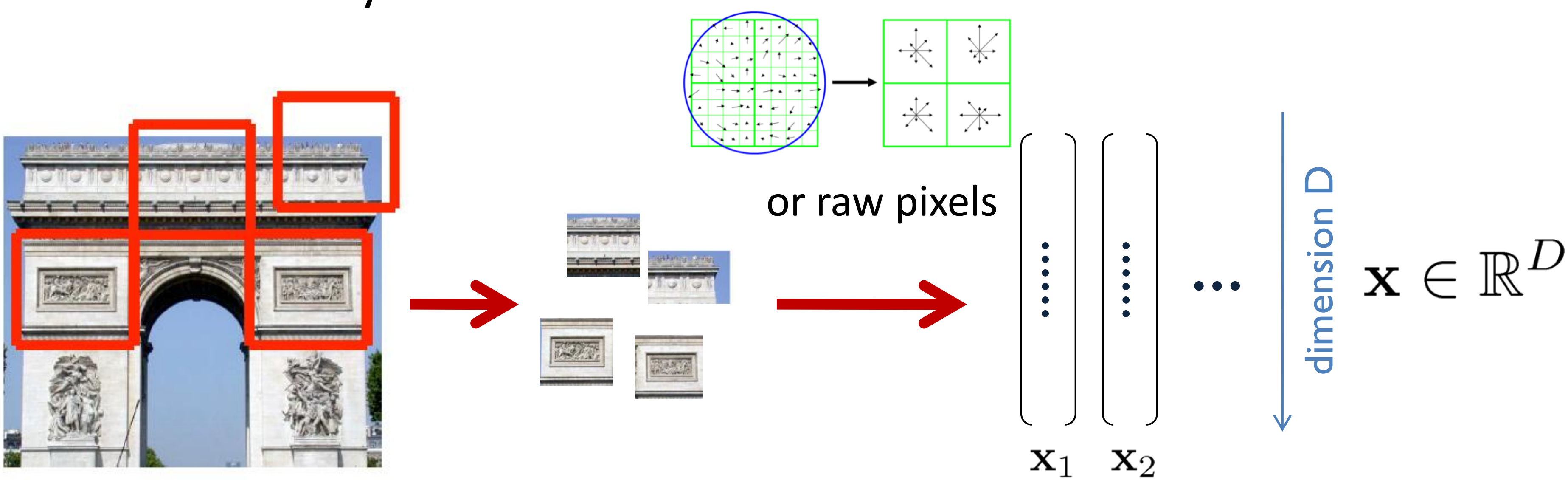
Visual Codebook

Tae-Kyun Kim
Sidney Sussex College

Visual Words

- Visual words are base elements to describe an image.
- Interest points are detected from an image
  - Corners, Blob detector, SIFT detector
- Image patches are represented by descriptor
  - SIFT (Scale-Invariant Feature Transform) or Raw pixel intensity

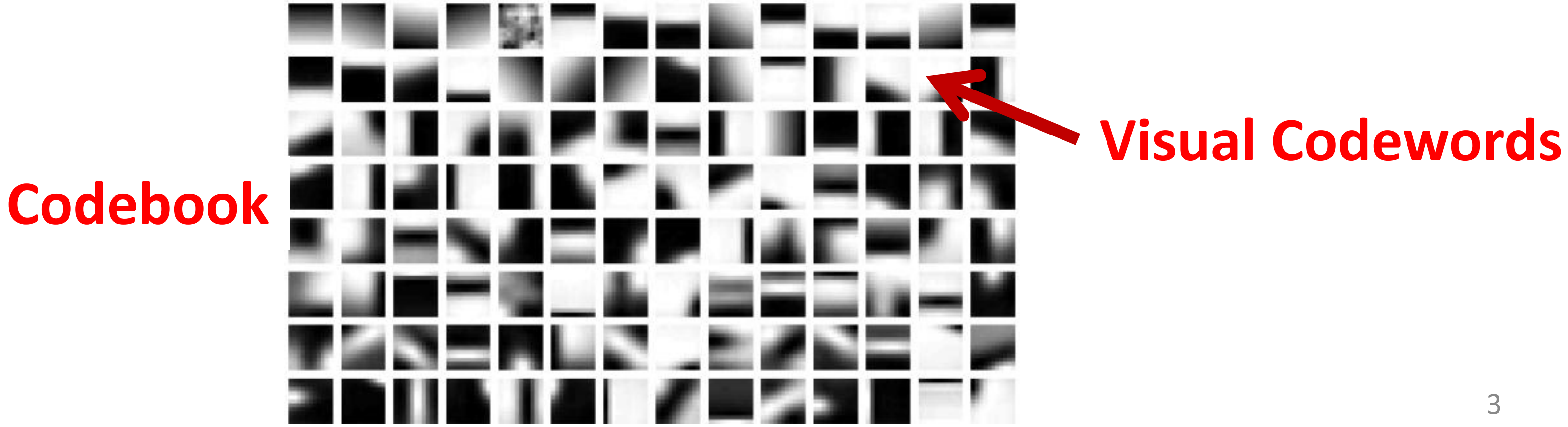


Building a Visual Codebook (Dictionary)

- Visual words (real-valued vectors) can be compared using Euclidean distance:

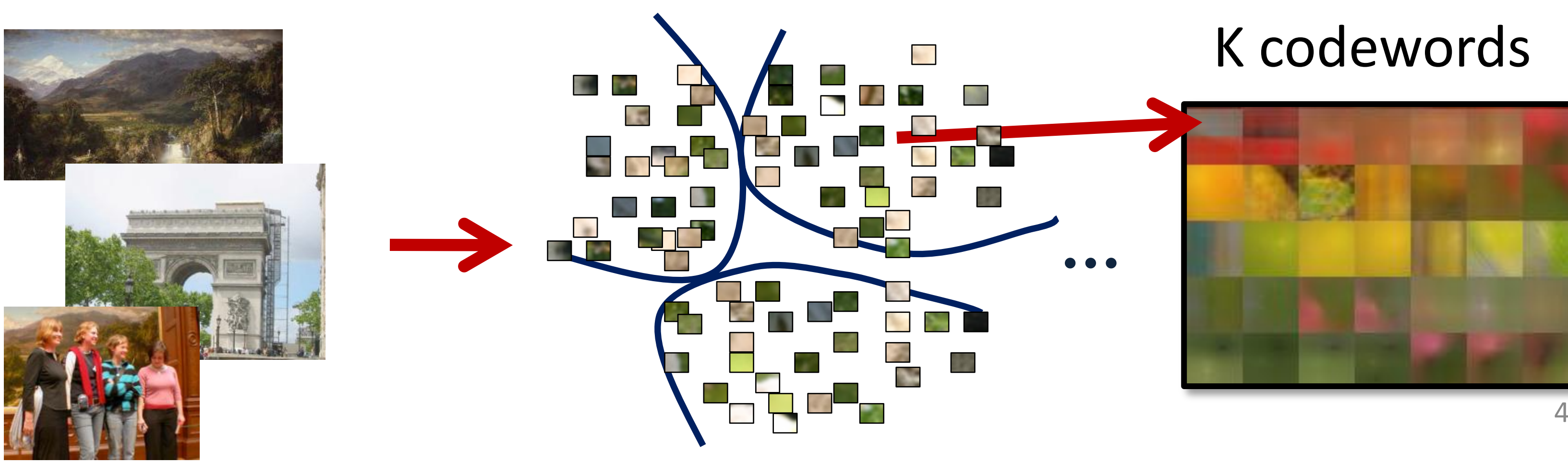
$$E(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_d (\mathbf{x}_1^d - \mathbf{x}_2^d)^2}$$

- These vectors are divided into groups which are similar, essentially clustered together, to form a codebook.



K-means Clustering

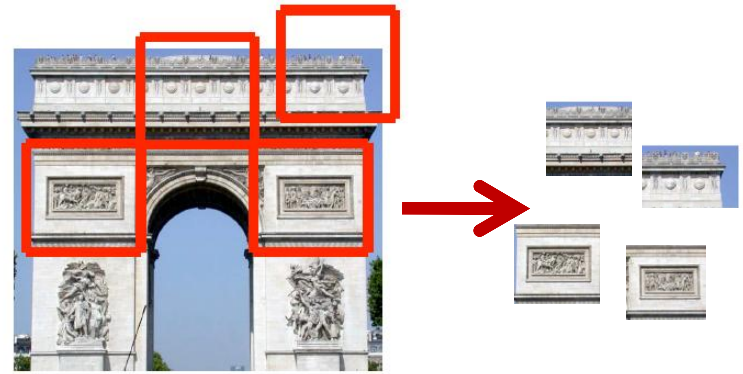
- The two steps repeated until no vector changes membership:
    1. Compute a cluster center for each cluster as the mean of the cluster members
    2. Reassign each data point to the cluster whose center is nearest
- The cluster centers (mean values) now form a visual dictionary.



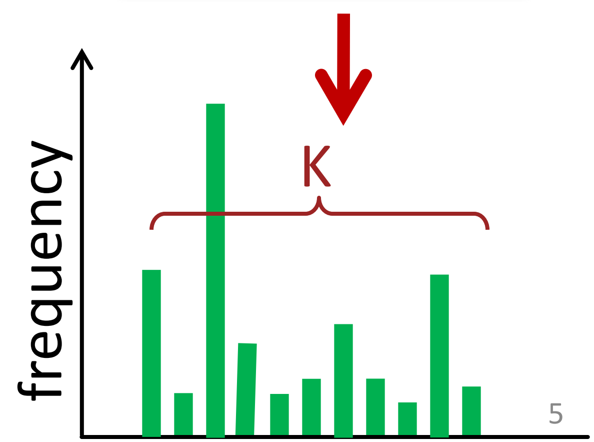
Histogram of Visual Words

- Every visual word is compared with codewords and assigned to the nearest codeword.

- Histogram bins are codewords and each bin counts the number of words assigned to the codeword.

