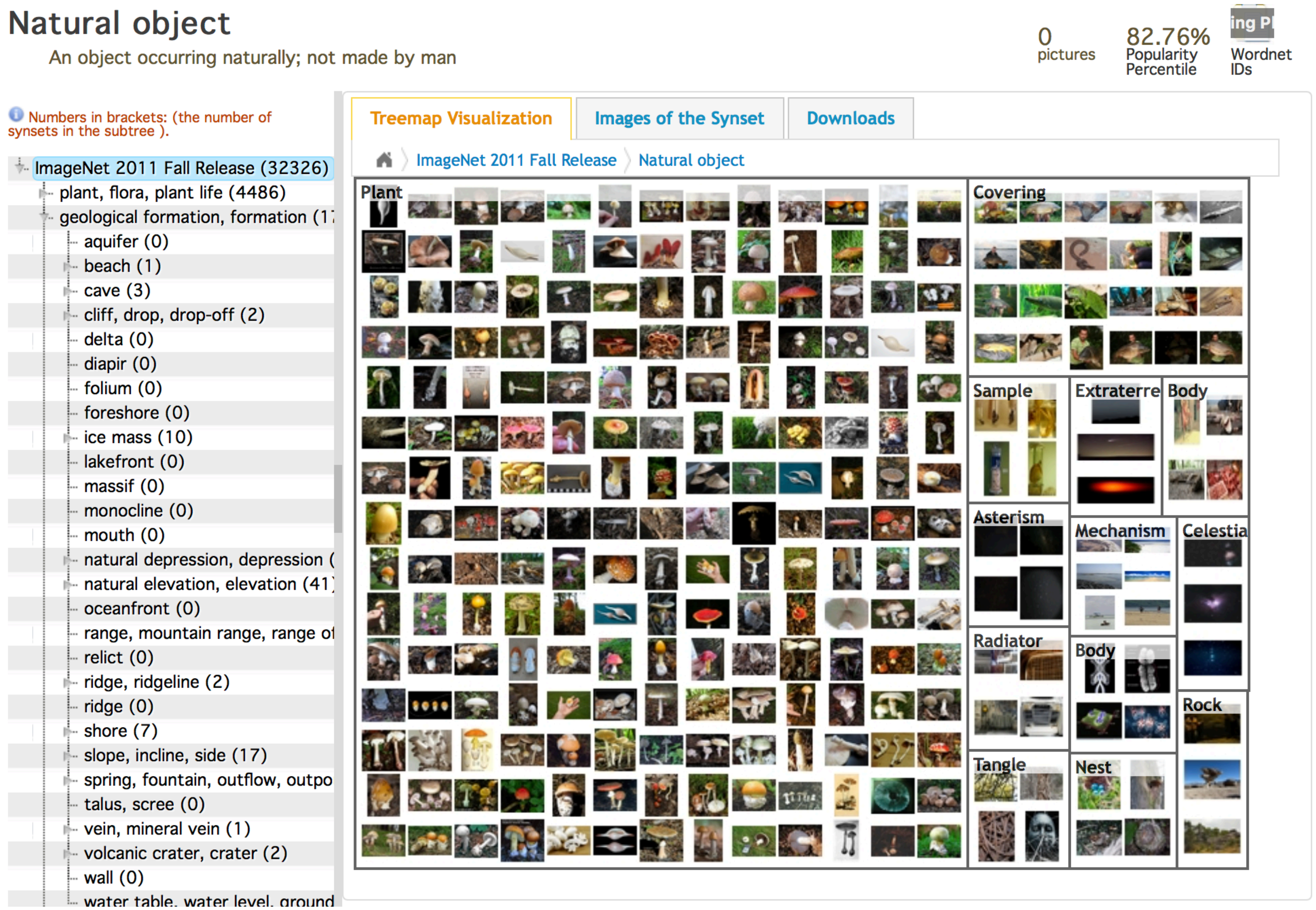


Example 3: Real Classification Problem

ImageNet Dataset

- 14 Million images

- 21K **object** categories

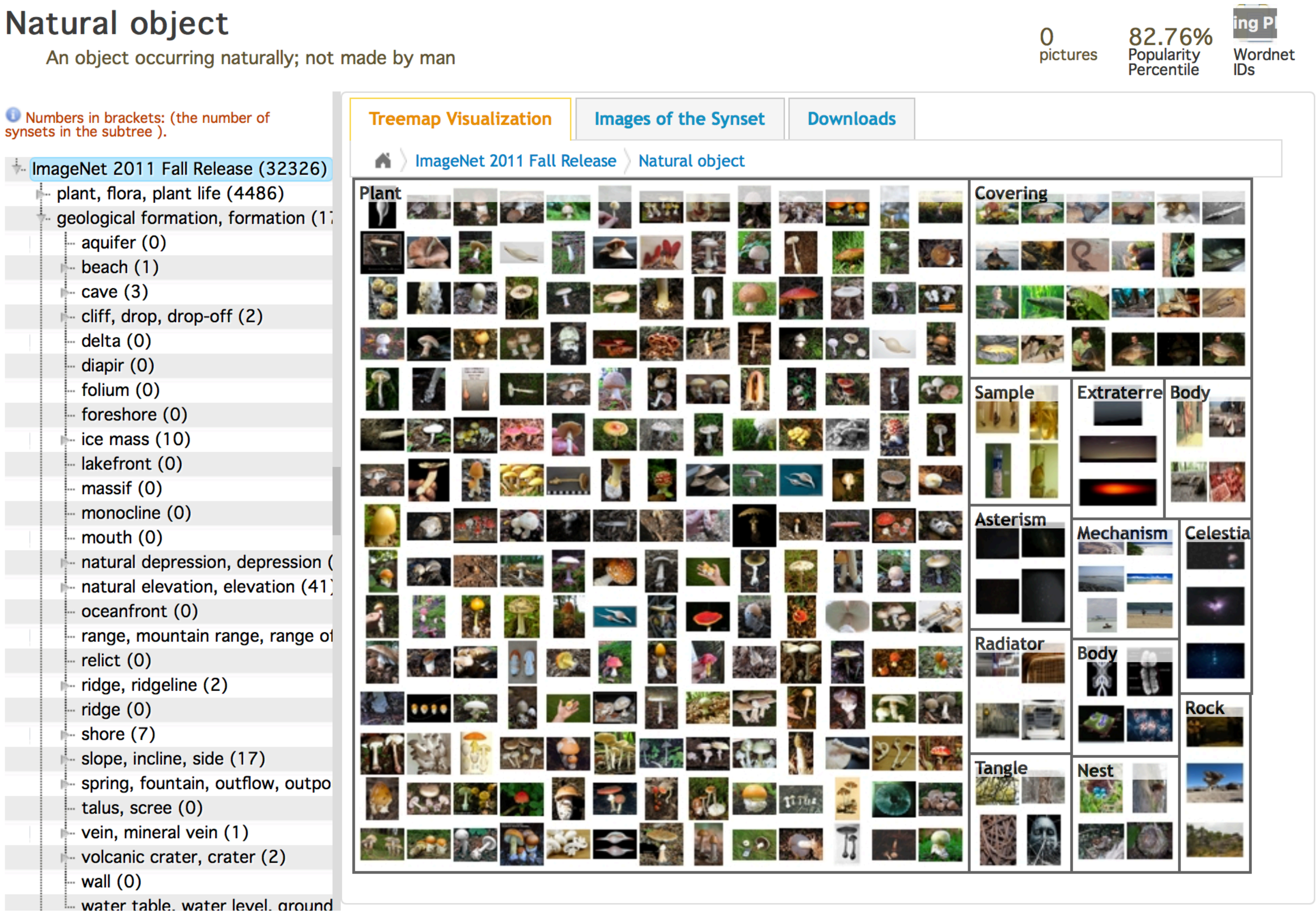


Example 3: Real Classification Problem

ImageNet Dataset

- 14 Million images

- 21K **object** categories



Closed-world problem

Issue: Classification assumes that incoming image belongs to one of k classes. However, in practice it is impossible to enumerate all relevant classes in the world, nor would doing so be useful. So how do we deal with images which don't belong?

Solution: Create an "unknown" or "irrelevant" class.

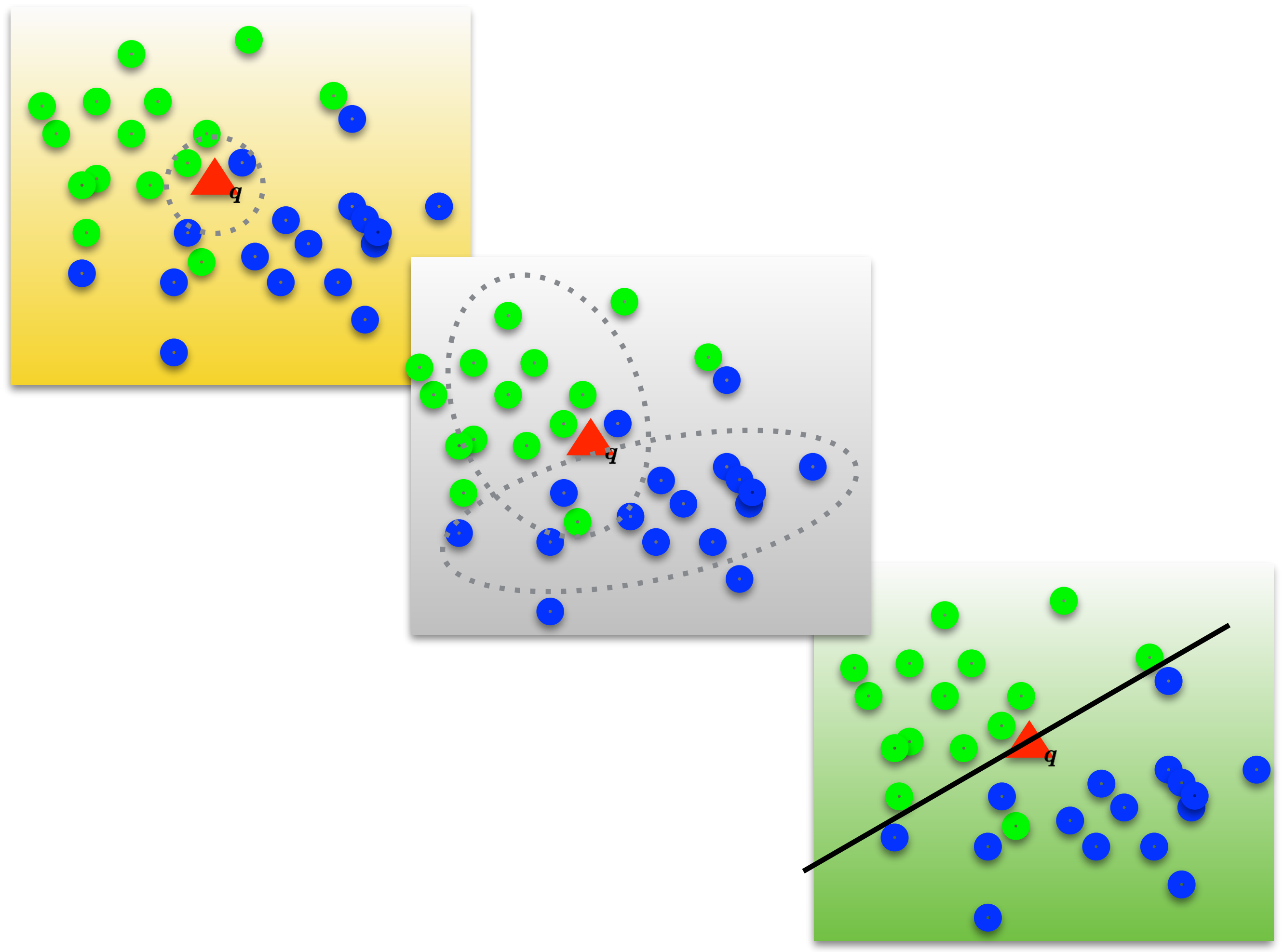
Image Classification

Classification Algorithms

— Bayes’ Classifier

— Nearest Neighbor Classifier

— SVM Classifier



Representation of Images

— Image pixels directly

— Bag of Words

Bayes Rule (Review and Definitions)

Let c be the **class label** and let x be the **measurement** (i.e., evidence)

$$\underbrace{P(c|x)}_{\text{posterior probability}} = \frac{P(x|c)p(c)}{P(x)}$$

Bayes Rule (Review and Definitions)

Let c be the **class label** and let x be the **measurement** (i.e., evidence)

class–conditional probability (a.k.a. likelihood)

prior probability

$$P(c|x) = \frac{P(x|c)p(c)}{P(x)}$$

posterior probability

unconditional probability (a.k.a. marginal likelihood)

Bayes Rule (Review and Definitions)

Let c be the **class label** and let x be the **measurement** (i.e., evidence)

Simple case:

— binary classification; i.e., $c \in \{1, 2\}$

— features are 1D; i.e., $x \in \mathbb{R}$

$$P(c|x) = \frac{P(x|c)p(c)}{P(x)}$$

Bayes Rule (Review and Definitions)

Let c be the **class label** and let x be the **measurement** (i.e., evidence)

Simple case:

— binary classification; i.e., $c \in \{1, 2\}$

— features are 1D; i.e., $x \in \mathbb{R}$

$$P(c|x) = \frac{P(x|c)p(c)}{P(x)}$$

Classify ***x*** as

1 if p(1|***x***) > p(2|***x***)

2 if p(1|***x***) < p(2|***x***)

Bayes Rule (Review and Definitions)

Let c be the **class label** and let x be the **measurement** (i.e., evidence)

Simple case:

— binary classification; i.e., $c \in \{1, 2\}$

— features are 1D; i.e., $x \in \mathbb{R}$

$$P(c|x) = \frac{P(x|c)p(c)}{P(x)}$$

General case:

— multi-class; i.e., $c \in \{1, ..., 1000\}$

— features are high-dimensional; i.e., $x \in \mathbb{R}^{2,000+}$

Example: Discrete Bayes Classifier

Assume we have two classes: $c_1 = \textbf{male}$ $c_2 = \textbf{female}$

We have a person who's gender we don't know, who's name is *drew*

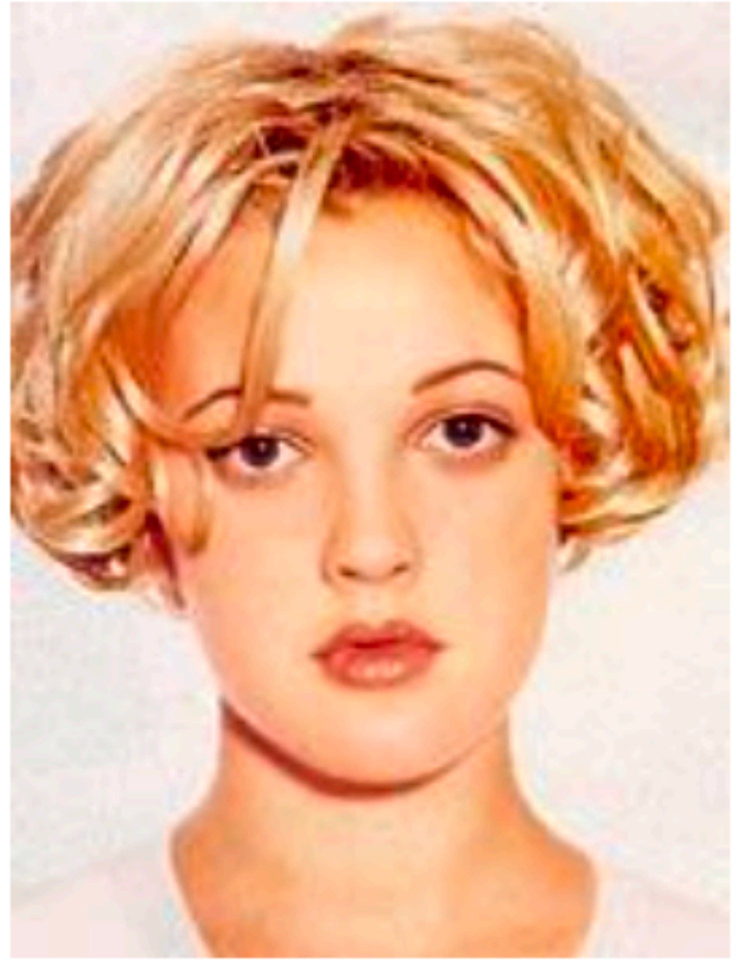
Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

Assume we have two classes: $c_1 = \textbf{male}$ $c_2 = \textbf{female}$

We have a person who's gender we don't know, who's name is *drew*

Drew Carey



Drew Barrymore

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

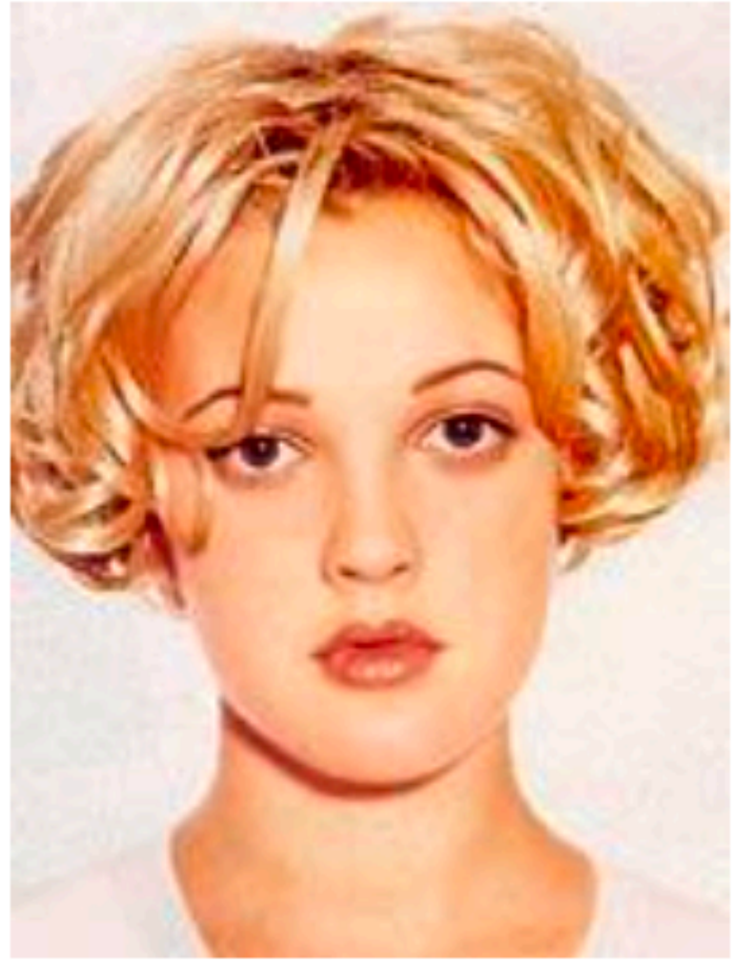
Assume we have two classes: $c_1 = \textbf{male}$ $c_2 = \textbf{female}$

We have a person who's gender we don't know, who's name is *drew*

Classifying drew as being male or female is equivalent to asking is it more probable that *drew* is male or female, i.e. which is greater $p(\textbf{male}|drew)$

$p(\textbf{female}|drew)$

Drew Carey



Drew Barrymore

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

Assume we have two classes: $c_1 = \textbf{male}$ $c_2 = \textbf{female}$

We have a person who's gender we don't know, who's name is *drew*

Classifying drew as being male or female is equivalent to asking is it more probable that *drew* is male or female, i.e. which is greater $p(\textbf{male}|drew)$

$p(\textbf{female}|drew)$

$$p(\textbf{male}|drew) = \frac{p(drew|\textbf{male})p(\textbf{male})}{p(drew)}$$

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

Name	Gender
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$p(\mathbf{male}|drew) = \frac{p(drew|\mathbf{male})p(\mathbf{male})}{p(drew)}$$

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

$$p(\mathbf{male}) =$$

$$p(drew|\mathbf{male}) =$$

$$p(drew) =$$

$$p(\mathbf{male}|drew) = \frac{p(drew|\mathbf{male})p(\mathbf{male})}{p(drew)}$$

Name	Gender
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

$$p(\mathbf{male}) = \frac{3}{8}$$

$$p(drew|\mathbf{male}) =$$

$$p(drew) =$$

$$p(\mathbf{male}|drew) = \frac{p(drew|\mathbf{male})p(\mathbf{male})}{p(drew)}$$

Name	Gender
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

Name	Gender
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$p(\mathbf{male}) = \frac{3}{8}$$

$$p(drew|\mathbf{male}) = \frac{1}{3}$$

$$p(drew) =$$

$$p(\mathbf{male}|drew) = \frac{p(drew|\mathbf{male})p(\mathbf{male})}{p(drew)}$$

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

Name	Gender
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$p(\mathbf{male}) = \frac{3}{8}$$

$$p(drew|\mathbf{male}) = \frac{1}{3}$$

$$p(drew) = \frac{3}{8}$$

$$p(\mathbf{male}|drew) = \frac{p(drew|\mathbf{male})p(\mathbf{male})}{p(drew)}$$

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

$$p(\mathbf{male}) = \frac{3}{8}$$

$$p(drew|\mathbf{male}) = \frac{1}{3}$$

$$\cancel{p(drew) = \frac{3}{8}}$$

$$p(\mathbf{male}|drew) = \frac{p(drew|\mathbf{male})p(\mathbf{male})}{\cancel{p(drew)}} = 0.125$$

Name	Gender
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

Example from: Eamonn Keogh

Example: Discrete Bayes Classifier

Name	Gender
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$p(\mathbf{male}) = \frac{3}{8} \qquad p(\mathbf{female}) = \frac{5}{8}$$

$$p(drew|\mathbf{male}) = \frac{1}{3} \qquad p(drew|\mathbf{female}) = \frac{2}{5}$$

$$\cancel{p(drew) = \frac{3}{8}}$$

$$p(\mathbf{male}|drew) = \frac{p(drew|\mathbf{male})p(\mathbf{male})}{\cancel{p(drew)}} = 0.125$$

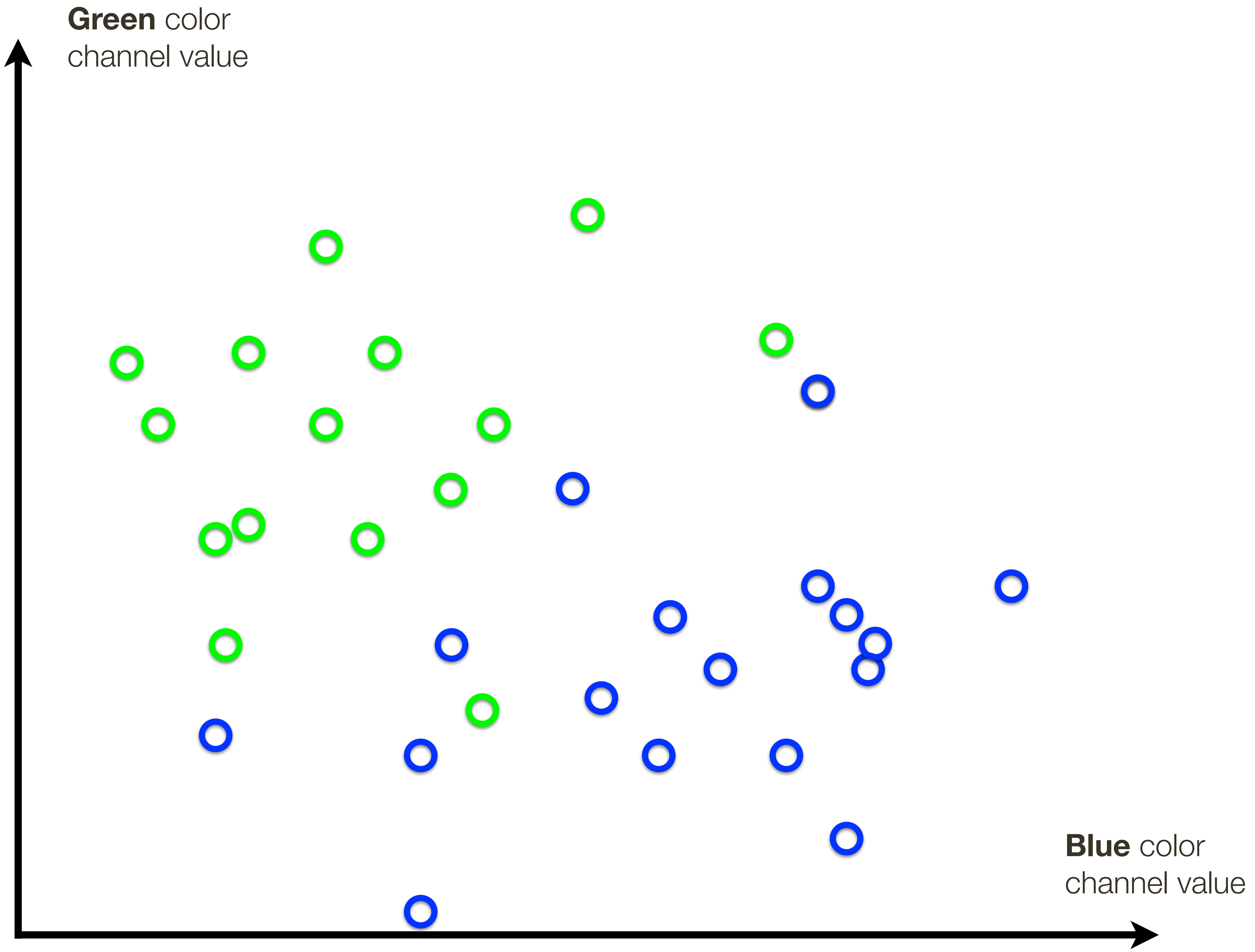
$$p(\mathbf{female}|drew) = \frac{p(drew|\mathbf{female})p(\mathbf{female})}{\cancel{p(drew)}} = 0.25$$

Example from: Eamonn Keogh

Example: 2D Bayes Classifier

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

- 17 samples of grass
- 15 samples of sky

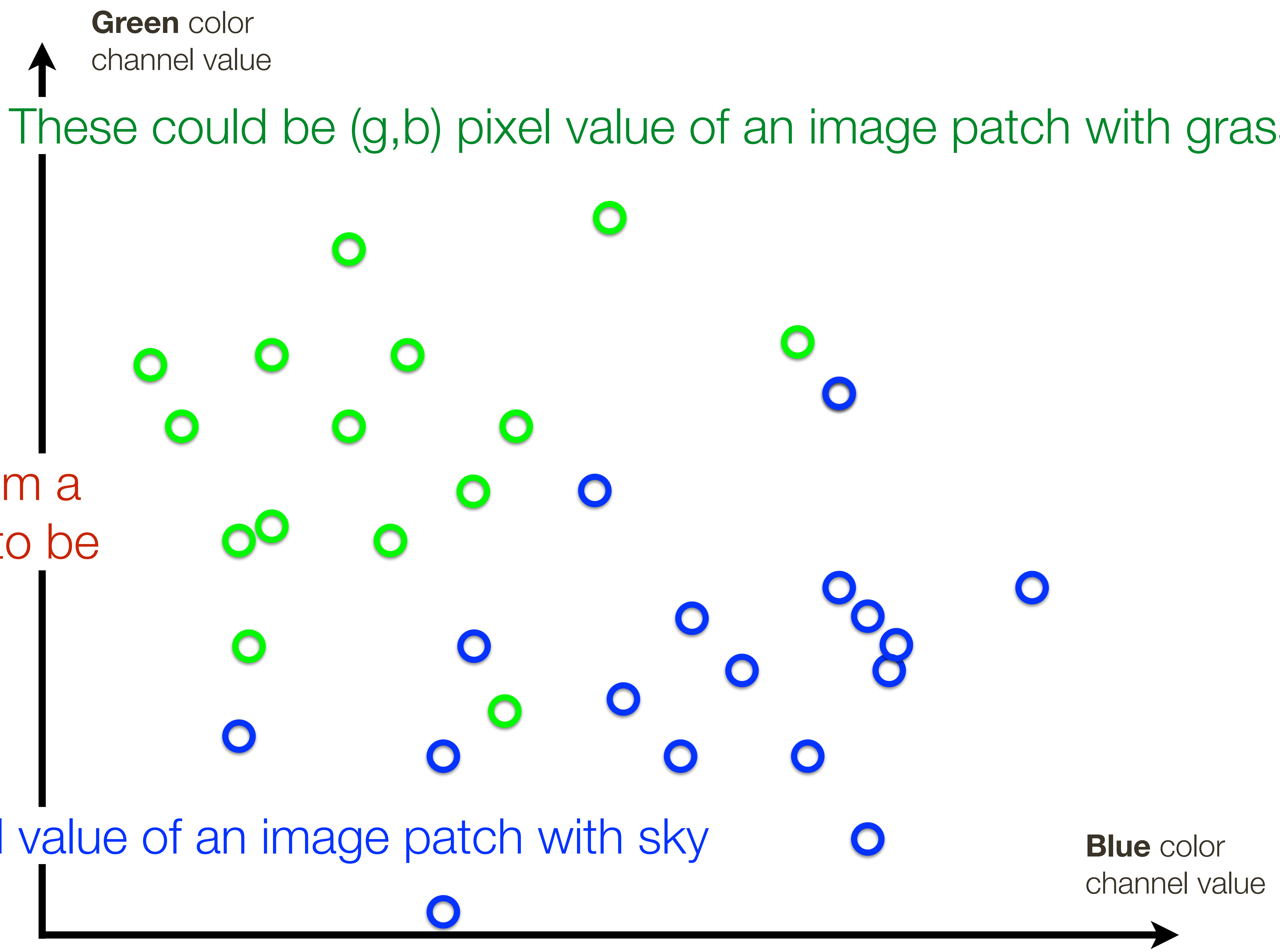


Example: 2D Bayes Classifier

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

- 17 samples of grass
- 15 samples of sky

These could be (g,b) pixel value of an image patch with grass



Given a (g,b) pixel value from a new patch is it more likely to be be grass or sky?

