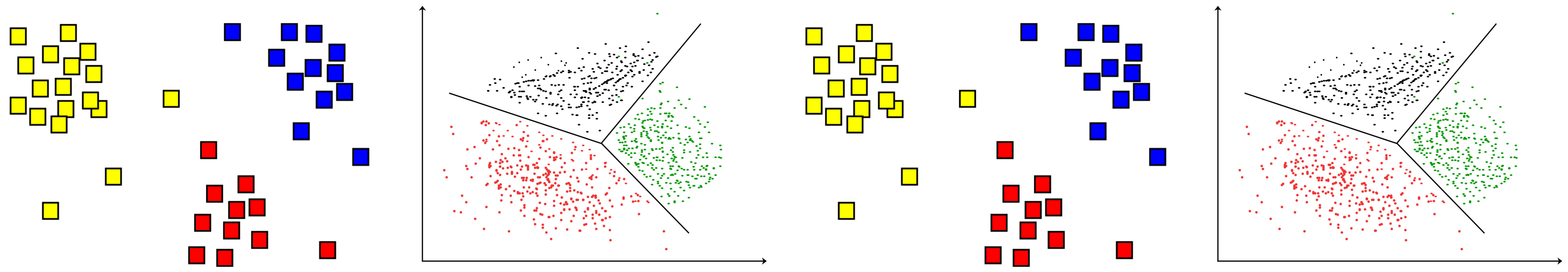
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 30: Classification (cont.)

Menu for Today (**November 19, 2018**)

Topics:

— Decision Trees
— Boosting
— Object Detection
— Face Detection

Redings:

— **Today's** Lecture: Forsyth & Ponce (2nd ed.) 17.1
— **Next** Lecture: Forsyth & Ponce (2nd ed.) 17.2

Reminders:

— **Assignment 5**: Scene Recognition with Bag of Words due **last day of classes**

Plan for Remainder of the Course

This week

- Finish classification
- Object detection

Next week

- Deep learning
- Final review

Final (December 11th)

- I will be away between December 2nd and December 9th
- I will have additional office hours next week and on December 10th
- I will have printed Midterm solutions in my office on Friday

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

Follow-up work to previous lecture's example of **Eulerian** video magnification

Lecture 29: Re-cap

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

The **spatial pyramid** partitions the image and counts visual words within each grid box; this is repeated at multiple levels

Lecture 29: Re-cap **Bag-of-Words** Representation

Dictionary Learning:
Learn Visual Words using clustering

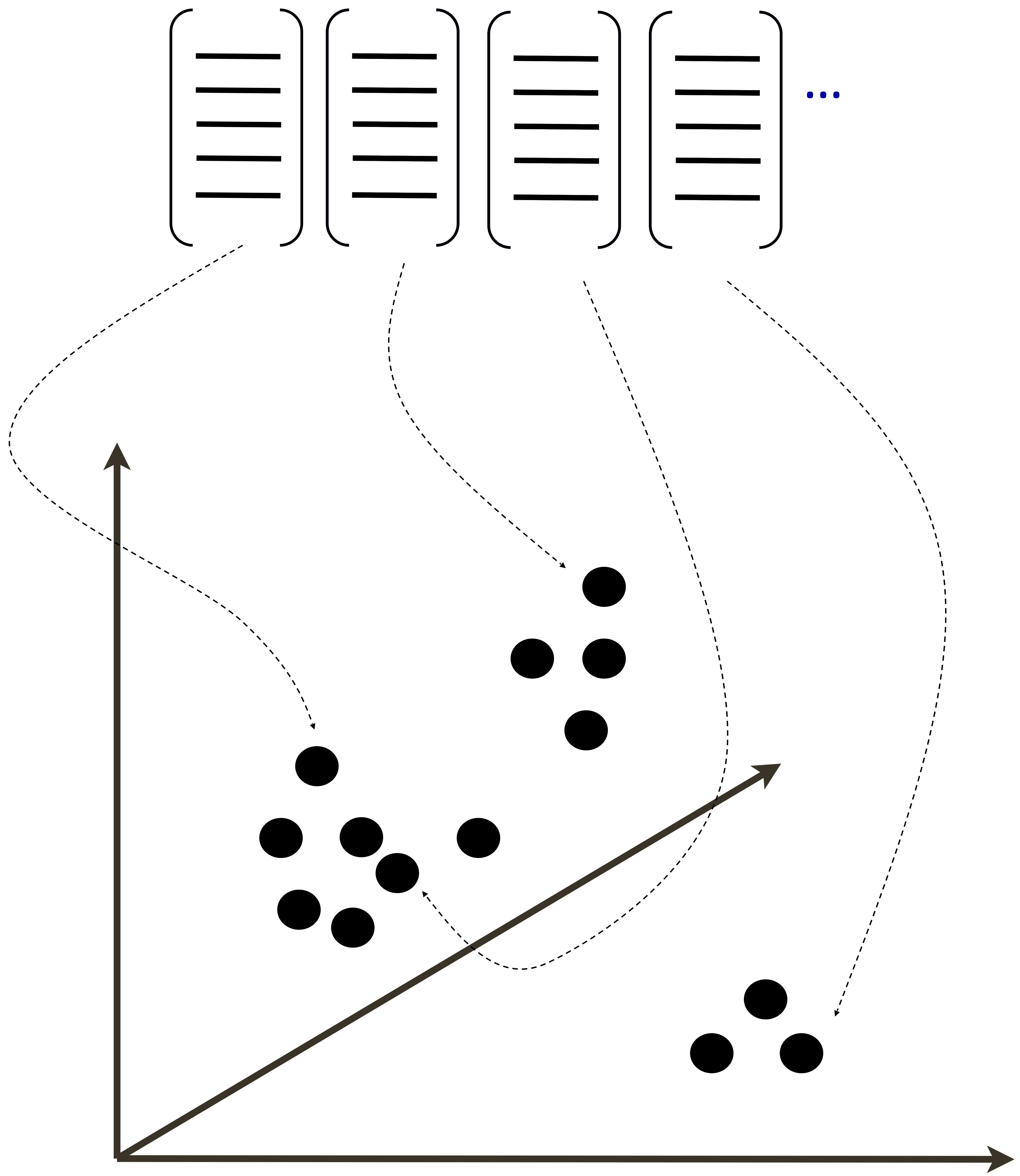
Encode:
build Bags-of-Words (BOW) vectors
for each image

