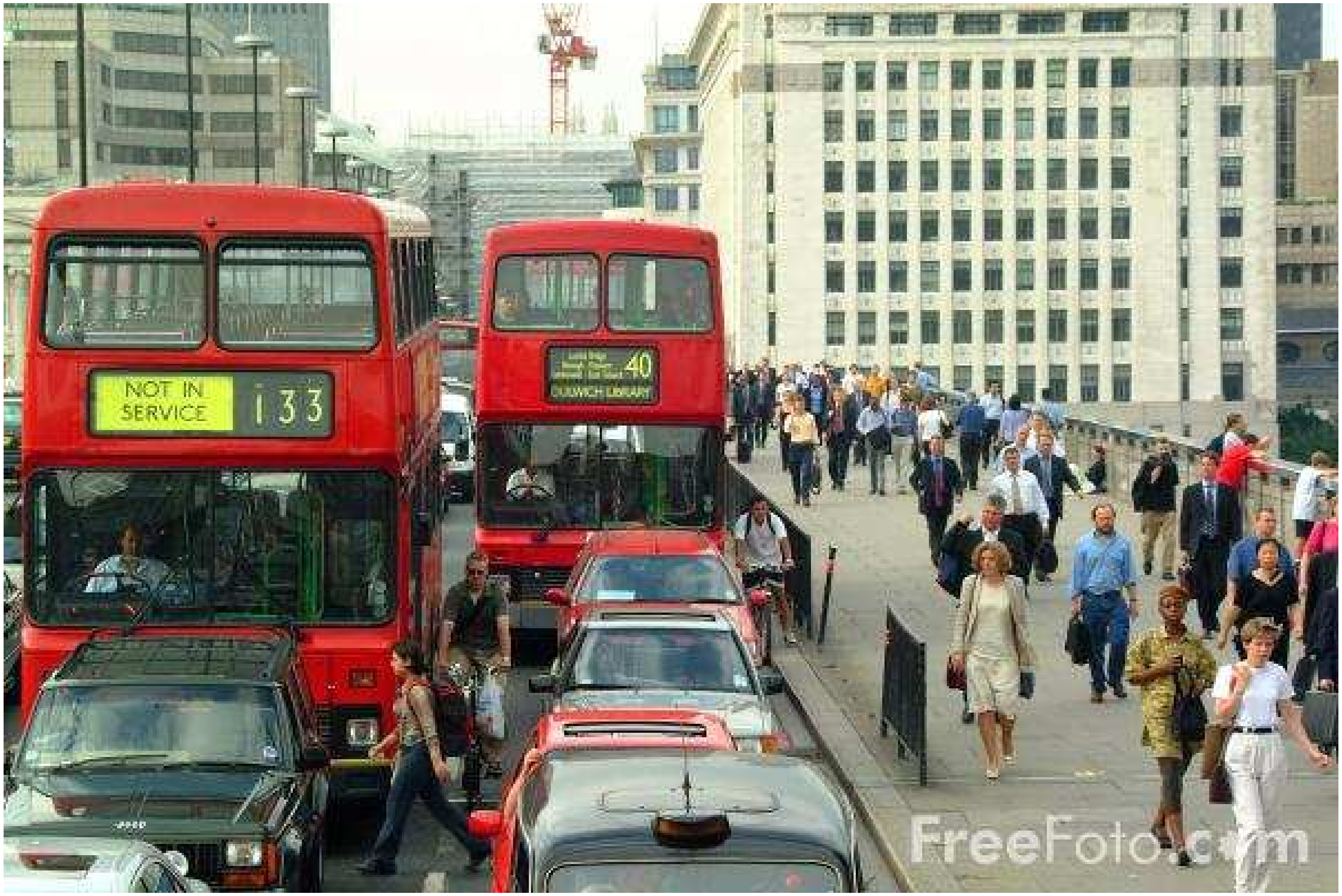
A Unified Approach to Object Category Recognition