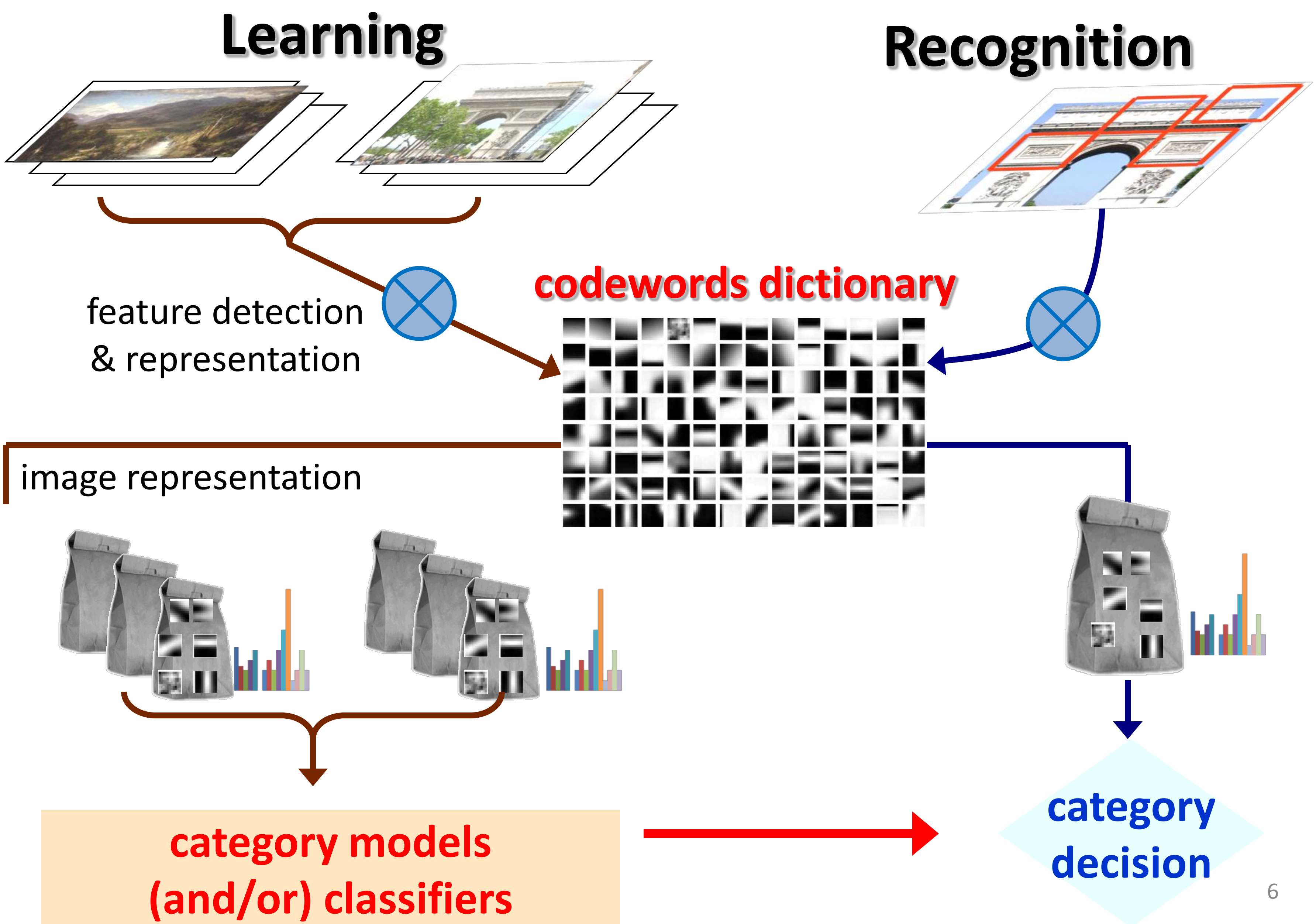


Nearest Neighbour Matching



"bag of words"

Categorisation

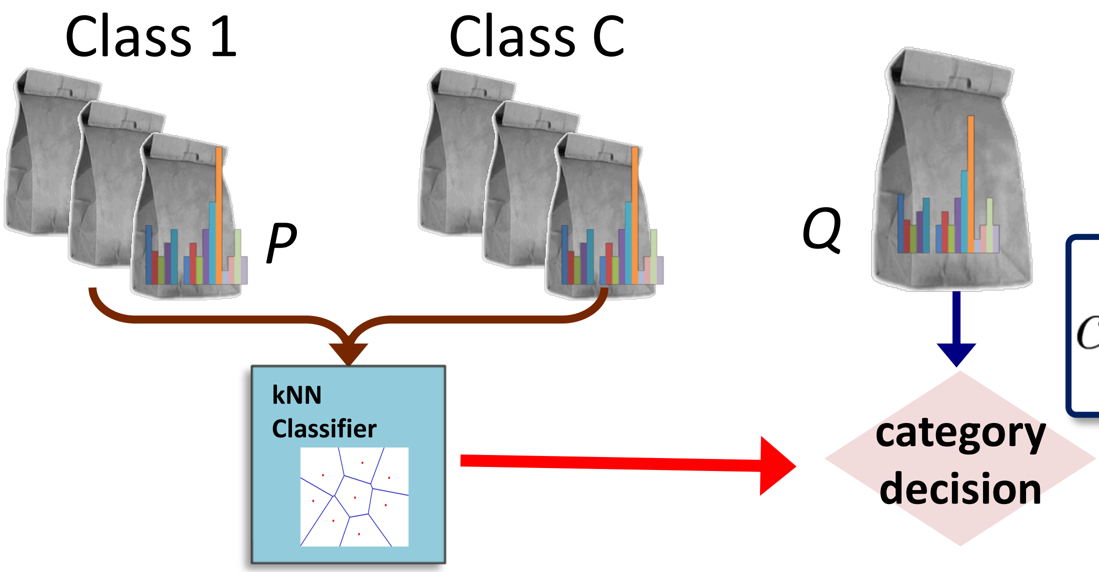


Categorisation

- Histogram matching of a pair of images *P* and *Q* is

$$\boxed{\mathcal{I}(P,Q) = \sum_{n=1}^{K} \min\left(H^{(n)}(P), H^{(n)}(Q)\right)}$$

- Based on HMK, we can use various classifiers such as kNN, SVM, Naïve-Bayes classifiers or pLSA, LDA.



Class assignment by

$$\boxed{class\left(\arg\max_{i} \mathcal{I}(P_i, Q)\right)}$$

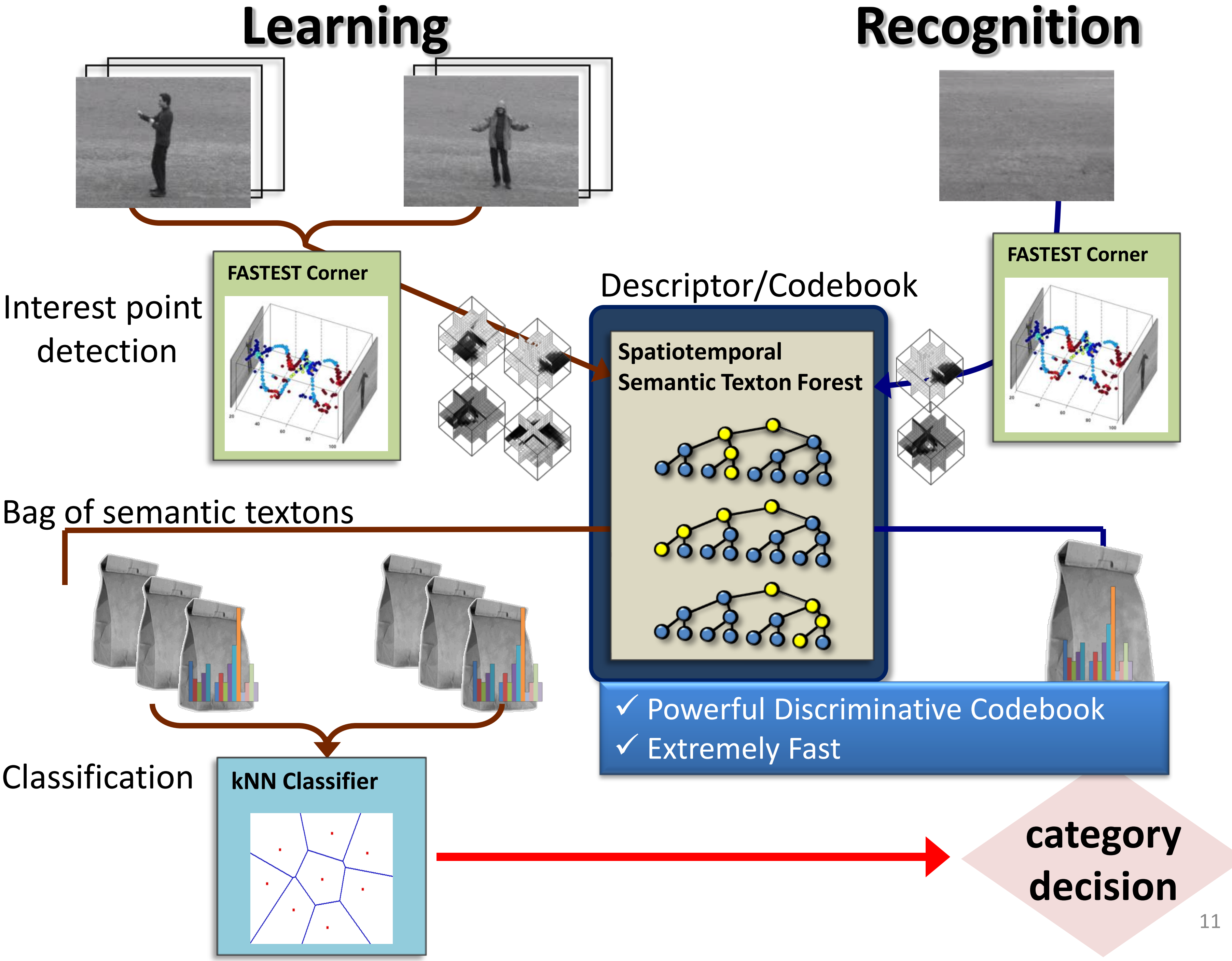
Summary of K-means Codebook

- The histogram is 1-D, a highly compact and robust representation of images.
- The histogram representation greatly facilitates modelling images and their categories.
- Codeword assignment (quantisation process) by K-means is time-demanding.
- K-means is an unsupervised learning method.

Case Study:
Video (action) Categorisation by RF Codebook

In collaboration with Tsz Ho Yu

Demo video: Action categorisation


Learning
Recognition
Interest point detection
FASTEST Corner
FASTEST Corner
Descriptor/Codebook
Spatiotemporal Semantic Texton Forest
Bag of semantic textons
✓ Powerful Discriminative Codebook
✓ Extremely Fast
Classification
kNN Classifier
category decision

Relation to the previous

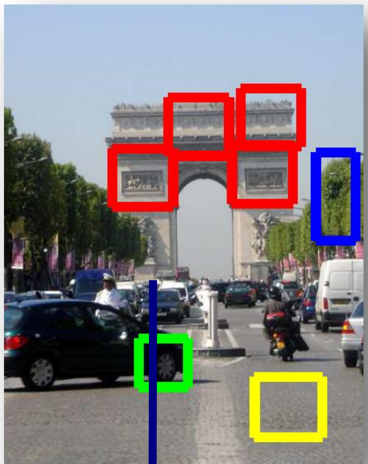
	2D → 3D	
Input	Images	Videos
Interest Point	Corners	3D corners
Descriptor	Patches	Space-time volumes
	SIFT	Raw pixels
Codebook	K-means	***Randomised Decision Forests***
Representation	Histogram of visual words	
Classifier	kNN classifier	

Skip

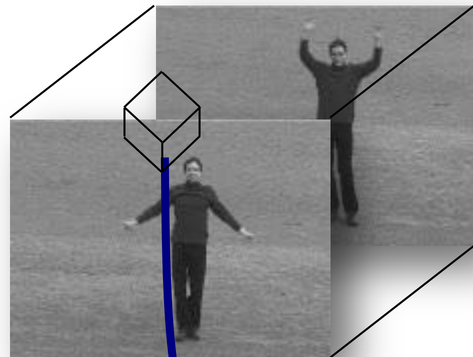
Relation to the previous

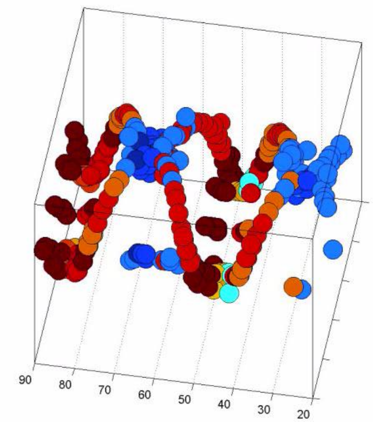
Patches ↔ Space-Time Volumes

Image (2D)

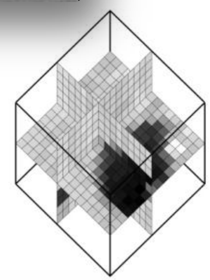


Video (3D)

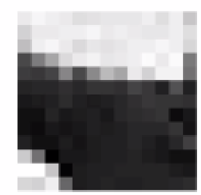


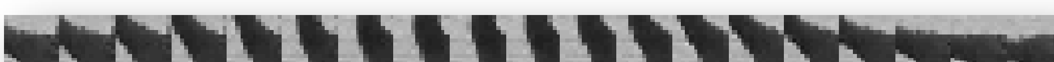


↔



or





Patch

Cuboid

$$\begin{bmatrix} \vdots \end{bmatrix} \quad \mathbf{x} \in \mathbb{R}^D$$

"visual words"