Classify:
Train and test data using BOWs

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Lecture 29: Re-cap

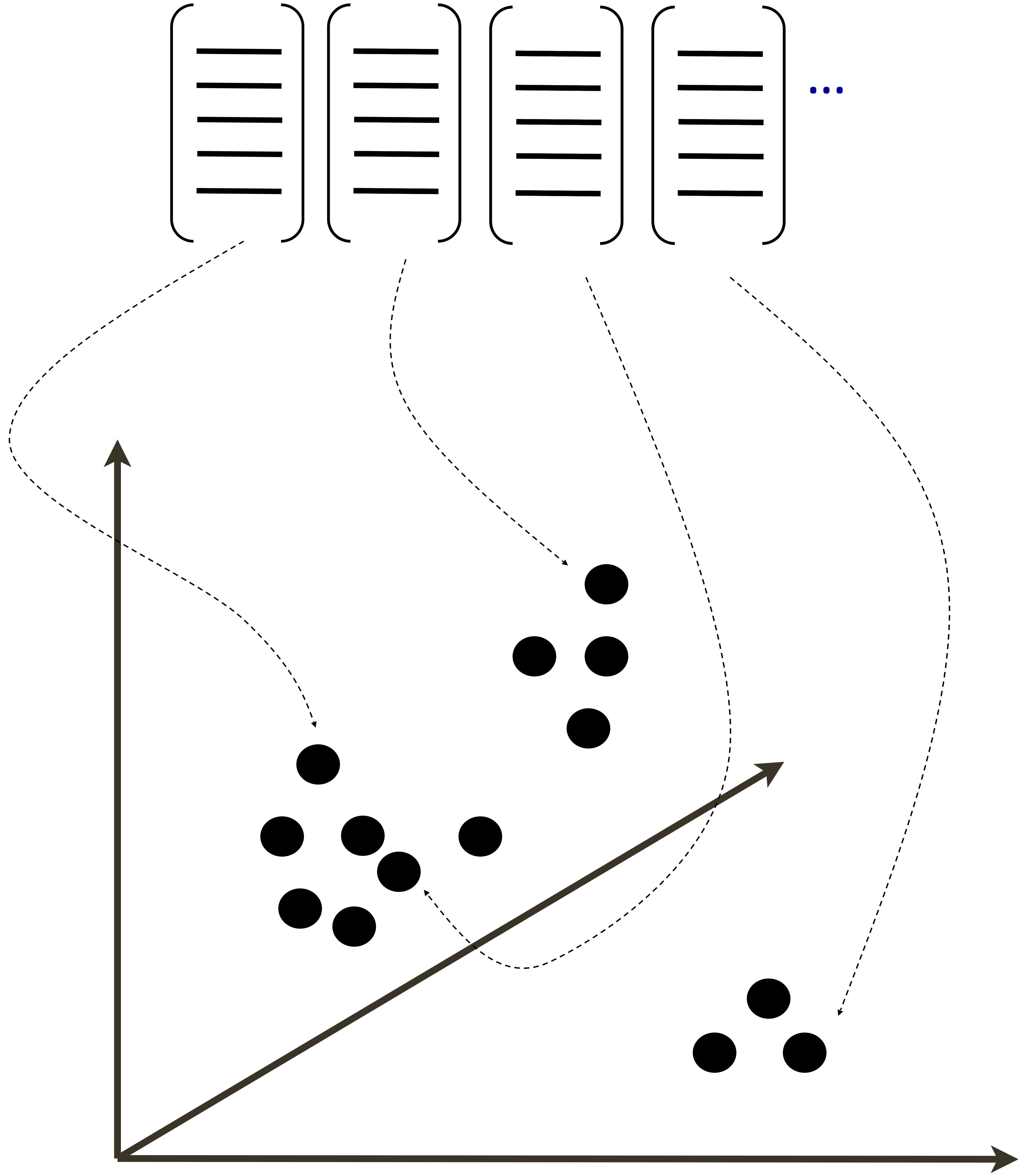
Bag-of-Words Representation

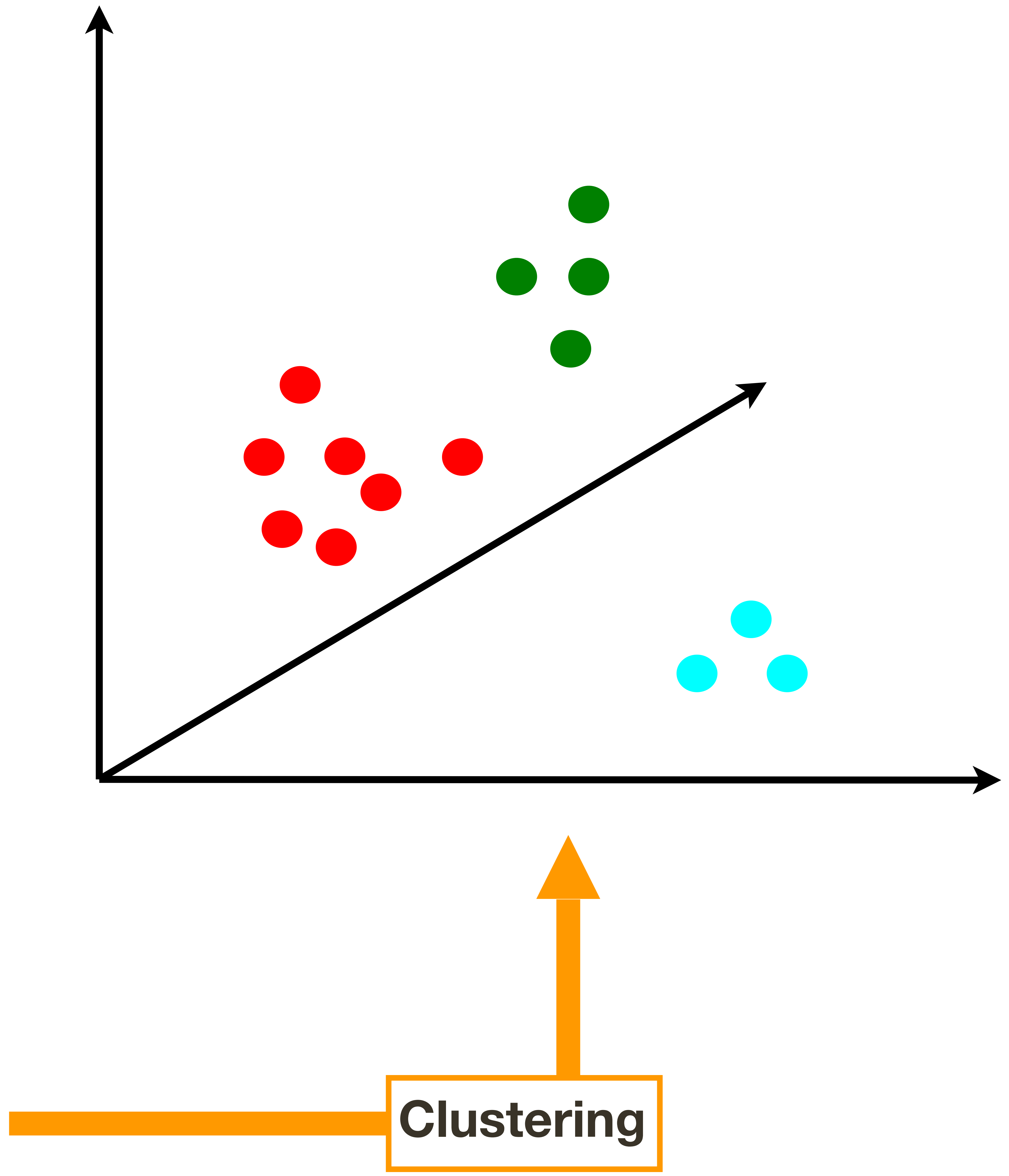


Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Lecture 29: Re-cap

Bag-of-Words Representation

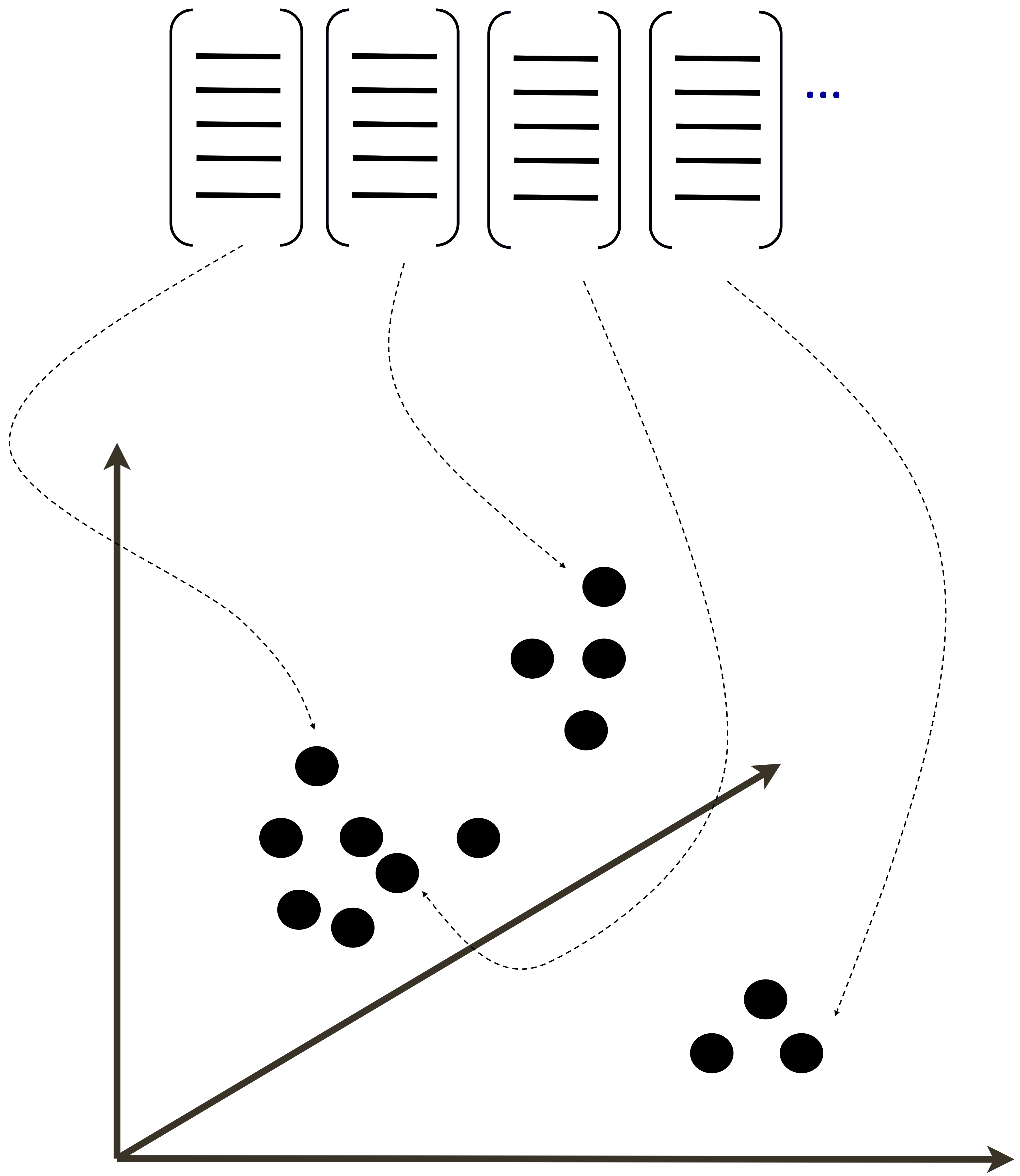


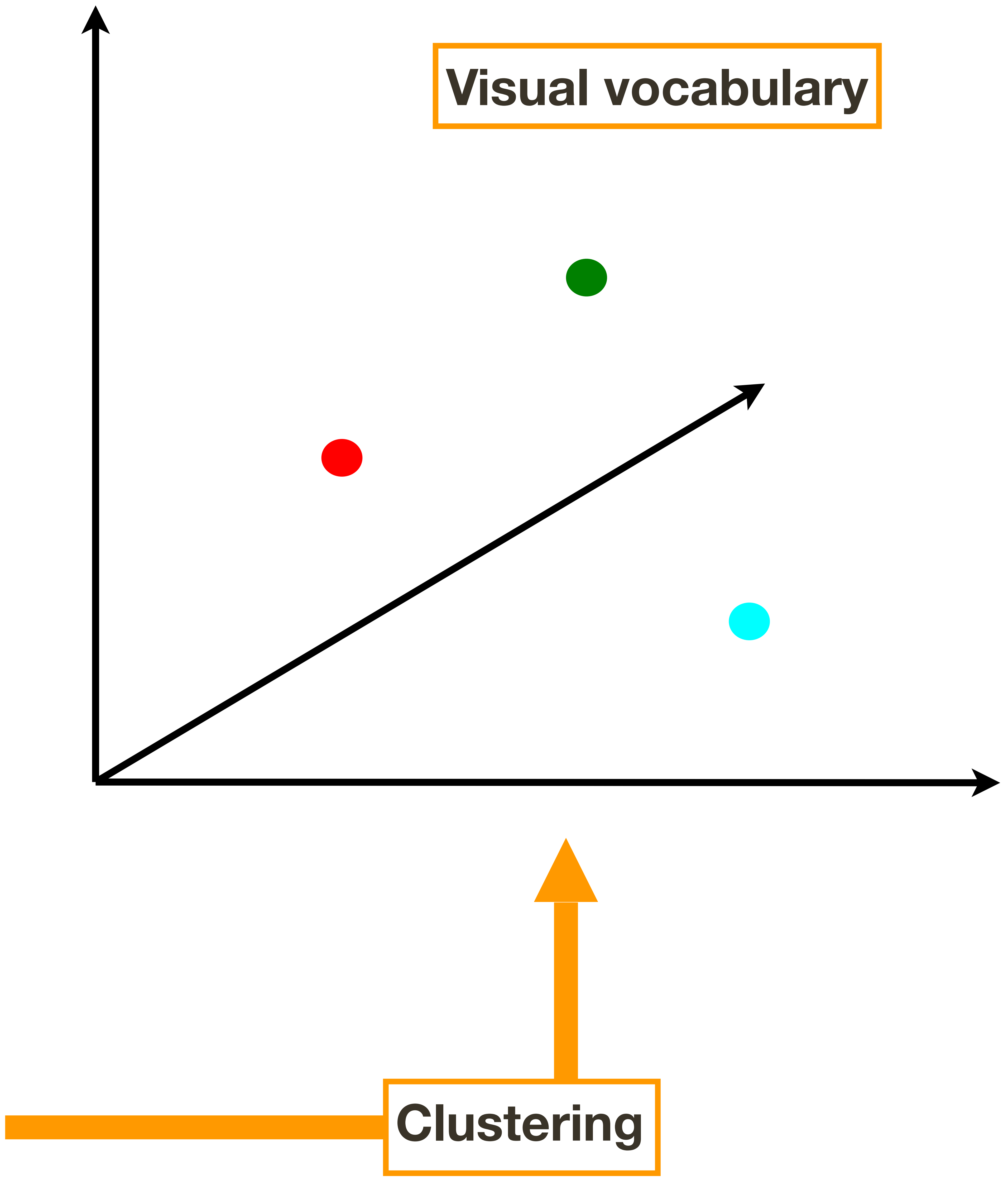


Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Lecture 29: Re-cap

Bag-of-Words Representation



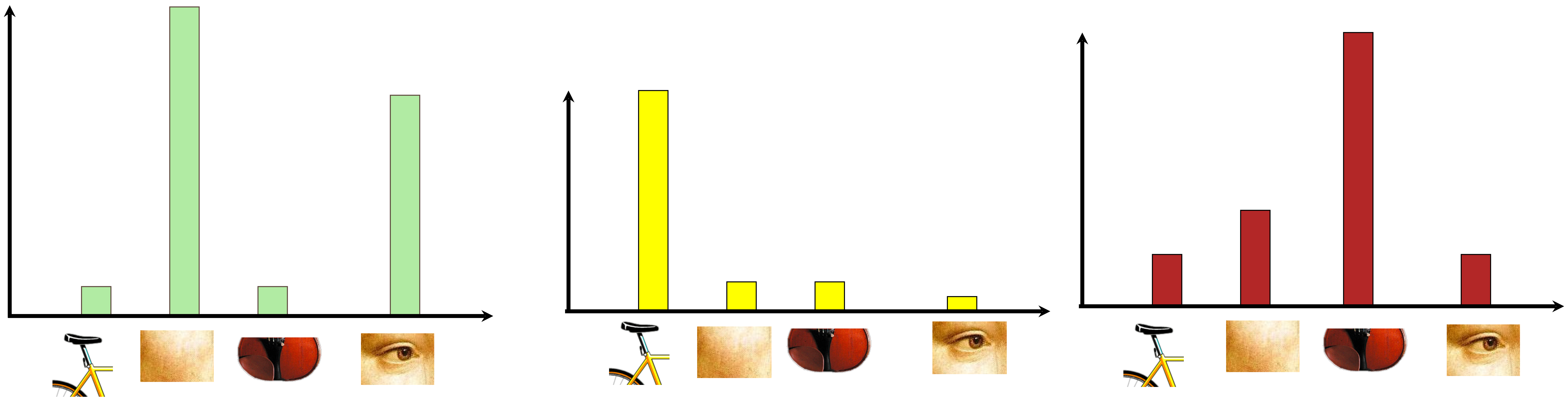


Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bag-of-Words Representation

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