These could be (g,b) pixel value of an image patch with sky

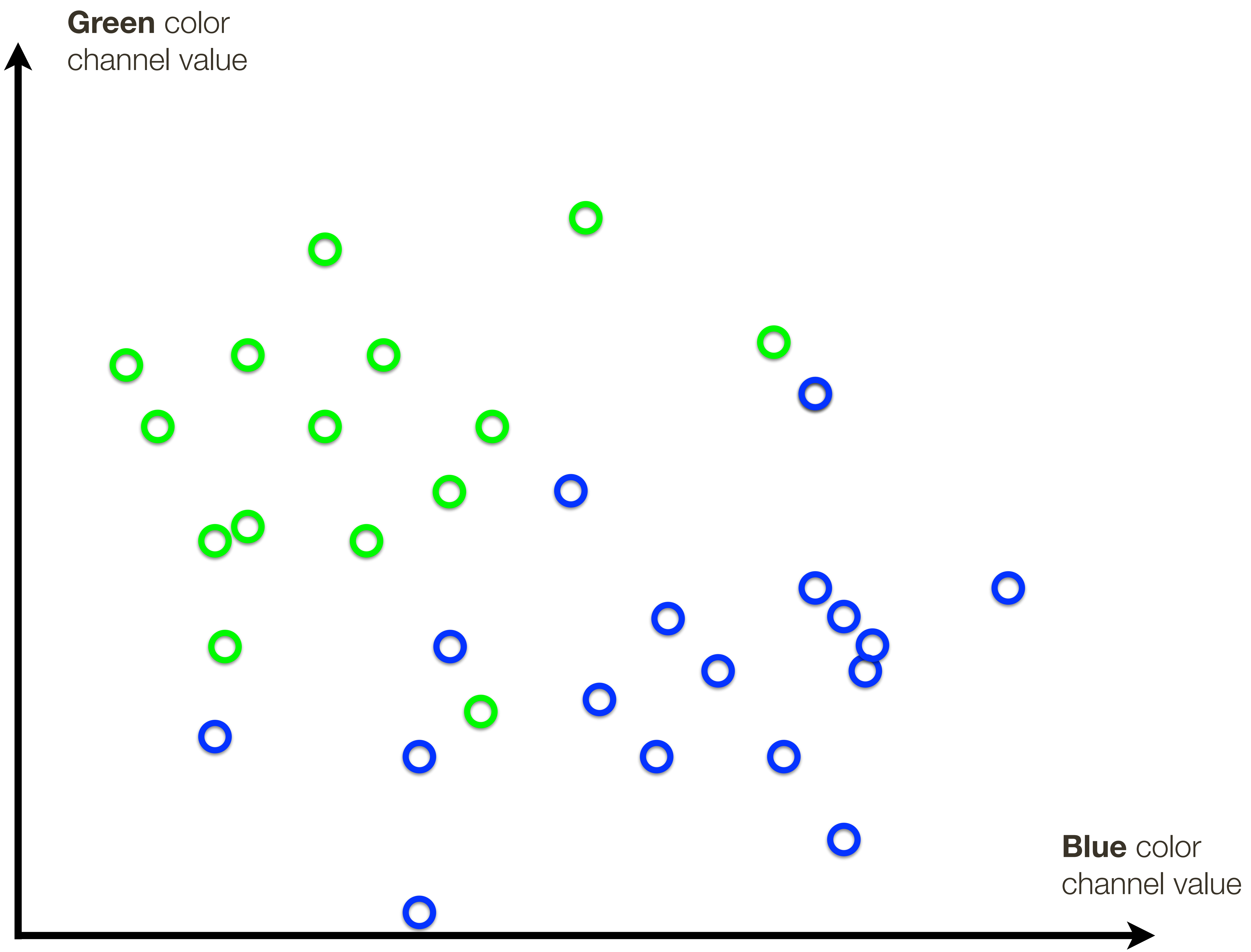
Example: 2D Bayes Classifier

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

- 17 samples of grass
- 15 samples of sky

$$p(blue) = \frac{17}{17+15}$$

$$p(green) = \frac{15}{17+15}$$



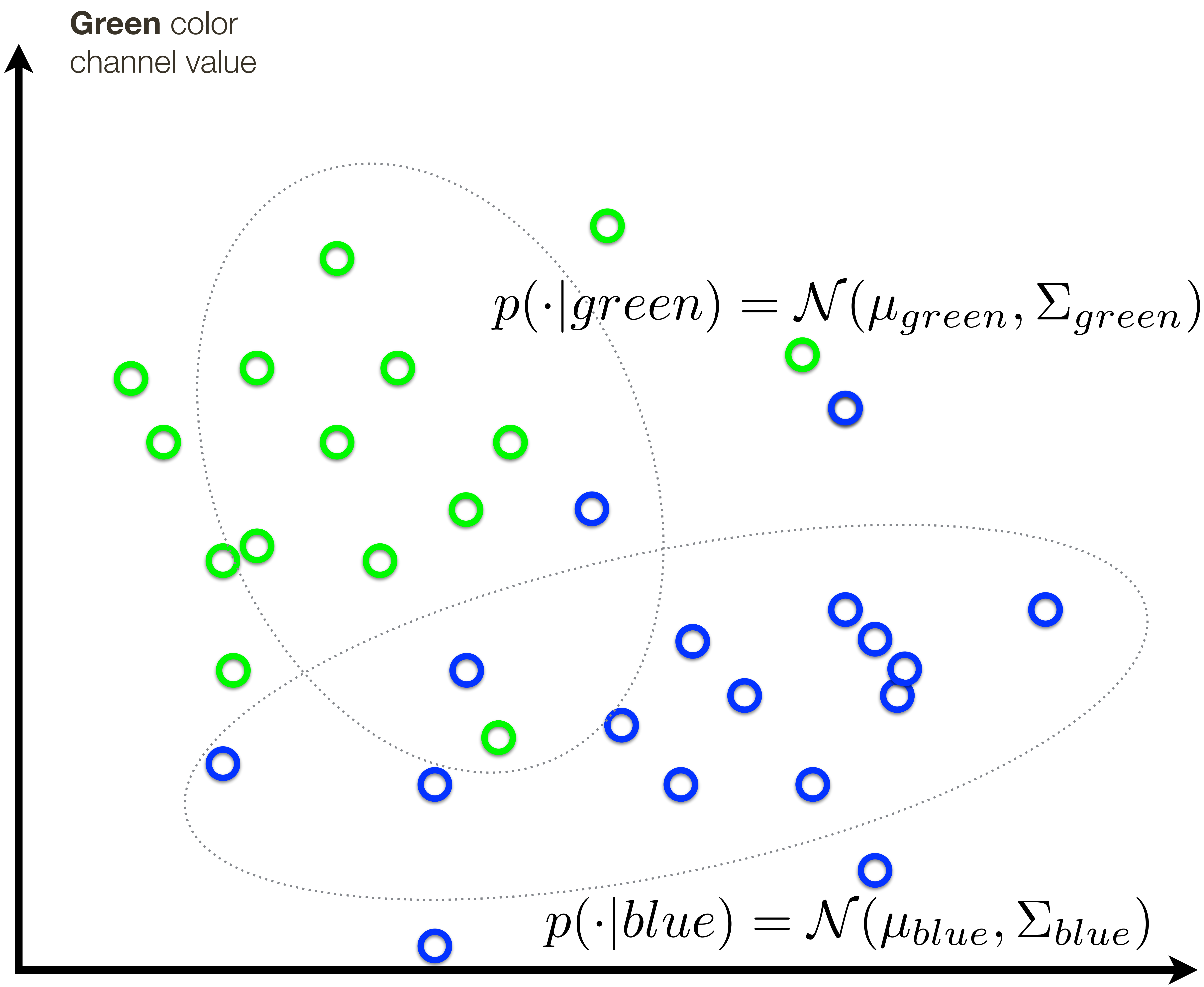
Example: 2D Bayes Classifier

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

- 17 samples of grass
- 15 samples of sky

$$p(blue) = \frac{17}{17+15}$$

$$p(green) = \frac{15}{17+15}$$



Example: 2D Bayes Classifier

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

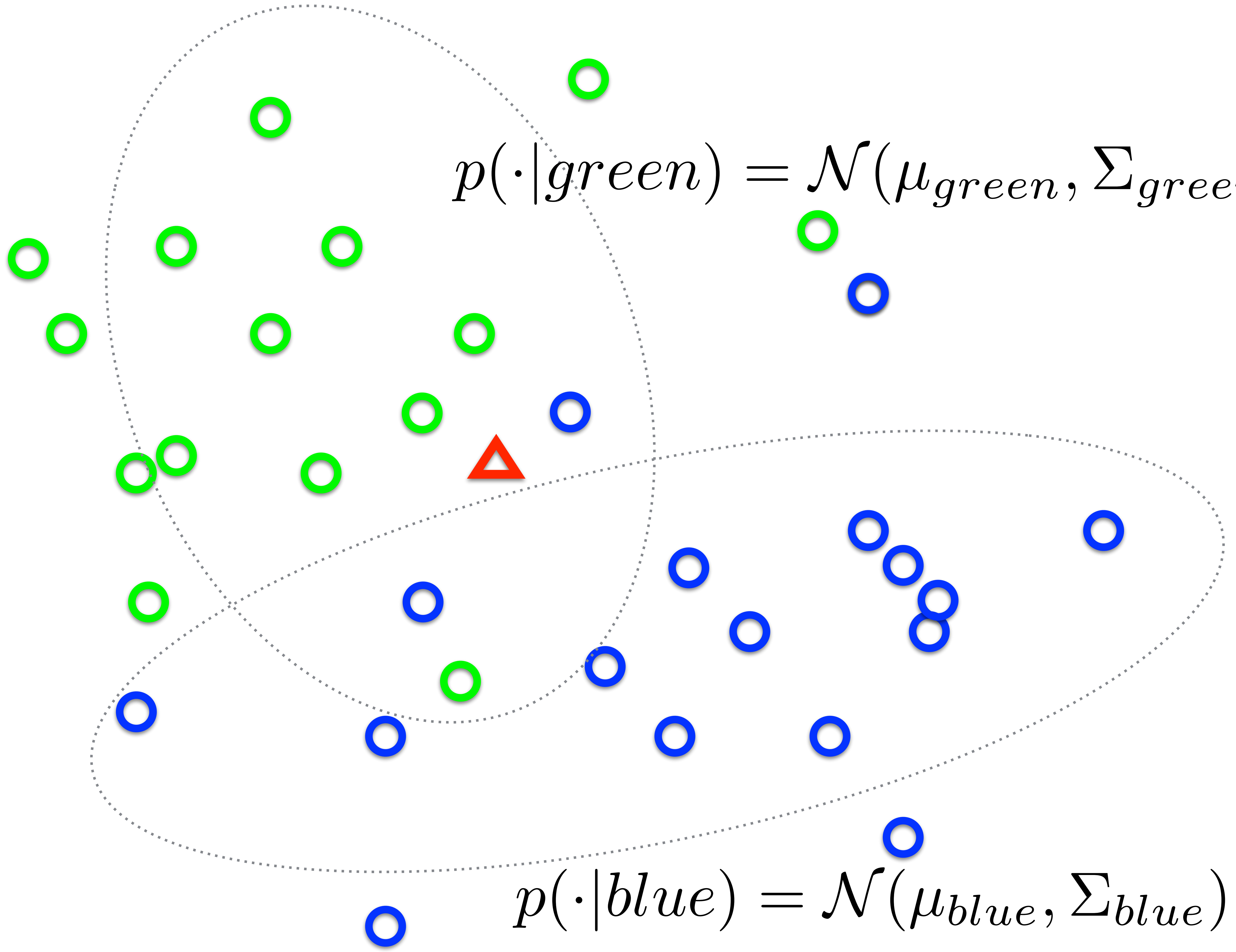
$$p(green|\triangle) \propto \mathcal{N}(\triangle; \mu_{green}, \Sigma_{green})p(green)$$

$$p(blue|\triangle) \propto \mathcal{N}(\triangle; \mu_{blue}, \Sigma_{blue})p(blue)$$

- 17 samples of grass
- 15 samples of sky

$$p(blue) = \frac{17}{17+15}$$

$$p(green) = \frac{15}{17+15}$$



$$p(\cdot|green) = \mathcal{N}(\mu_{green}, \Sigma_{green})$$

$$p(\cdot|blue) = \mathcal{N}(\mu_{blue}, \Sigma_{blue})$$

Bayes Rule (Review and Definitions)

Let c be the **class label** and let x be the **measurement** (i.e., evidence)

Simple case:

— binary classification; i.e., $c \in \{1, 2\}$

— features are 1D; i.e., $x \in \mathbb{R}$

$$P(c|x) = \frac{P(x|c)p(c)}{P(x)}$$

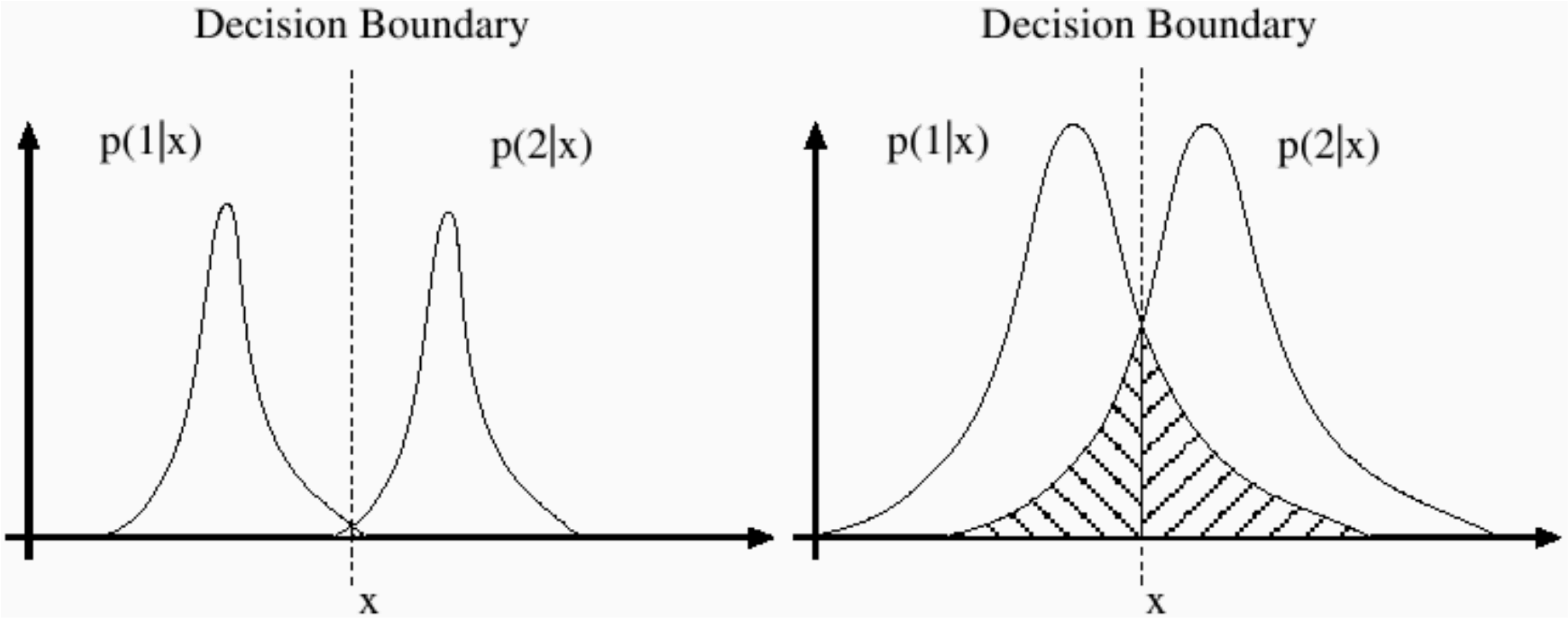
General case:

— multi-class; i.e., $c \in \{1, ..., 1000\}$

— features are high-dimensional; i.e., $x \in \mathbb{R}^{2,000+}$

Bayes' Risk

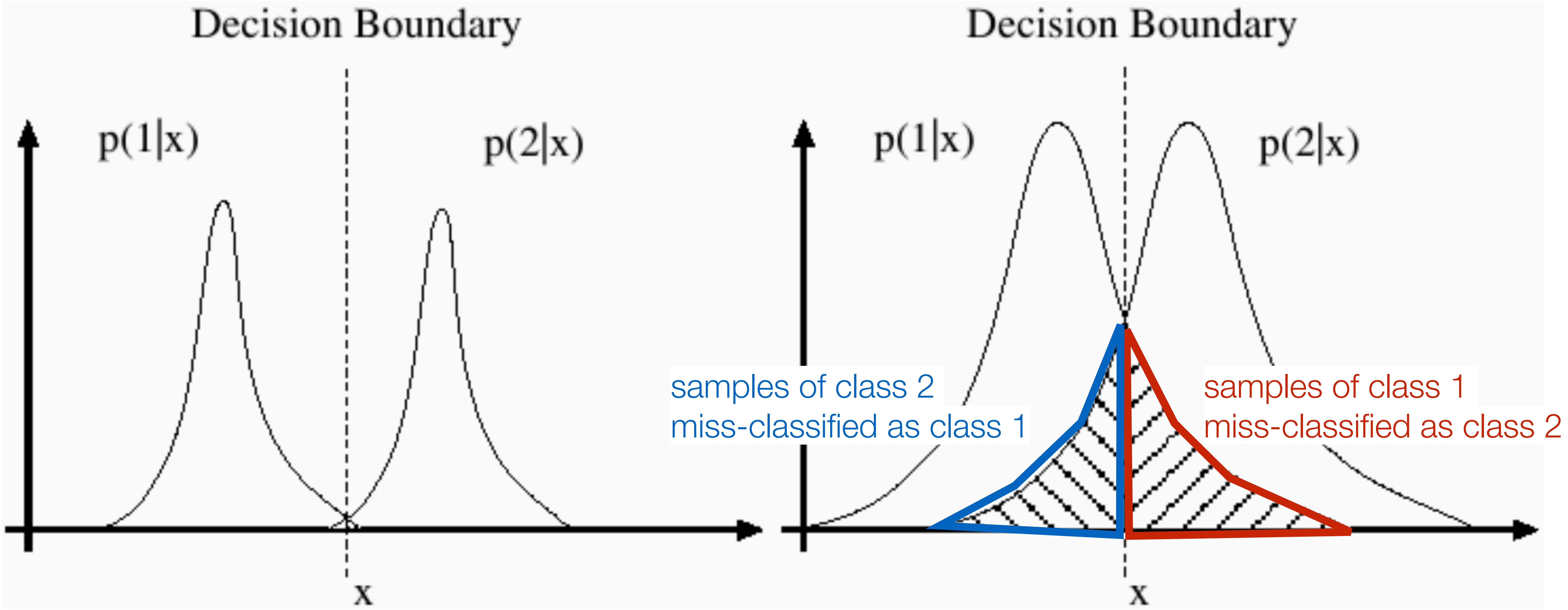
Some errors may be inevitable: the minimum risk (shaded area) is called the **Bayes' risk**



Forsyth & Ponce (2nd ed.) Figure 15.1

Bayes' Risk

Some errors may be inevitable: the minimum risk (shaded area) is called the **Bayes' risk**



Forsyth & Ponce (2nd ed.) Figure 15.1

Loss Functions and Classifiers

Loss

— Some errors may be more expensive than others

Example: A fatal disease that is easily cured by a cheap medicine with no side-effects. Here, false positives in diagnosis are better than false negatives

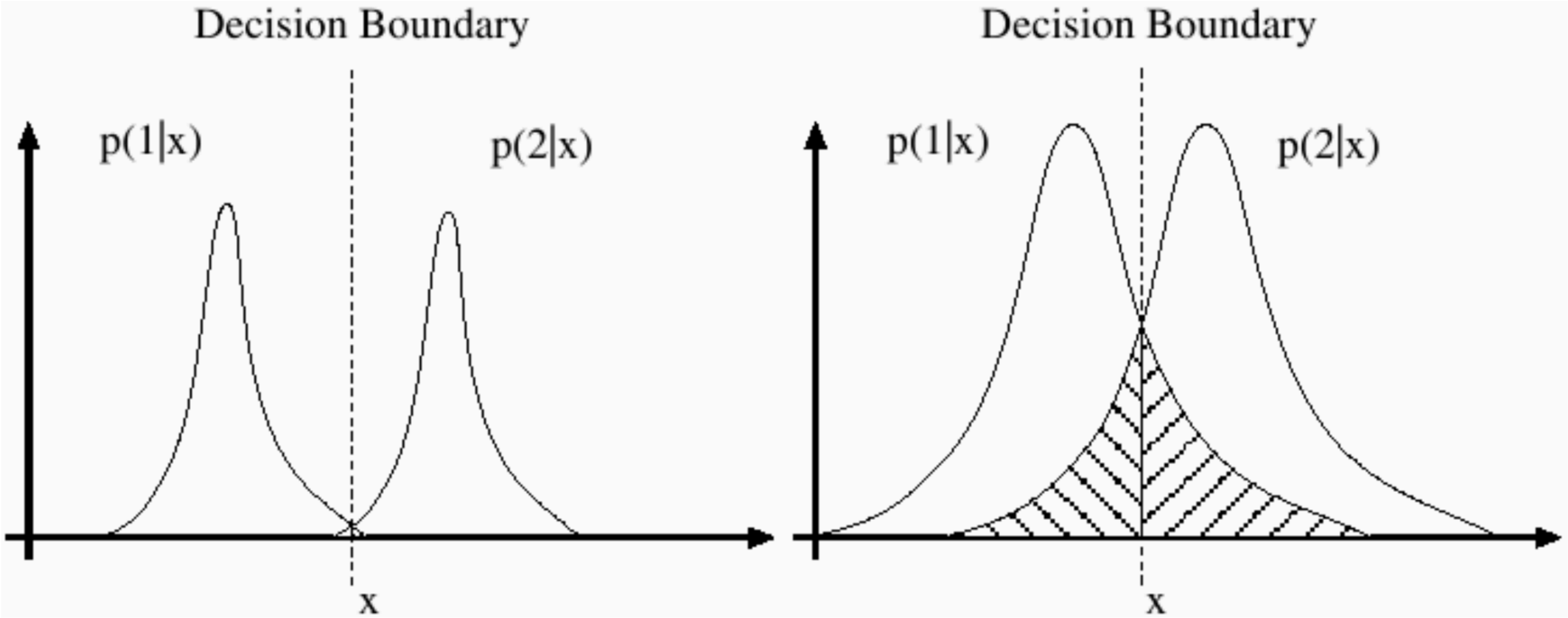
— We discuss two class classification:
$L(1 \rightarrow 2)$ is the loss caused by calling 1 a 2

Total risk of using classifier ***s*** is

$$R(s) = \underbrace{\Pr\{1 \rightarrow 2 \mid \text{using } \boldsymbol{s}\}}_{\text{Probability of Miss-classification}} \underbrace{L(1 \rightarrow 2)}_{\substack{\text{Loss} \\ \text{(i.e. cost of miss-classification)}}} + \underbrace{\Pr\{2 \rightarrow 1 \mid \text{using } \boldsymbol{s}\}}_{\text{Probability of Miss-classification}} \underbrace{L(2 \rightarrow 1)}_{\substack{\text{Loss} \\ \text{(i.e. cost of miss-classification)}}}$$

Bayes' Risk

Some errors may be inevitable: the minimum risk (shaded area) is called the **Bayes' risk**



Forsyth & Ponce (2nd ed.) Figure 15.1

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]

Training Error, **Testing** Error, and Overfitting

Training error is the error a classifier makes on the training set

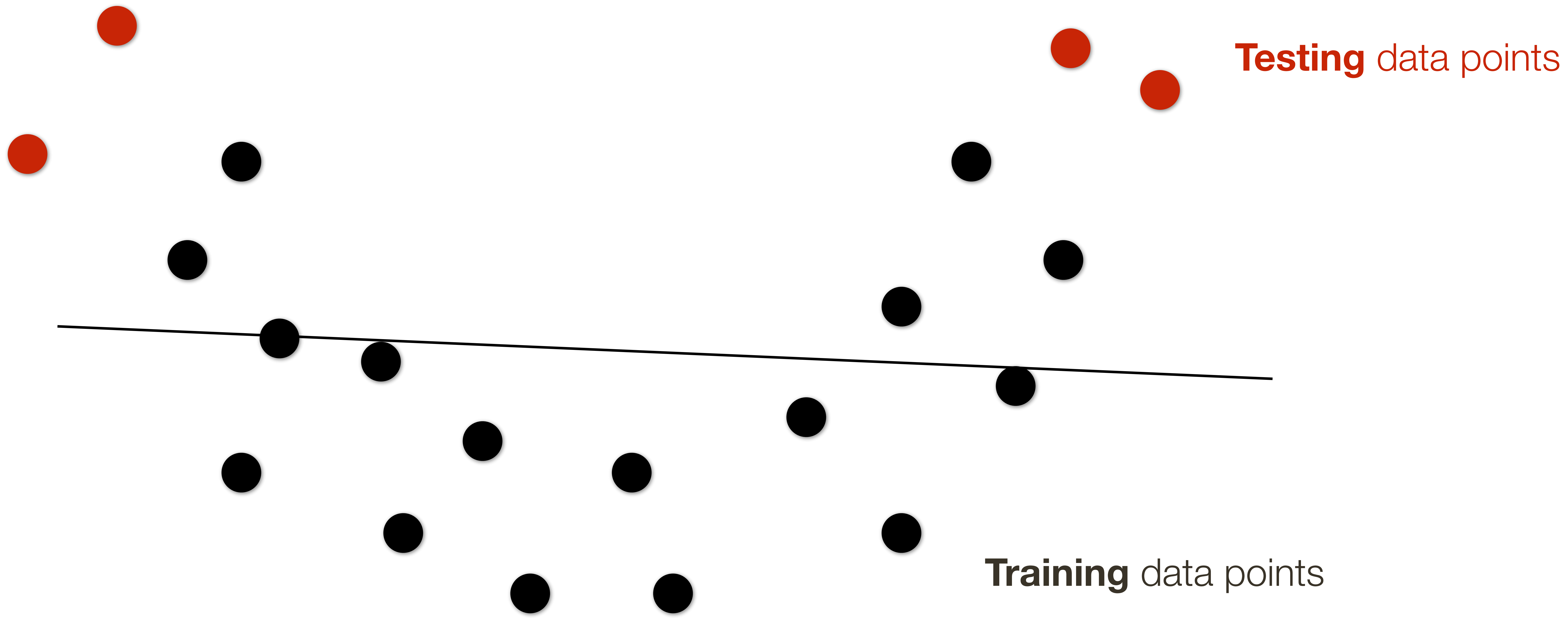
We want to minimize the **testing error** – the error the classifier makes on an unseen testing set