Data Set and Interest Point

Action data set

- http://www.nada.kth.se/cvap/actions/
- 25 subjects, 2391 sequences – the largest public action DB

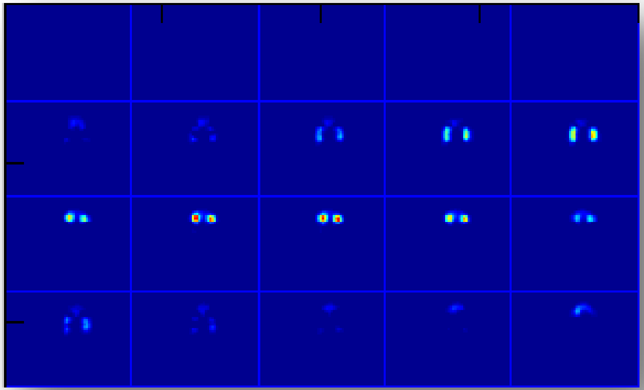
walking jogging running boxing Hand waving Hand clapping

s1
s2
s3
s4

Space-Time Interest Point

- In Dollar et al. 2005, separable linear filters are applied.
- The response function is

$$\boxed{R = (I * G * h_{ev})^2 + (I * G * h_{od})^2}$$



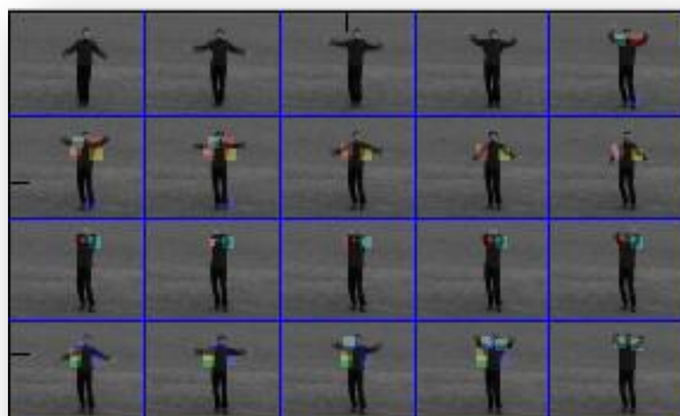
  - $G(x, y; \sigma)$ is **2D Gaussian smoothing** kernel
  - h_{ev} / h_{od} are a pair of **1D Gabor filters temporally applied** as

$$h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2} \quad h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$$

Space-Time Interest Point

- For multiple scales, it runs the detector over a set of spatial and temporal scales.

- Interest points are detected as **local maxima** of the response-function.

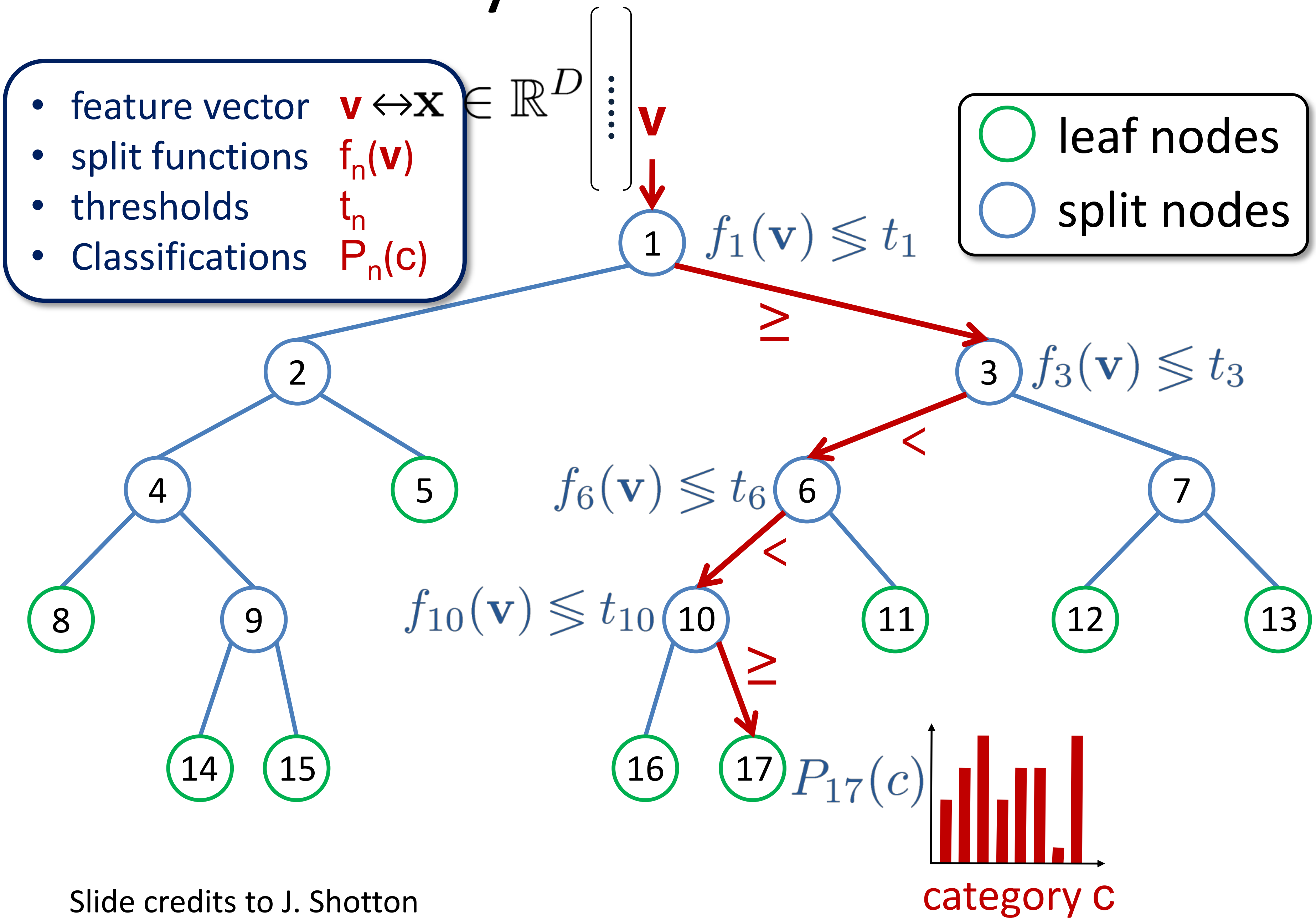


Matlab Demo:

Space-Time Interest Points

Background:
Randomised Decision Forest

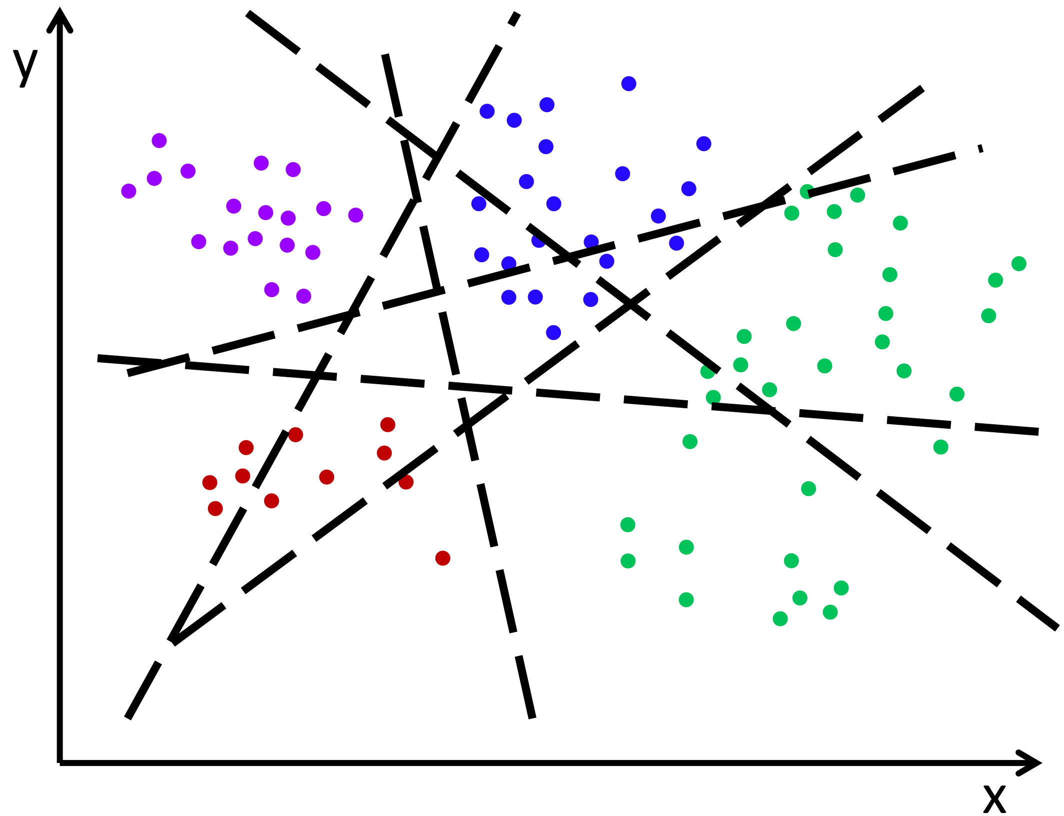
Binary Decision Trees



Slide credits to J. Shotton

Toy Learning Example

- **Try several lines, chosen at random**
- **Keep line that best separates data**
  - information gain
- **Recurse**

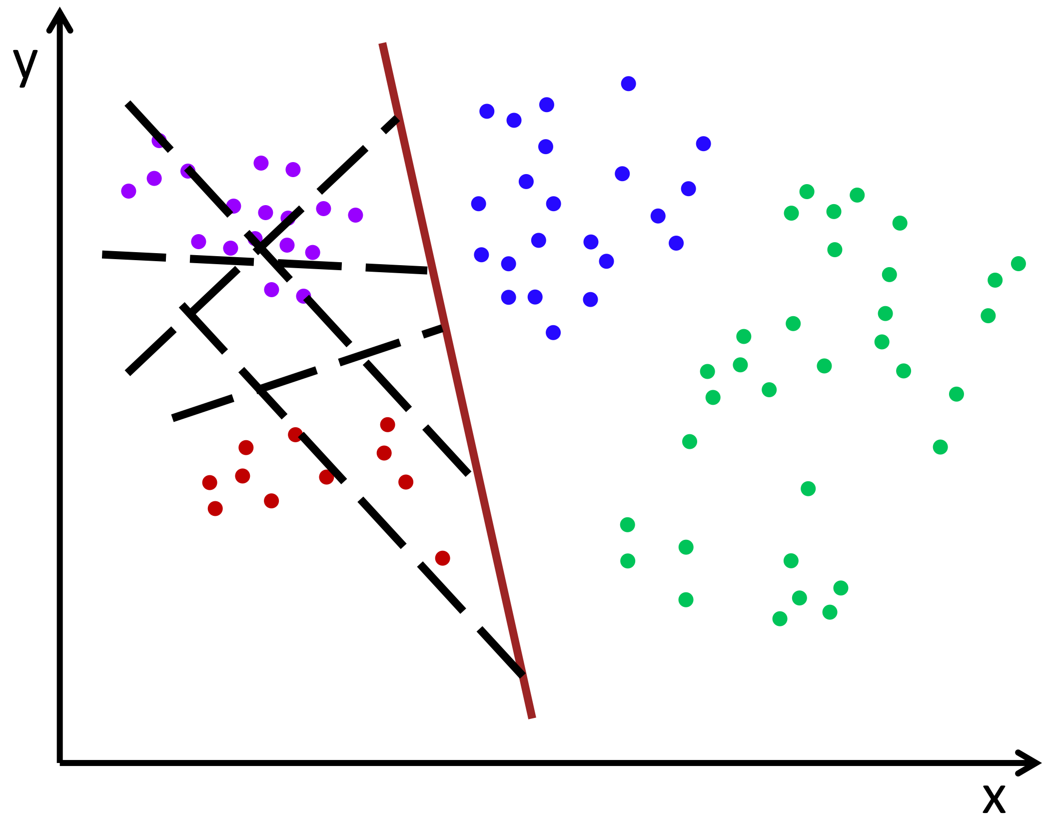


- feature vectors are x, y coordinates: $\mathbf{v} = [x, y]^T$
- split functions are lines with parameters a, b: $f_n(\mathbf{v}) = ax + by$
- threshold determines intercepts: t_n
- four classes: purple, blue, red, green