Histogram: count the number of visual word occurrences



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Spatial Pyramid

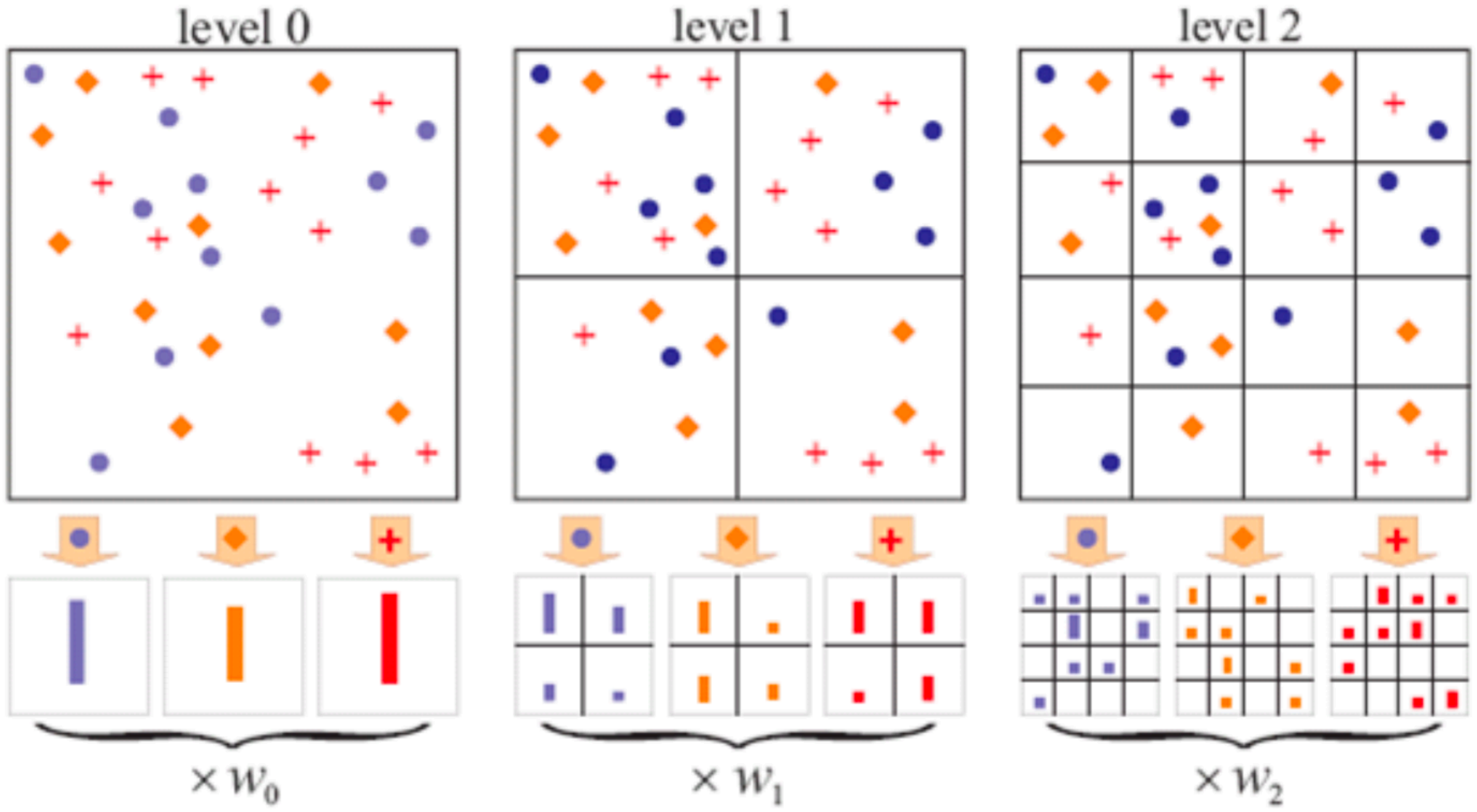


Fig. 16.8 in Forsyth & Ponce (2nd ed.).
Original credit: Lazebnik et al., 2006

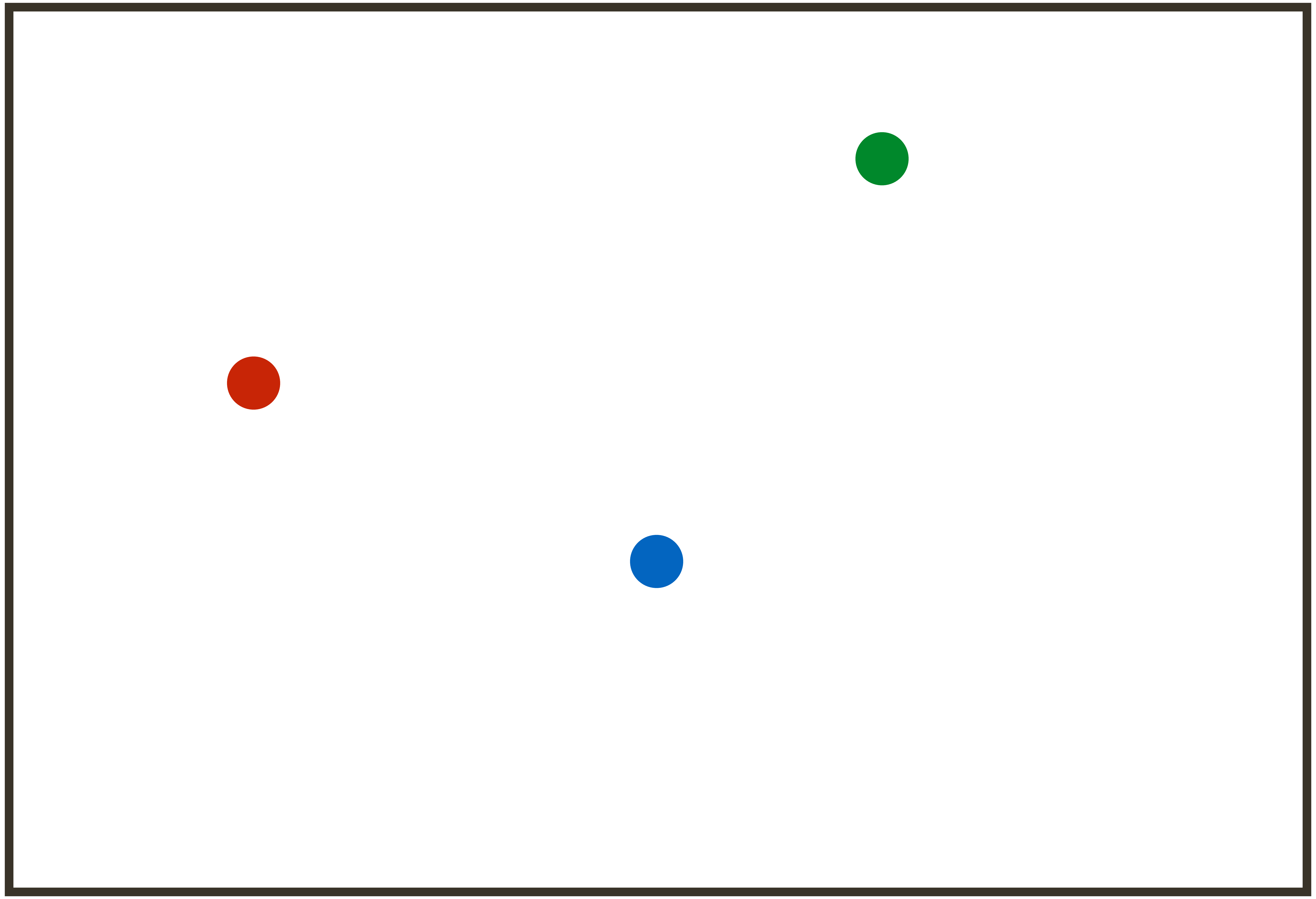
VLAD (Vector of Locally Aggregated Descriptors)

There are more advanced ways to ‘count’ visual words than incrementing its histogram bin

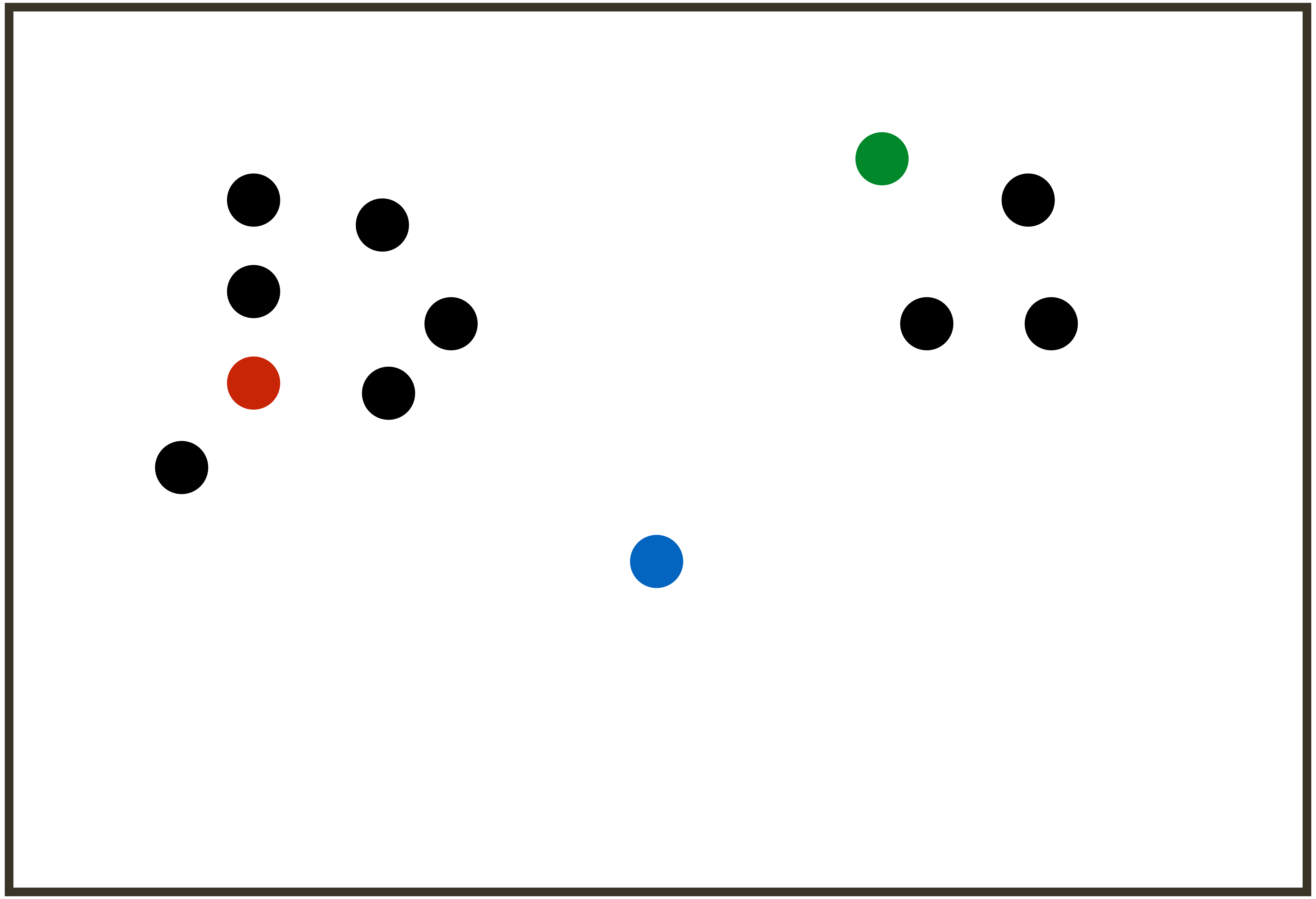
For example, it might be useful to describe how local descriptors are quantized to their visual words

In the VLAD representation, instead of incrementing the histogram bin by one, we increment it by the **residual** vector $\boldsymbol{x - c(x)}$