Classifiers that have small training error may not necessarily have small testing error

The phenomenon that causes testing error to be worse than training error is called **overfitting**

Training Error, Testing Error, and **Overfitting**

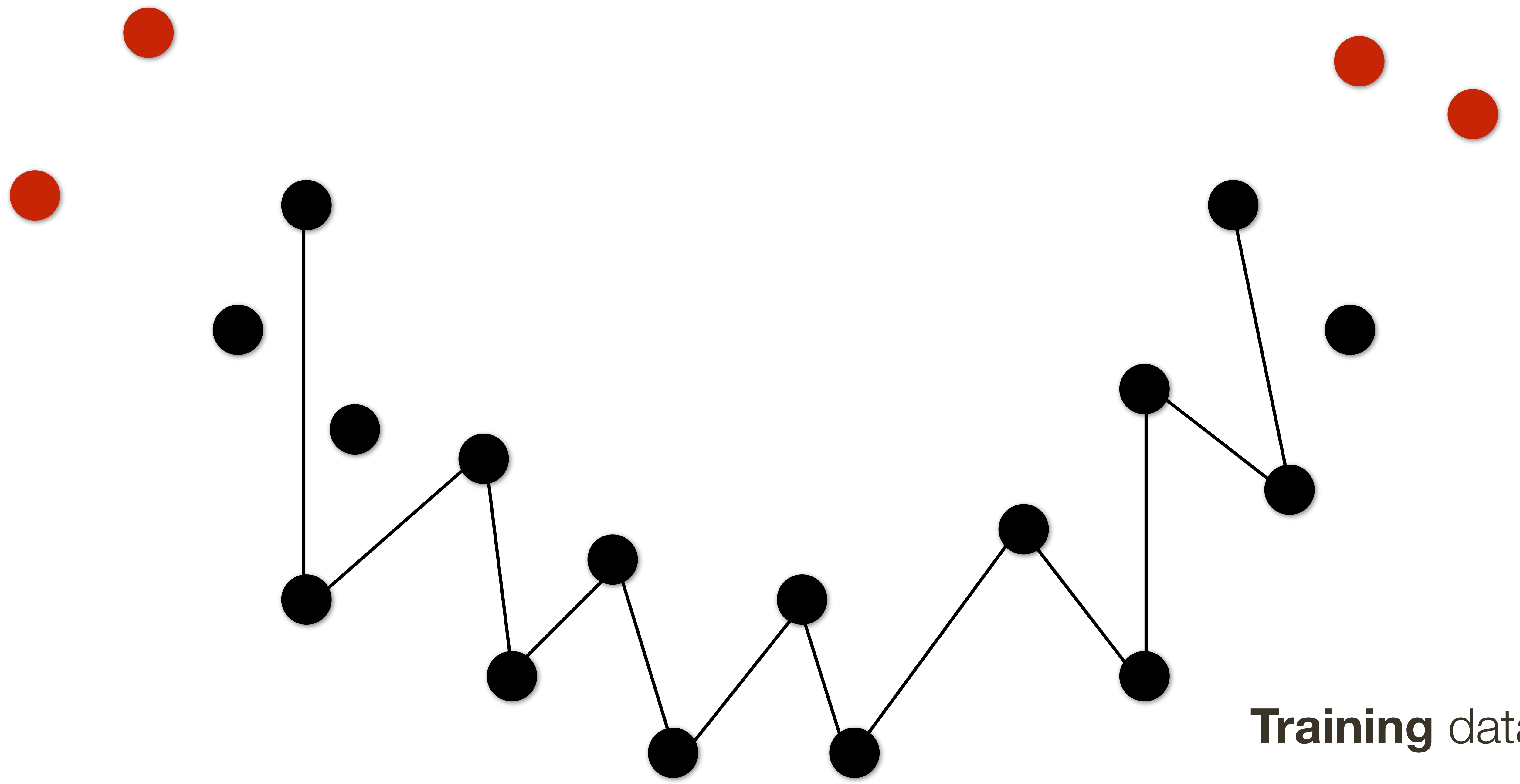
Underfitting: model is too simple to represent all the relevant class characteristics



Training Error, Testing Error, and **Overfitting**

Underfitting: model is too simple to represent all the relevant class characteristics

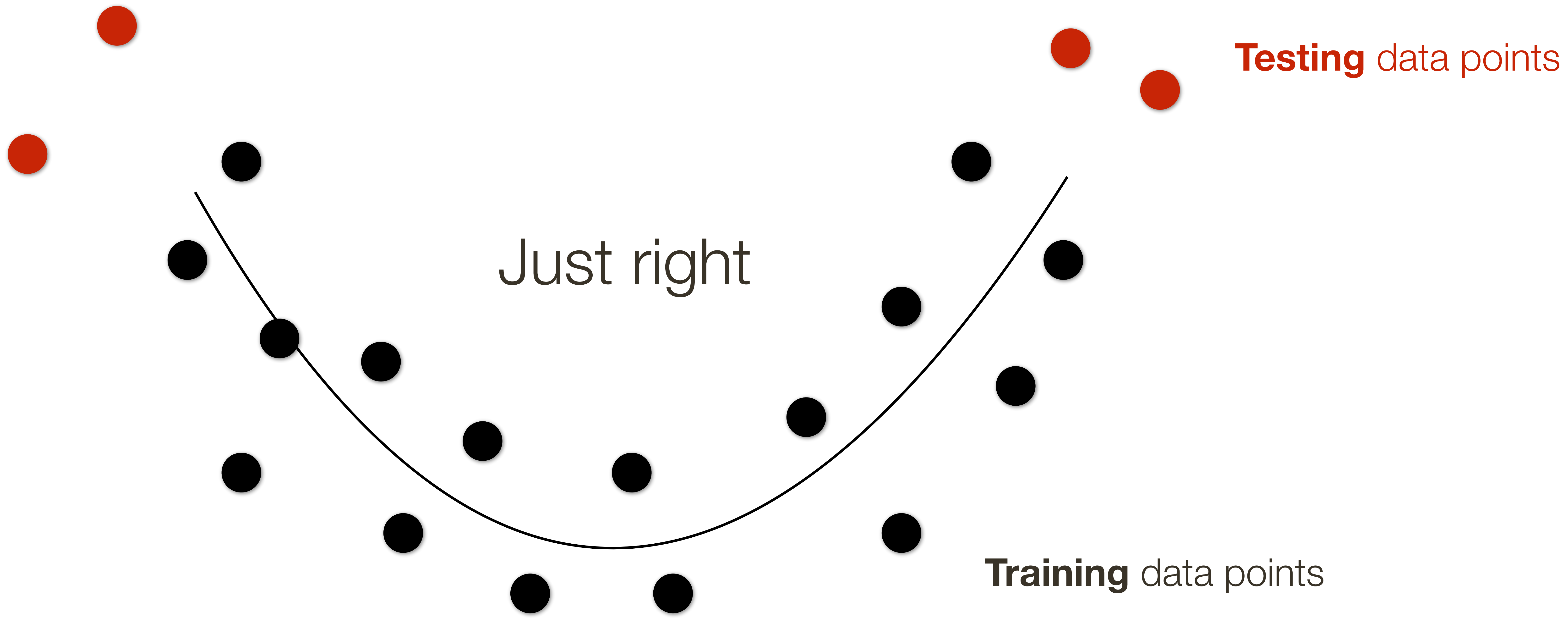
Overfitting: model is too complex and fits irrelevant characteristics (noise) in the data



Training Error, Testing Error, and **Overfitting**

Underfitting: model is too simple to represent all the relevant class characteristics

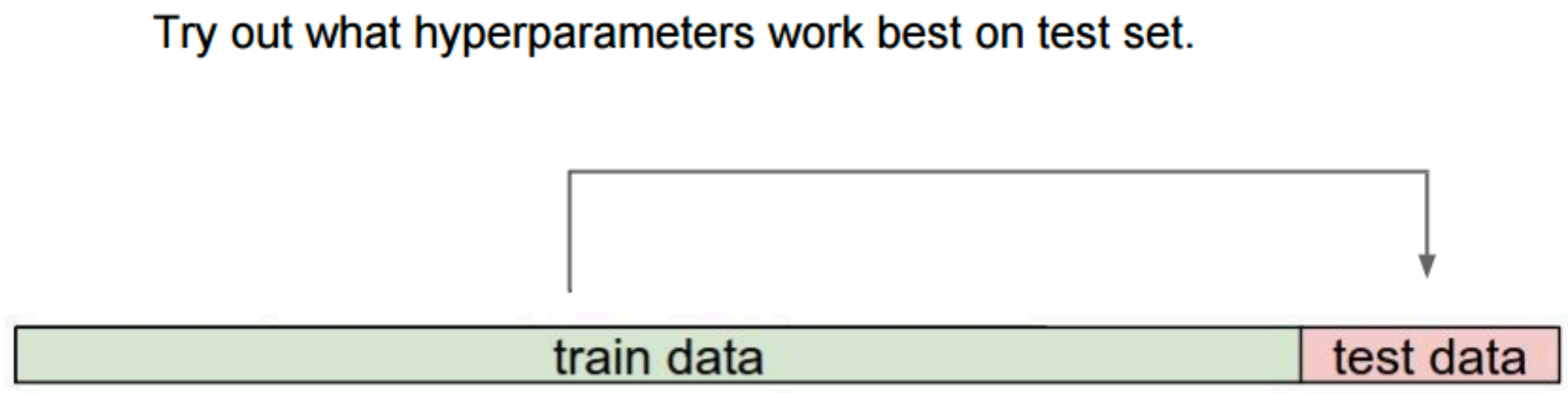
Overfitting: model is too complex and fits irrelevant characteristics (noise) in the data



Cross-Validation

We cannot reliably estimate the error rate of the classifier using the training set

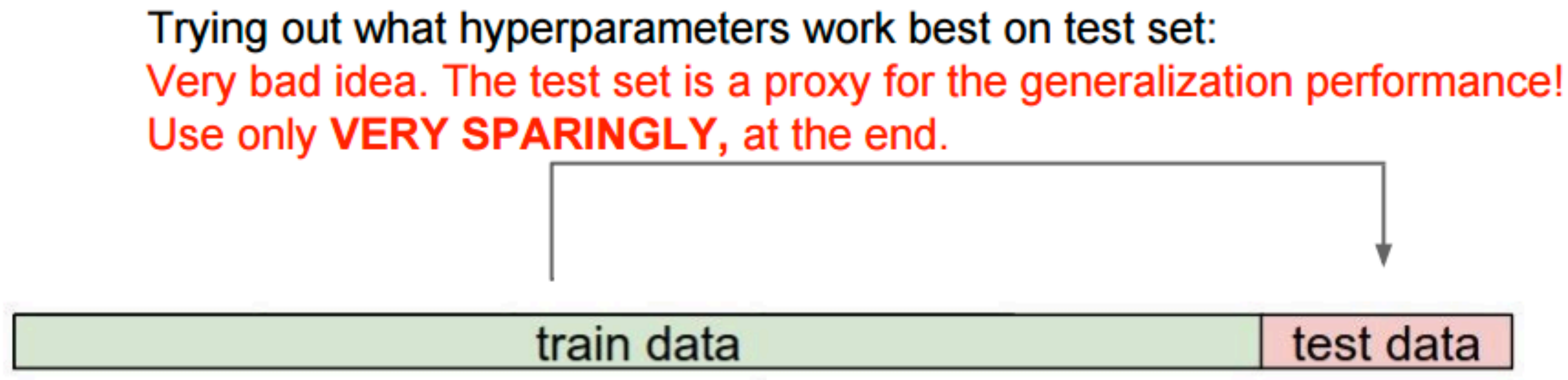
An alternative is to split some training data to form a **validation** set, then train the classifier on the rest of the data and evaluate on the validation set



Cross-Validation

We cannot reliably estimate the error rate of the classifier using the training set

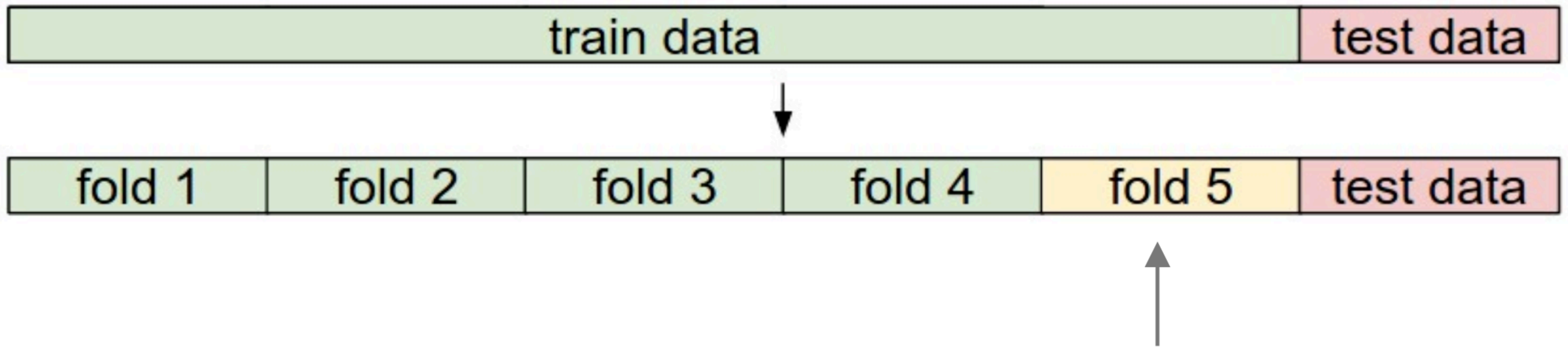
An alternative is to split some training data to form a **validation** set, then train the classifier on the rest of the data and evaluate on the validation set



Cross-Validation

We cannot reliably estimate the error rate of the classifier using the training set

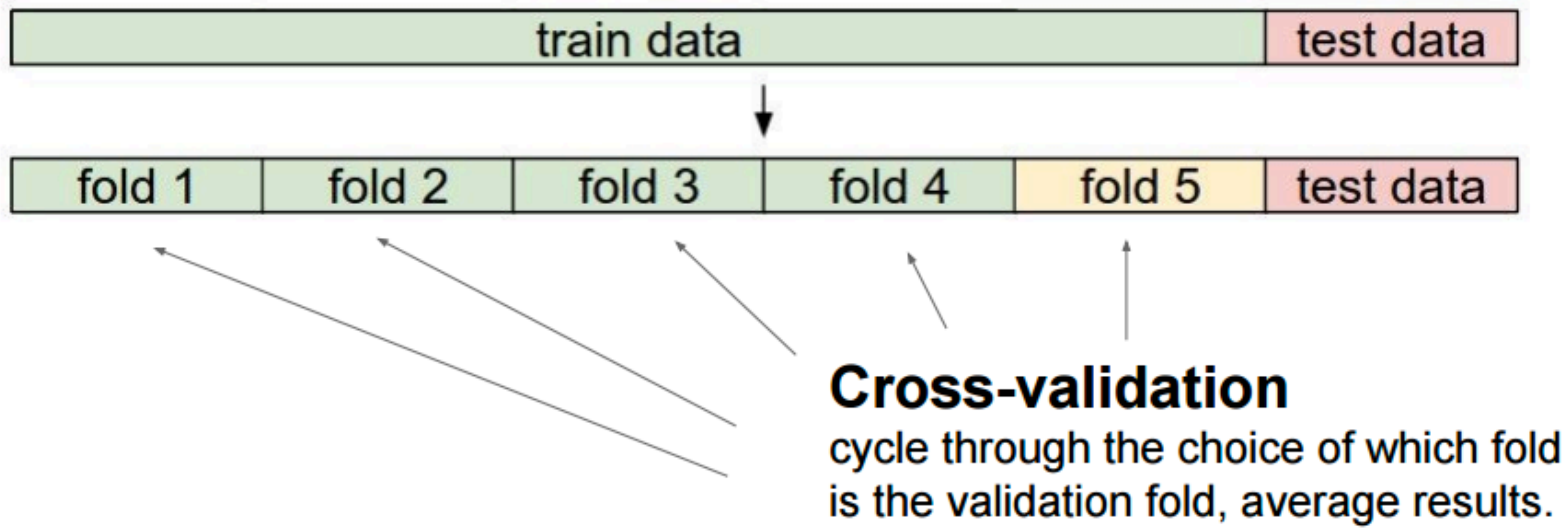
An alternative is to split some training data to form a **validation** set, then train the classifier on the rest of the data and evaluate on the validation set



Validation data
use to tune hyperparameters
evaluate on test set ONCE at the end

Cross-Validation

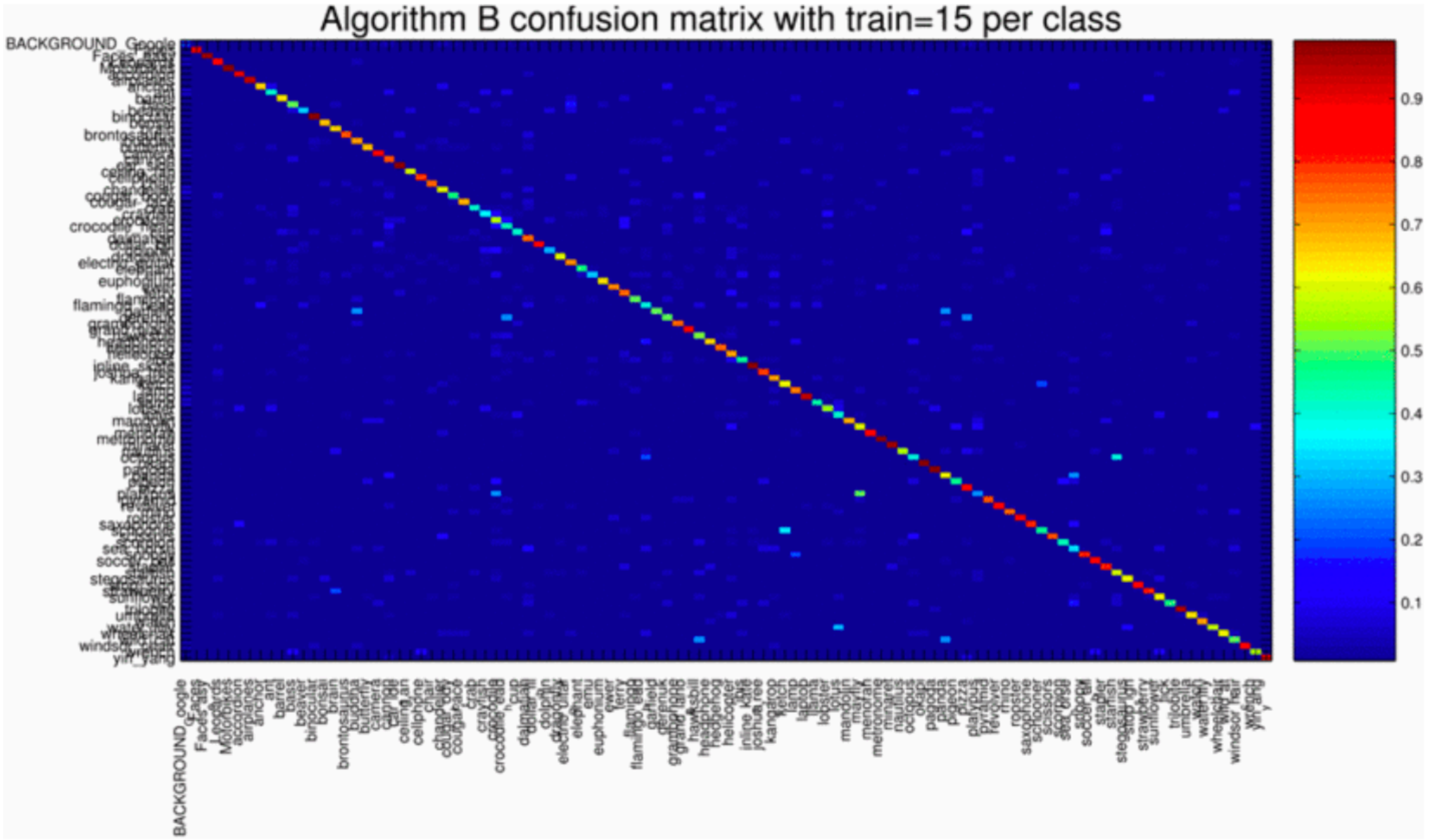
Cross-validation involves performing multiple splits and averaging the error over all splits



Confusion Matrix

When evaluating a multi-class classifier, it may be useful to know how often certain classes are often misclassified as others.

A **confusion matrix** is a table whose (i,j)th entry is the frequency (or proportion) an item of true class i was labelled as j by the classifier.



Forsyth & Ponce (2nd ed.) Figure 15.3. Original credit: H. Zhang et al., 2006.

Classifier Strategies

Classification strategies fall under two broad types: **parametric** and **non-parametric**.

Classifier Strategies

Classification strategies fall under two broad types: **parametric** and **non-parametric**.

Parametric classifiers are **model driven**. The parameters of the model are learned from training examples. New data points are classified by the learned model.

— fast, compact

— flexibility and accuracy depend on model assumptions

Classifier Strategies

Classification strategies fall under two broad types: **parametric** and **non-parametric**.

Parametric classifiers are **model driven**. The parameters of the model are learned from training examples. New data points are classified by the learned model.
— fast, compact
— flexibility and accuracy depend on model assumptions

Non-parametric classifiers are **data driven**. New data points are classified by comparing to the training examples directly. "The data is the model".
— slow
— highly flexible decision boundaries

Nearest Neighbor Classifier

Given a new data point, assign the label of nearest training example in feature space.

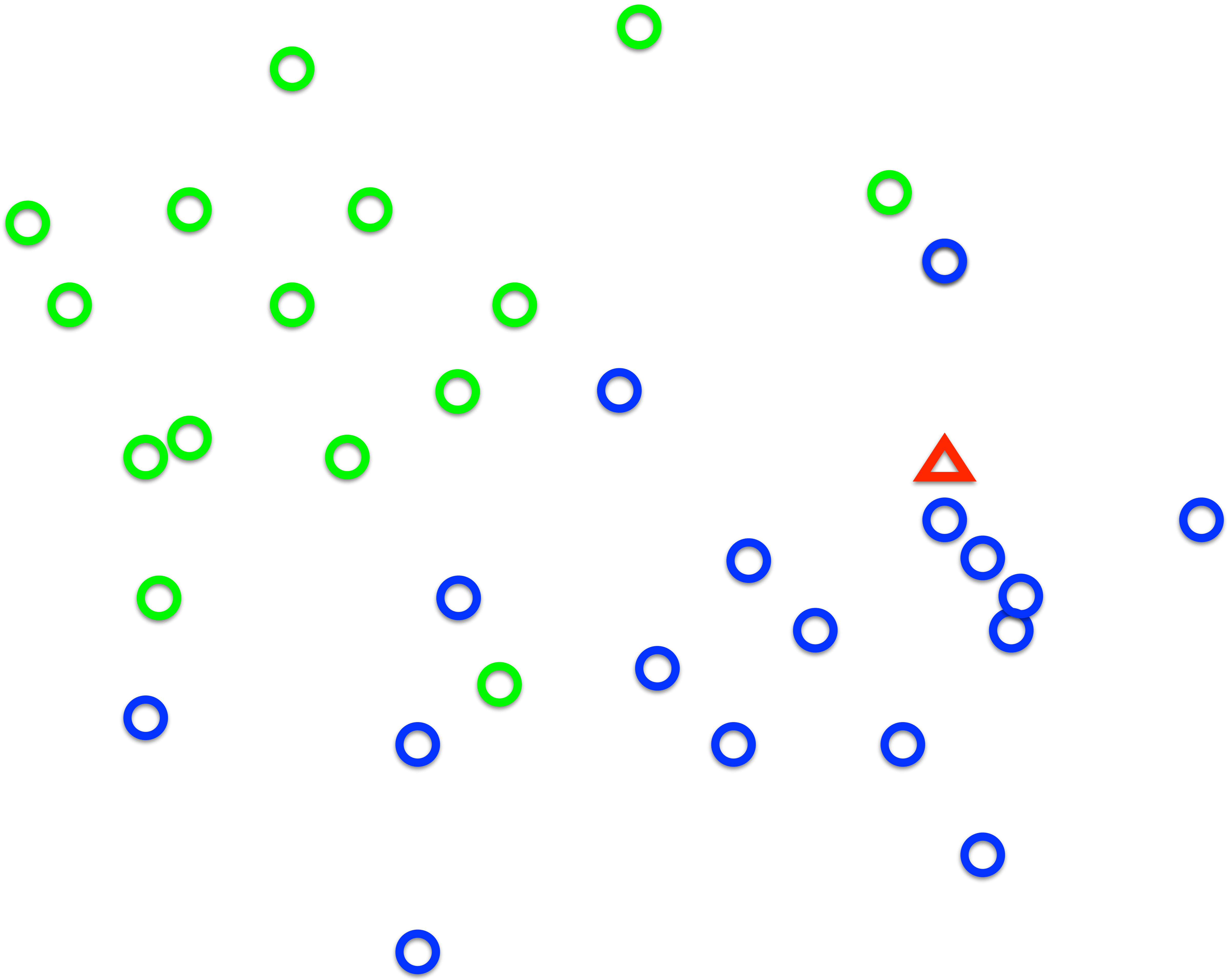


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Nearest Neighbor Classifier

Given a new data point, assign the label of nearest training example in feature space.