Slide credits to J. Shotton

Toy Learning Example

- **Try several lines, chosen at random**
- **Keep line that best separates data**
  - information gain
- **Recurse**

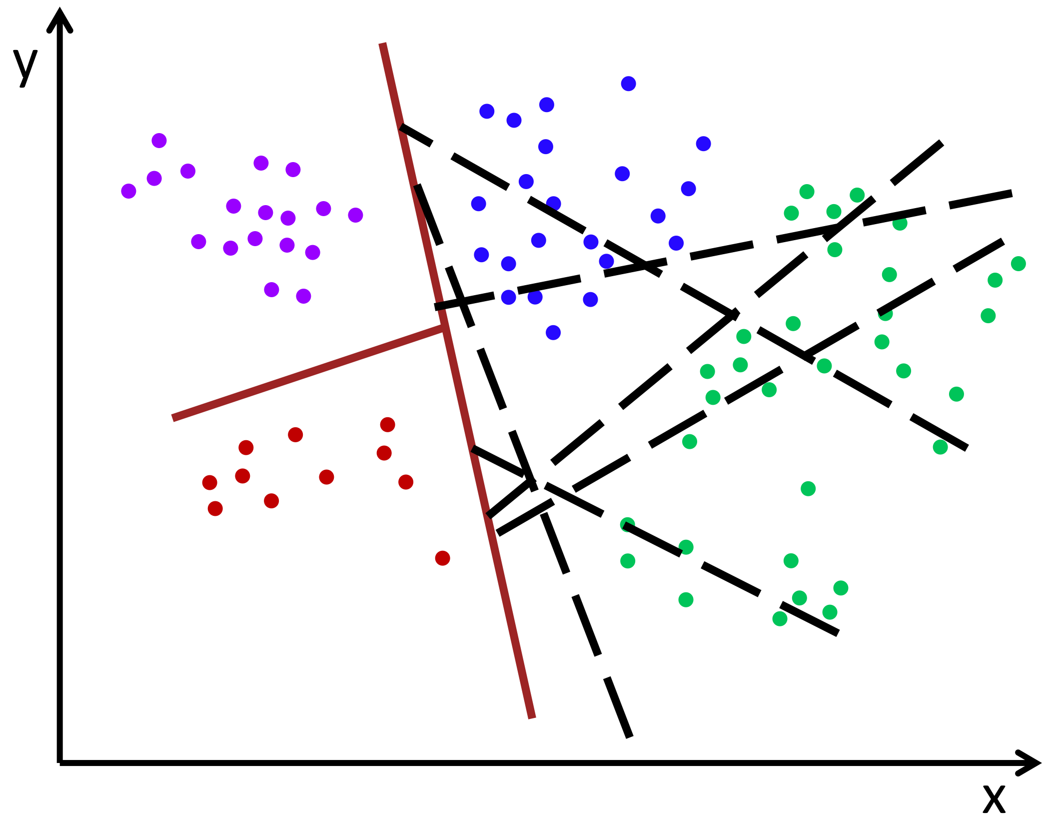


- feature vectors are x, y coordinates: $\mathbf{v} = [x, y]^T$
- split functions are lines with parameters a, b: $f_n(\mathbf{v}) = ax + by$
- threshold determines intercepts: t_n
- four classes: purple, blue, red, green

Slide credits to J. Shotton

Toy Learning Example

- **Try several lines, chosen at random**
- **Keep line that best separates data**
  - information gain
- **Recurse**

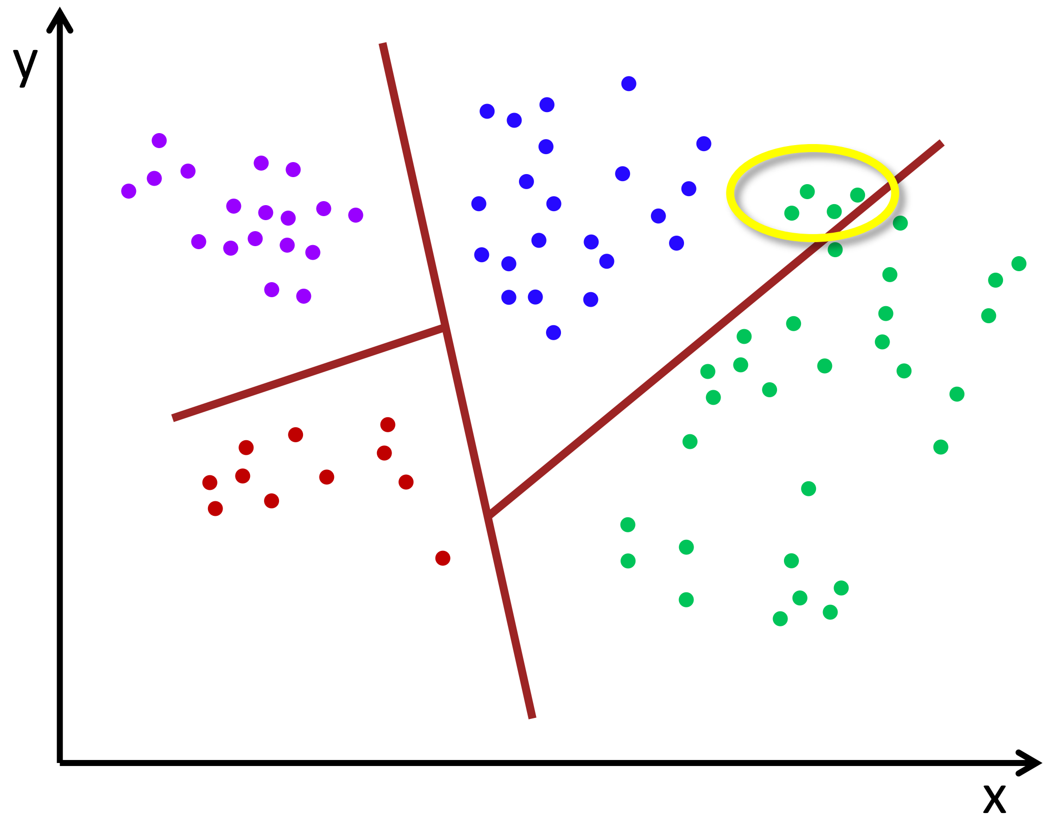


- feature vectors are x, y coordinates: $\mathbf{v} = [x, y]^T$
- split functions are lines with parameters a, b: $f_n(\mathbf{v}) = ax + by$
- threshold determines intercepts: t_n
- four classes: purple, blue, red, green

Slide credits to J. Shotton

Toy Learning Example

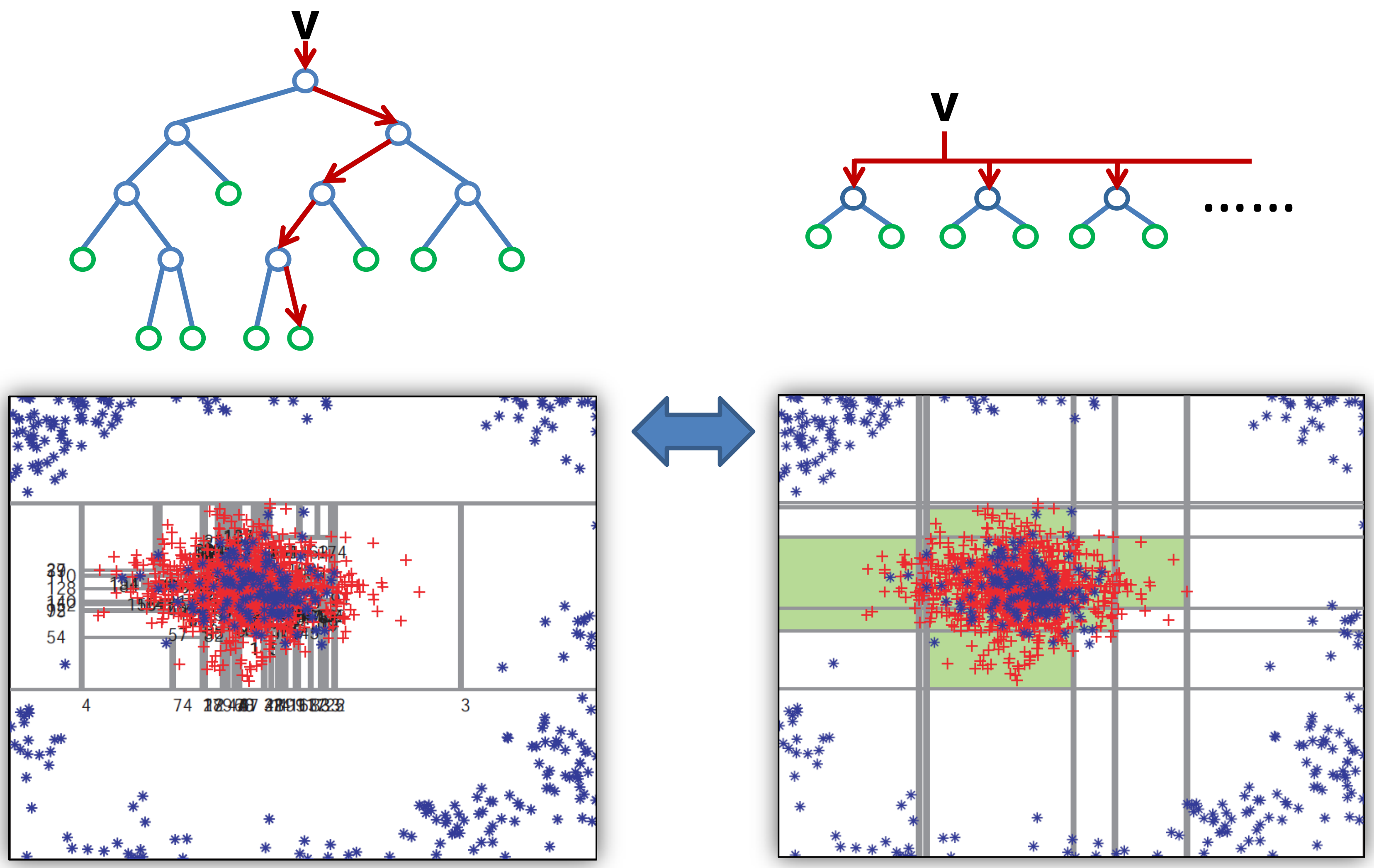
- Try several lines, chosen at random
- Keep line that best separates data
  - information gain
- Recurse



- feature vectors are x, y coordinates: $\mathbf{v} = [x, y]^T$
- split functions are lines with parameters a, b: $f_n(\mathbf{v}) = ax + by$
- threshold determines intercepts: t_n
- four classes: purple, blue, red, green