Example: VLAD

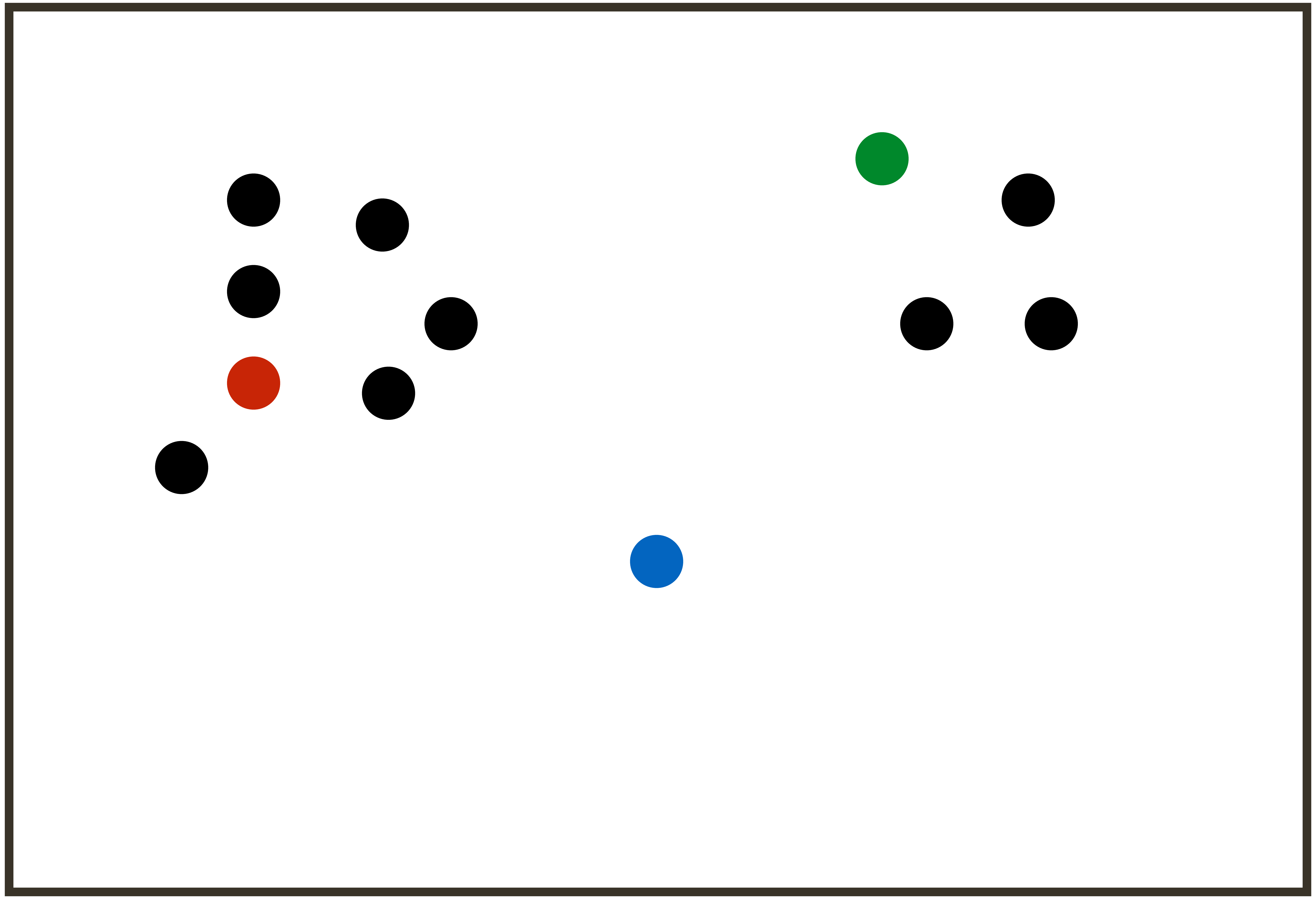


Example: VLAD

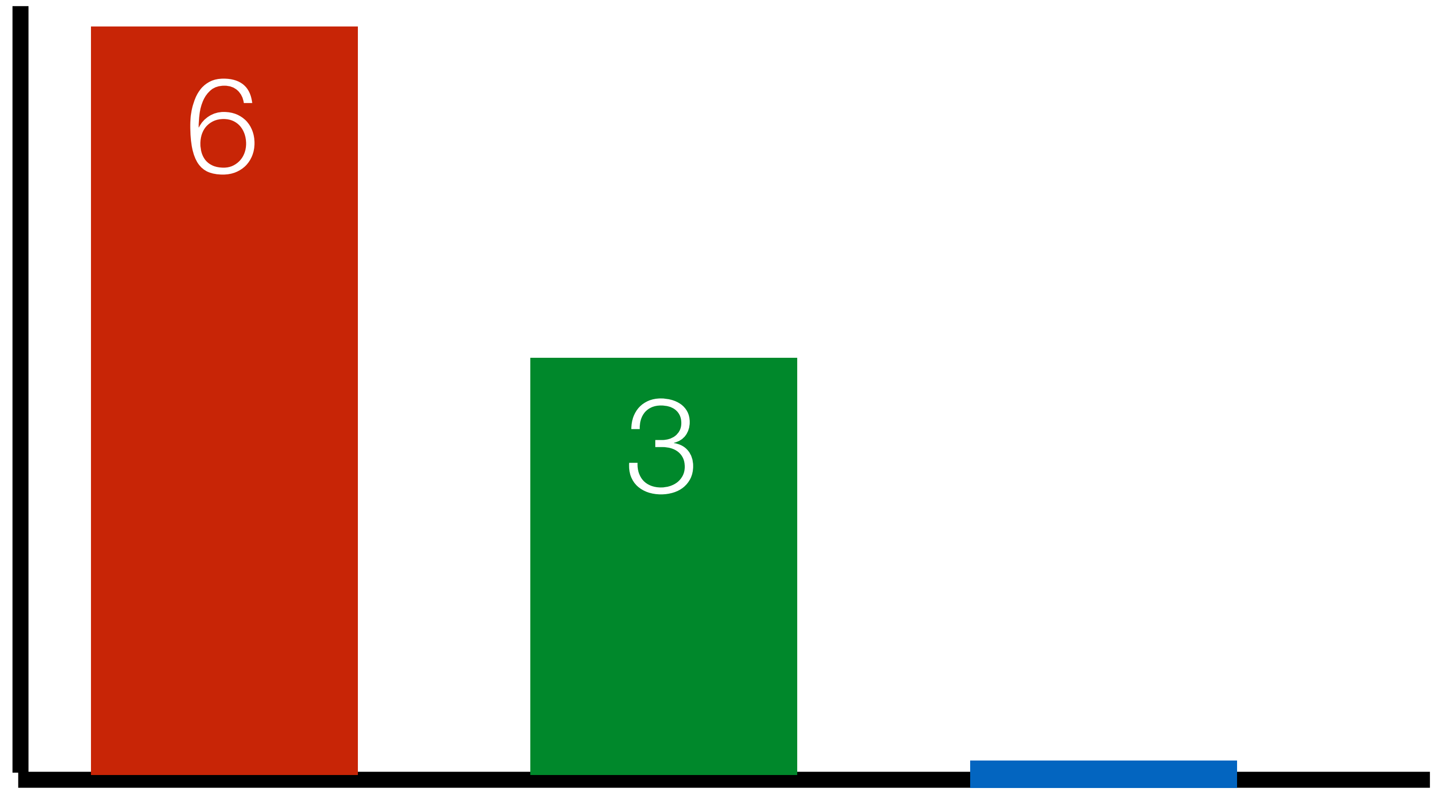


Bag of Word

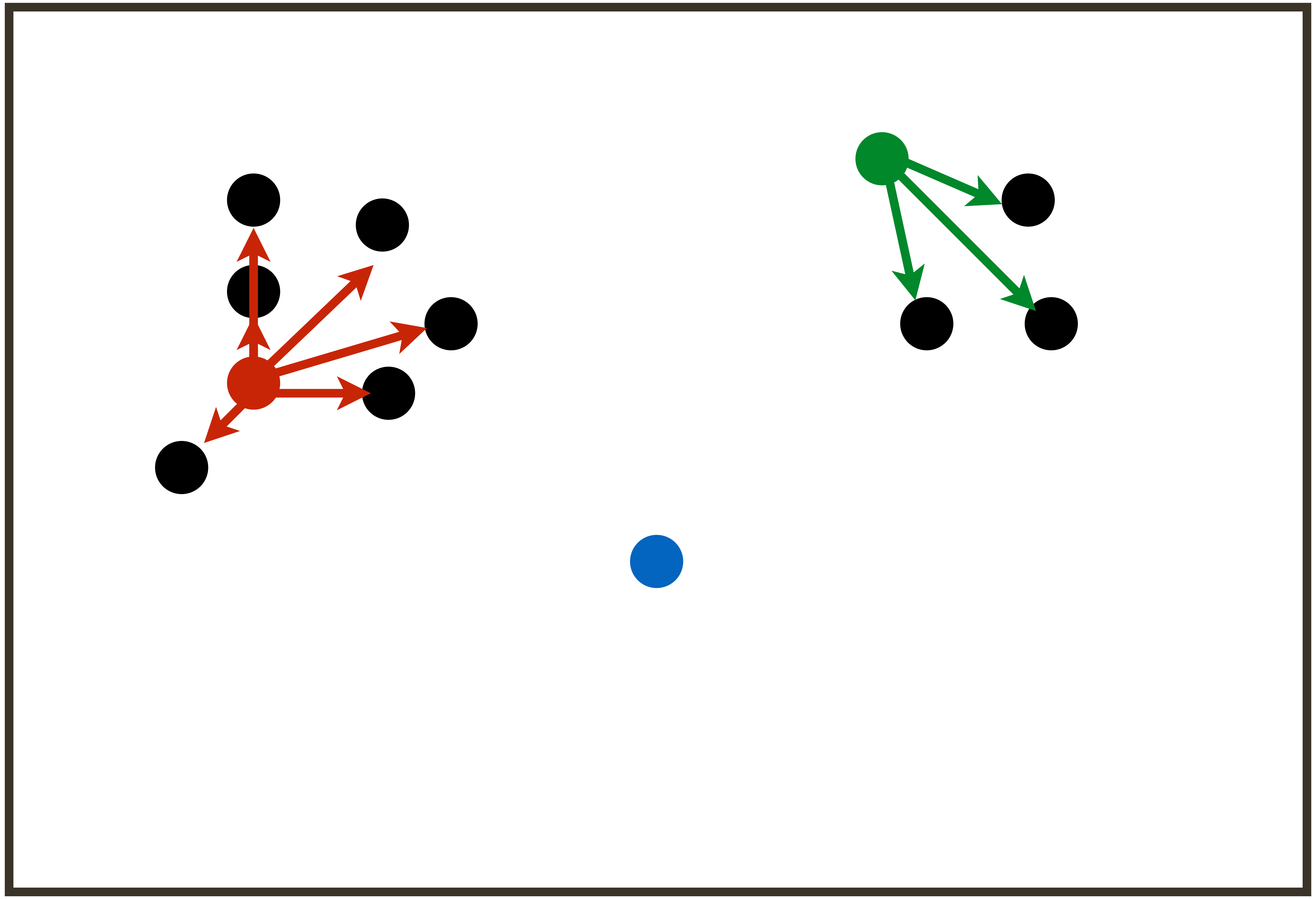
Example: VLAD



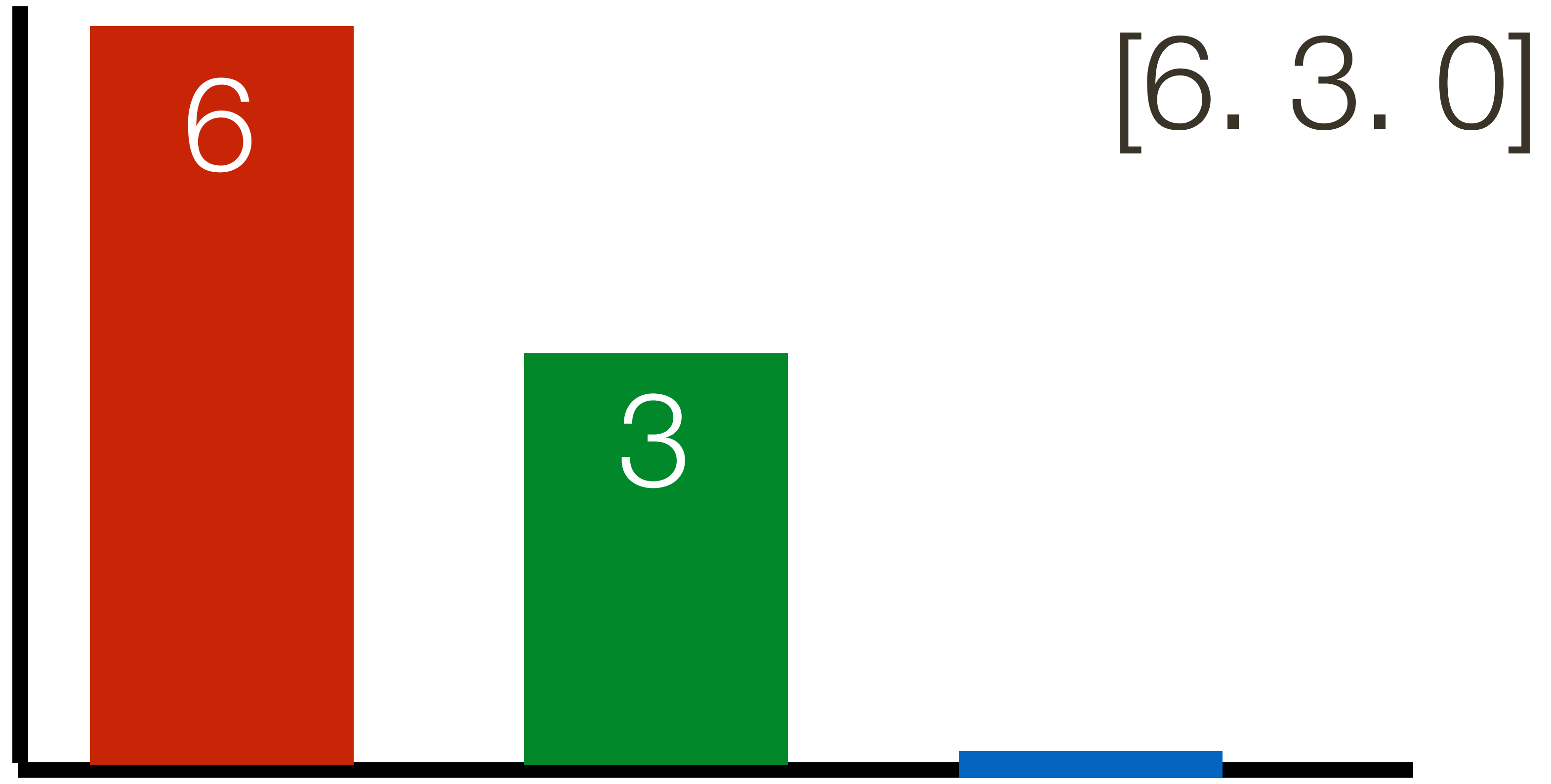
Bag of Word



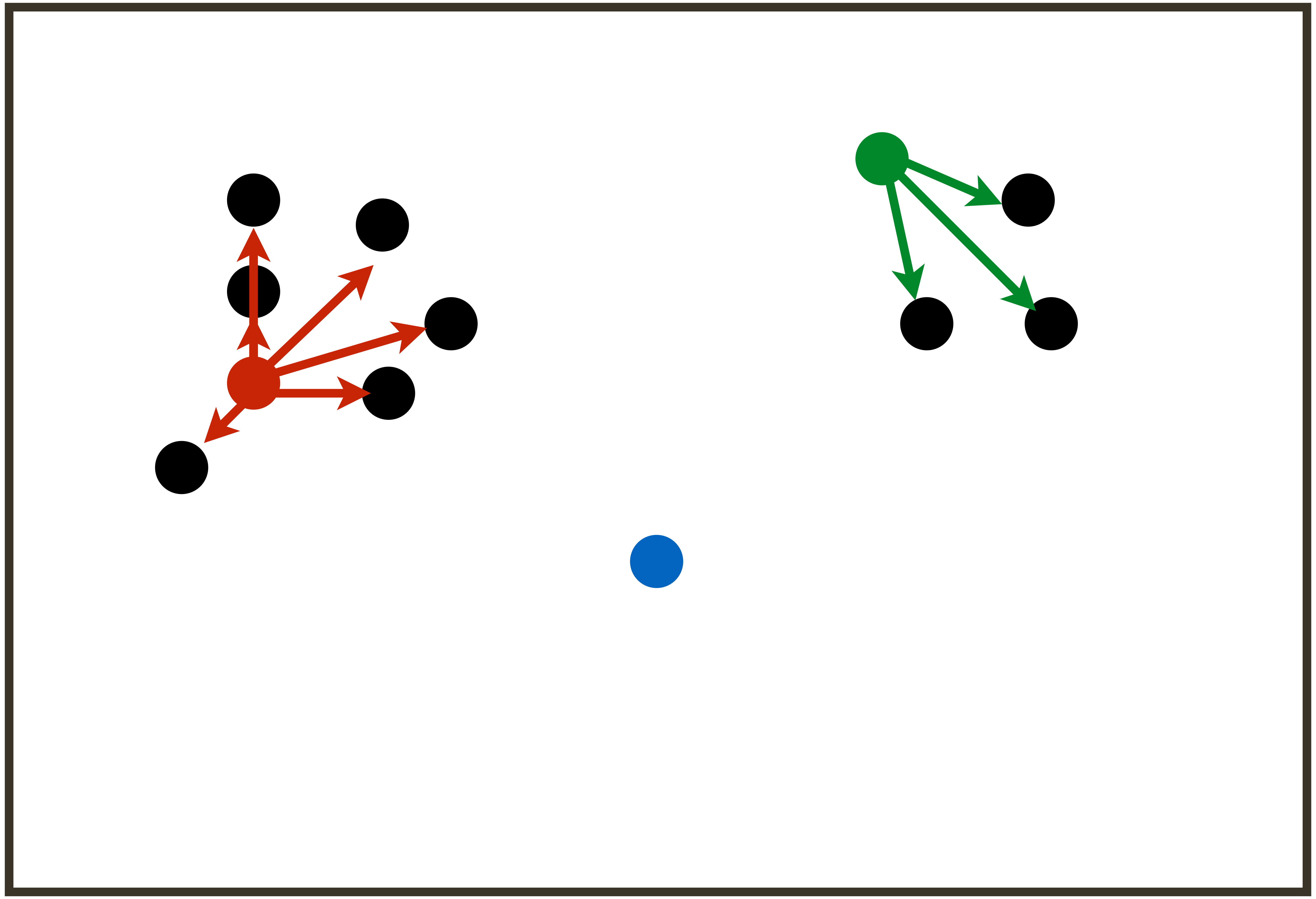
Example: VLAD

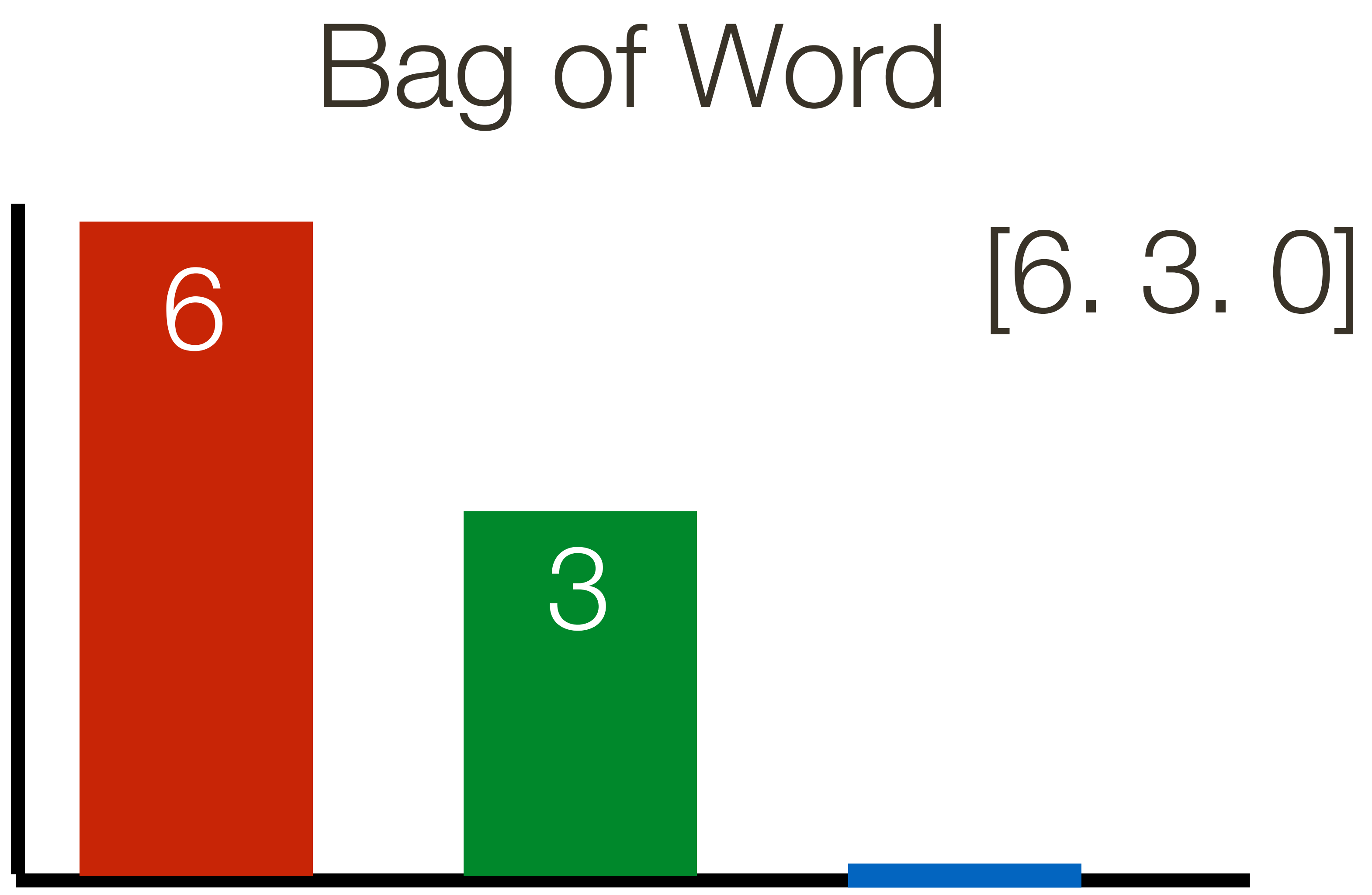


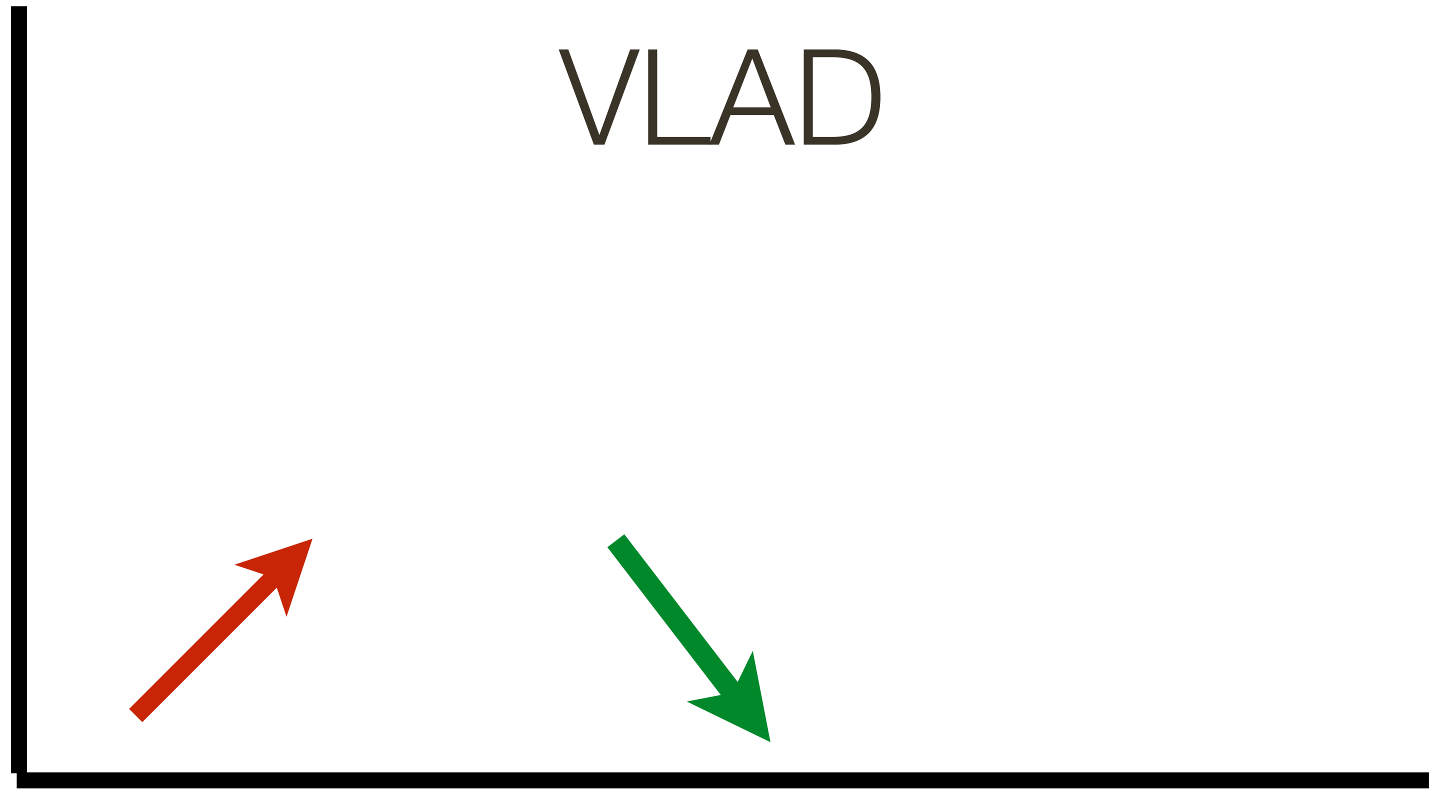
Bag of Word



Example: VLAD







VLAD (Vector of Locally Aggregated Descriptors)

The dimensionality of a **VLAD** descriptor is Kd

— K : number of codewords

— d : dimensionality of the local descriptor

VLAD characterizes the distribution of local descriptors with respect to the codewords

Back to **Classification**

Decision Tree

A **decision tree** is a simple non-linear parametric classifier

Consists of a tree in which each internal node is associated with a feature test

A data point starts at the root and recursively proceeds to the child node determined by the feature test, until it reaches a leaf node

The leaf node stores a class label or a probability distribution over class labels

Decision Tree