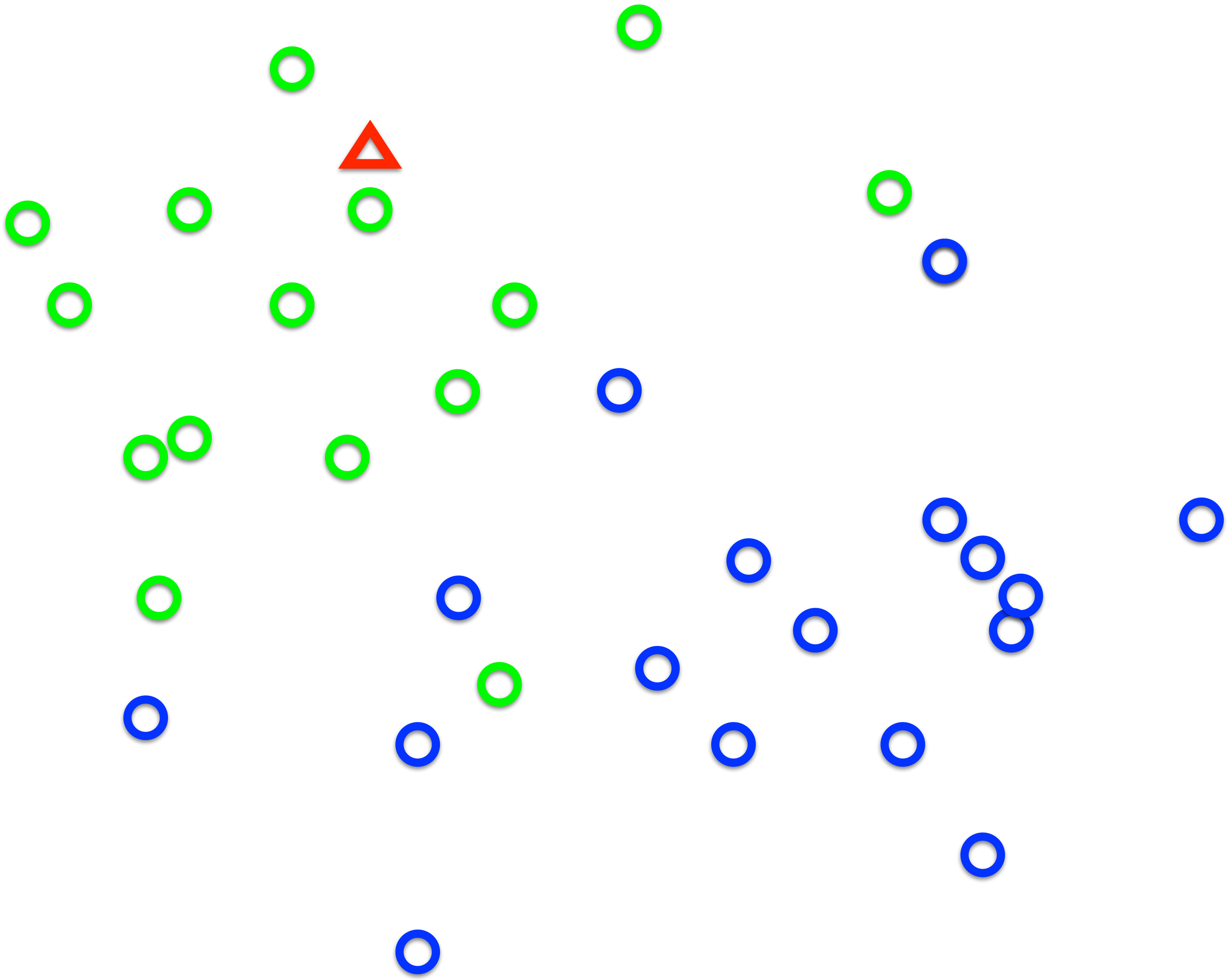


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

k-**Nearest Neighbor** (kNN) Classifier

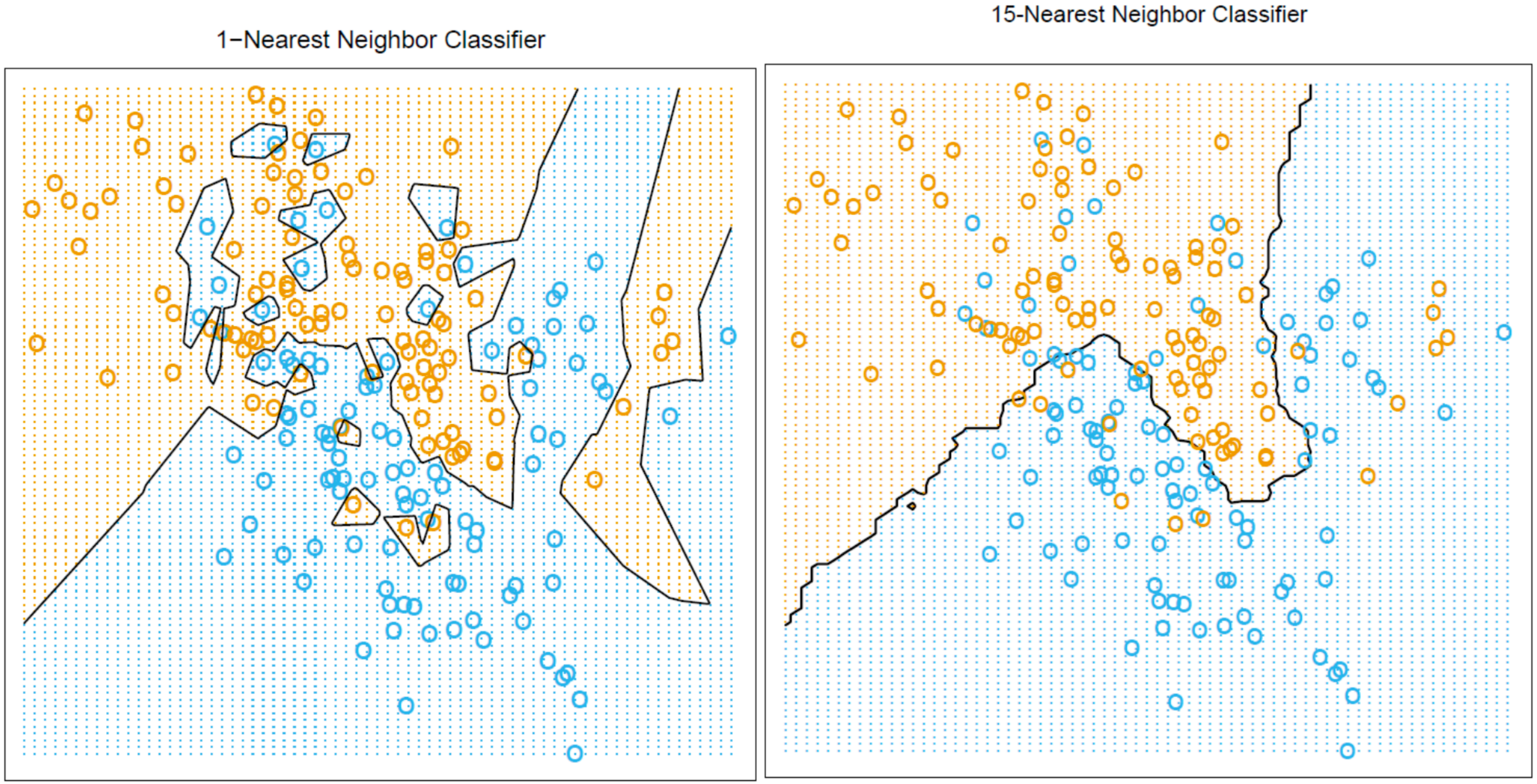
We can gain some robustness to noise by voting over **multiple** neighbours.

Given a **new** data point, find the k nearest training examples. Assign the label by **majority vote**.

Simple method that works well if the **distance measure** correctly weights the various dimensions

For **large data sets**, as k increases kNN approaches optimality in terms of minimizing probability of error

k-**Nearest Neighbor** (kNN) Classifier



kNN decision boundaries respond to local clusters where one class dominates

Figure credit: Hastie, Tibshirani & Friedman (2nd ed.)

Classifier Strategies

Classification strategies fall under two broad types: **parametric** and **non-parametric**.

Parametric classifiers are **model driven**. The parameters of the model are learned from training examples. New data points are classified by the learned model.
— fast, compact
— flexibility and accuracy depend on model assumptions

Non-parametric classifiers are **data driven**. New data points are classified by comparing to the training examples directly. "The data is the model".
— slow
— highly flexible decision boundaries

Support Vector Machines (SVM)

Idea: Try to obtain the decision boundary directly

The decision boundary is parameterized as a **separating hyperplane** in feature space.
— e.g. a separating line in 2D

We choose the hyperplane that is as far as possible from each class - that maximizes the distance to the closest point from either class.

Linear Classifier

Defines a score function:

$$f(\mathbf{x}_i, \mathbf{W}, \mathbf{b}) = \underbrace{\mathbf{W}}_{\text{weights (parameters)}} \overbrace{\mathbf{x}_i}^{\text{image features}} + \underbrace{\mathbf{b}}_{\text{bias vector}}$$