Dr. Rahul Sukthankar
Intel Research and Carnegie Mellon

UCF Vision Class Guest Lecture
Nov 13, 2008

Collaborators: L. Yang, R. Jin, F. Jurie
Details in IEEE CVPR 2008 paper

Object Category Recognition

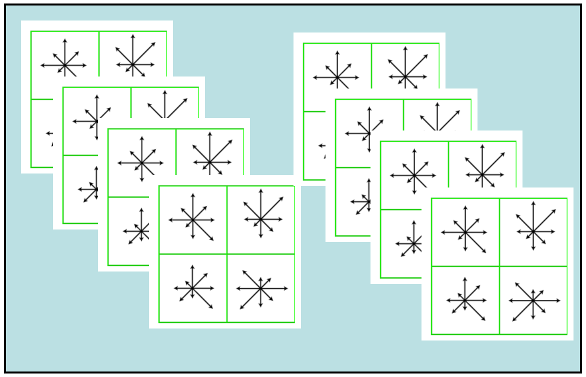


- sheep? ✗
- bus? ✓
- cat? ✗
- bicycle? ✓
- car? ✓
- cow? ✗
- dog? ✗
- horse? ✗
- mbike? ✓
- person? ✓

Standard Approach
(adopted from text IR)

[Fei-Fei *et al.*, 2005];
[Sivic *et al.*, 2005];
and many others

Feature extraction
and representation
(e.g., SIFT)



Quantization
+ histogram

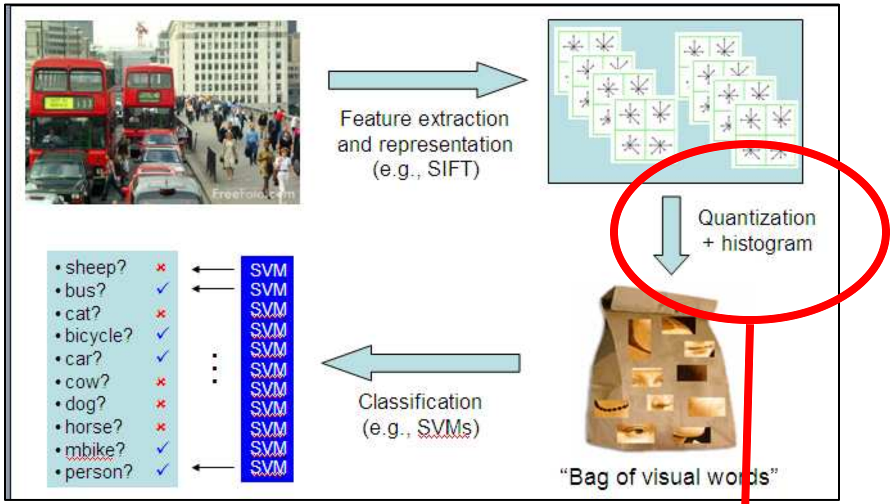
“Bag of visual words”

Classification
(e.g., SVMs)

SVM
SVM
SVM
SVM
SVM
SVM
SVM
SVM
SVM
SVM
SVM

• sheep?	✗
• bus?	✓
• cat?	✗
• bicycle?	✓
• car?	✓
• cow?	✗
• dog?	✗
• horse?	✗
• mbike?	✓
• person?	✓