Slide credits to J. Shotton

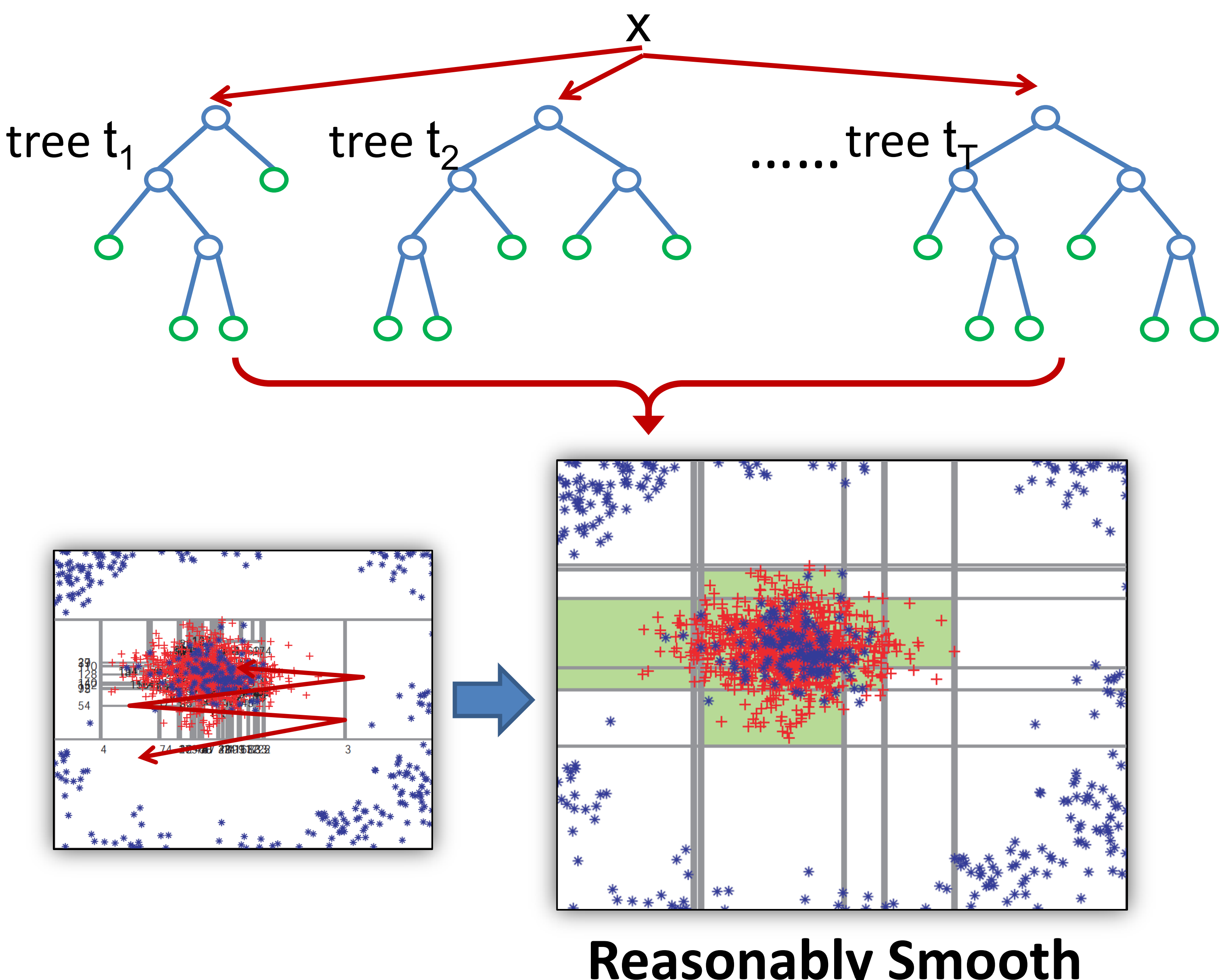
Generalisation



Overfit

Reasonably Smooth

Generalisation



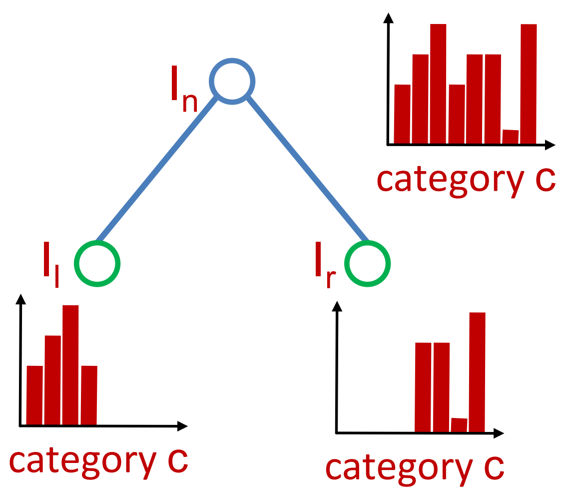
Randomized Tree Learning

- Train data: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \quad y_i \in \{1, 2, ..., C\}$
- Recursive algorithm
  - set I_n of training examples that reach node n is split:

$$
\begin{aligned}
\text{left split} \quad I_\mathrm{l} &= \{i \in I_n \mid f(\mathbf{v}_i) < t\} \\
\text{right split} \quad I_\mathrm{r} &= I_n \setminus I_\mathrm{l}
\end{aligned}
$$

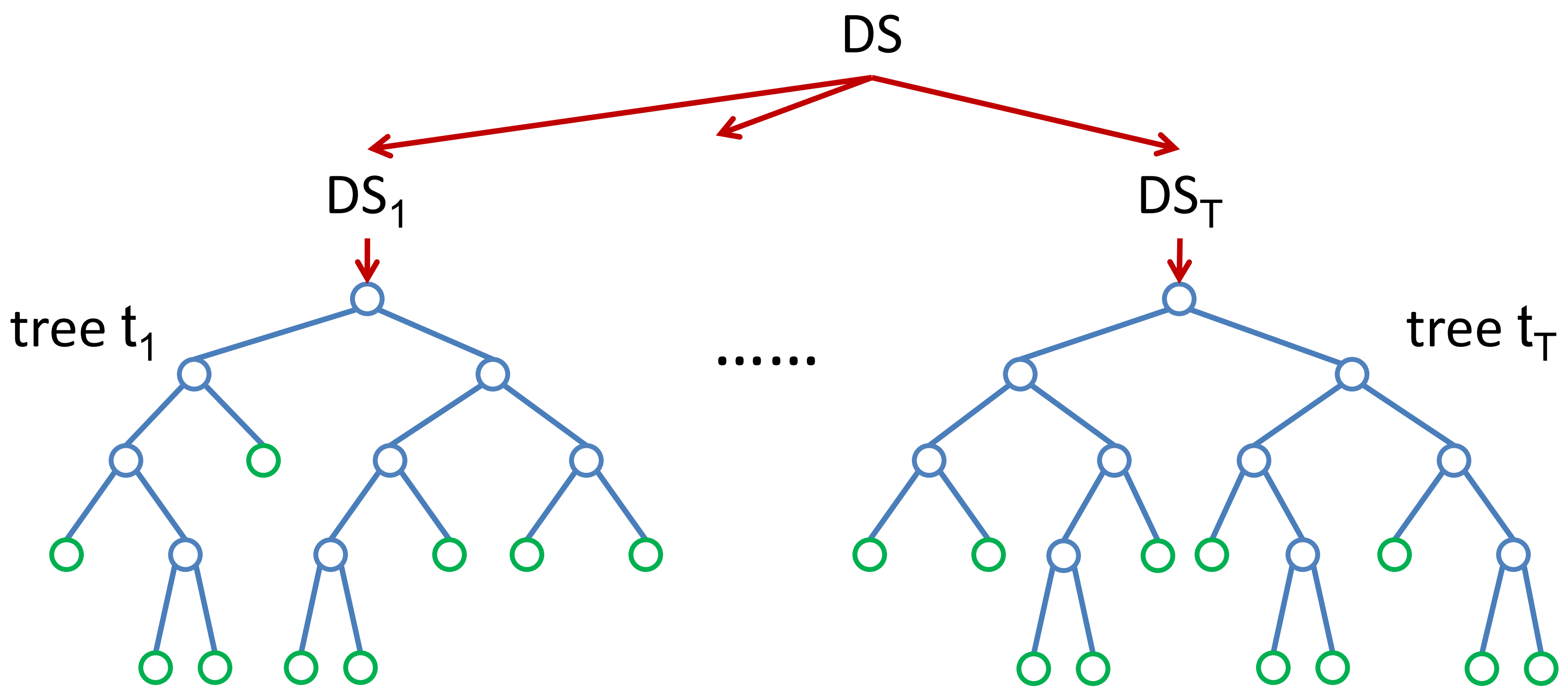
- Features f and thresholds t chosen at random
- Choose f and t to maximize gain in information

$$\Delta E = -\frac{|I_\mathrm{l}|}{|I_n|} E(I_\mathrm{l}) - \frac{|I_\mathrm{r}|}{|I_n|} E(I_\mathrm{r})$$



A Forest of Trees : Learning

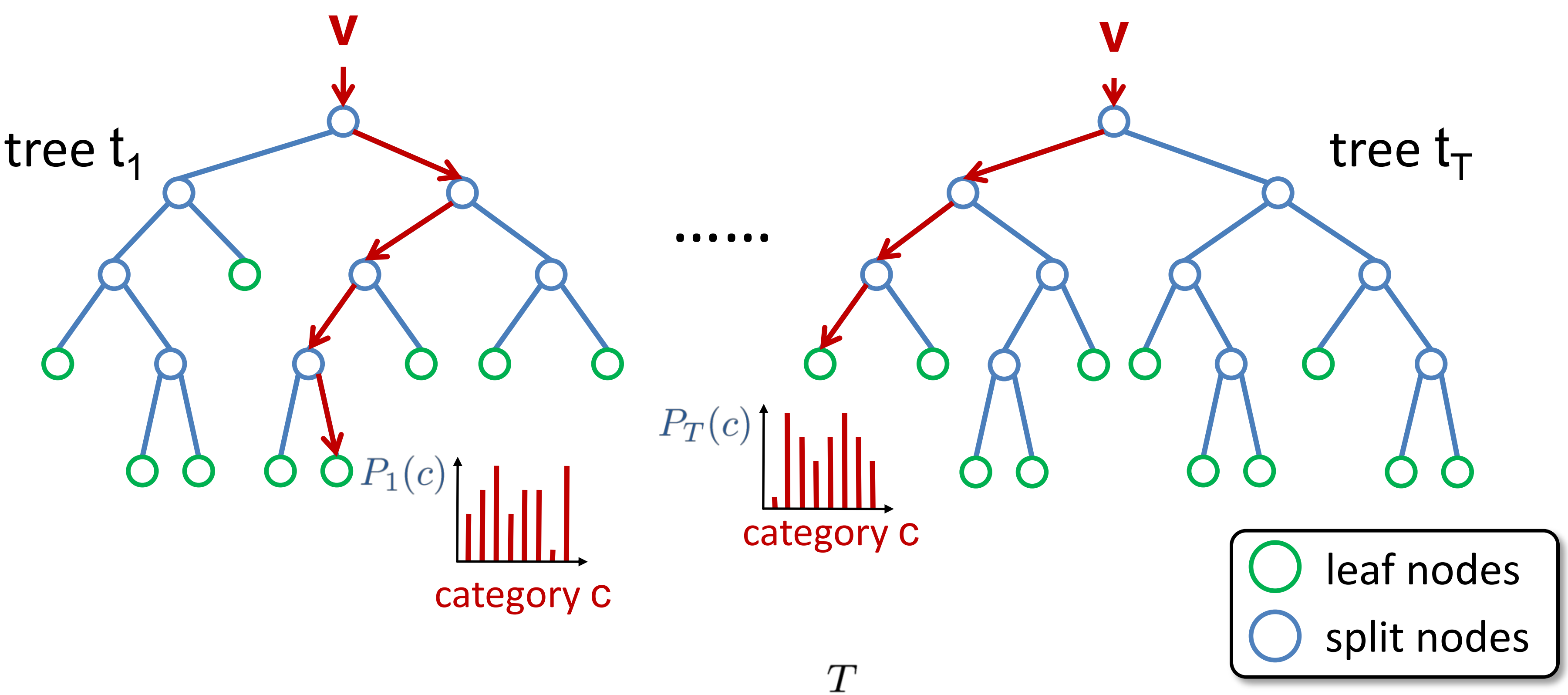
- Forest is an ensemble of decision trees



- Bagging (Bootstrap AGGregatING)
  - For each set, it randomly draws examples from the uniform dist. allowing duplication and missing.