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Classifier

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

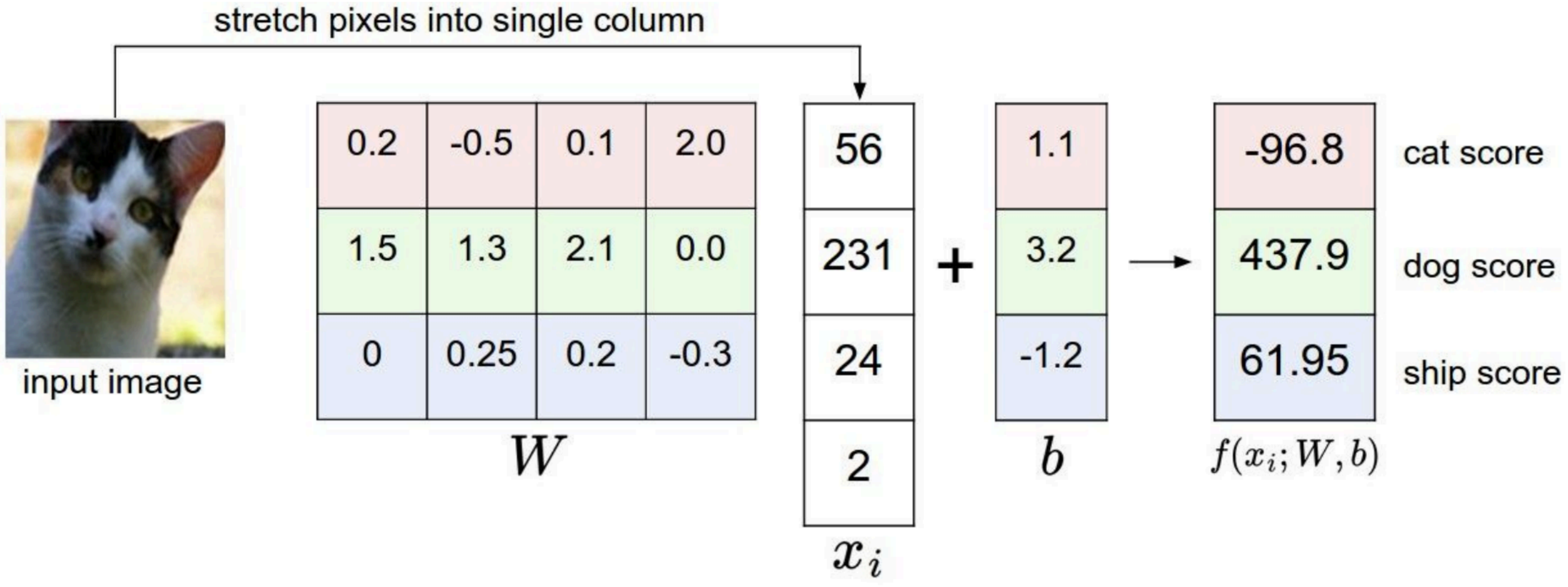


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

Learn the decision boundary

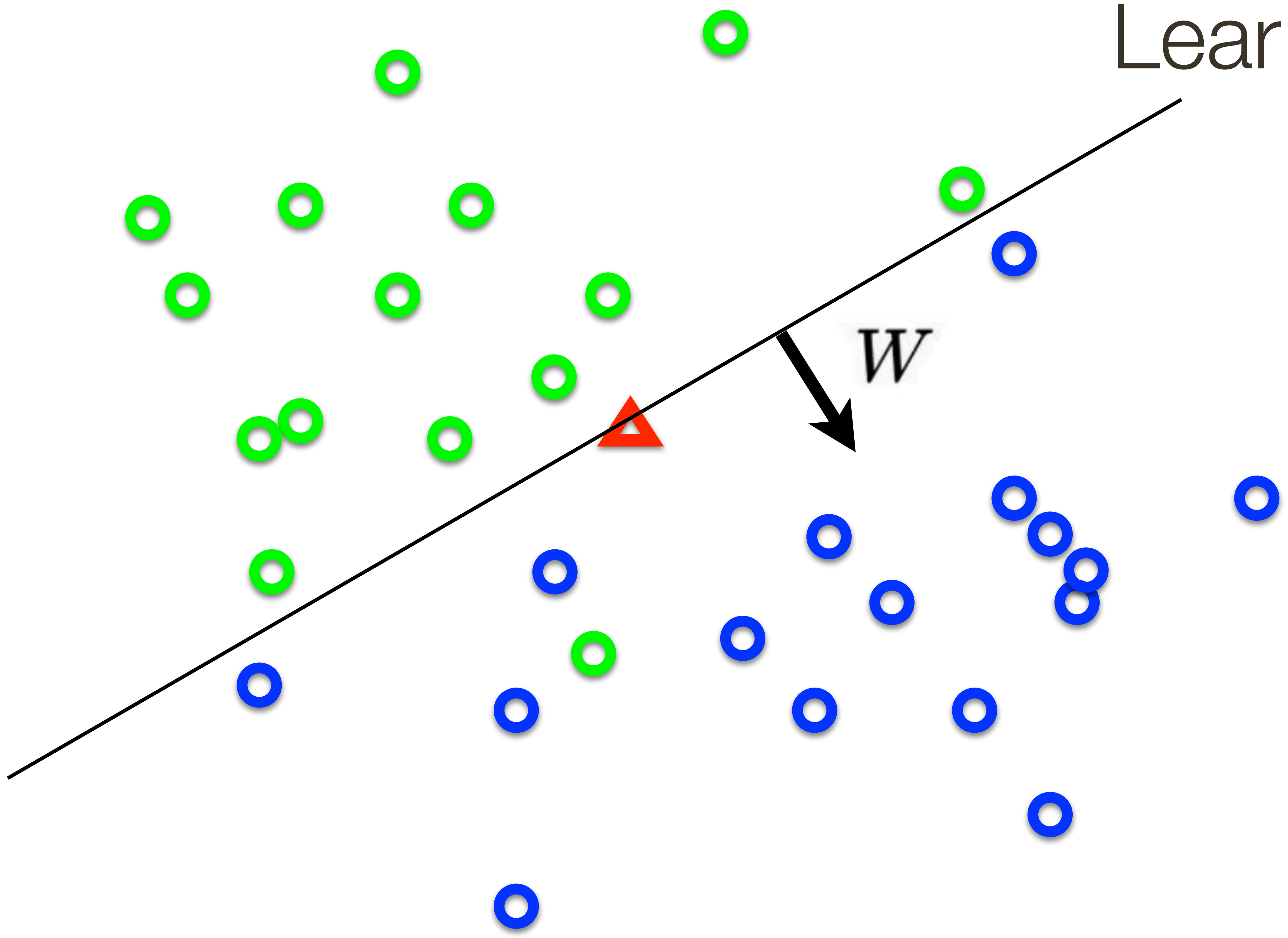


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

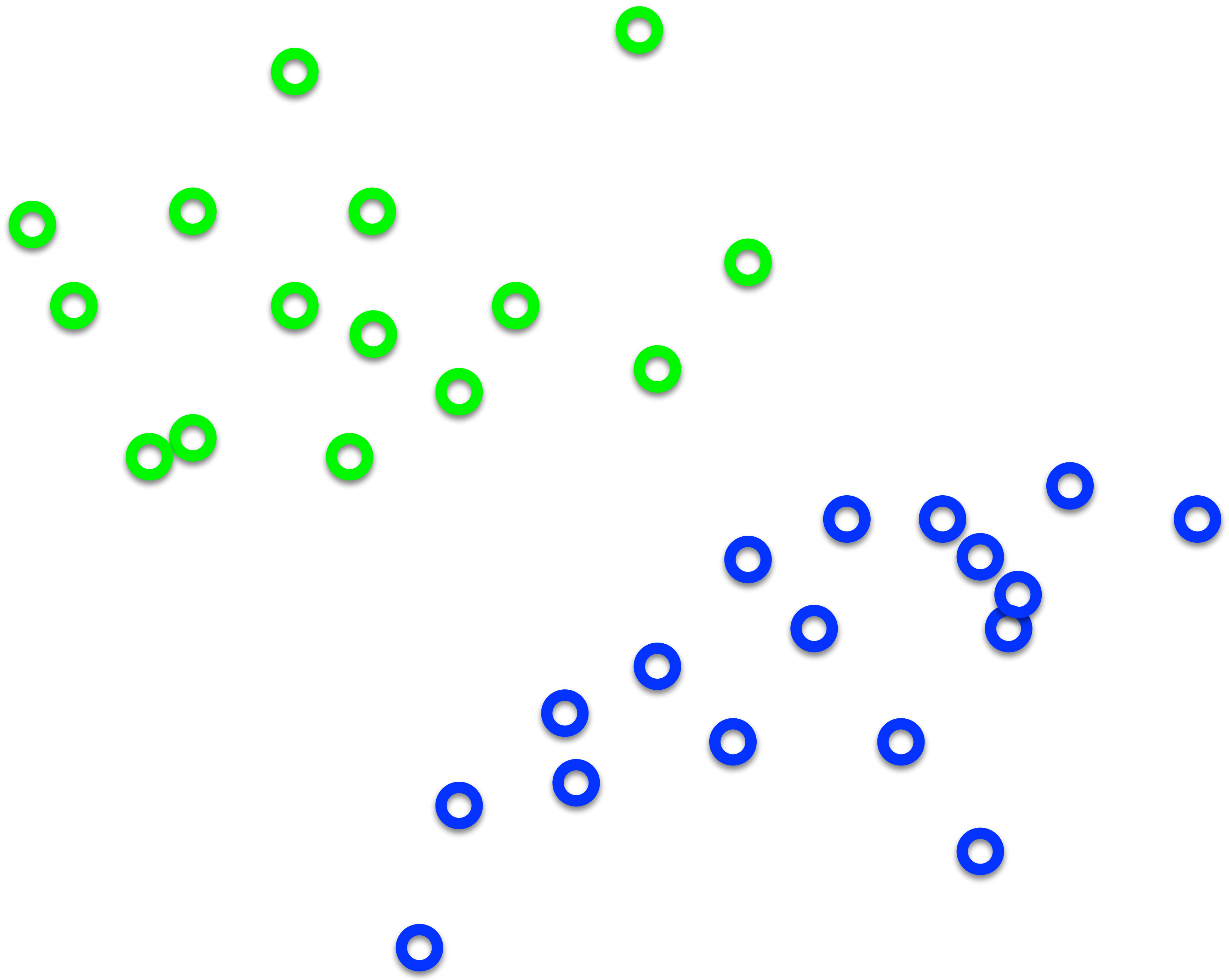


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

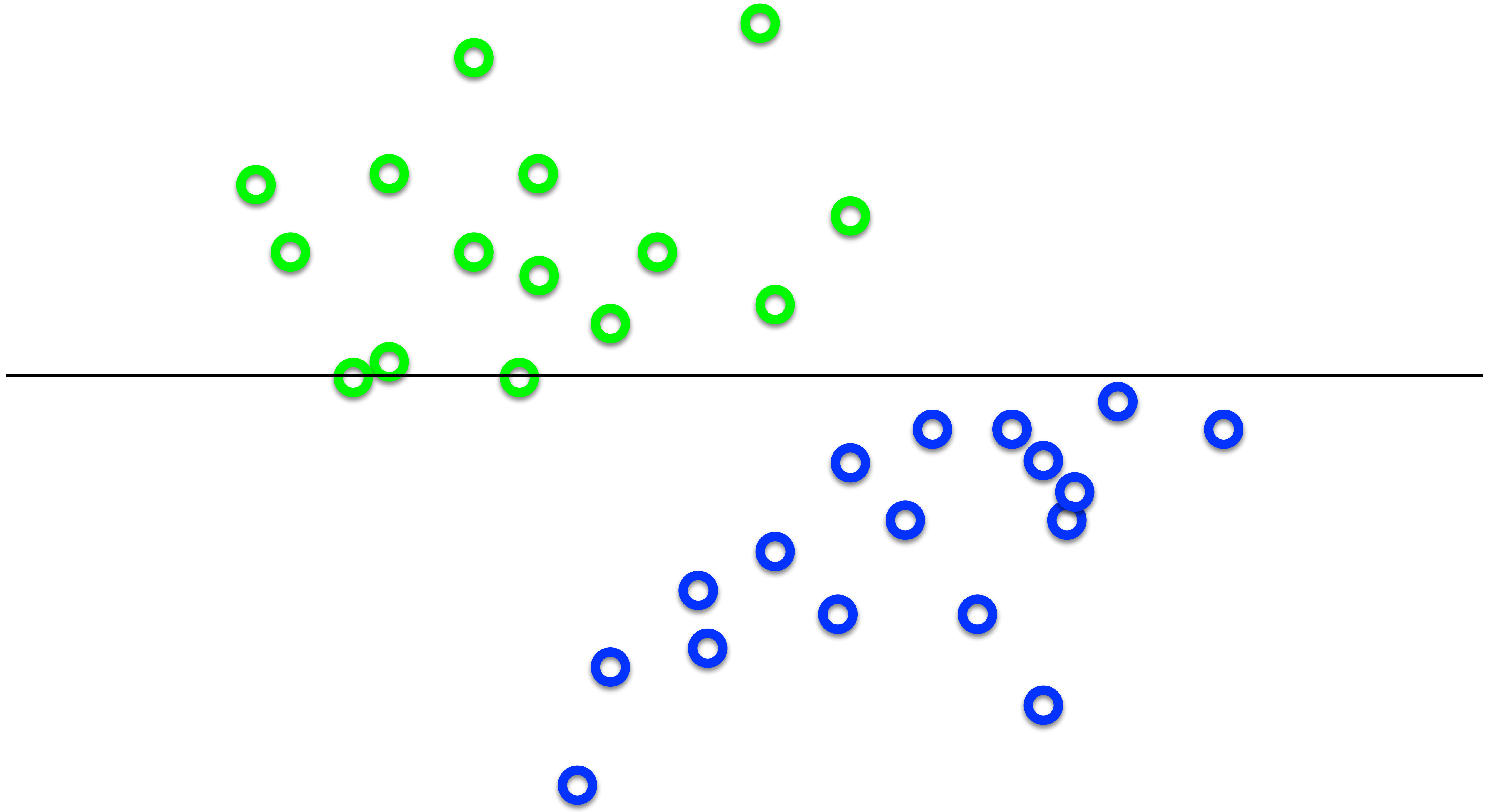


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

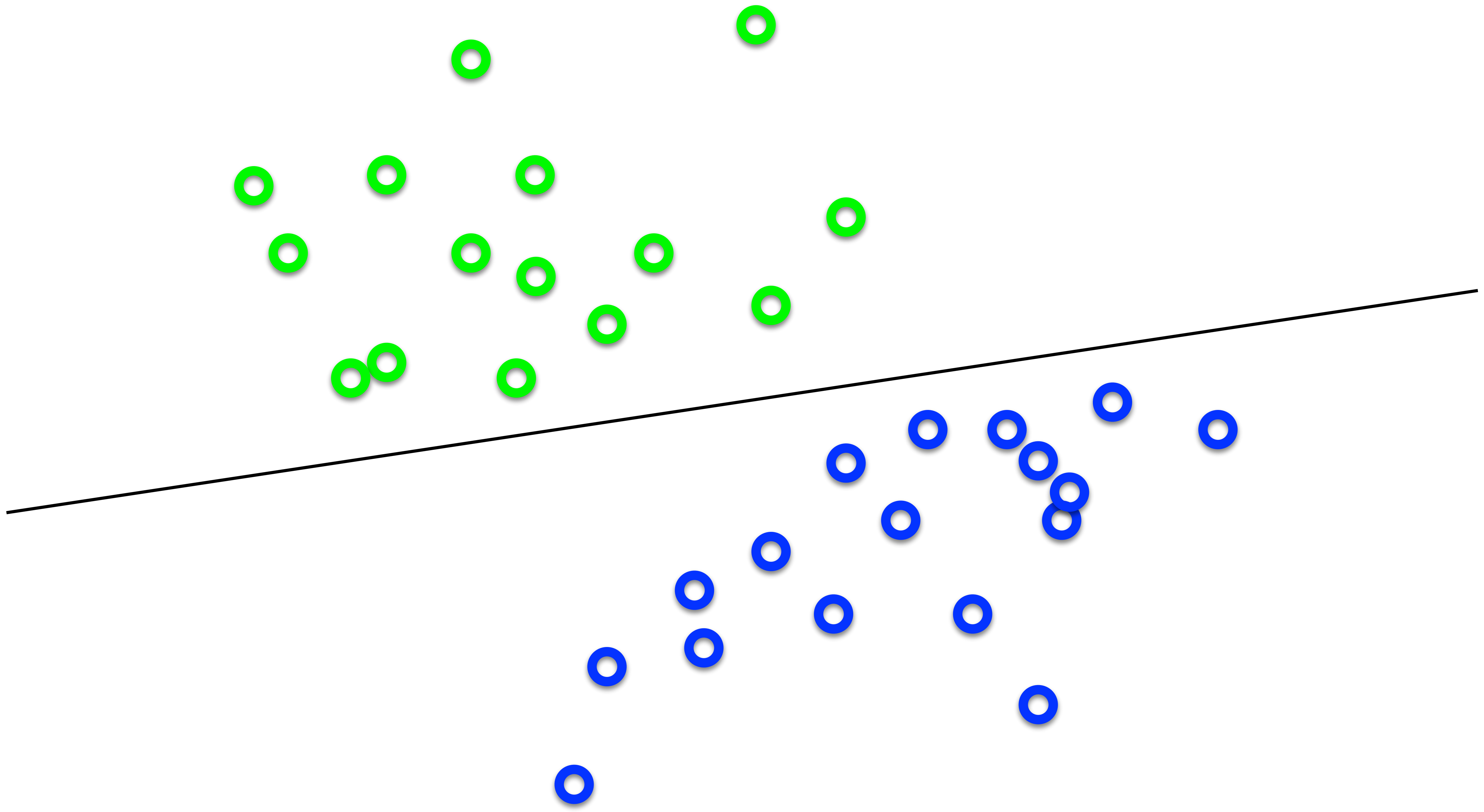


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

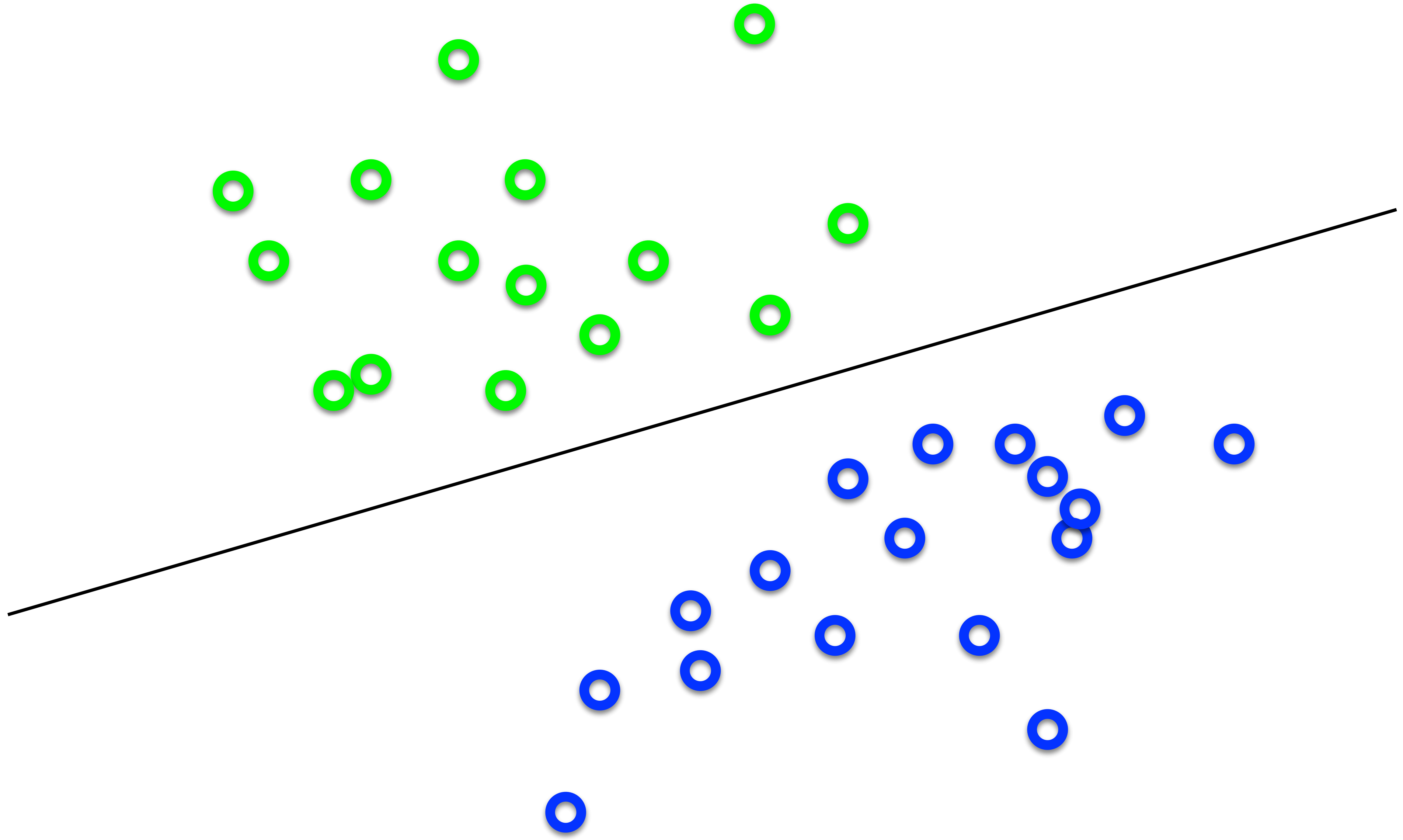


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

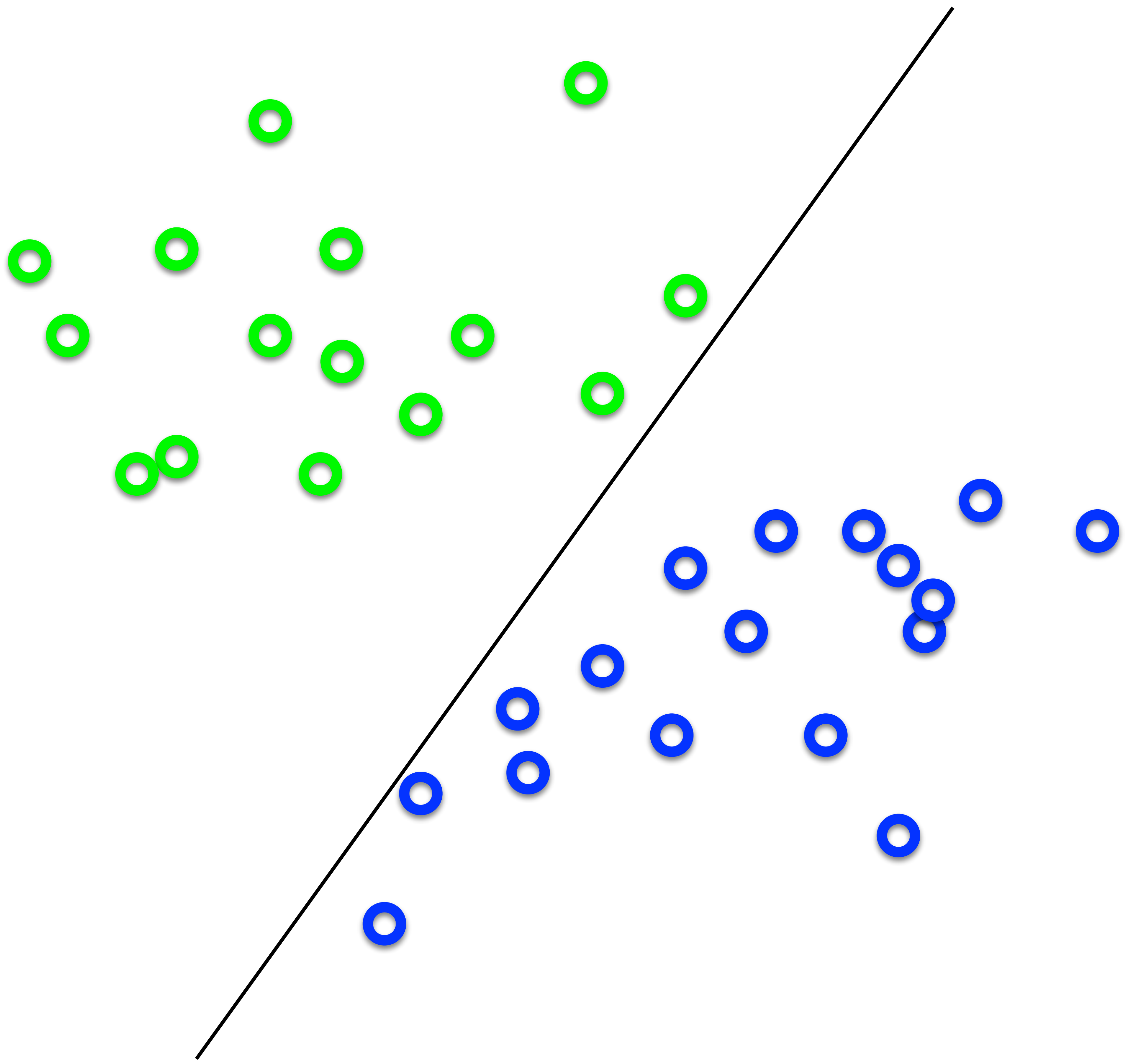
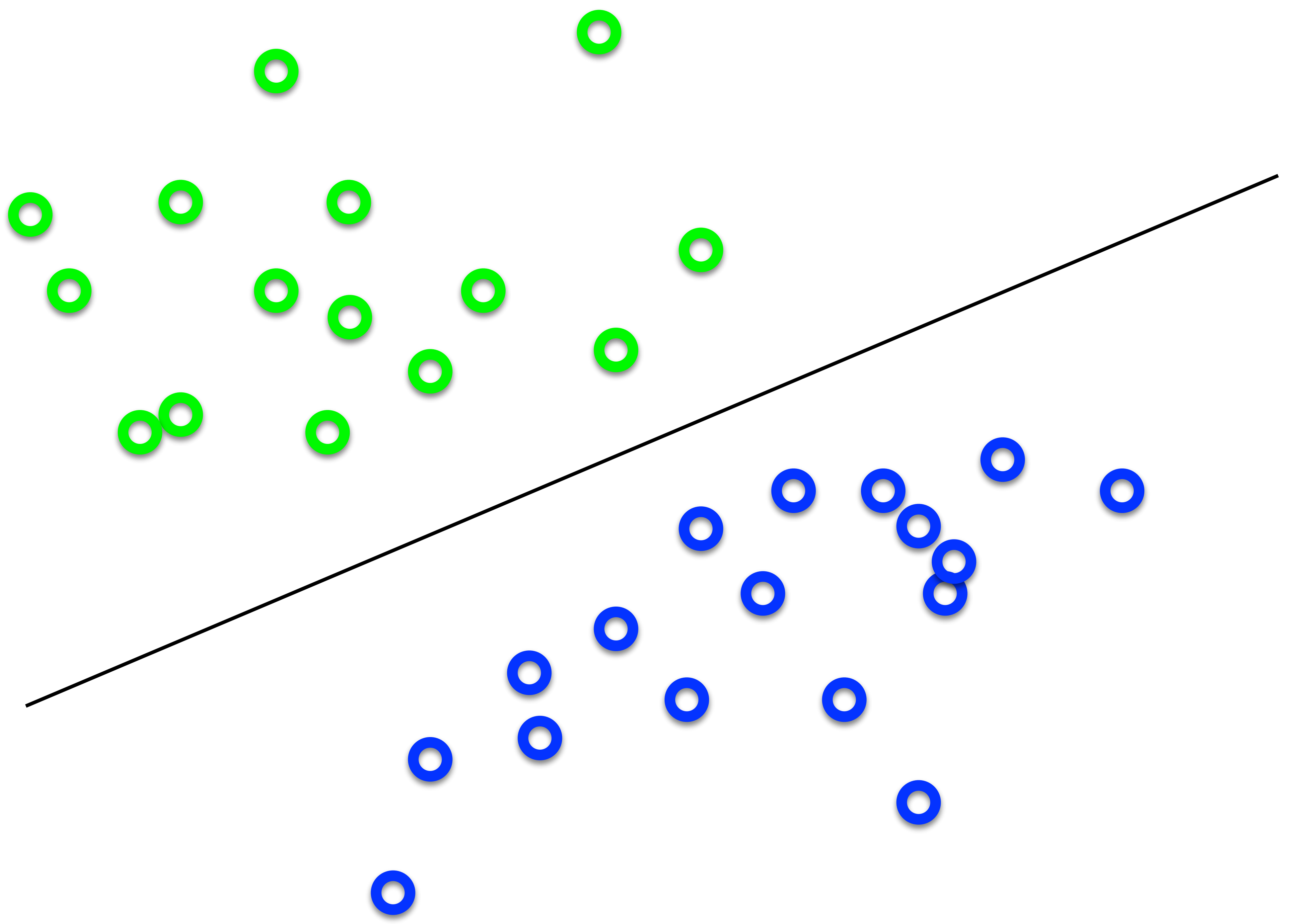


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

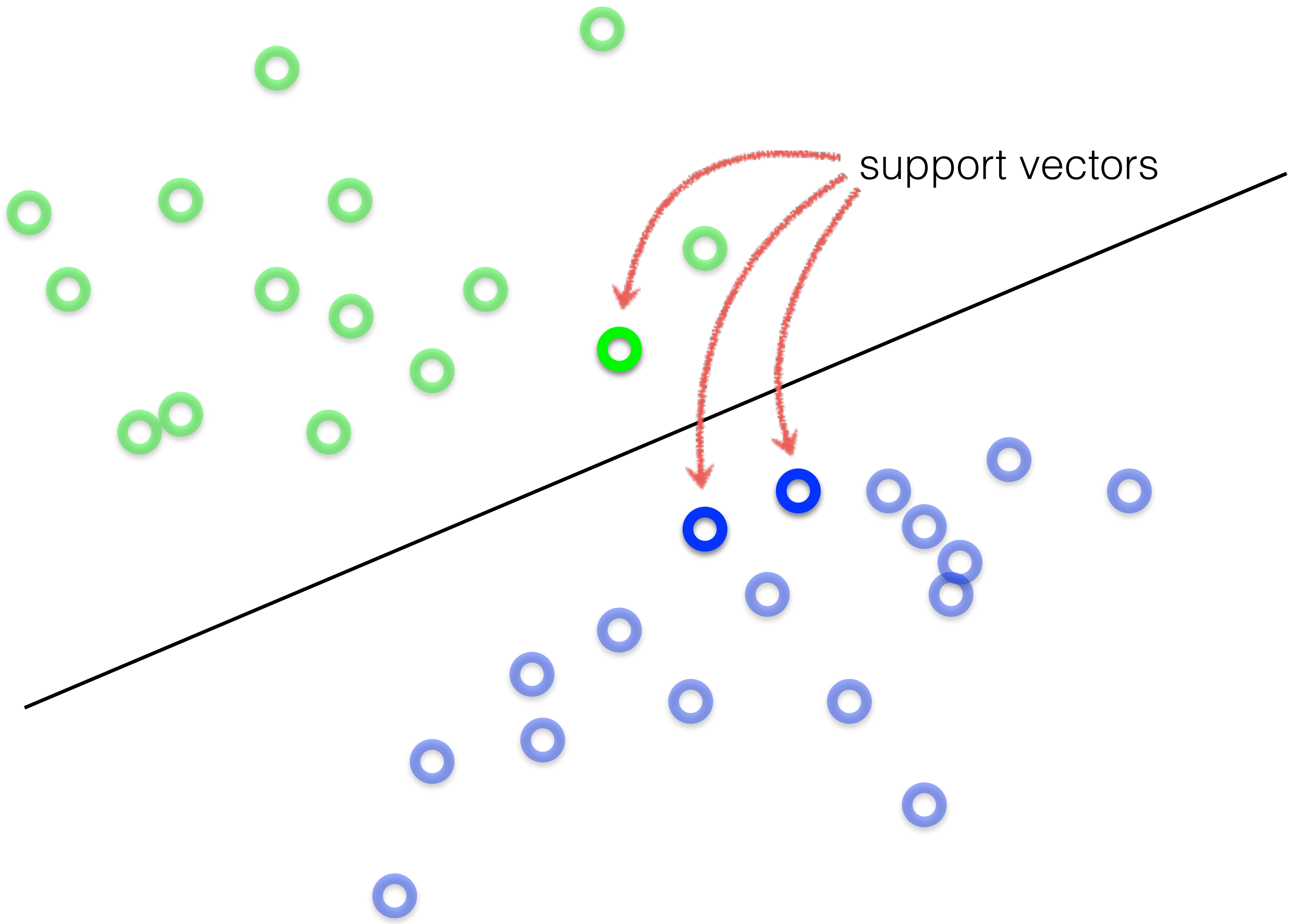


Intuitively, the line that is the farthest from all interior points

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

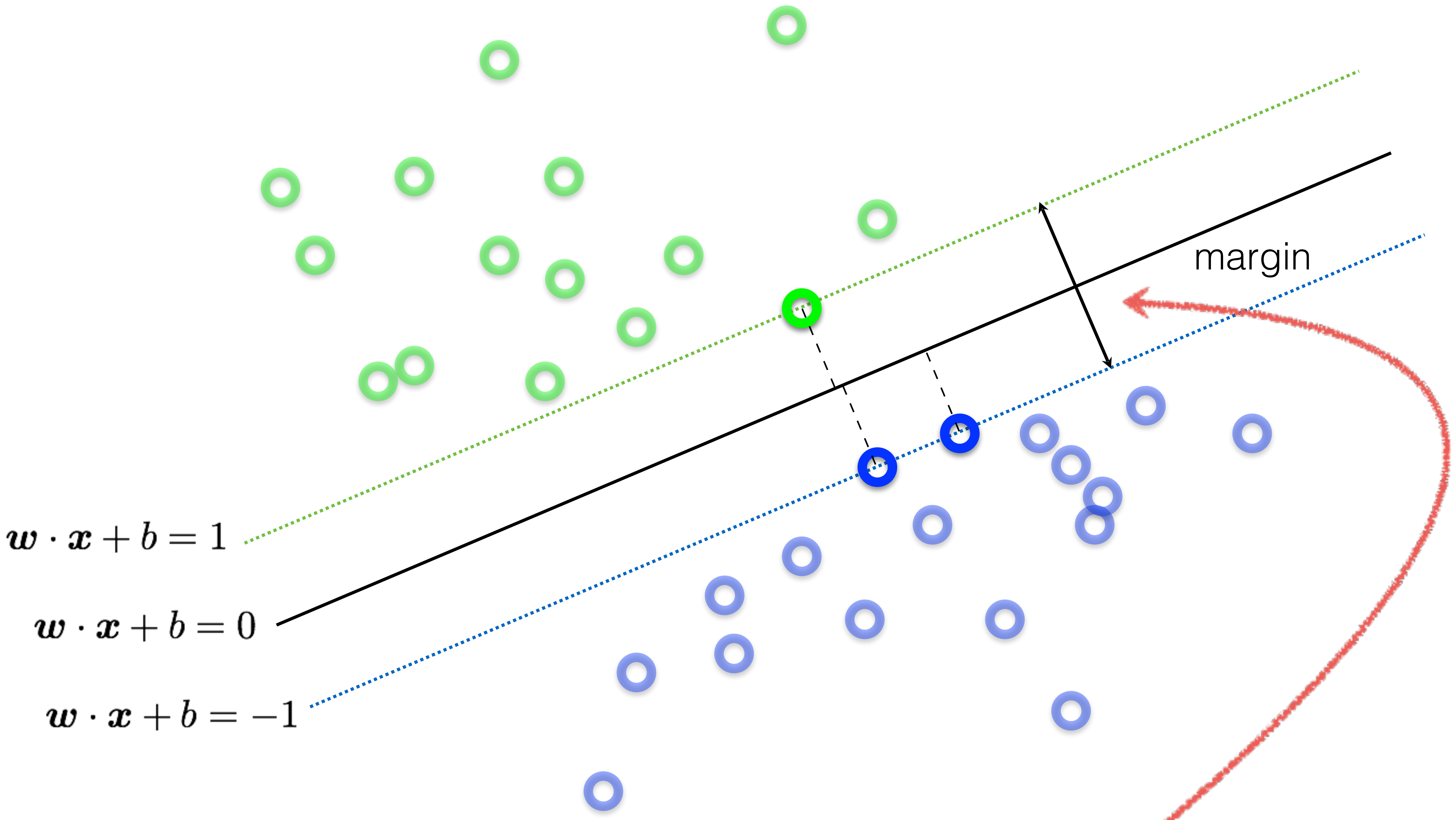


Want a hyperplane that is far away from 'inner points'

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

Find hyperplane **w** such that …



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

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

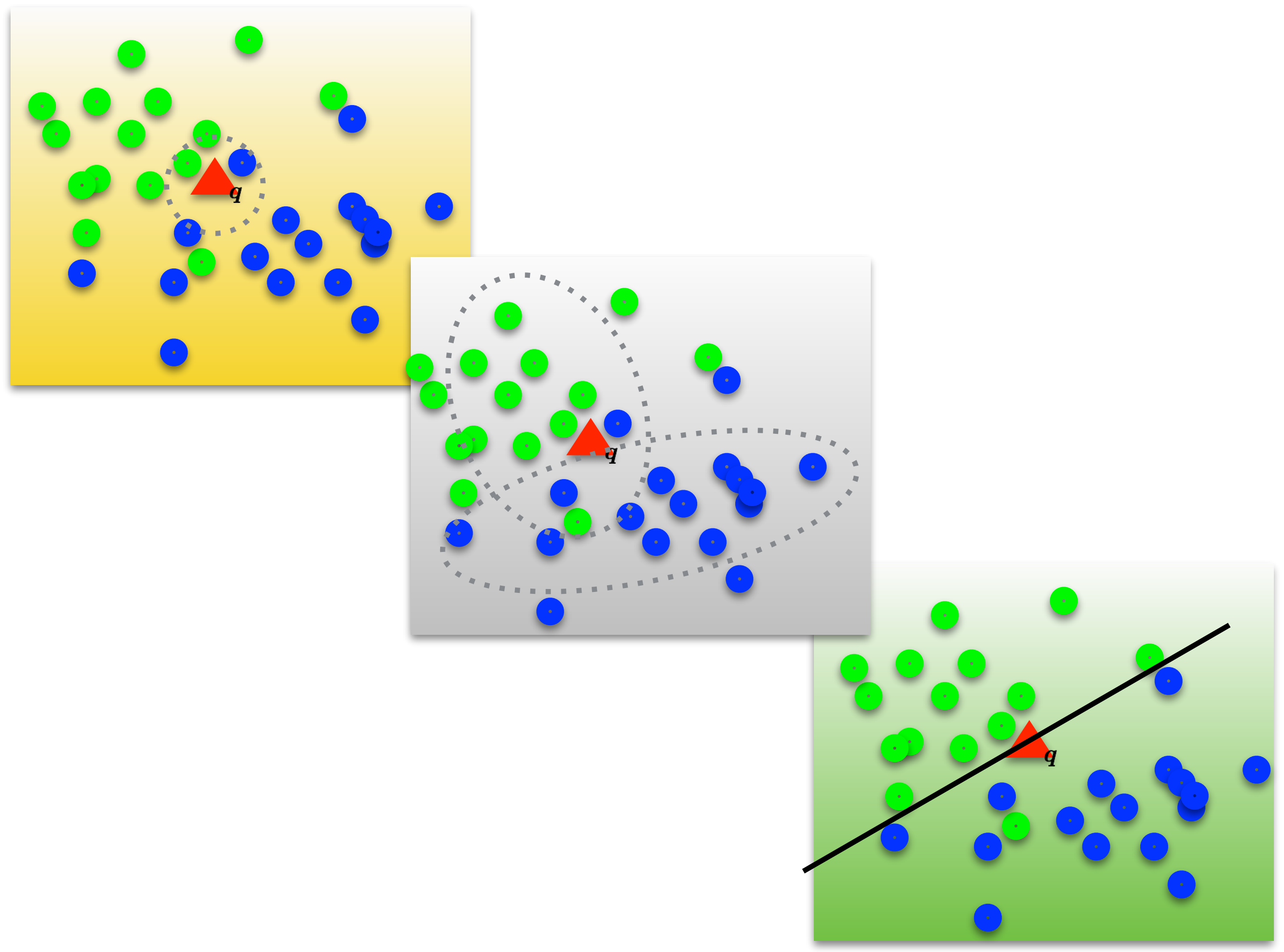
Image Classification

Classification Algorithms

— Bayes' Classifier

— Nearest Neighbor Classifier

— SVM Classifier



Representation of Images

— Image pixels directly

— Bag of Words

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

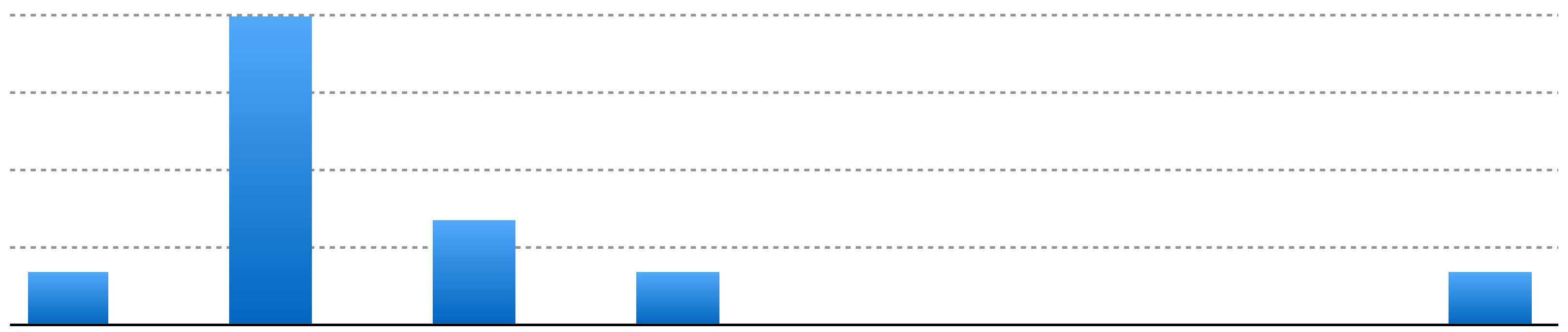
G. Salton. ‘Mathematics and Information Retrieval’ Journal of Documentation, 1979


The Newspa

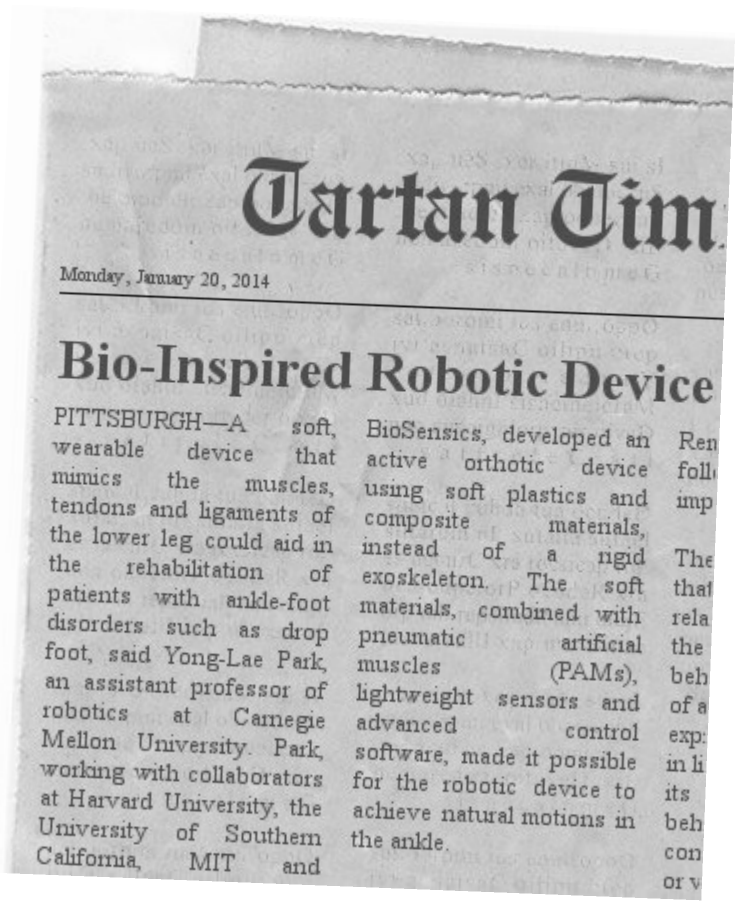
Sunday, December 22, 2013

DARPA Selects Carnegie Me

The Tartan Rescue Team from Carnegie Mellon University's National Robotics Engineering Center ranked third among teams competing in the Defense Advanced Research Projects Agency (DARPA) Robotics Challenge Trials this weekend in Homestead, Fla., and was selected by the agency as one of eight teams eligible for DARPA funding to prepare for next December's finals. The team's four-limbed CMU Highly Intelligent Mobile Platform, or CHIMP, robot scored 18 out of a possible 32 points during the two-day trials. It demonstrated its ability to perform such tasks as removing debris, cutting a hole through a wall and closing a series of valves.



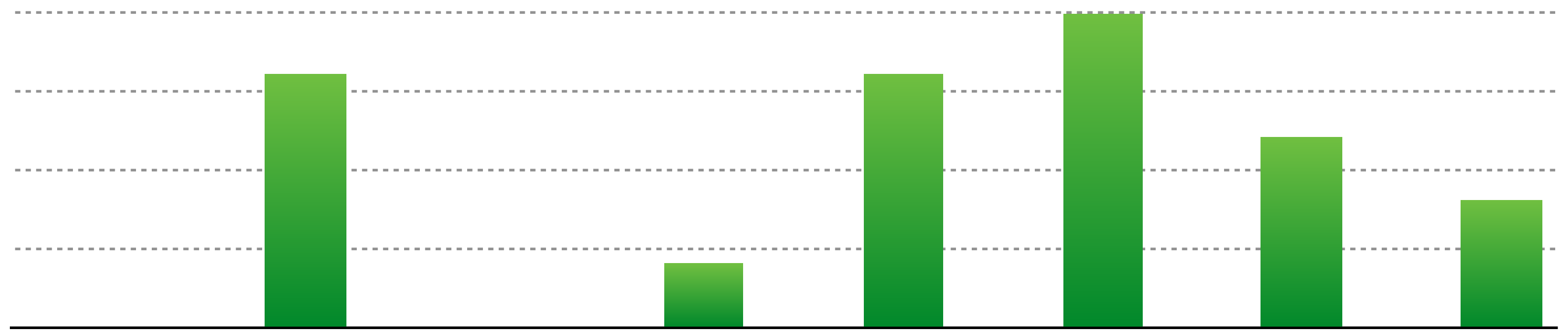
1	6	2	1	0	0	0	1
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor


Tartan Tim

Monday, January 20, 2014

Bio-Inspired Robotic Device

PITTSBURGH—A soft, wearable device that mimics the muscles, tendons and ligaments of the lower leg could aid in the rehabilitation of patients with ankle-foot disorders such as drop foot, said Yong-Lae Park, an assistant professor of robotics at Carnegie Mellon University. Park, working with collaborators at Harvard University, the University of Southern California, MIT and BioSensics, developed an active orthotic device using soft plastics and composite materials, instead of a rigid exoskeleton. The soft materials, combined with pneumatic artificial muscles (PAMs), lightweight sensors and advanced control software, made it possible for the robotic device to achieve natural motions in the ankle.