Codebook Construction by Clustering

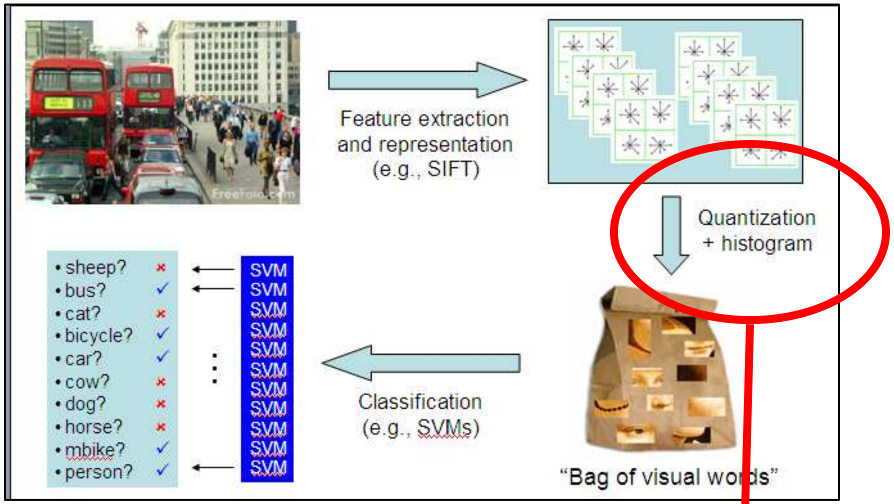


Limitation (I):

Universal dictionary → category independent

Unsupervised clustering → ignores labeling information

Codebook Construction by Clustering



[Farquar *et al*., 2005]
class-specific dictionaries (MAP)

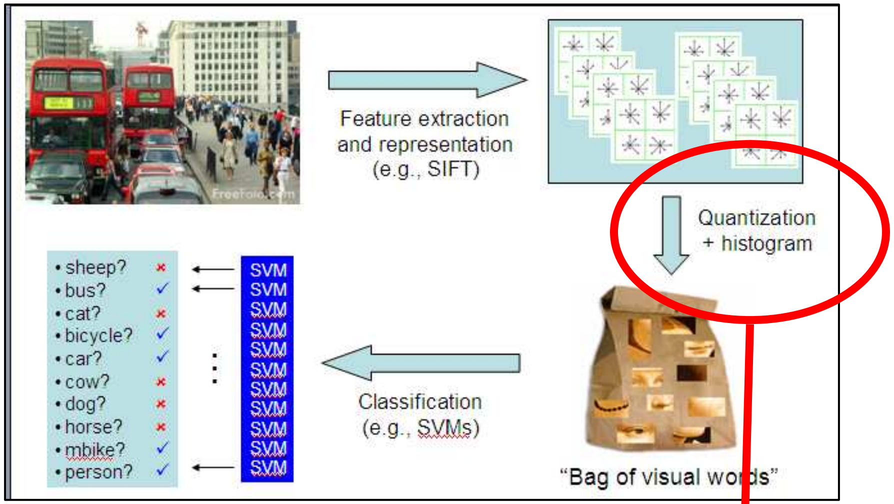
[Perronnin et al., 2006]
class-specific histograms

Limitation (I):

Universal dictionary → category independent

Unsupervised clustering → ignores labeling information

Codebook Construction by Clustering



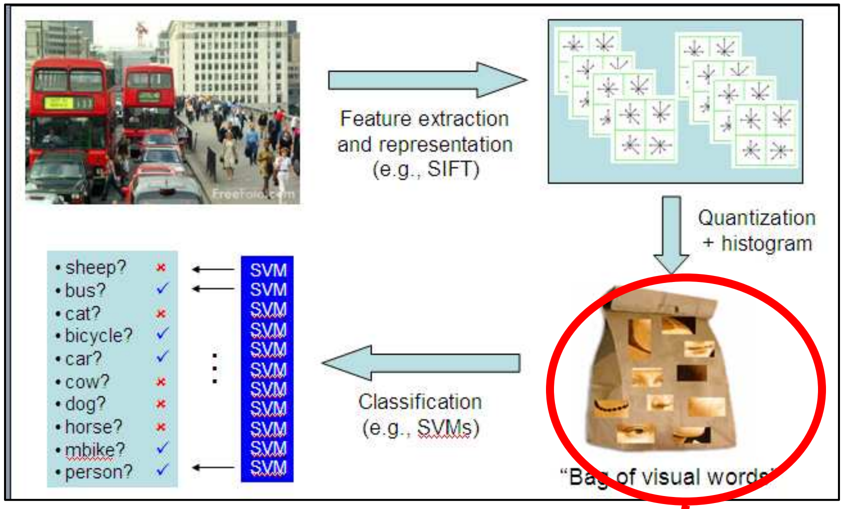
[Moosmann *et al*., 2006]
discriminative random forests