A Forest of Trees : Recognition

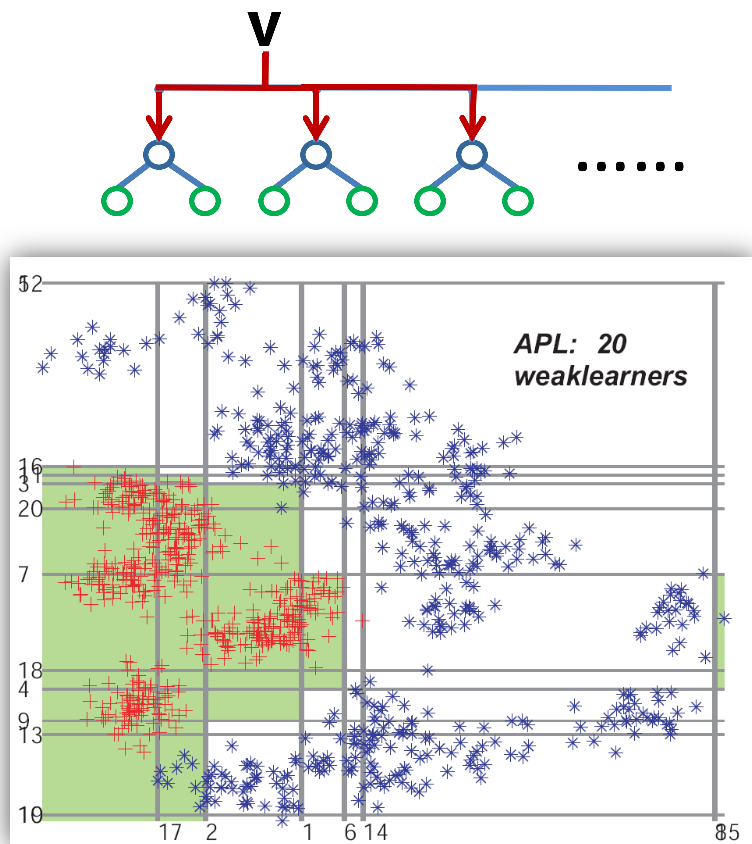
- Forest is an ensemble of decision trees



– classification is $$P(c|\mathbf{v}) = \sum_{t=1}^{T} P_t(c|\mathbf{v})$$

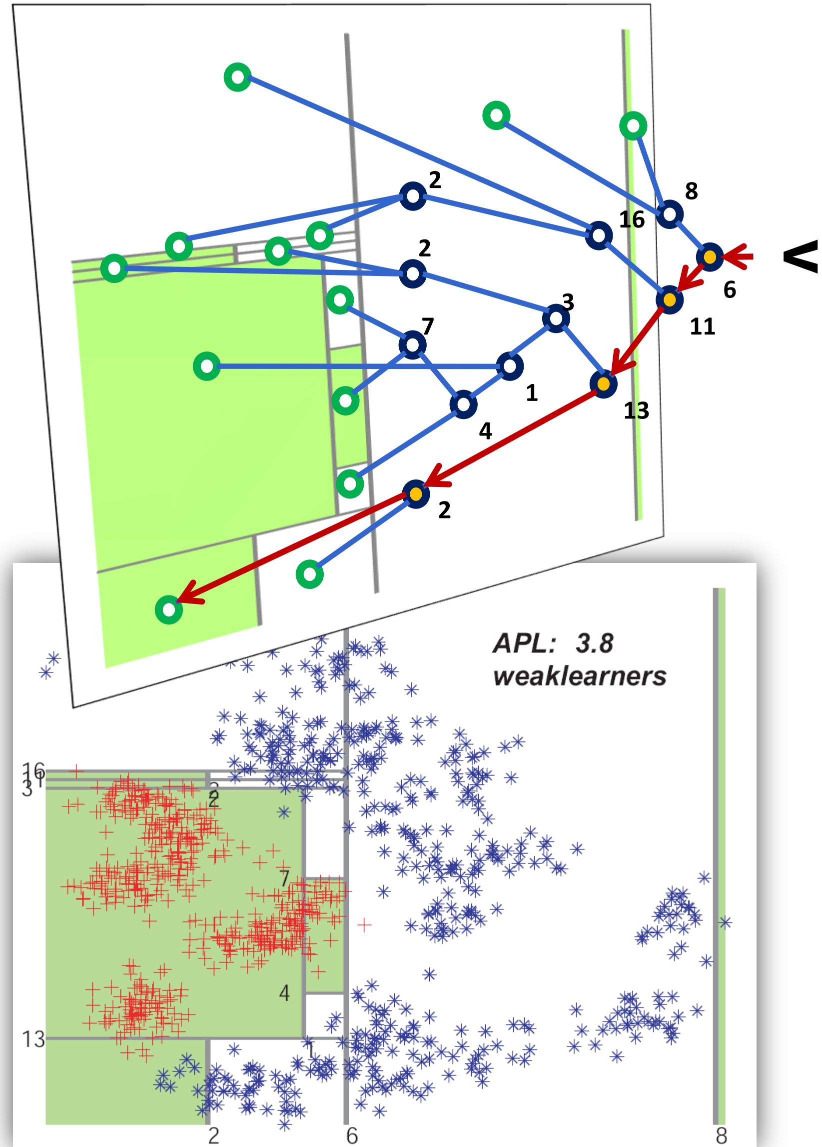
Slide credits to J. Shotton

Fast Evaluation



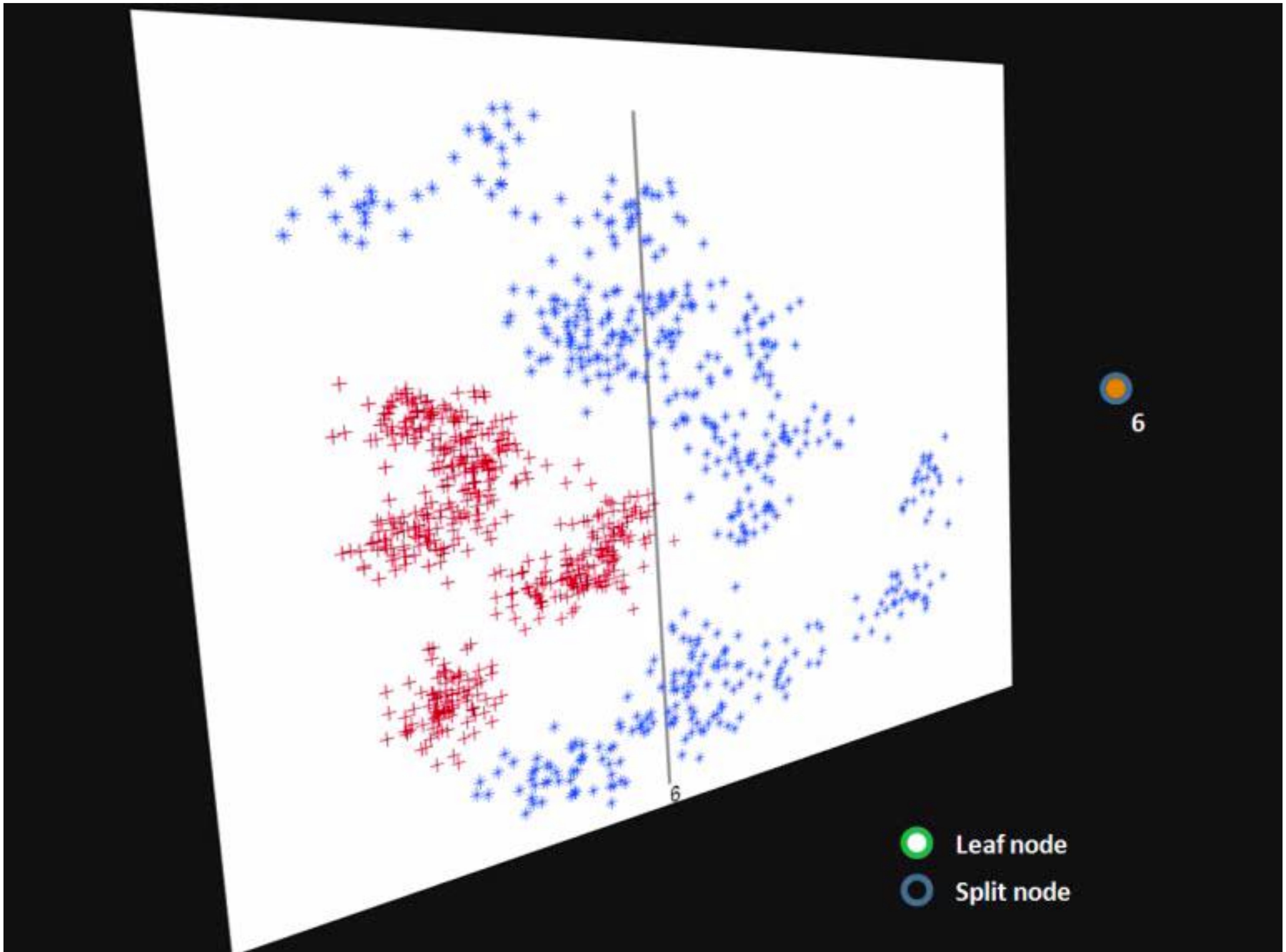
Flat structure

5 times
speed up



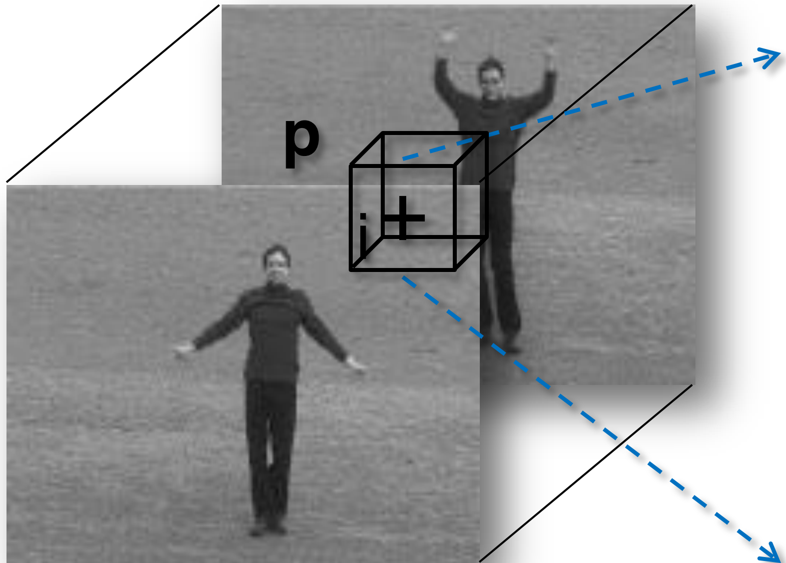
Tree structure

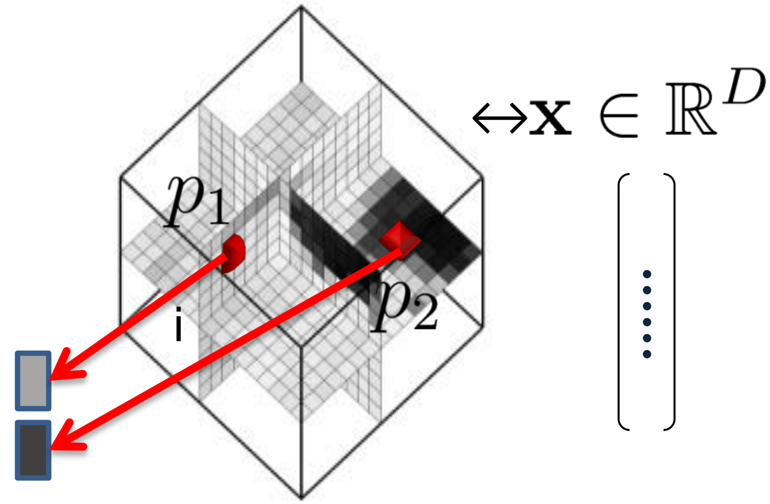
Demo video: Fast evaluation



Random Forest Codebook

Video Cuboid Features





Video pixel i gives cuboid p

(13x13x19 pixels in experiments)