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)

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

$n(\cdot)$ 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)

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

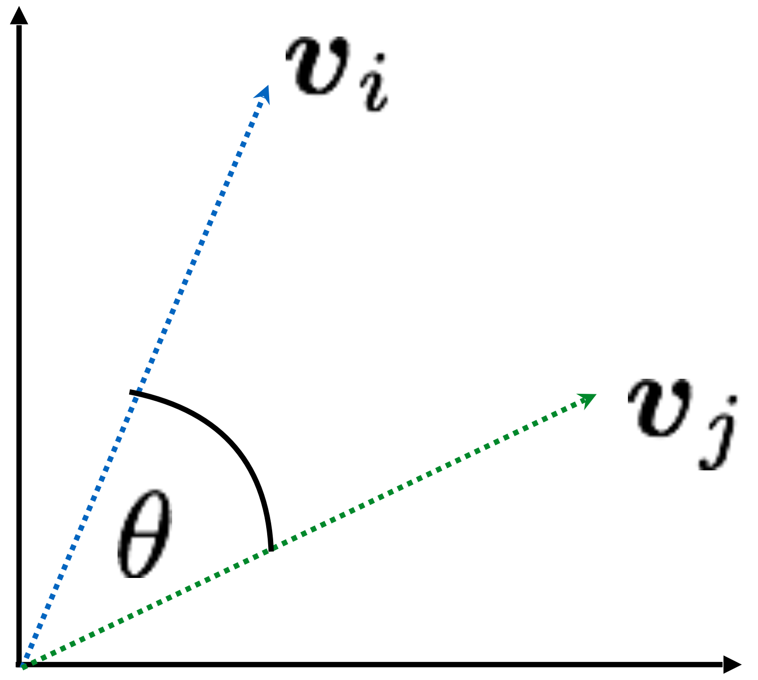
$n(\cdot)$ counts the number of occurrences

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:

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



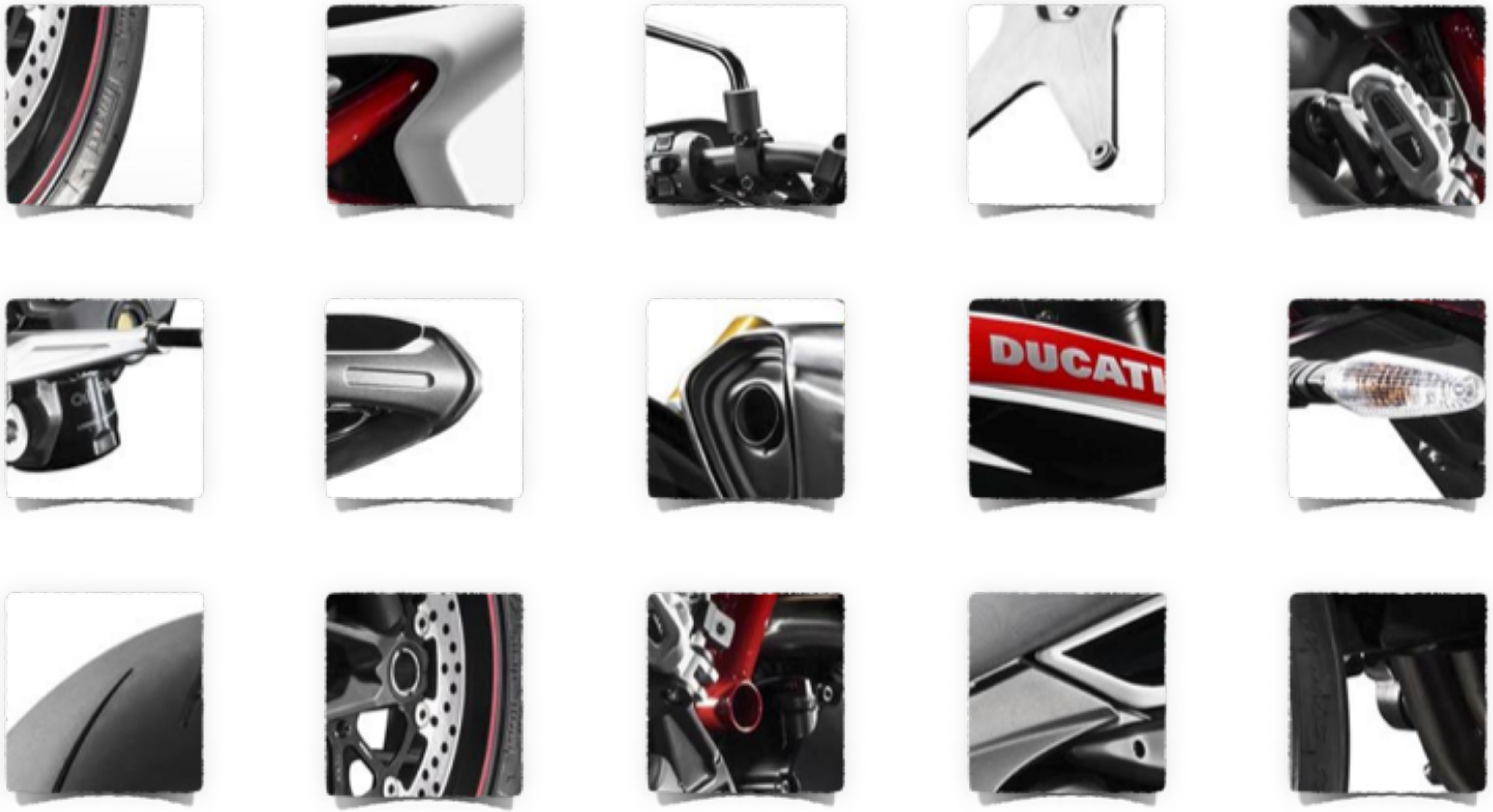
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

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.

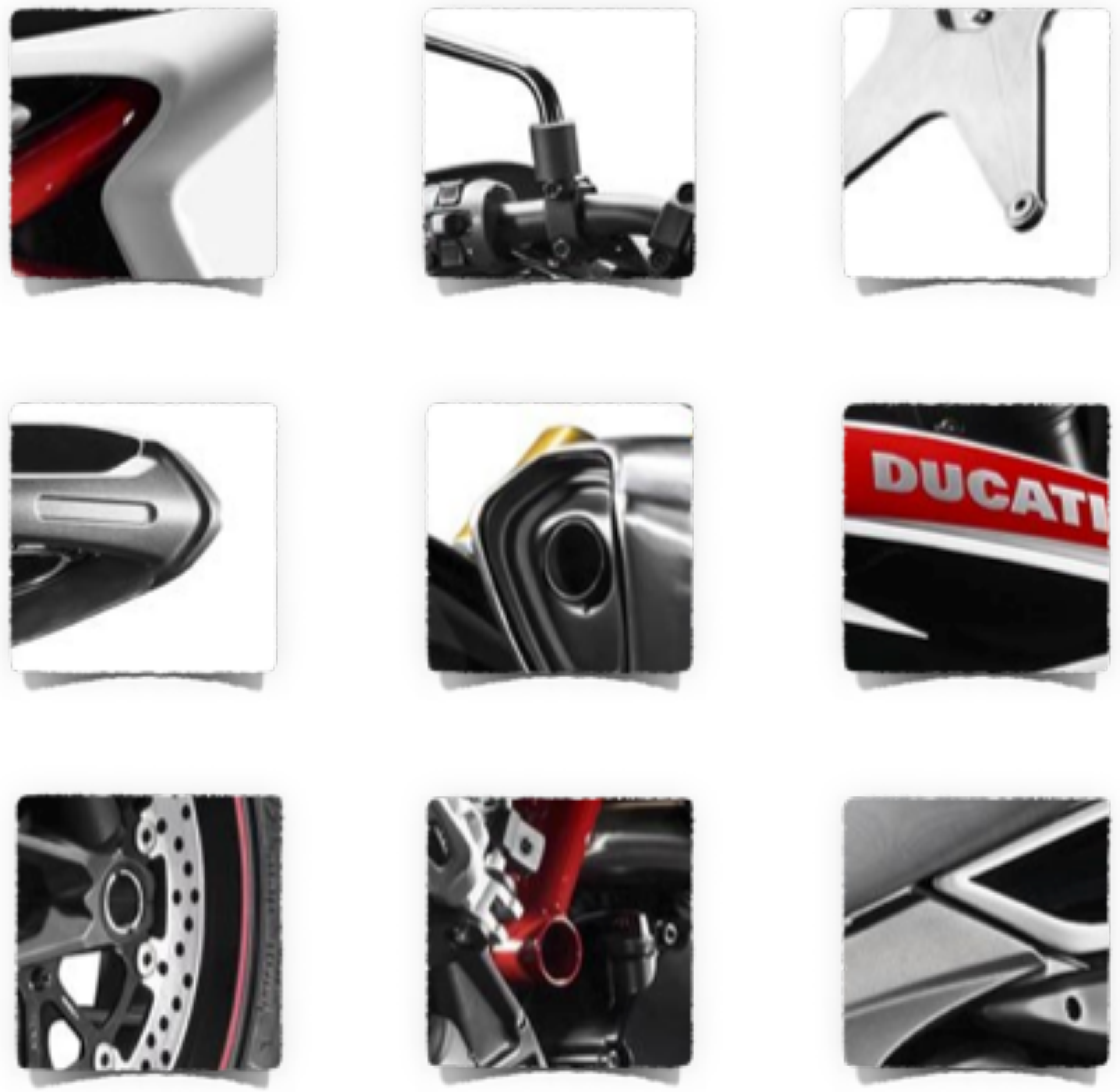
We construct a **vocabulary** or **codebook** of local descriptors, containing representative local descriptors.

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



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Some local feature are very informative



An object as

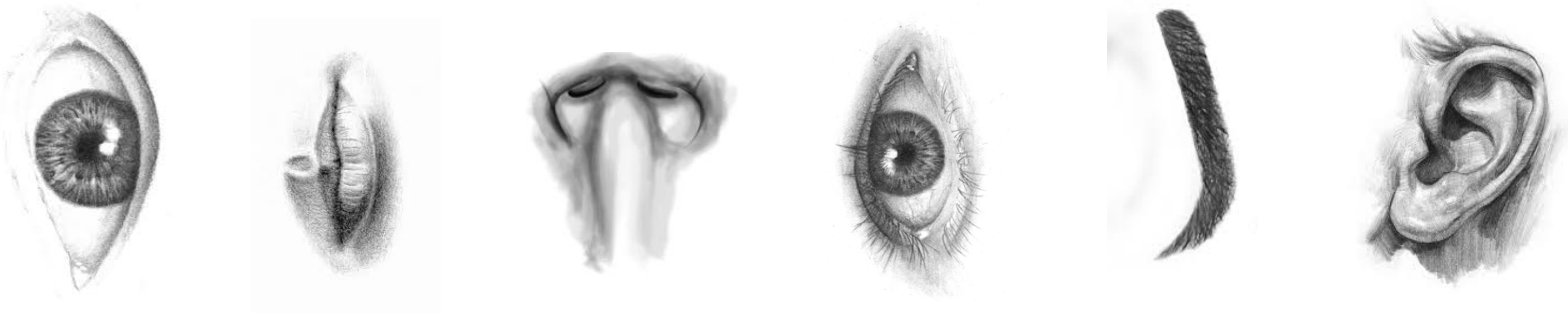
a collection of local features

(bag-of-features)

- deals well with occlusion
- scale invariant
- rotation invariant

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

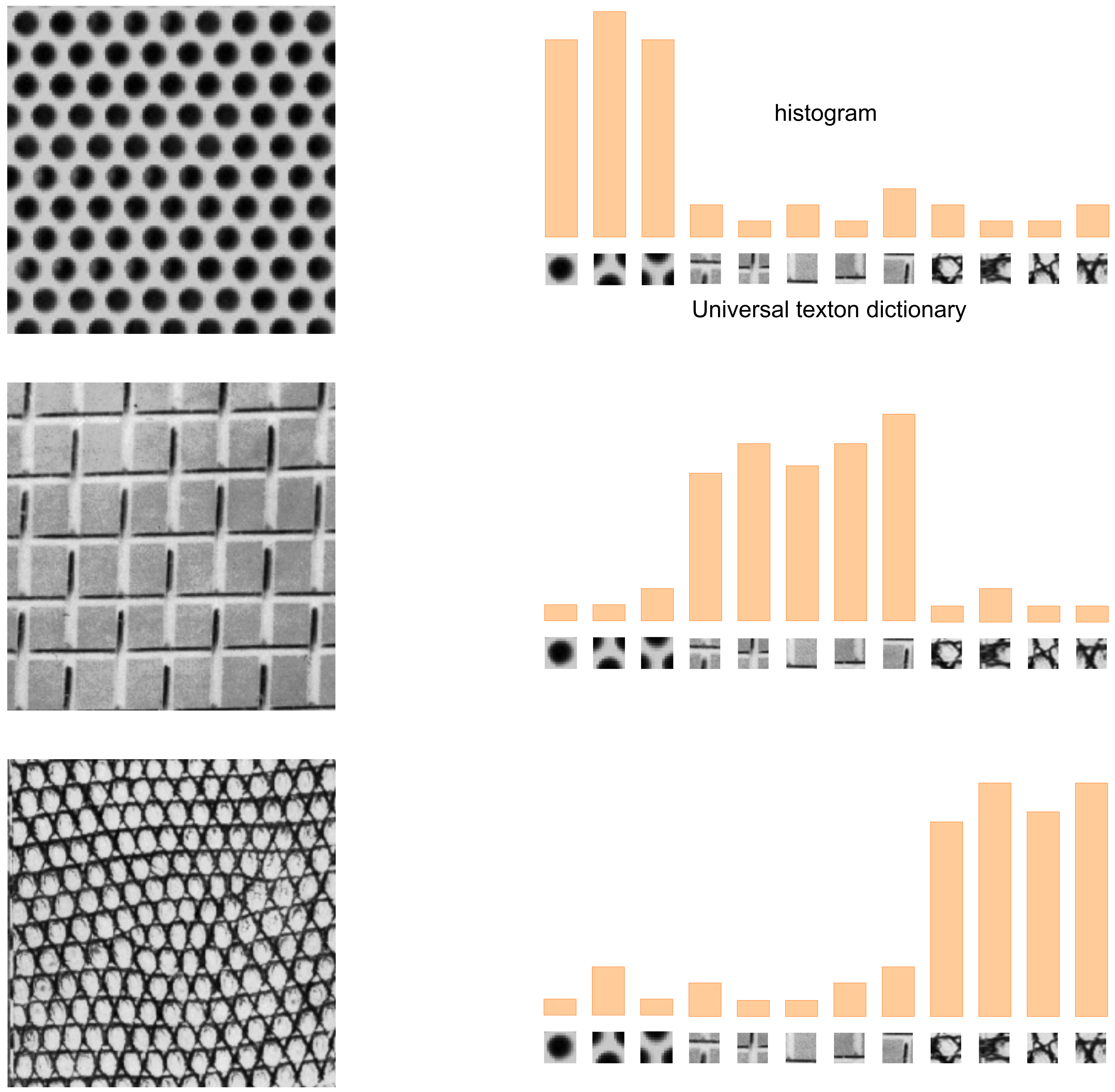
(**not so**) Crazy Assumption



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

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Recall: Texture Representation



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)

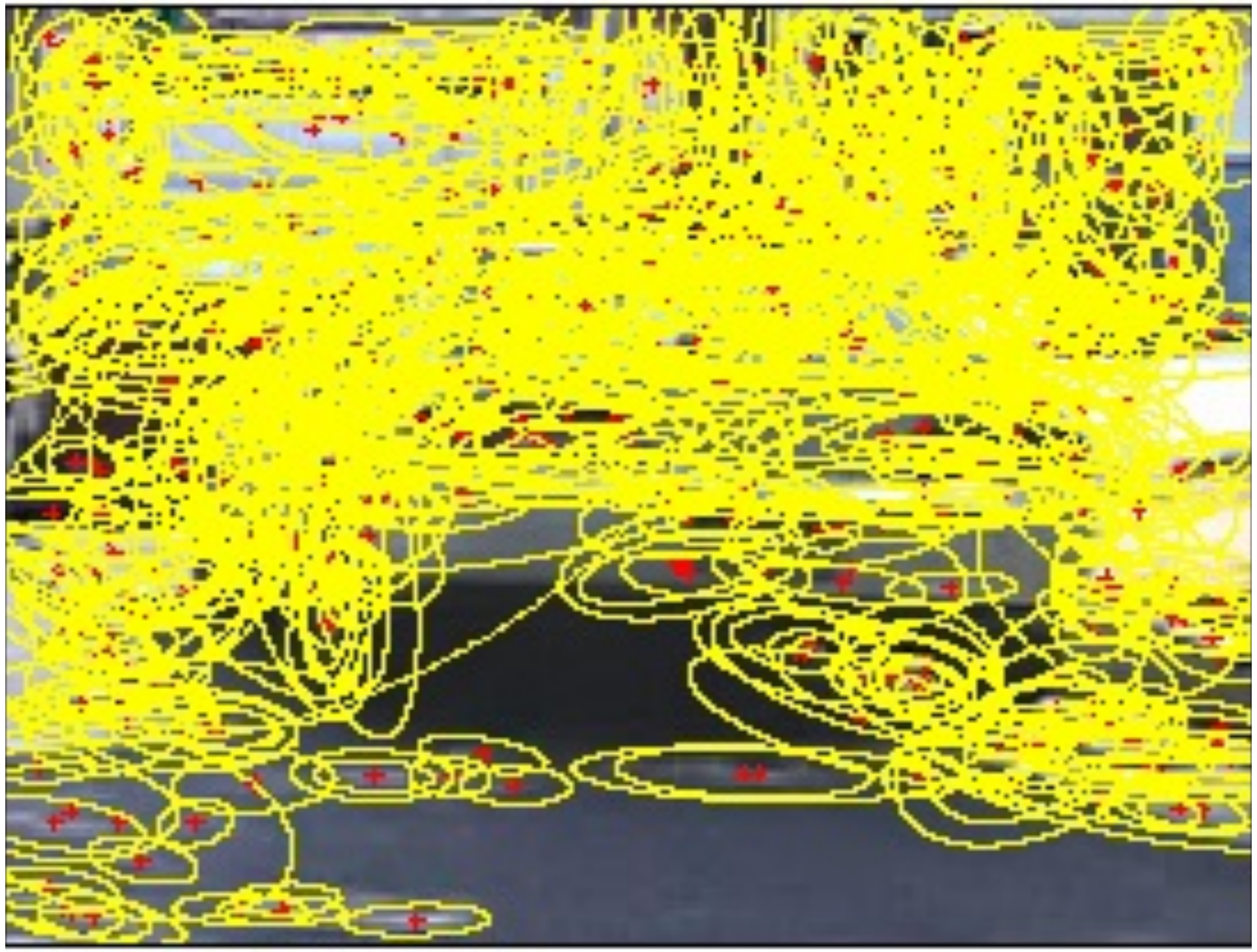
1. **Dictionary Learning**: Learn Visual Words using Clustering

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

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

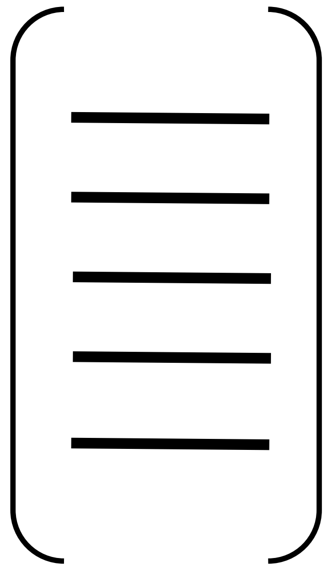
What **Features** Should We Extract?

- 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)



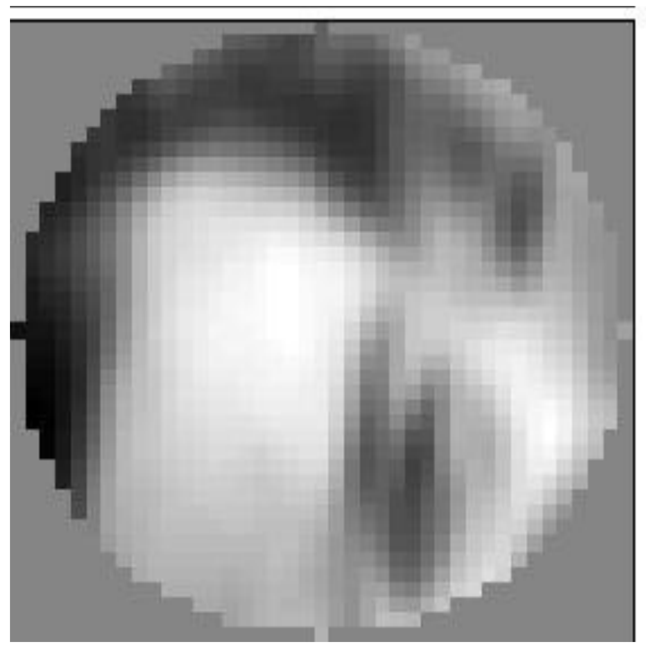
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Extracting **SIFT** Patches

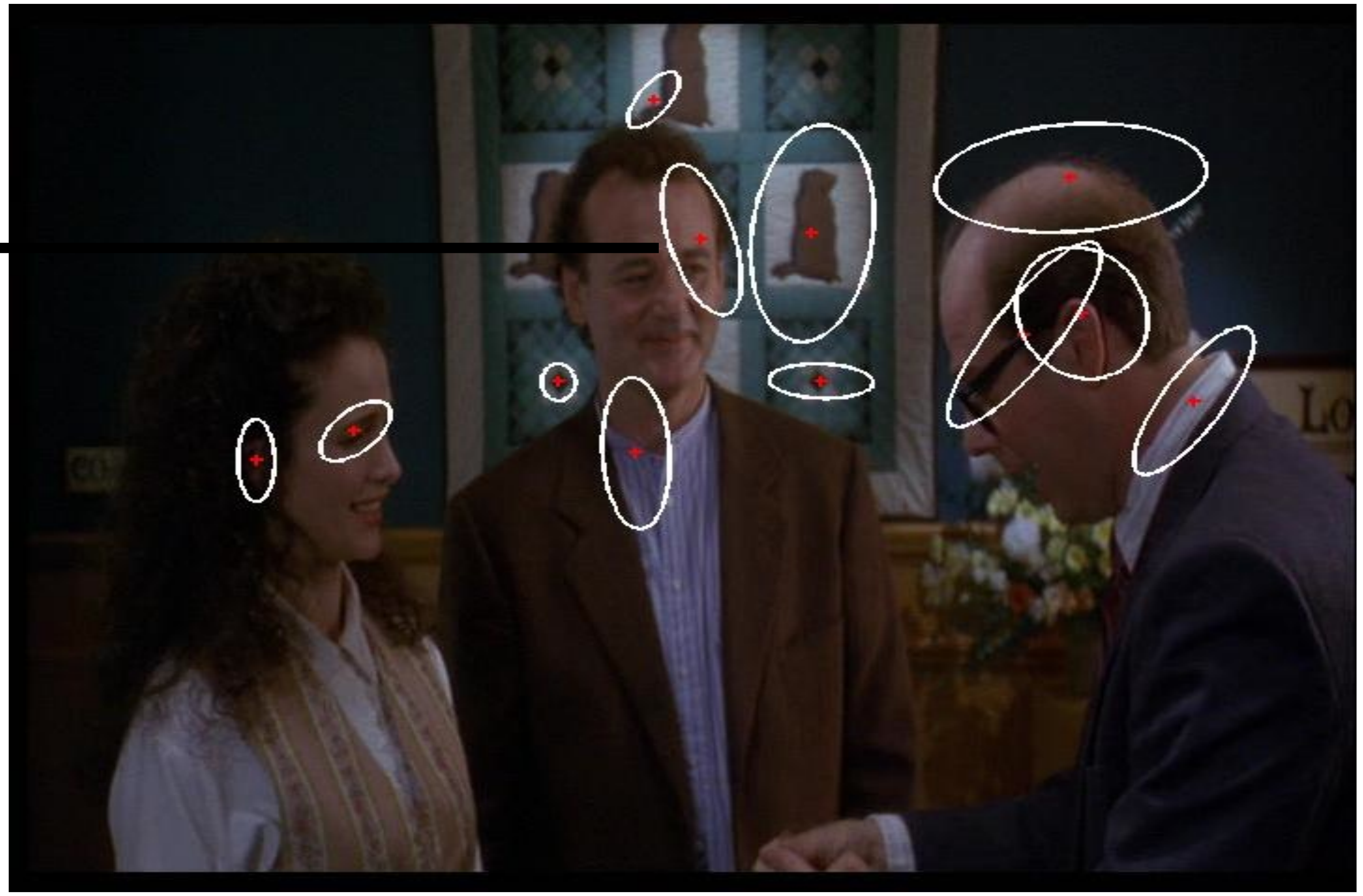


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

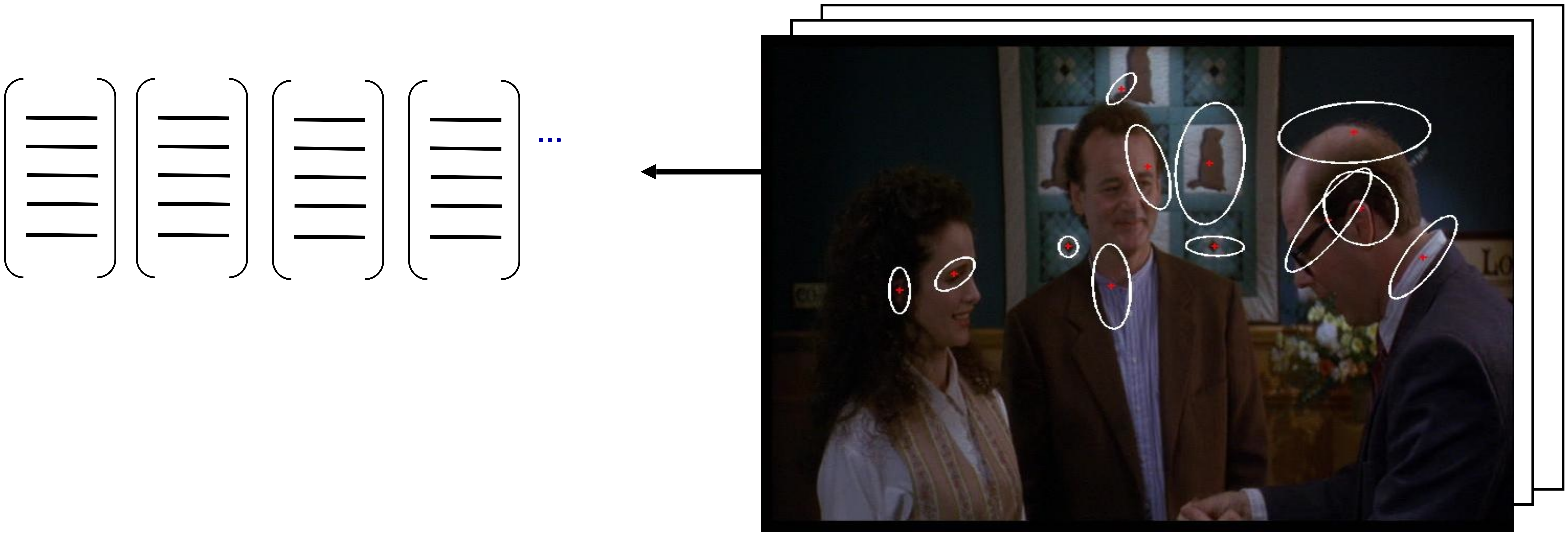
[Mikojaczyk and Schmid '02]

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

[Sivic & Zisserman, '03]