Learning a decision tree from a training set involves selecting an efficient sequence of feature tests

Example: Waiting for a restaurant table

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

Decision Tree

Which test is more helpful?

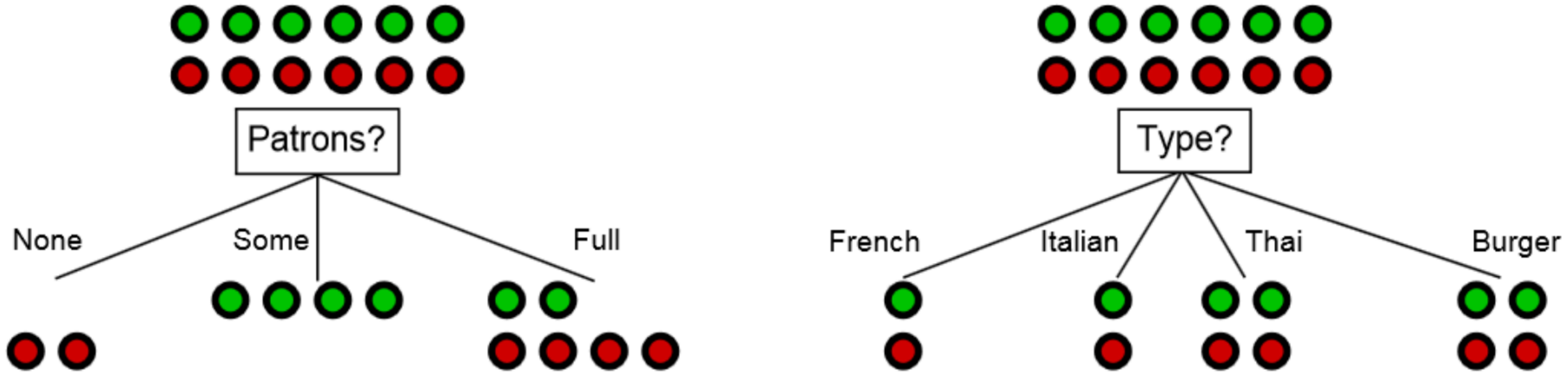


Figure credit: Russell and Norvig (3rd ed.)

Decision Tree

The **entropy** of a set S of data samples is defined as

$$H(S) = -\sum_{c \in C} p(c) \log(p(c))$$

where C is the set of classes represented in S, and $p(c)$ is the empirical distribution of class c in S

Entropy is highest when data samples are spread equally across all classes, and zero when all data samples are from the same class.

Decision Tree

In general we try to select the feature test that maximizes the i**nformation gain**:

$$I = H(S) - \sum_{i \in \{children\}} \frac{|S^i|}{|S|} H(S^i)$$

In the previous example, the information gains of the two candidate tests are:

$$I_{Patrons} = 0.541 \qquad\qquad I_{Type} = 0$$

So we choose the ‘Patrons’ test.

Decision Tree

Following this construction procedure we obtain the final decision tree:

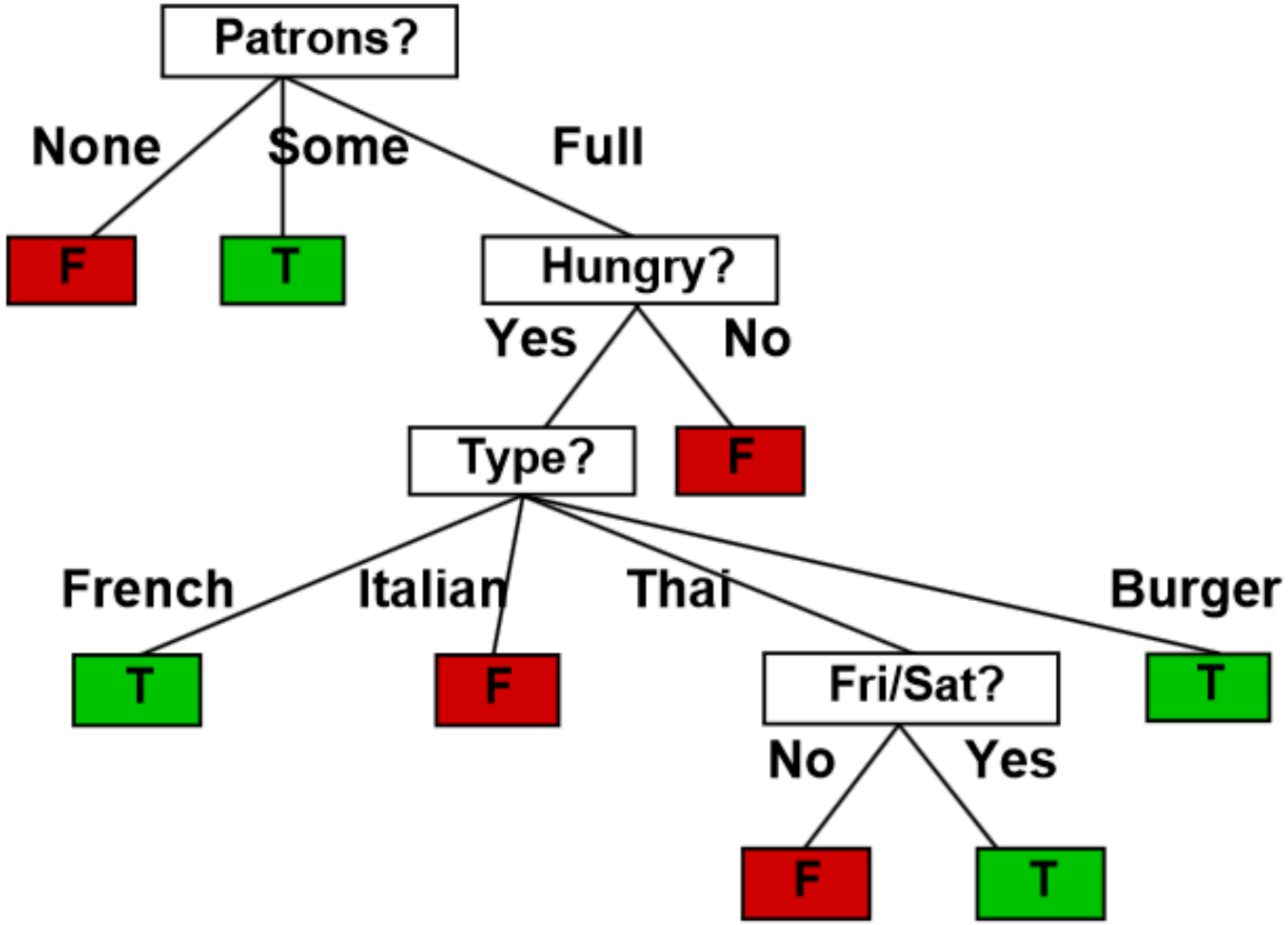


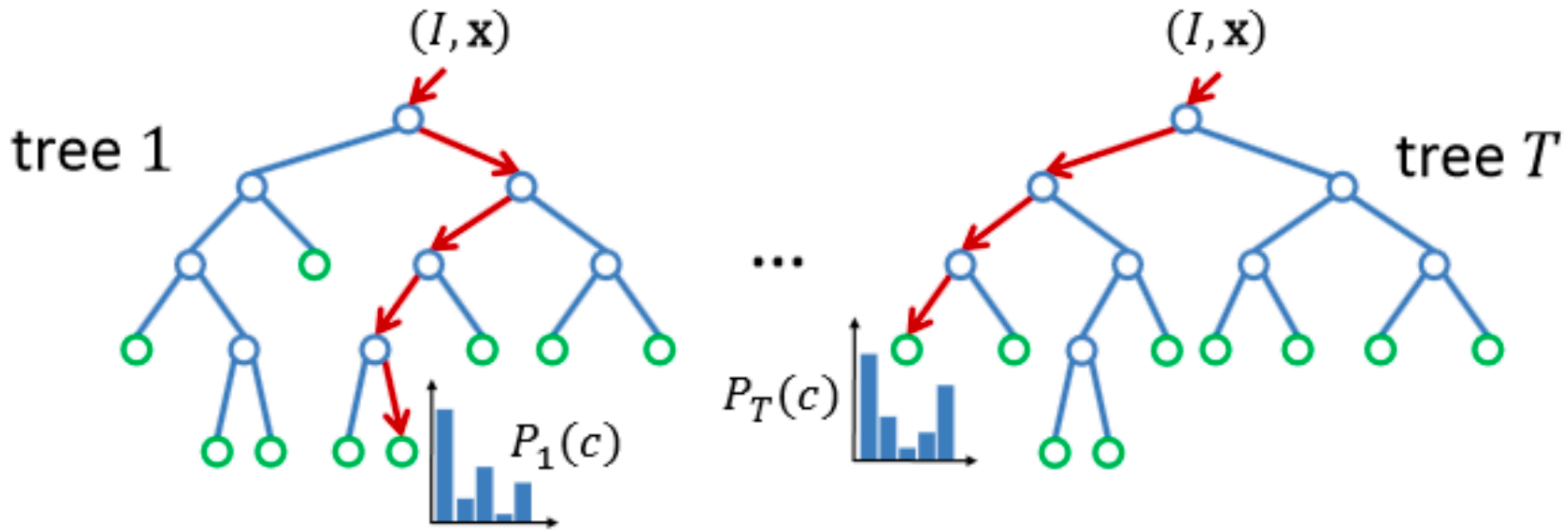
Figure credit: Russell and Norvig (3rd ed.)

Decision Tree

A **random forest** is an ensemble of decision trees.

Randomness is incorporated via training set sampling and/or generation of the candidate binary tests

The prediction of the random forest is obtained by averaging over all decision trees.



Forsyth & Ponce (2nd ed.) Figure 14.19. Original credit: J. Shotton et al., 2011

Example 1: Kinect

Kinect allows users of Microsoft’s Xbox 360 console to interact with games using natural body motions instead of a traditional handheld controller. The pose (joint positions) of the user is predicted using a random forest trained on depth features.

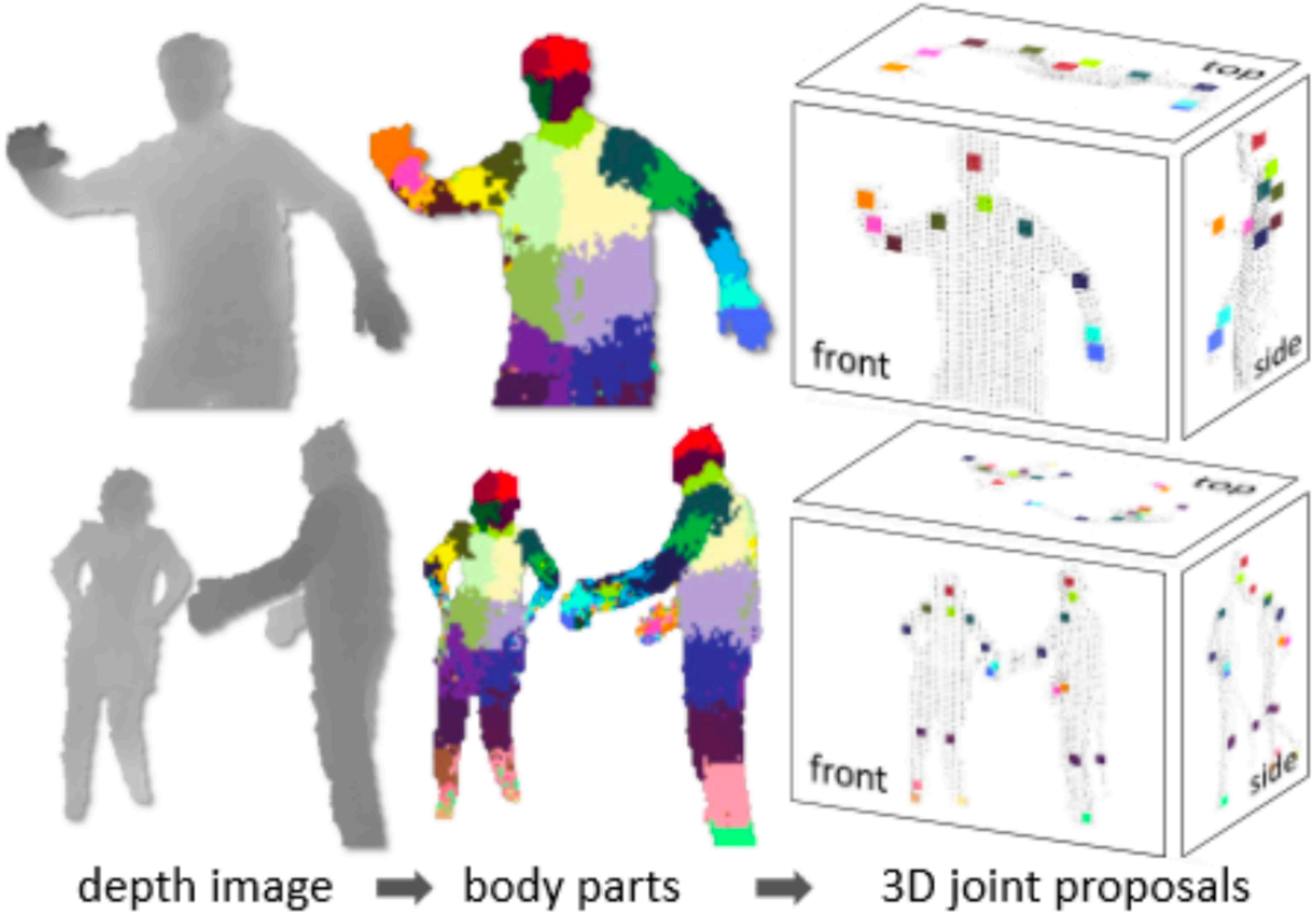
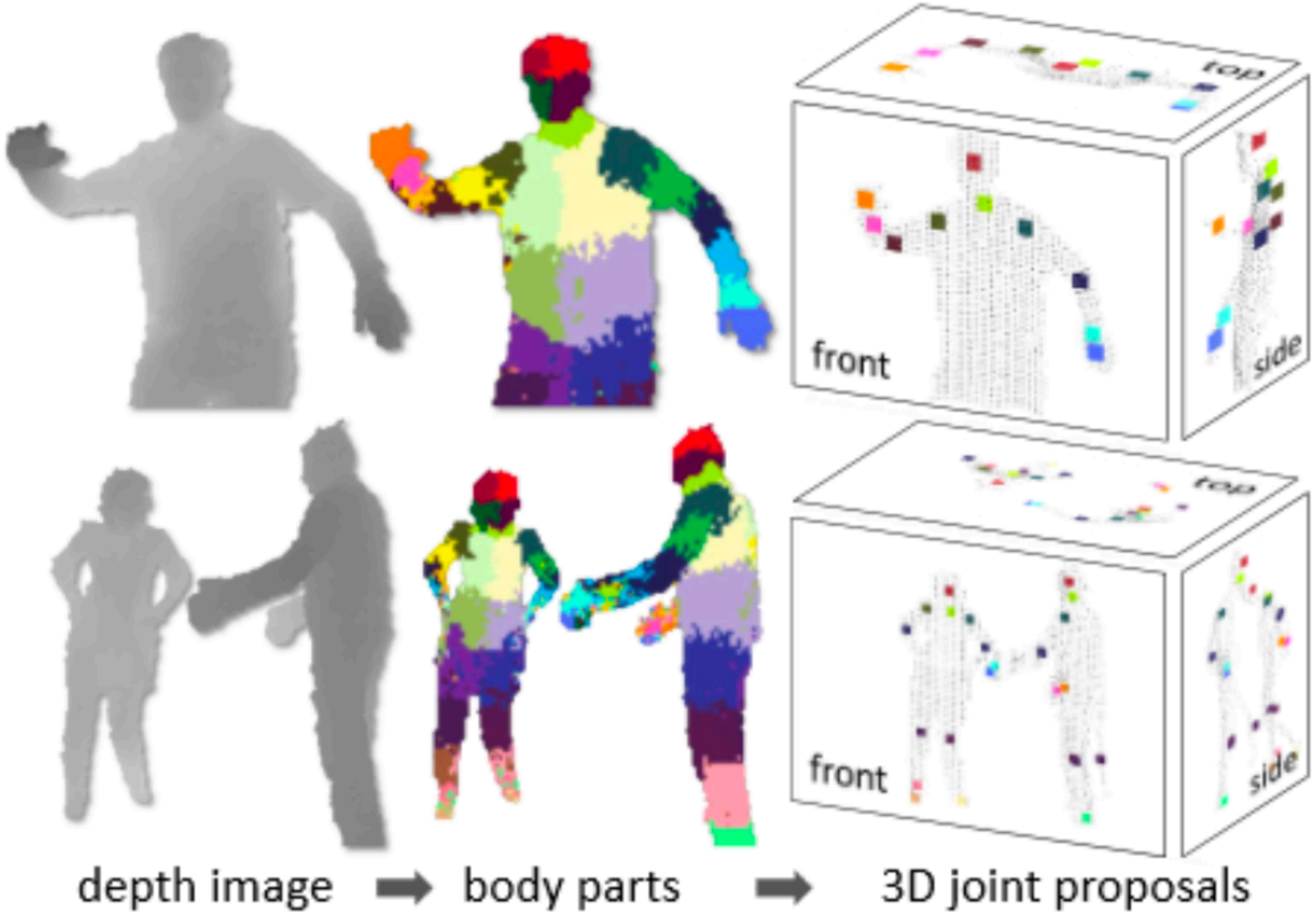


Figure credit: J. Shotton et al., 2011

Example 1: Kinect

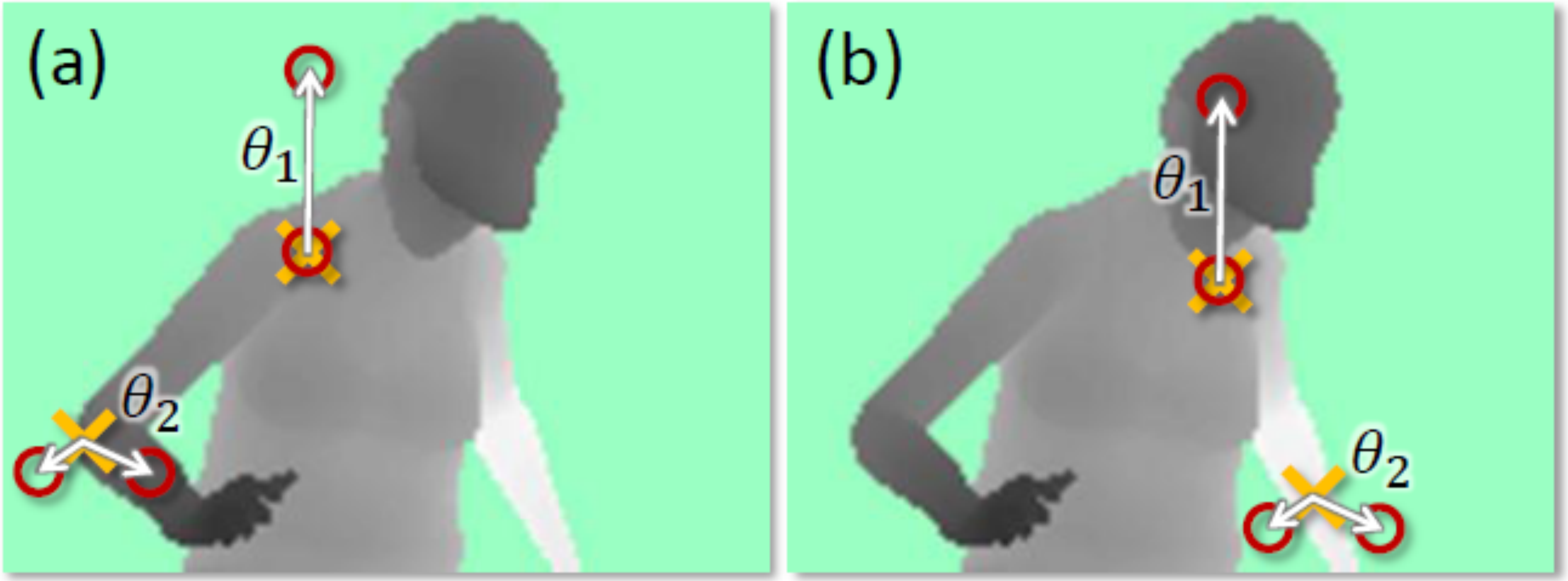
Kinect allows users of Microsoft's Xbox 360 console to interact with games using natural body motions instead of a traditional handheld controller. The pose (joint positions) of the user is predicted using a random forest trained on depth features.