$$f(\mathbf{p}) = p_{x_1,y_1,t_1}$$

$$f(\mathbf{p}) = p_{x_1,y_1,t_1} - p_{x_2,y_2,t_2}$$

$$f(\mathbf{p}) = p_{x_1,y_1,t_1} + p_{x_2,y_2,t_2}$$

tree split function

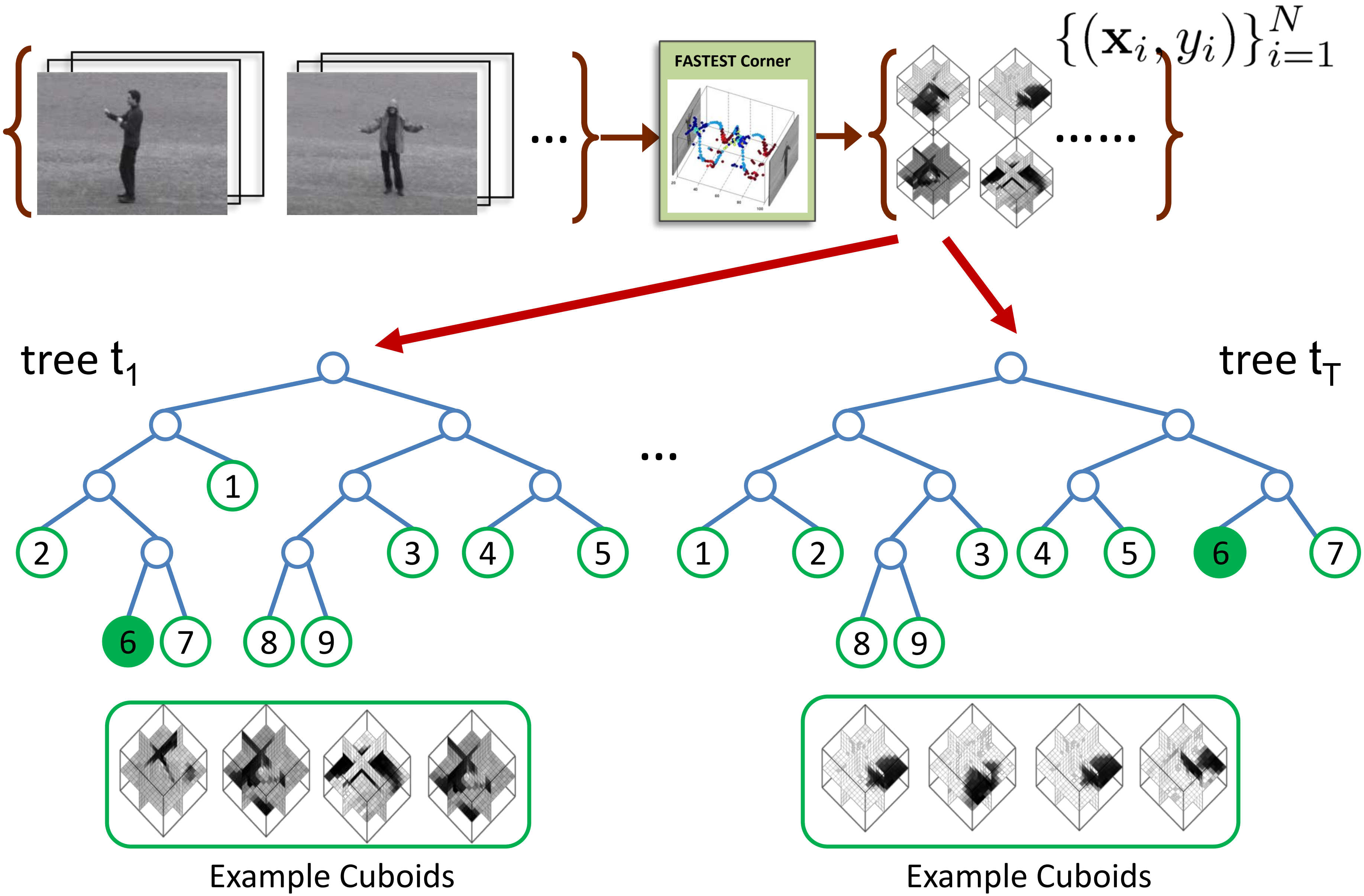
If f(**p**) > threshold

goto Left

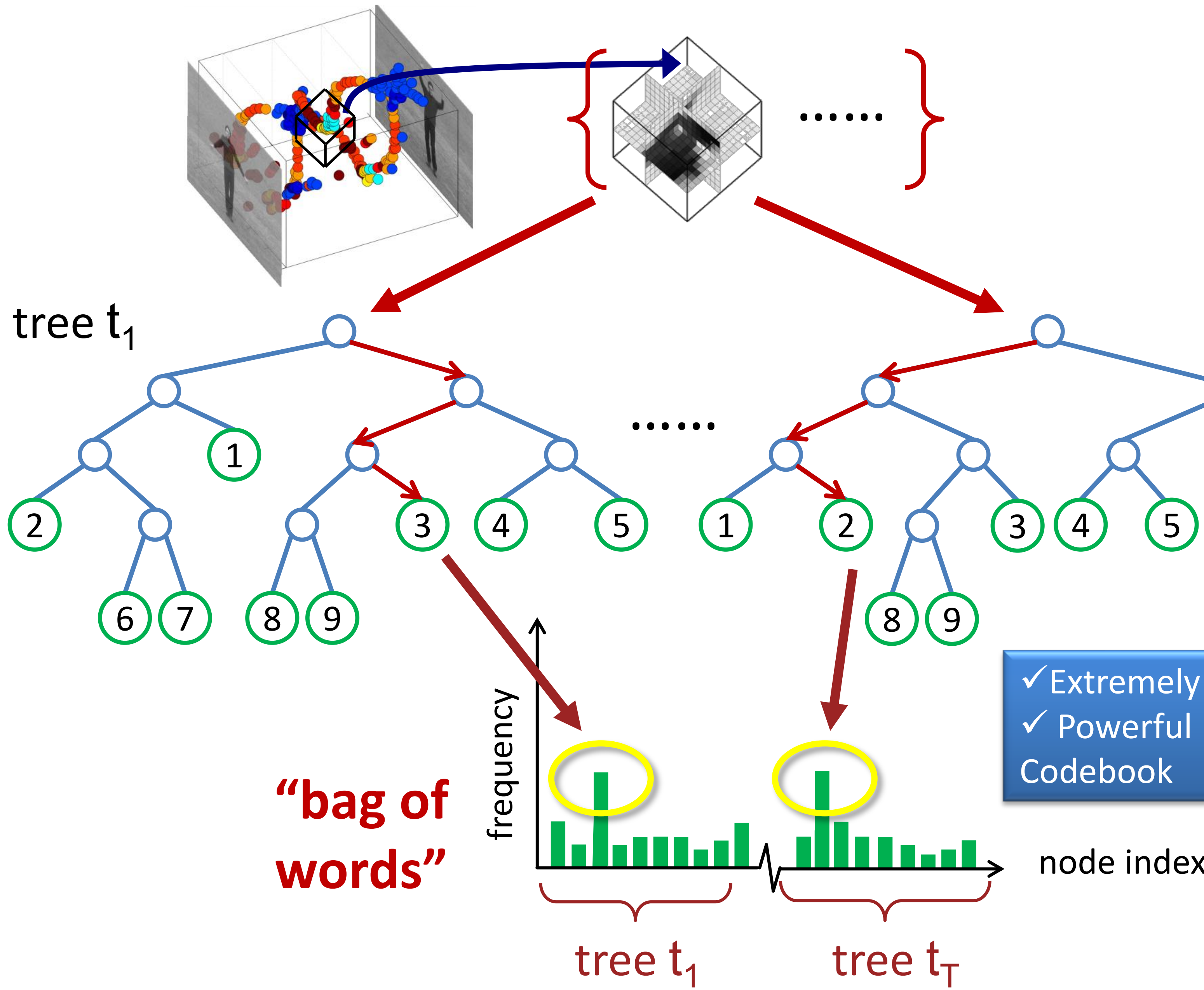
Else

goto Right

Randomized Forest Codebook



Histogram of Randomized Forest Codewords



✓ Extremely Fast
✓ Powerful Discriminative Codebook

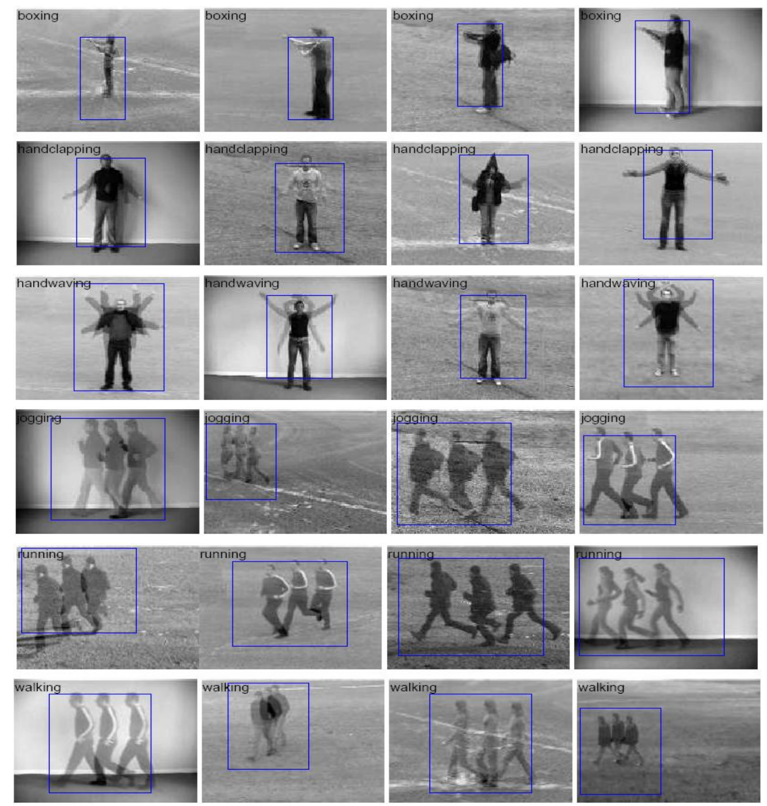
Matlab Demo:

Random Forest Codebook

Action Categorisation Result

- **Action categorization accuracy (%) on KTH dataset**
  - 6 types of actions, 25 subjects of 4 scenarios by leave-one-out protocol

	walking	running	jogging	hw	hc	boxing
walking	.79	.01	.14	.00	.06	.00
running	.01	.88	.11	.00	.00	.00
jogging	.11	.36	.52	.00	.01	.00
hand-waving	.00	.00	.00	.93	.01	.06
hand clapping	.00	.00	.00	.00	.77	.23
boxing	.00	.00	.00	.00	.00	1.00