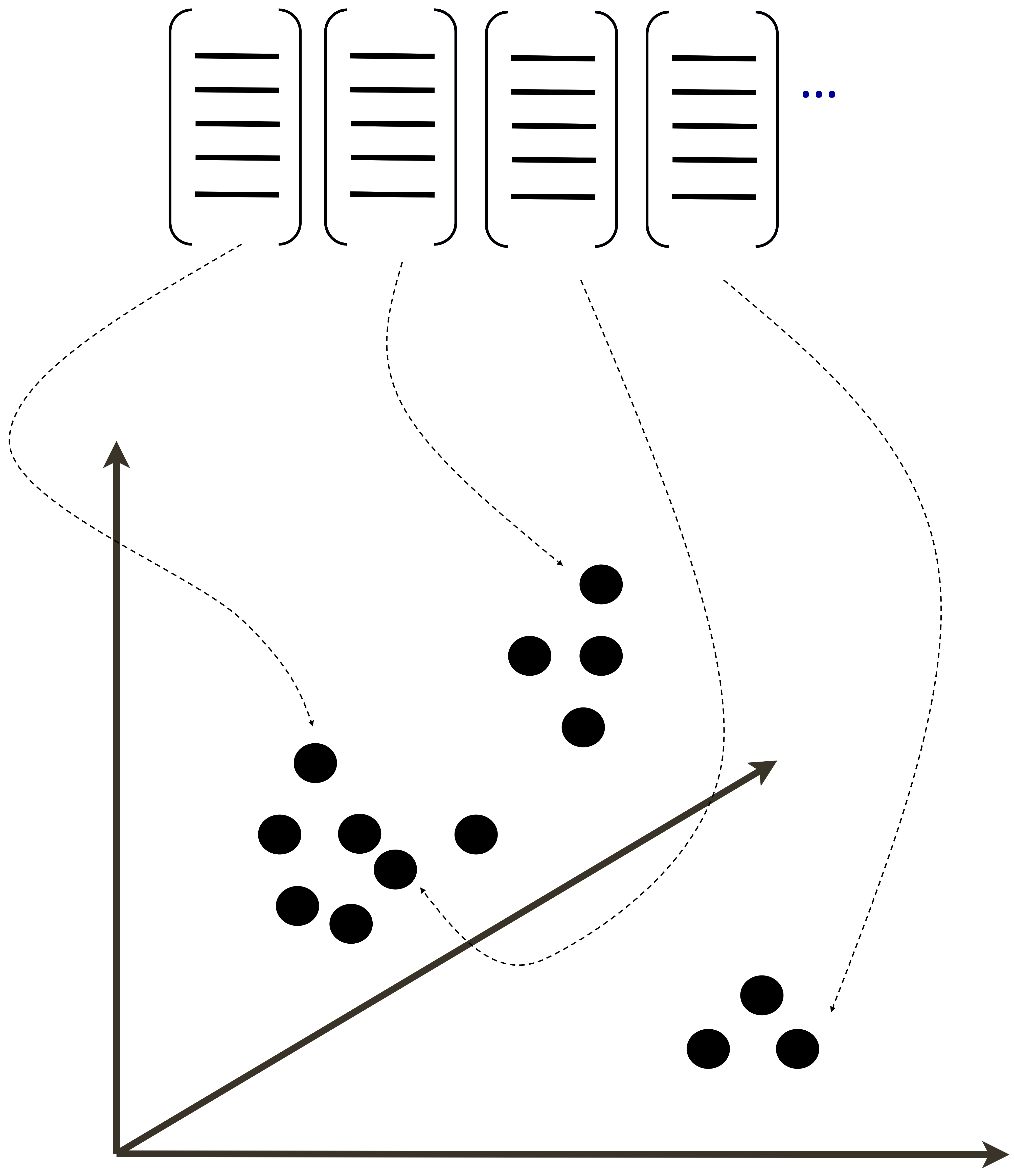
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Extracting **SIFT** Patches



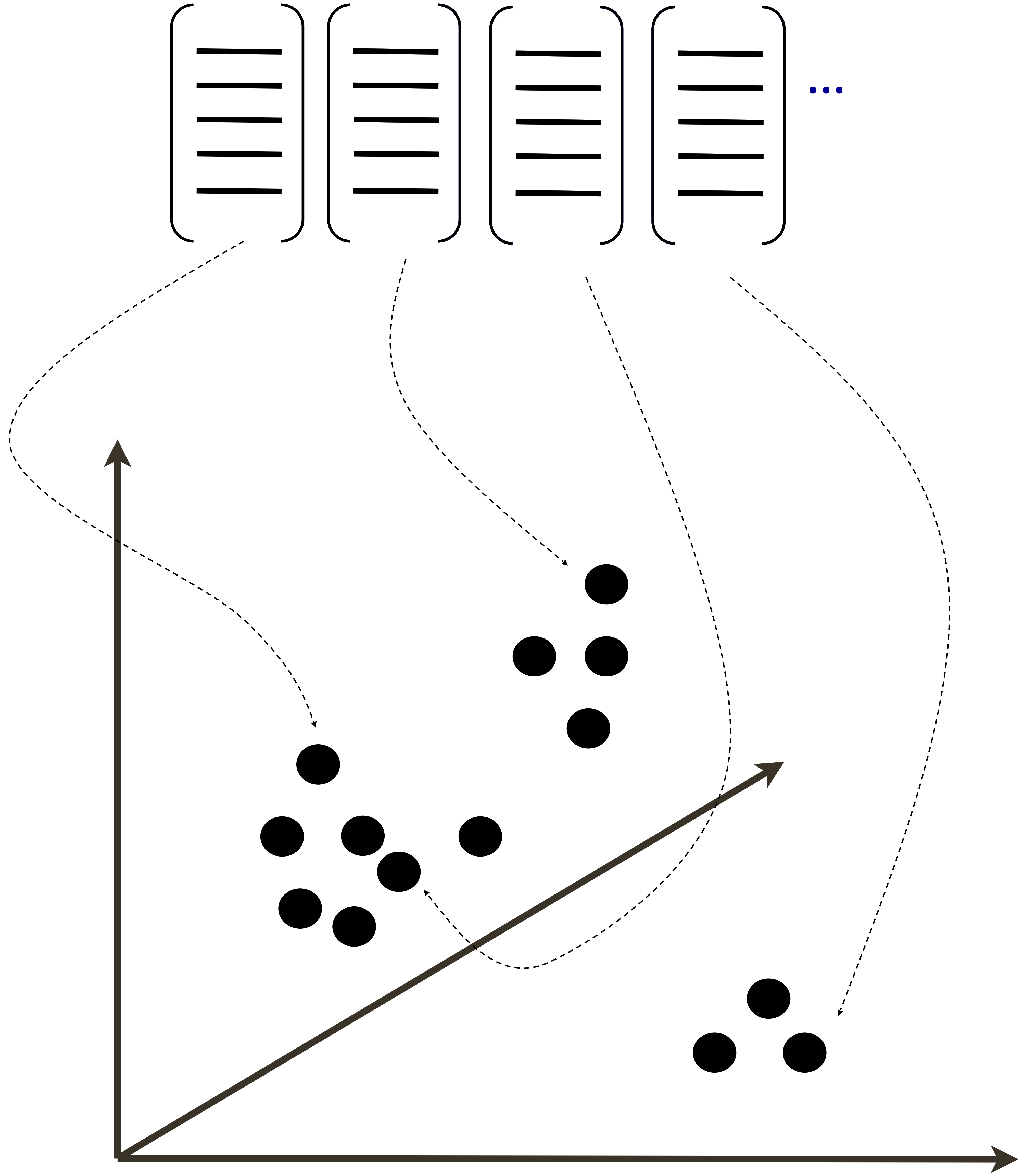
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

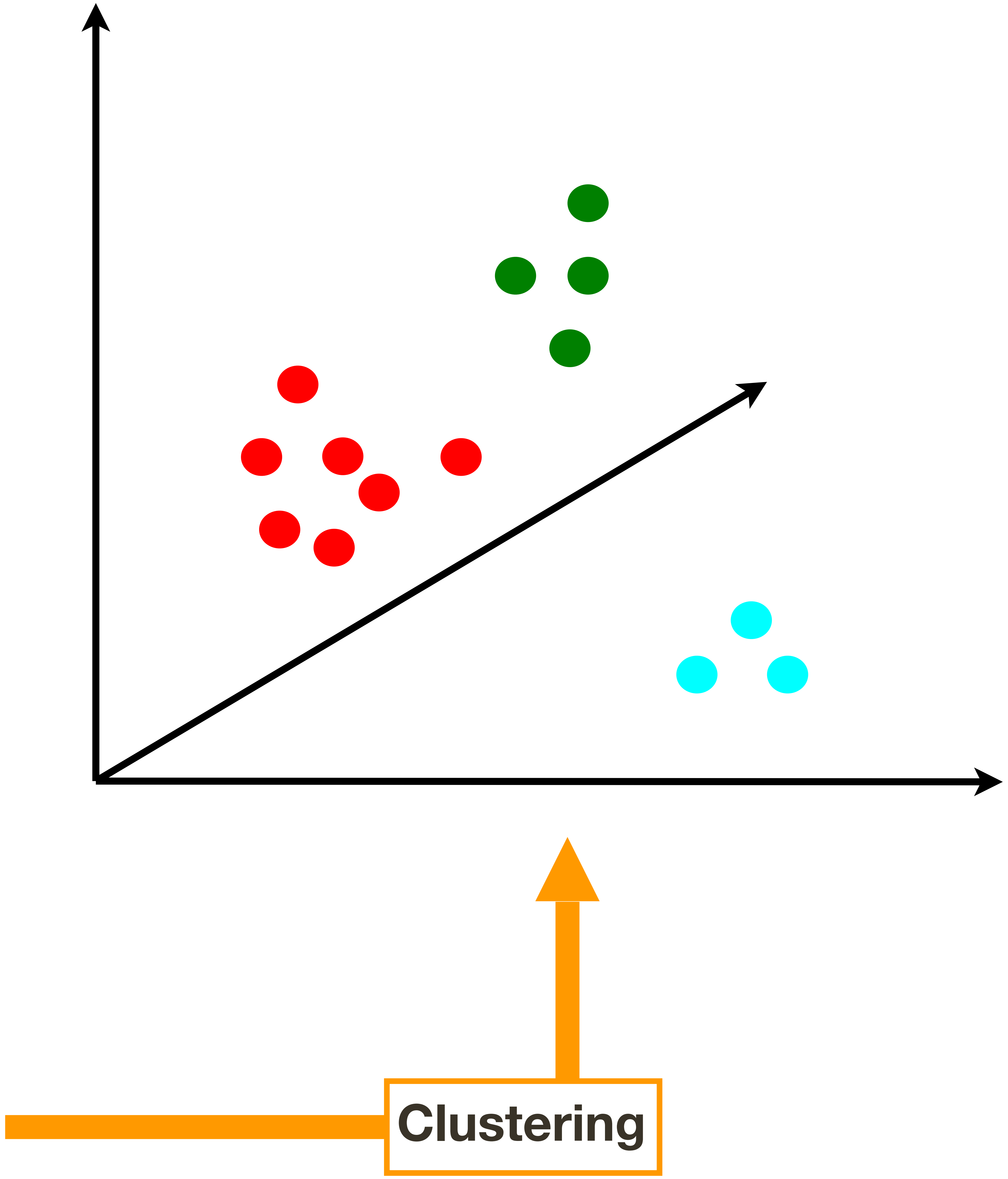
Creating **Dictionary**



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

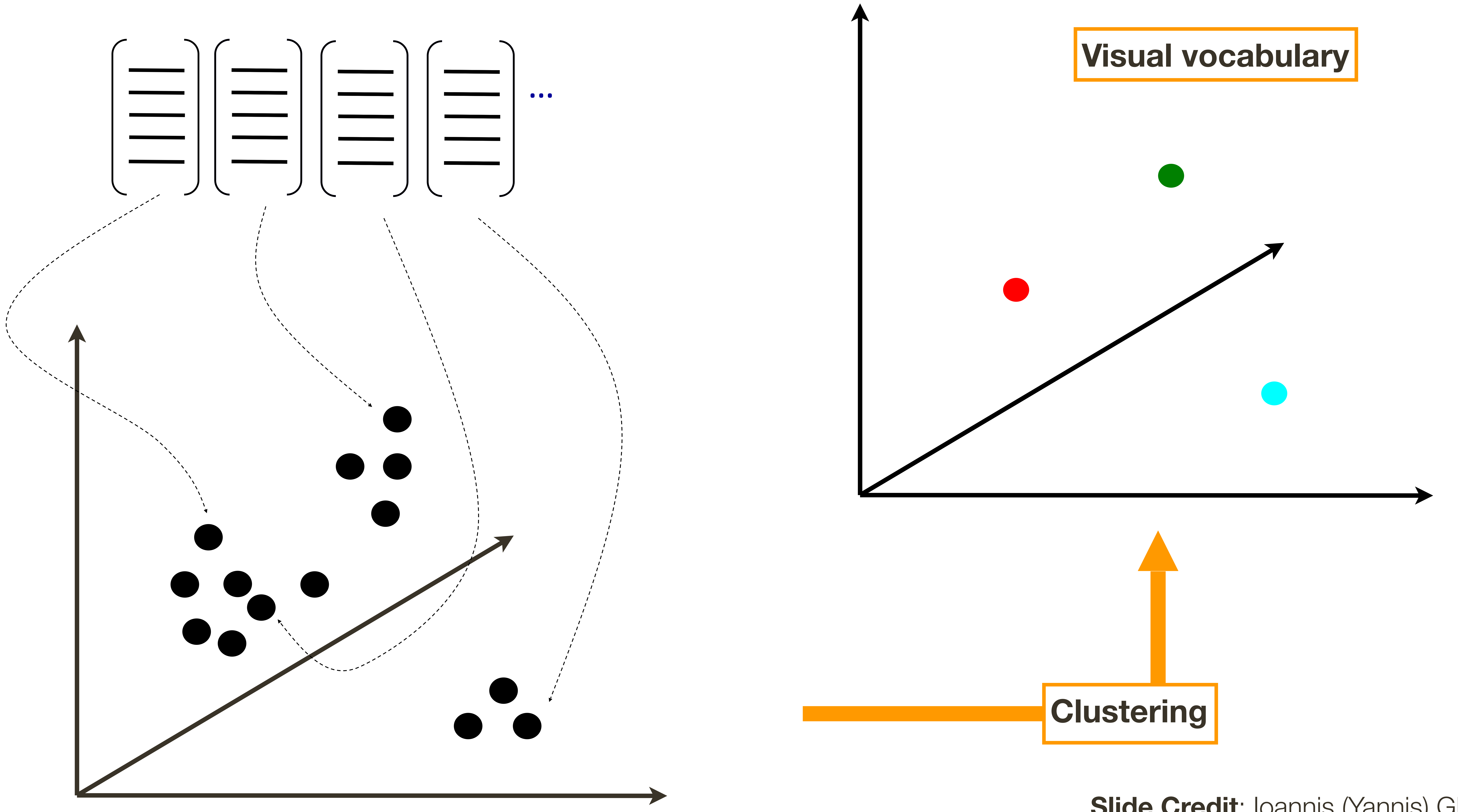
Creating **Dictionary**





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.

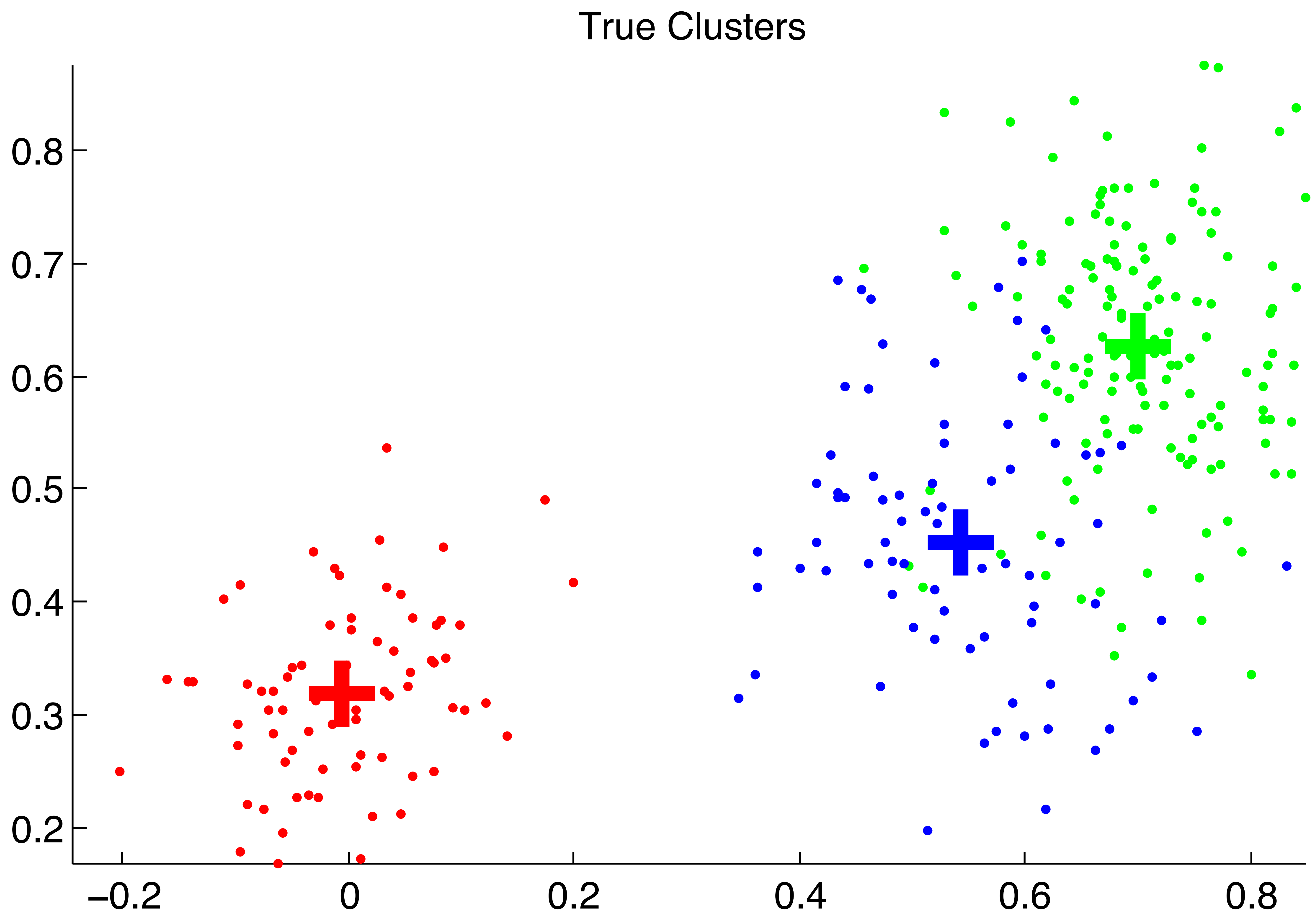
2. Assume the assignment of points to clusters is known (fixed). Compute the best center for each cluster, as the mean of the points assigned to the cluster.

The algorithm is initialized by choosing K random cluster centers

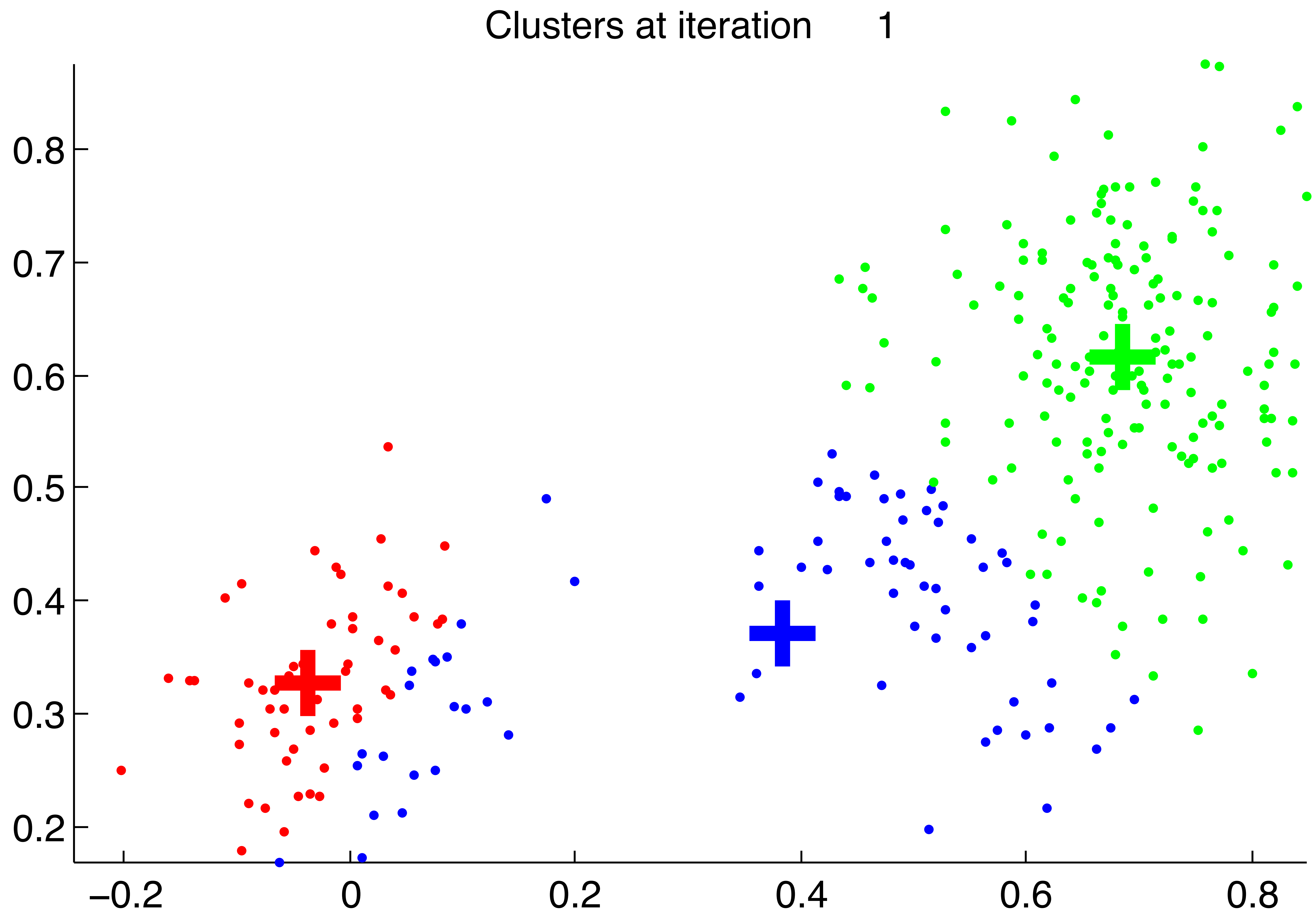
K-means converges to a local minimum of the objective function