Jamie Shotton

Figure credit: J. Shotton et al., 2011

Example 1: Kinect



$$f_\theta(I, \mathbf{x}) = d_I\left(\mathbf{x} + \frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{v}}{d_I(\mathbf{x})}\right)$$

Figure credit: J. Shotton et al., 2011

Example 1: Kinect

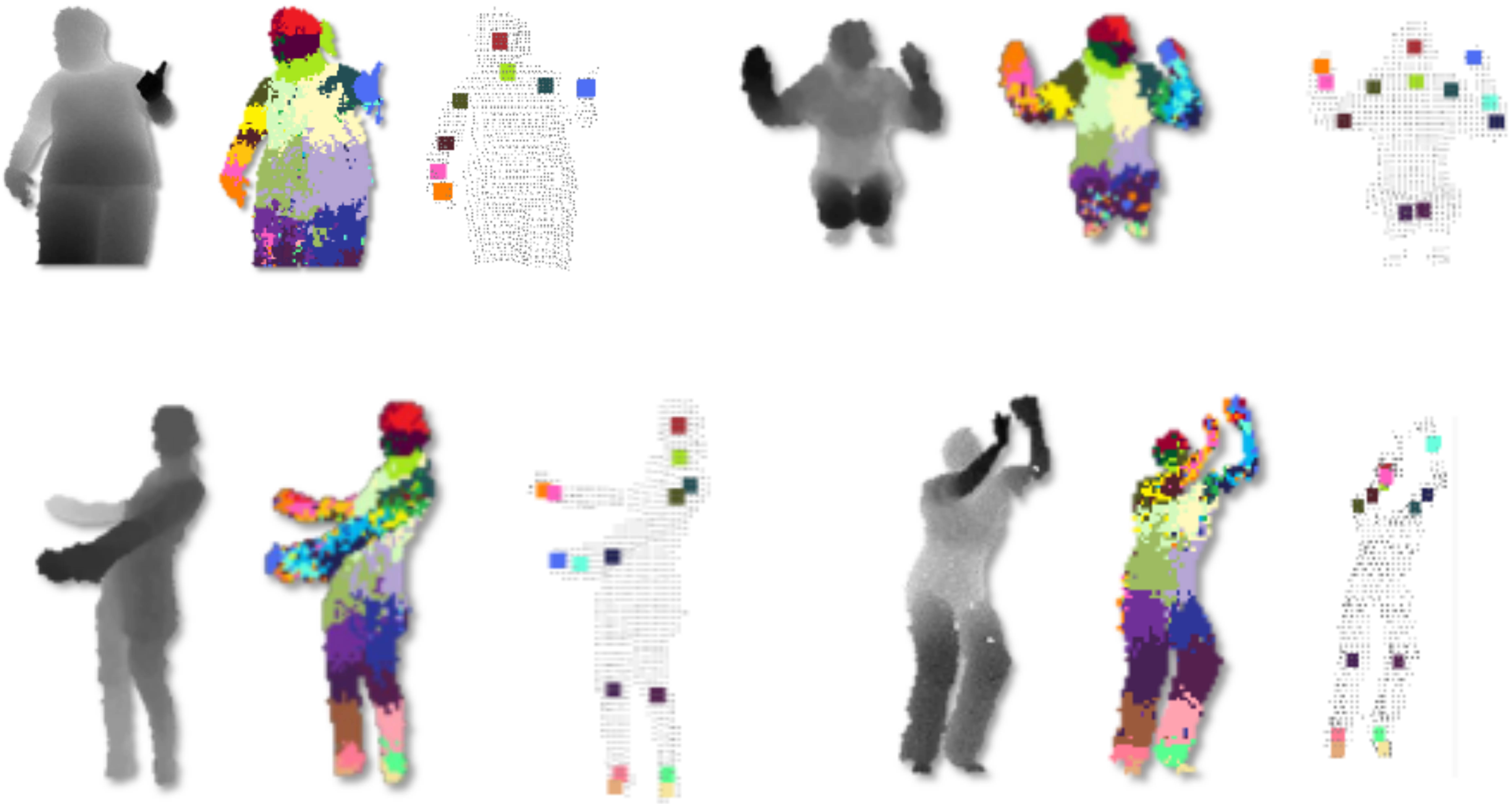


Figure credit: J. Shotton et al., 2011

Combining **Classifiers**

One common strategy to obtain a better classifier is to combine multiple classifiers.

A simple approach is to train an ensemble of independent classifiers, and average their predictions.

Boosting is another approach.
— Train an ensemble of classifiers sequentially.
— Bias subsequent classifiers to correctly predict training examples that previous classifiers got wrong.
— The final boosted classifier is a weighted combination of the individual classifiers.