Matlab Demo:

Action Categorisation

Take-home Message

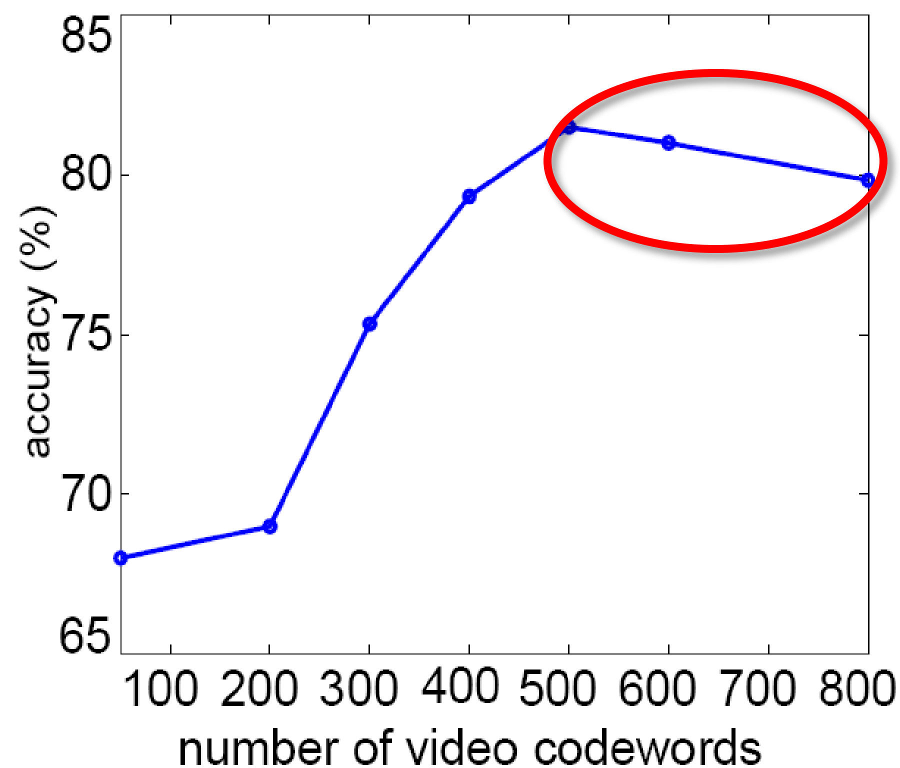
- Use of visual codebook greatly facilitates image and video categorisation tasks.
- The RF codebook is extremely fast as it only uses a small number of features in trees.
  - c.f. K-means codebook is in a flat structure. It compares an input with every codewords, which is time-demanding.
- The RF codebook is also a powerful discriminative codebook. It combines multiple decision trees showing good generalisation.
  - c.f. K-means is an unsupervised learning method.

Questions?

Action Recognition Result

- **Action categorization accuracy (%) on KTH dataset**
  - 6 types of actions, 25 subjects of 4 scenarios by leave-one-out protocol

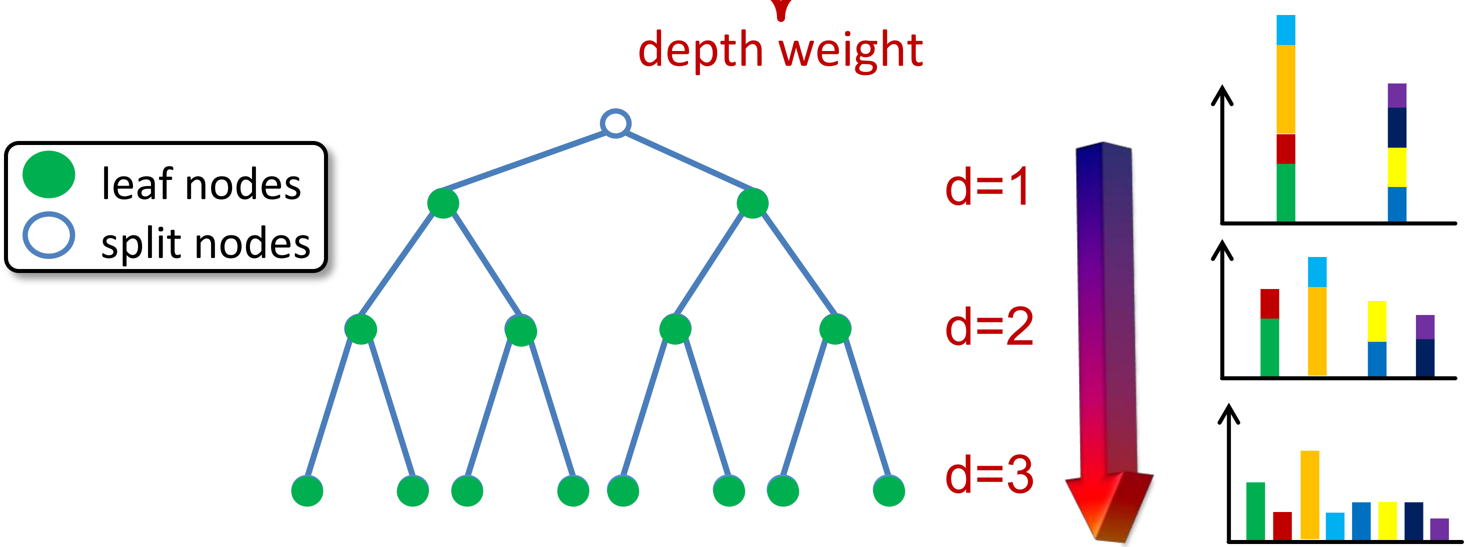
quantisation effect!



Pyramid Match Kernel

- PMK acts on semantic texton histogram
  - matches P and Q in *learned* hierarchical histogram space
  - deeper node matches are more important

$$K(P,Q) = \sum_{d=1}^{D} \underbrace{\frac{1}{2^{D-d+1}}}_{\text{depth weight}} (\overbrace{\mathcal{I}_d - \mathcal{I}_{d+1}}^{\text{increased similarity at depth d}})$$

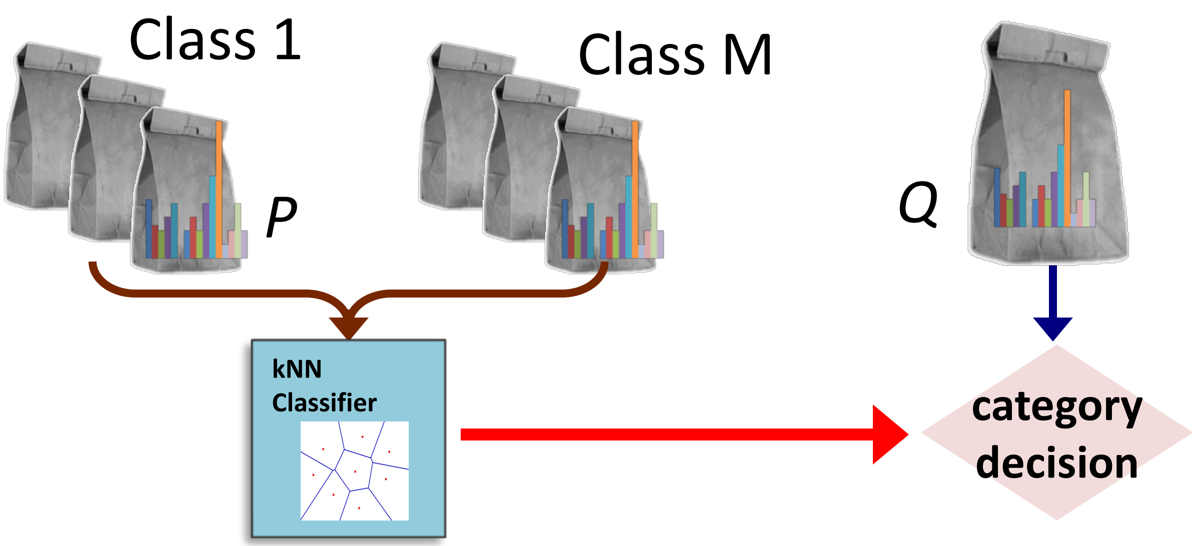


Categorisation

- Histogram matching of a pair of videos *P* and *Q* is

$$\boxed{\mathcal{I}_d(P,Q) = \sum_{n=1}^{b_d} \min\left(H_d(P^{(n)}), H_d(Q^{(n)})\right)}$$

- Based on HMK, we can use various classifiers such as kNN, SVM, Naïve-Bayes classifiers or pLSA, LDA.

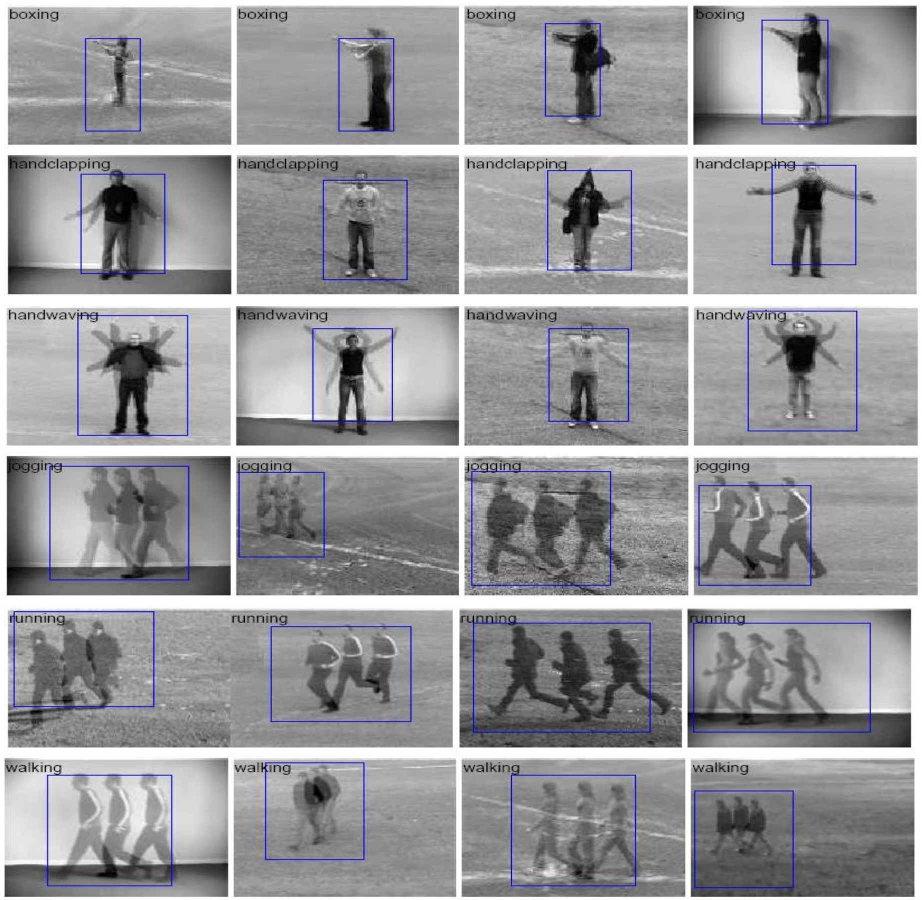


NN classification by

$$\boxed{\arg\max_c \mathcal{I}_d\left(P_c, Q\right)}$$

Action Recognition Result

- **Action categorization accuracy (%) on KTH dataset**
  - 6 types of actions, 25 subjects of 4 scenarios by leave-one-out protocol



	box	hclp	hwav	jog	run	walk
box	.98	.02	.00	.00	.00	.00
hclp	.00	1.0	.00	.00	.00	.00
hwav	.01	.02	.97	.00	.00	.00
jog	.00	.00	.00	.90	.10	.00
run	.00	.00	.00	.12	.88	.00
walk	.00	.00	.00	.01	.00	.99