— Results are initialization dependent

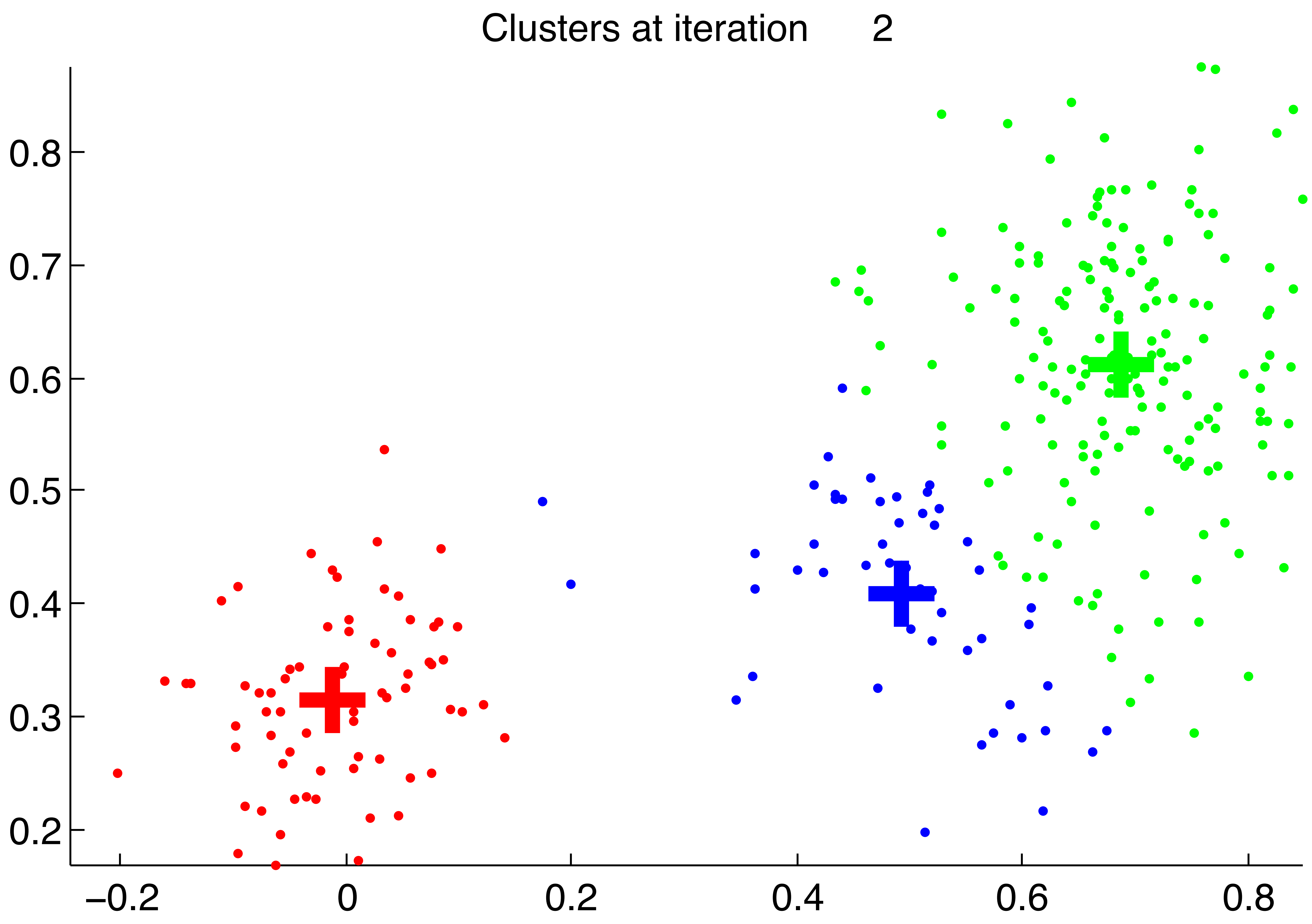
Example 1: K-Means Clustering



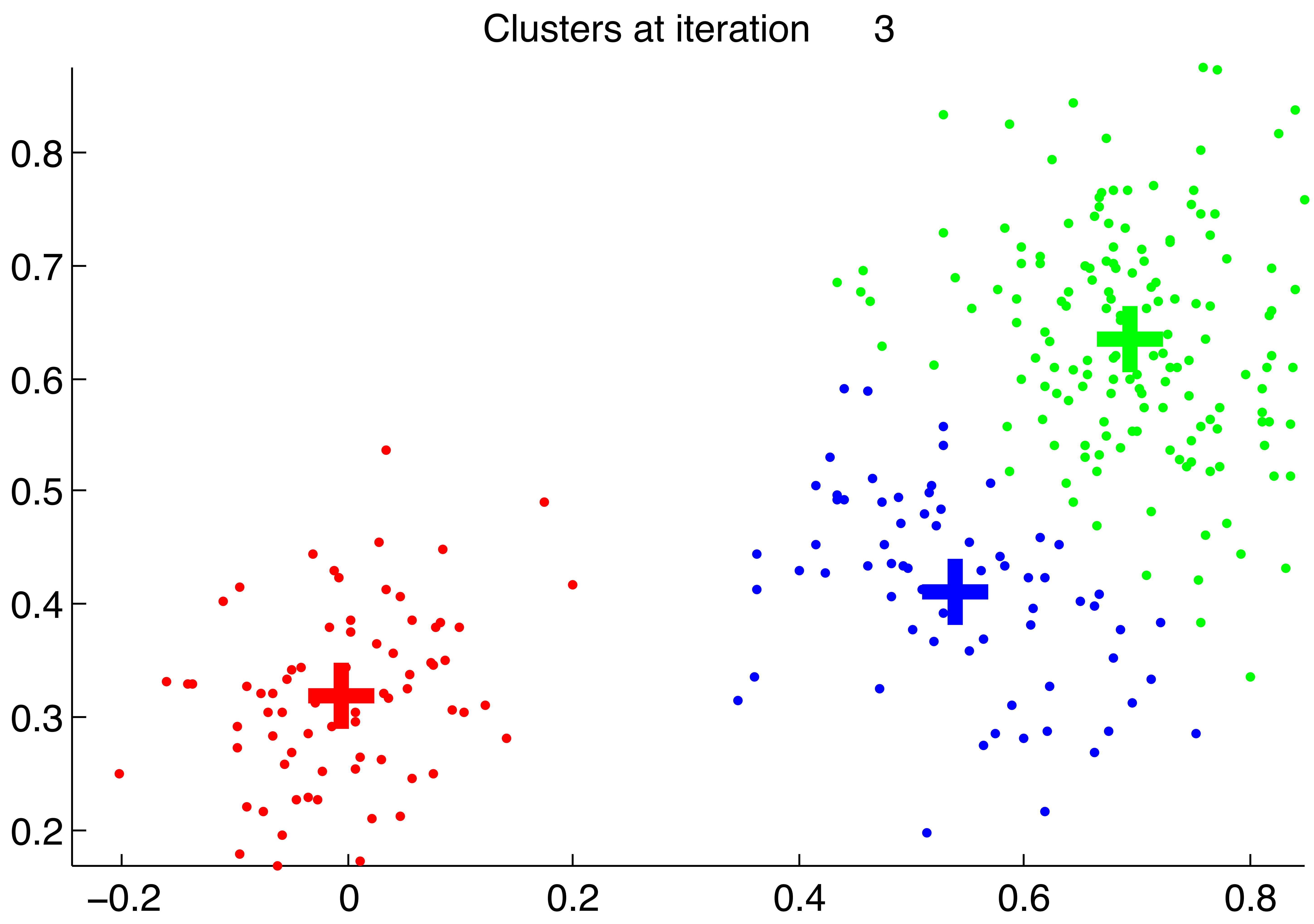
Example 1: K-Means Clustering



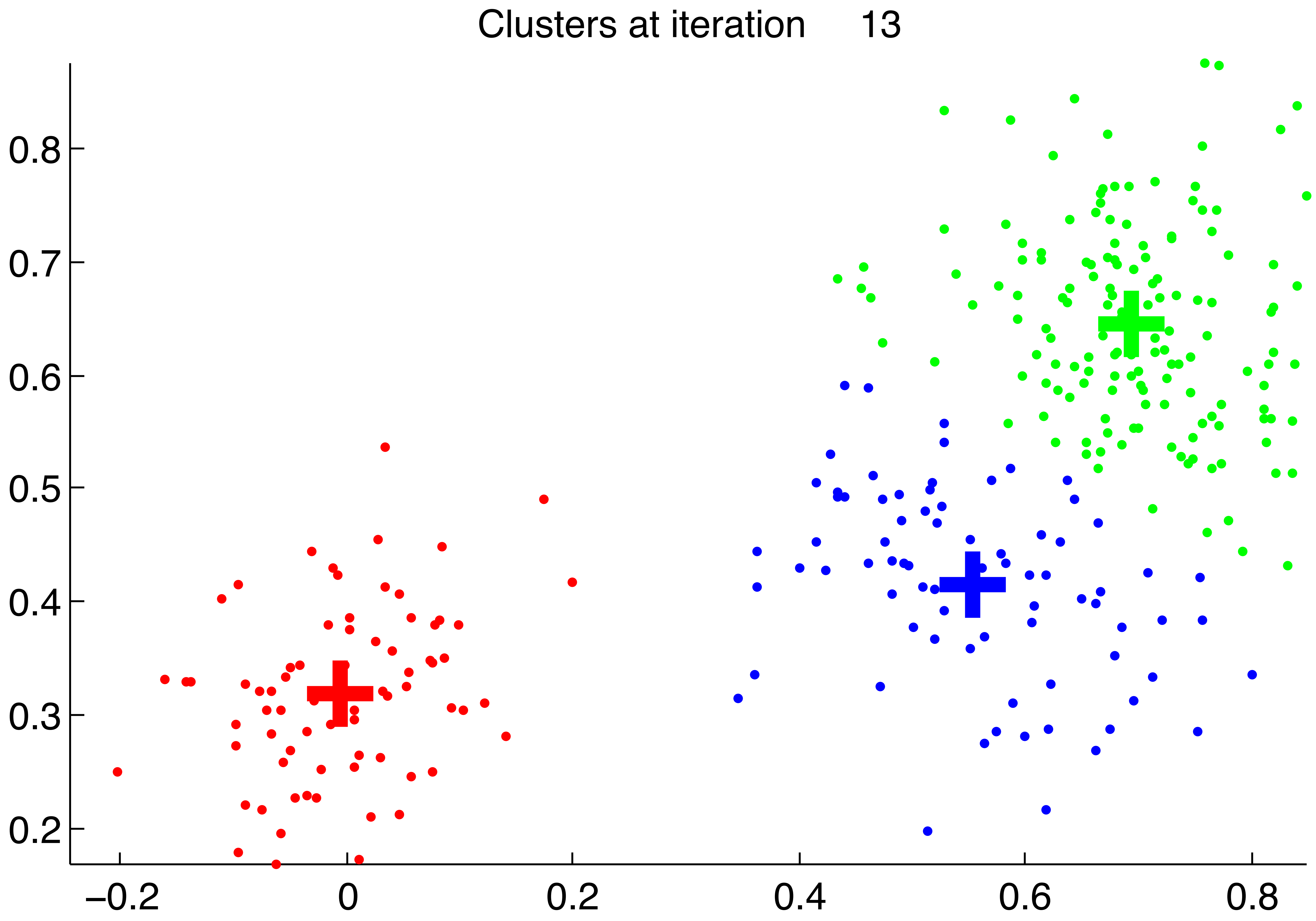
Example 1: K-Means Clustering



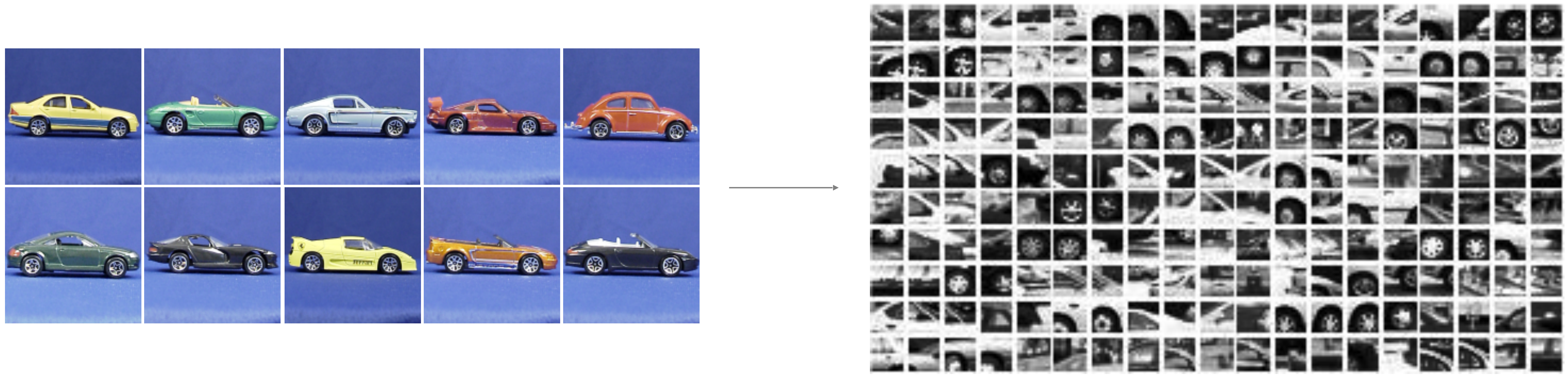
Example 1: K-Means Clustering

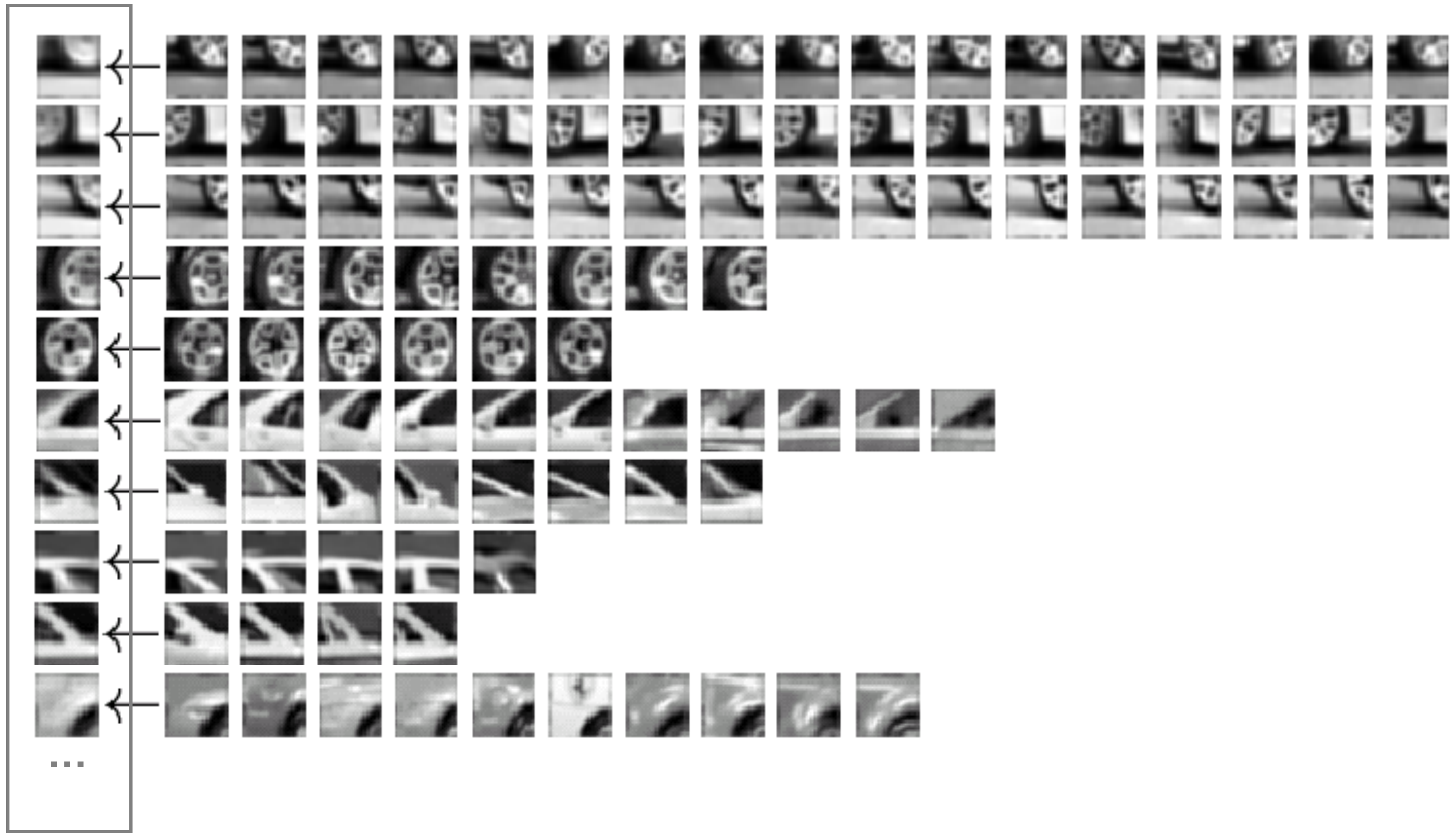


Example 1: K-Means Clustering



Example **Visual Dictionary**





Source: B. Leibe

Example **Visual Dictionary**

Source: B. Leibe

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

Dictionary Learning:
Learn Visual Words using clustering

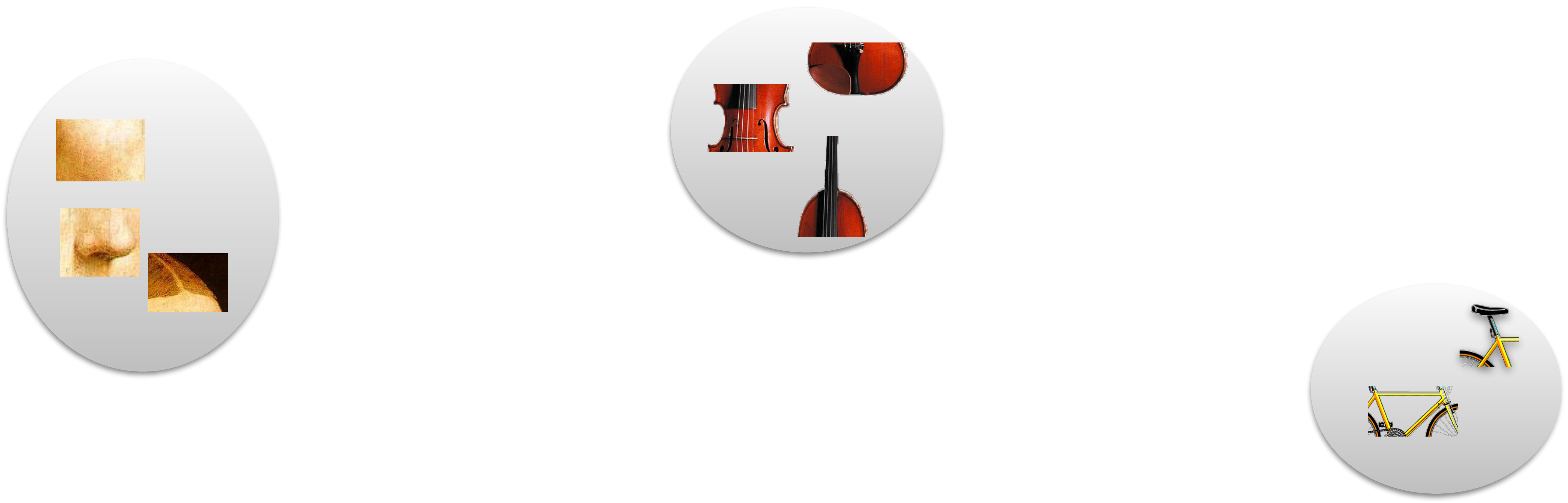
Encode:
build Bags-of-Words (BOW) vectors
for each image

Classify:
Train and test data using BOWs

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

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

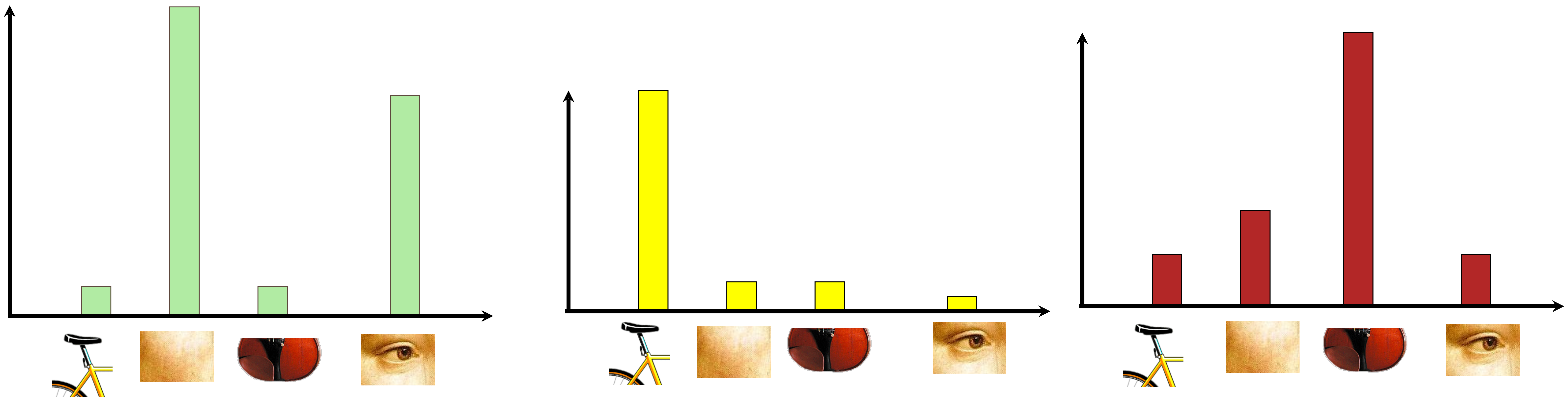
1. **Quantization**: image features gets associated to a visual word (nearest cluster center)



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

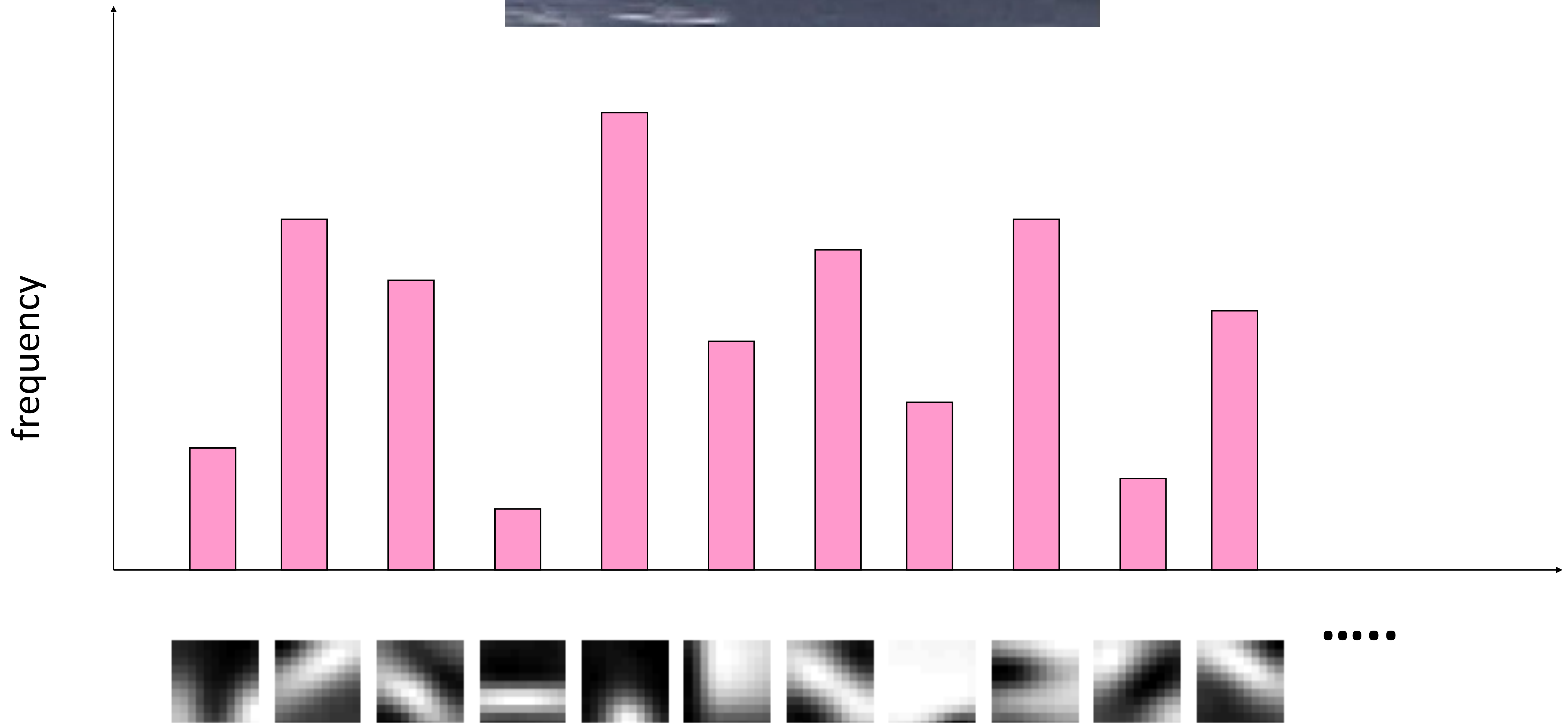
2. **Encode:** build Bag-of-Words (BOW) vectors for each image

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)

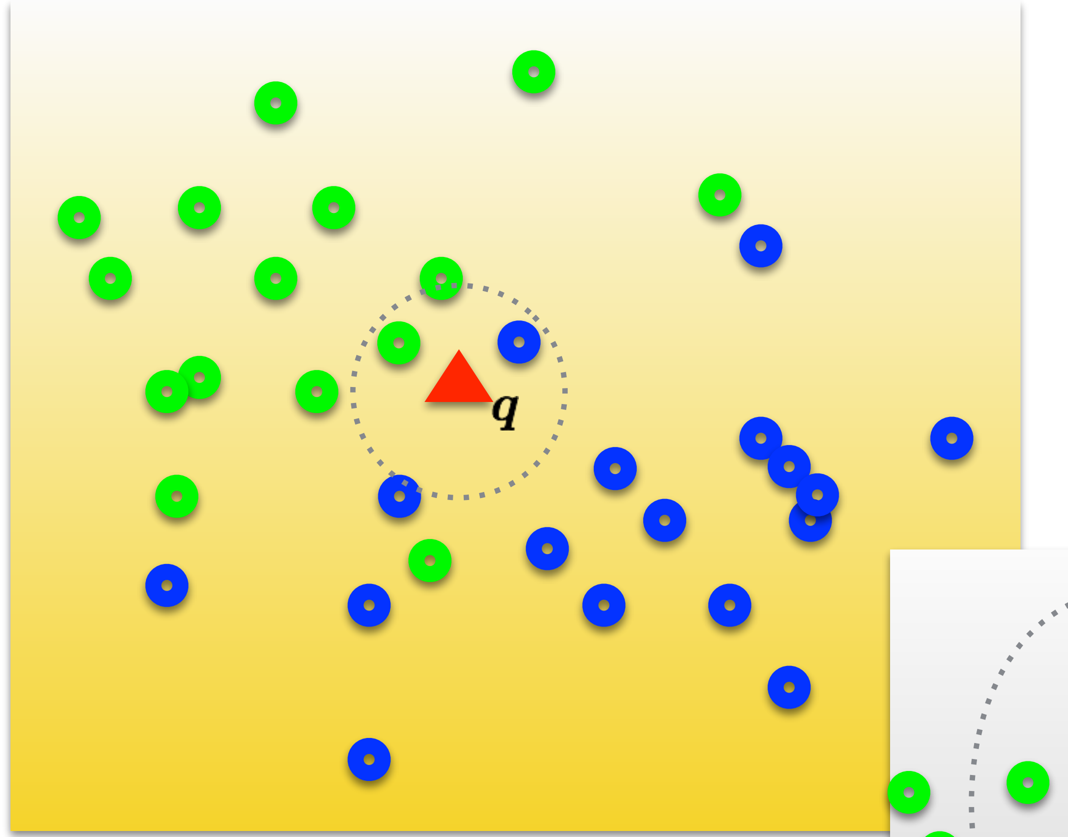
Dictionary Learning:
Learn Visual Words using clustering

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

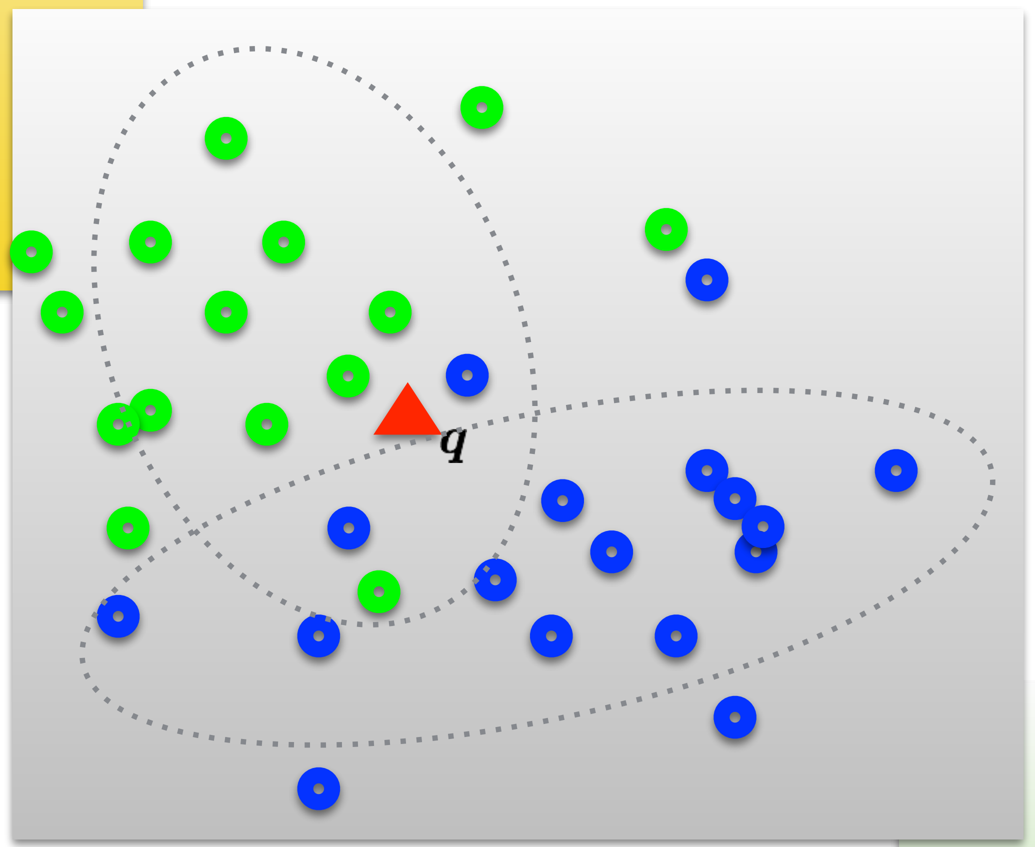
Classify:
Train and test data using BOWs

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

3. **Classify**: Train and text classifier using BOWs

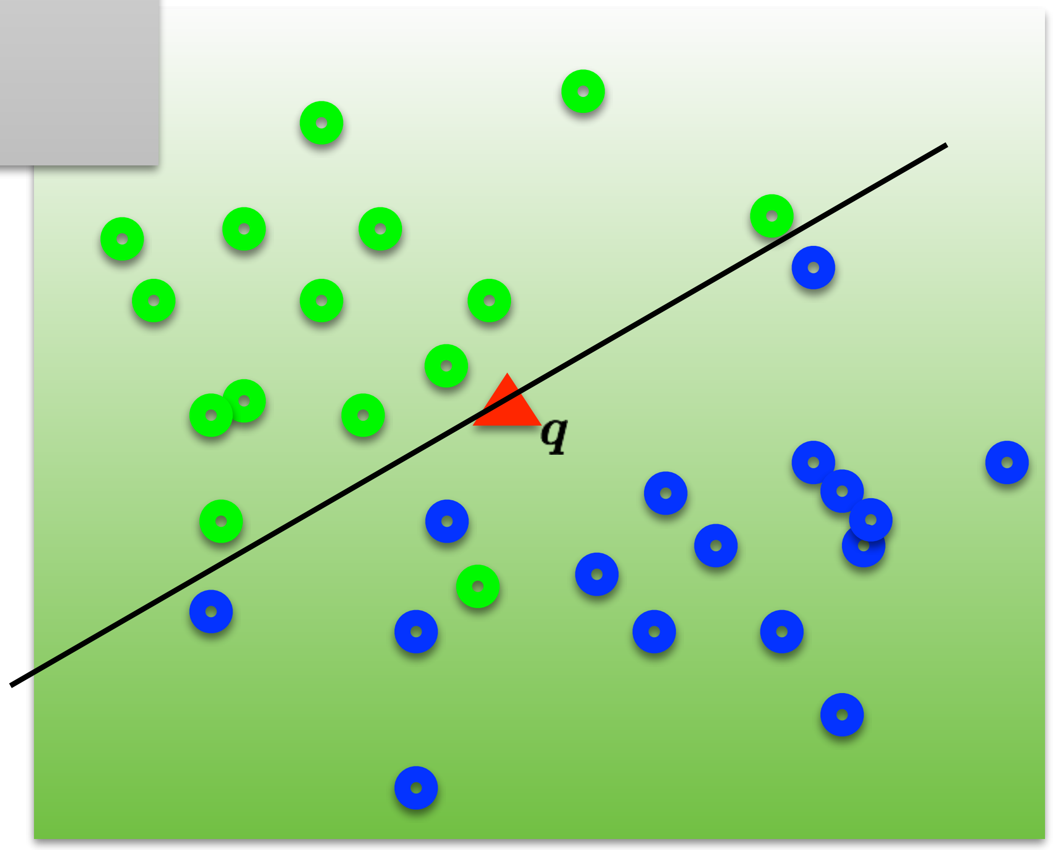


K nearest neighbors



Naïve Bayes

Support Vector Machine



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)