Combining Classifiers: **Boosting**

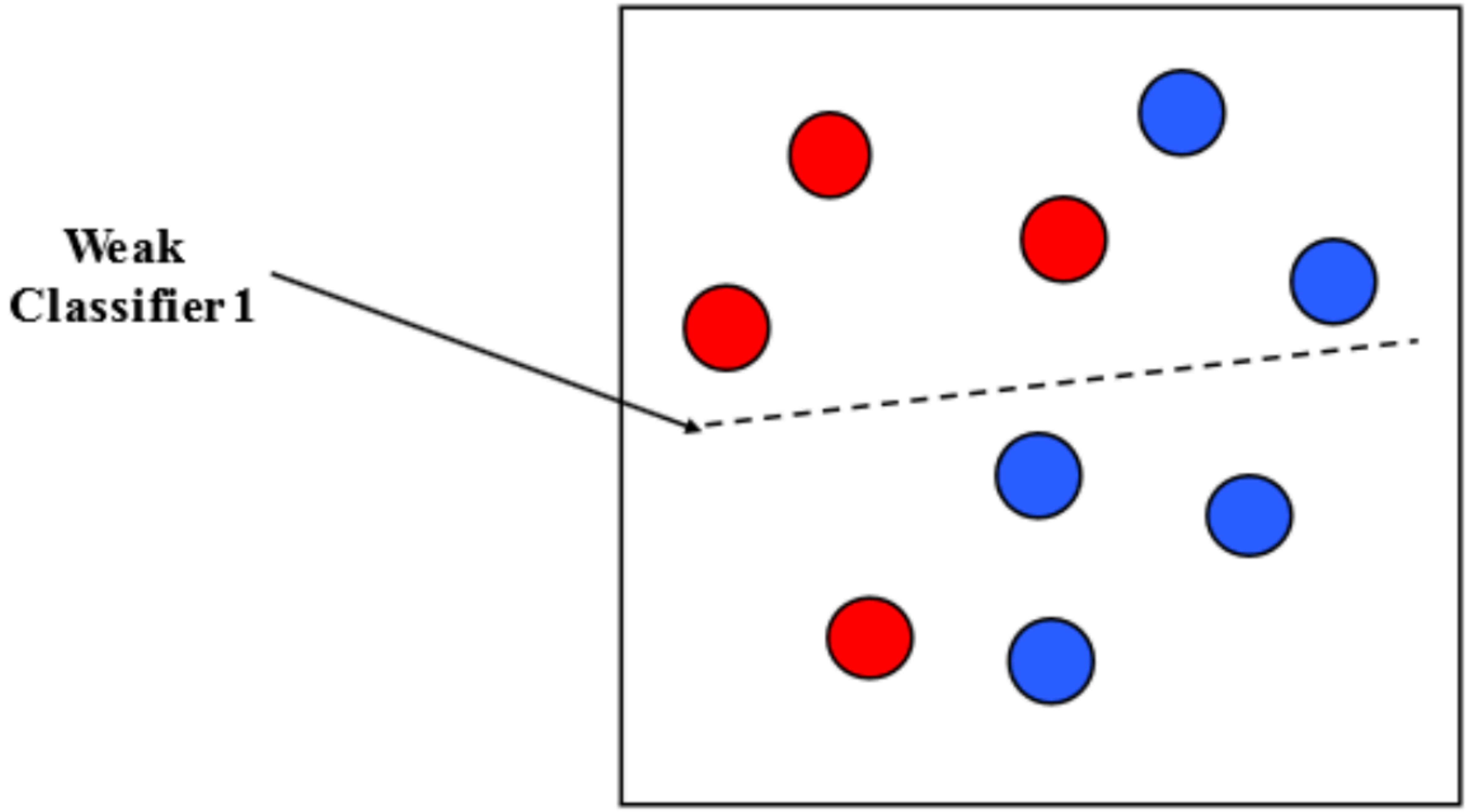


Figure credit: Paul Viola

Combining Classifiers: **Boosting**

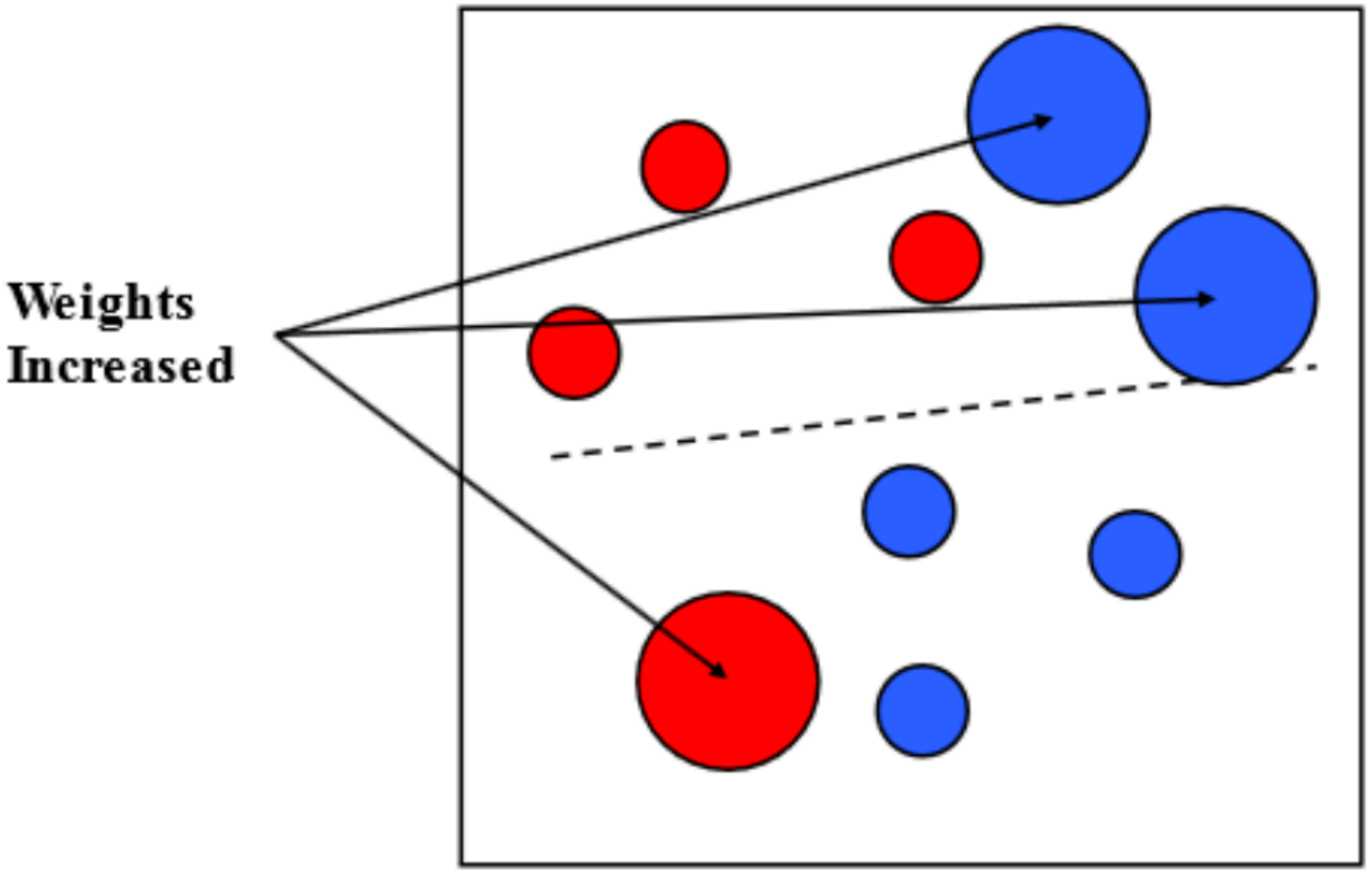


Figure credit: Paul Viola

Combining Classifiers: **Boosting**

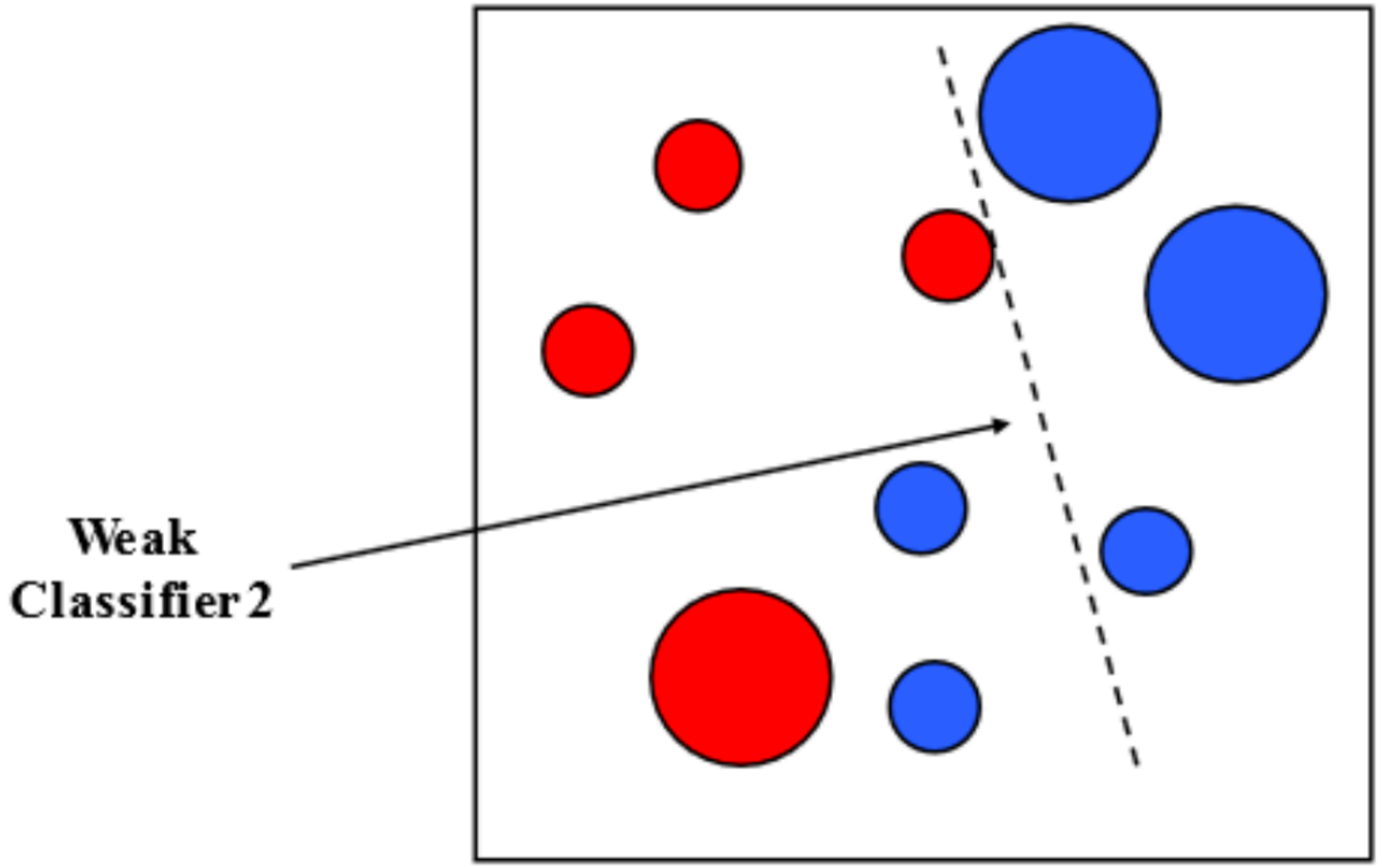


Figure credit: Paul Viola

Combining Classifiers: **Boosting**

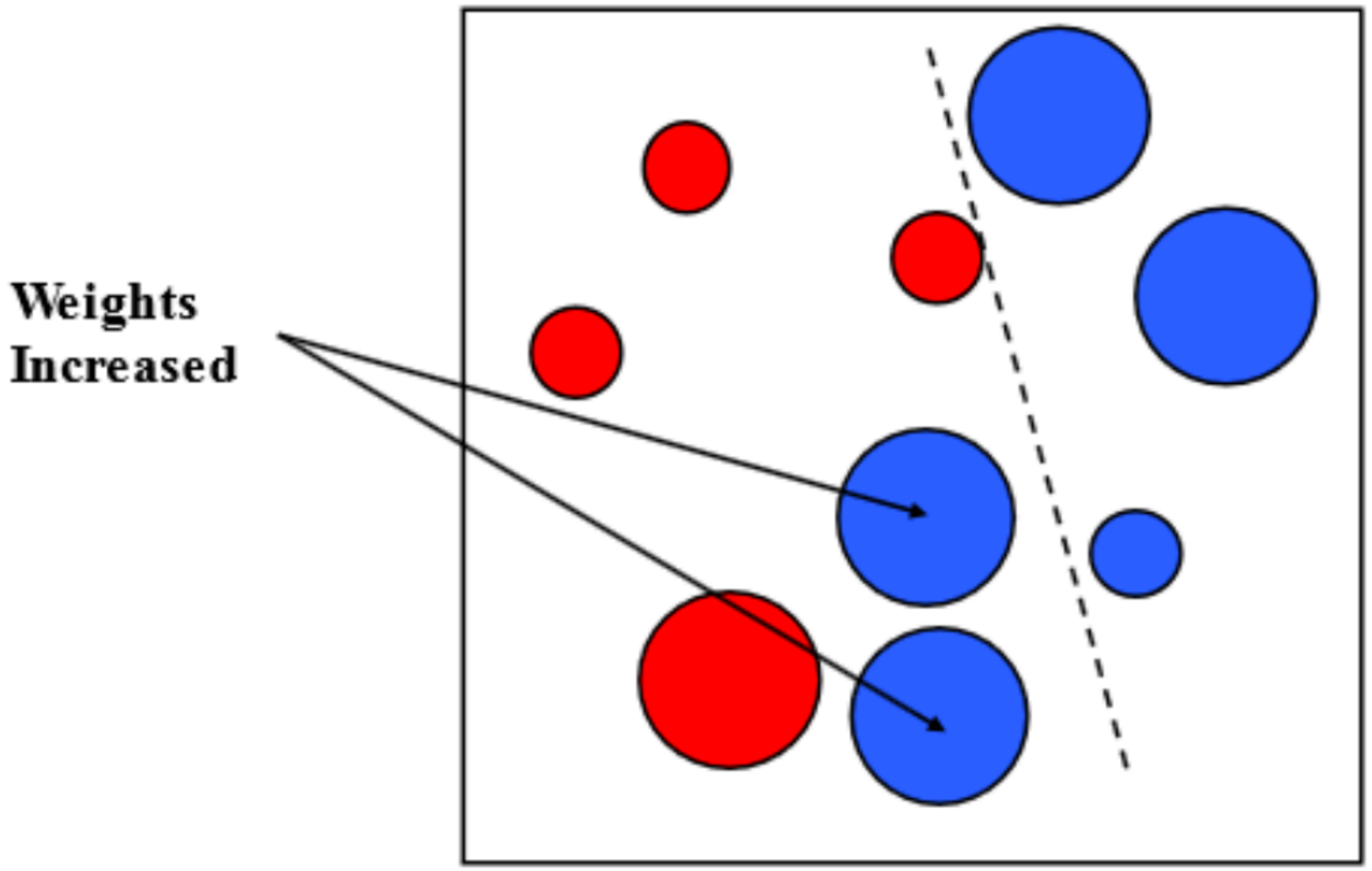


Figure credit: Paul Viola

Combining Classifiers: **Boosting**

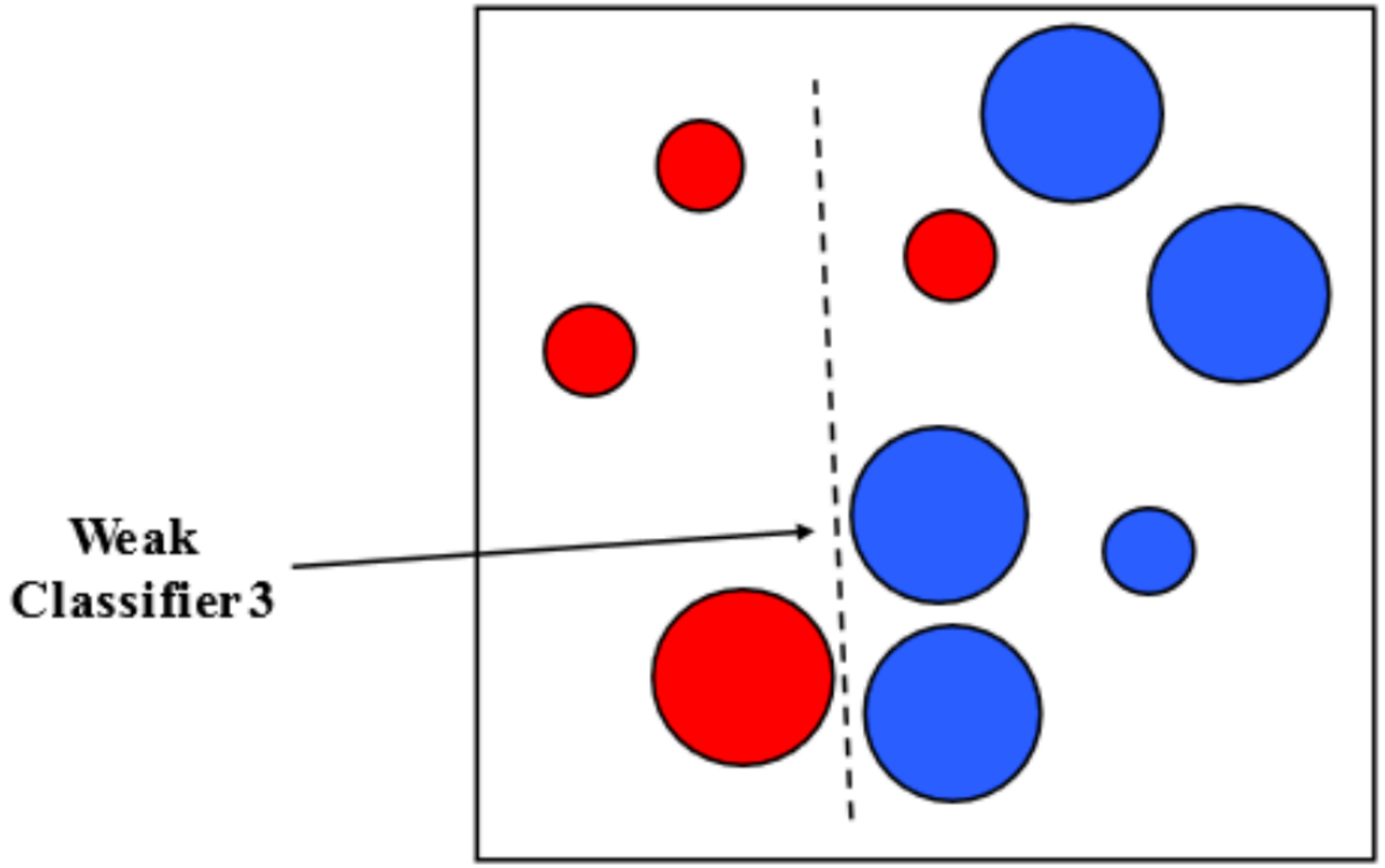


Figure credit: Paul Viola

Combining Classifiers: **Boosting**

Final classifier is a combination of weak classifiers

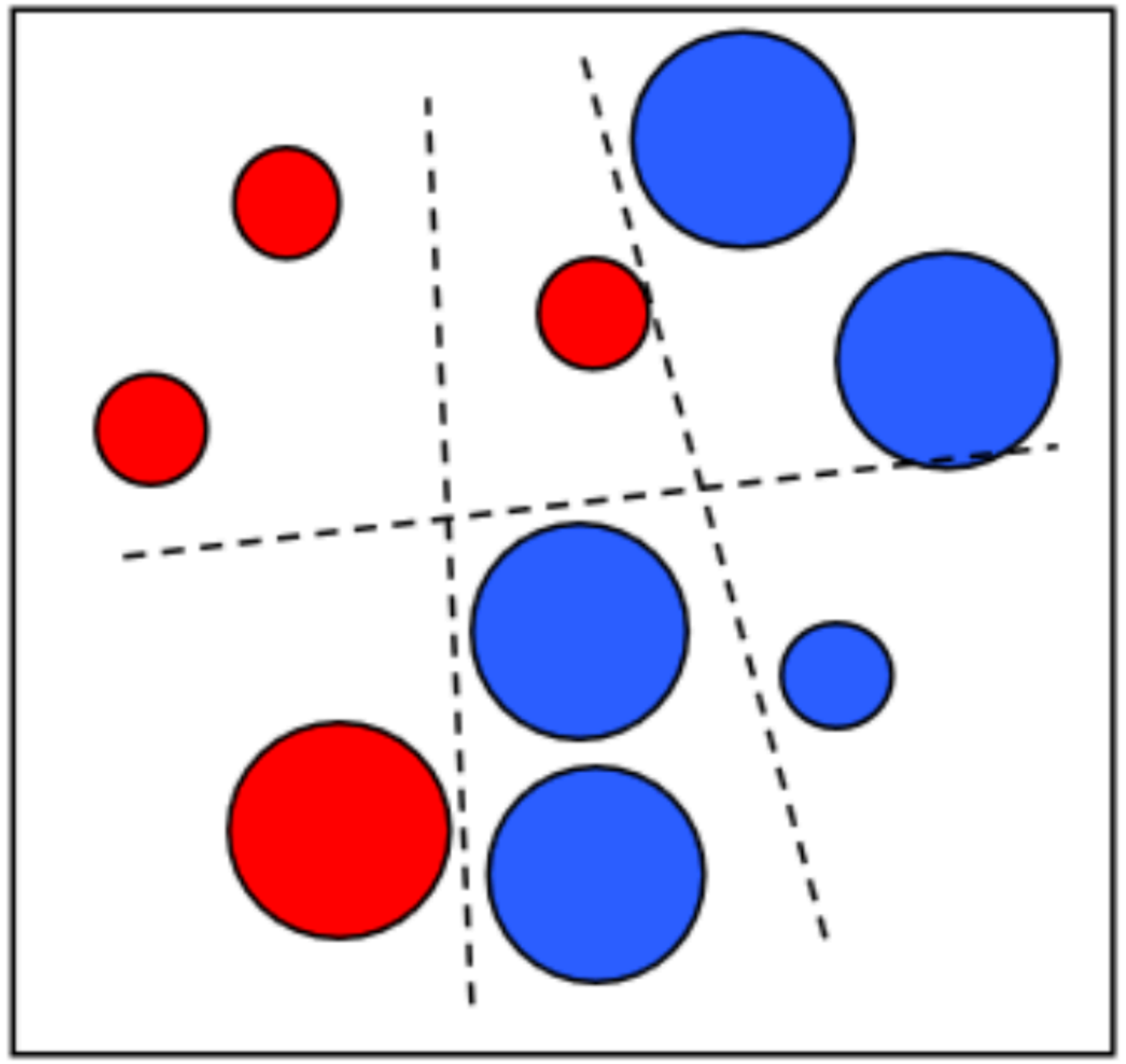


Figure credit: Paul Viola

Summary

A **decision tree** passes a data point through a sequence of feature tests. A random forest is an ensemble of decision trees.

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

The **spatial pyramid** partitions the image and counts visual words within each grid box; this is repeated at multiple levels