Limitation (I):

Universal dictionary → category independent

Unsupervised clustering → ignores labeling information

Codebook Construction by Clustering

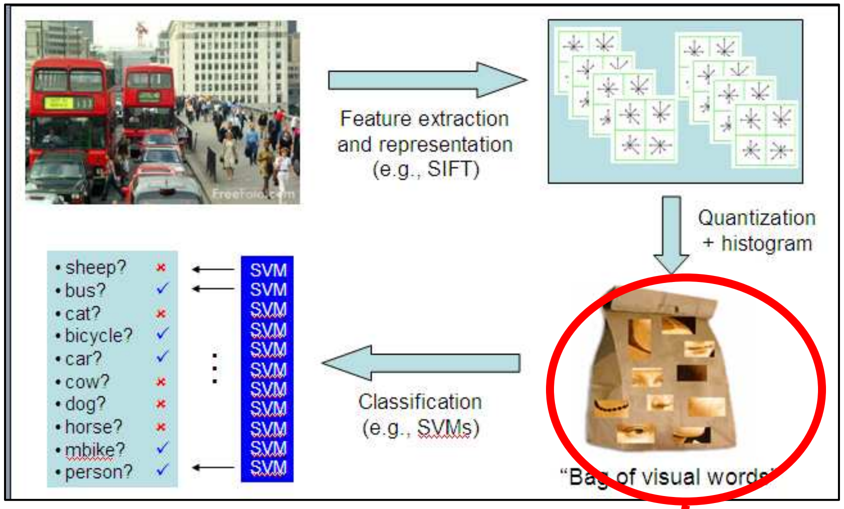


Limitation (II):

Every SIFT feature forced into one cluster
→ failure to capture partial similarity

Difficulty in deciding the number of clusters
→ wrong choice leads to poor dictionaries

Codebook Construction by Clustering



[Perronnin *et al*., 2007]
soft assignment to words

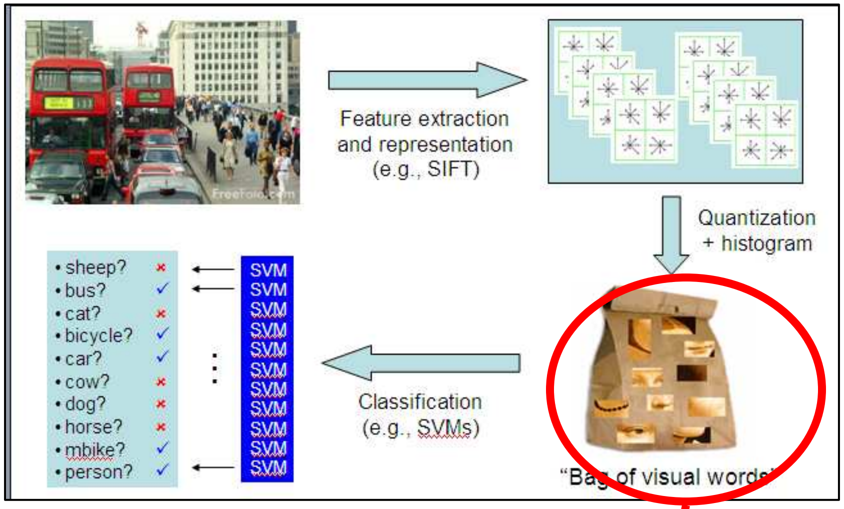
[Heller & Ghahramani, 2007]
overlapping clustering

Limitation (II):

Every SIFT feature forced into one cluster
→ failure to capture partial similarity

Difficulty in deciding the number of clusters
→ wrong choice leads to poor dictionaries

Codebook Construction by Clustering



[Winn *et al*., 2005]
pair-wise word merging

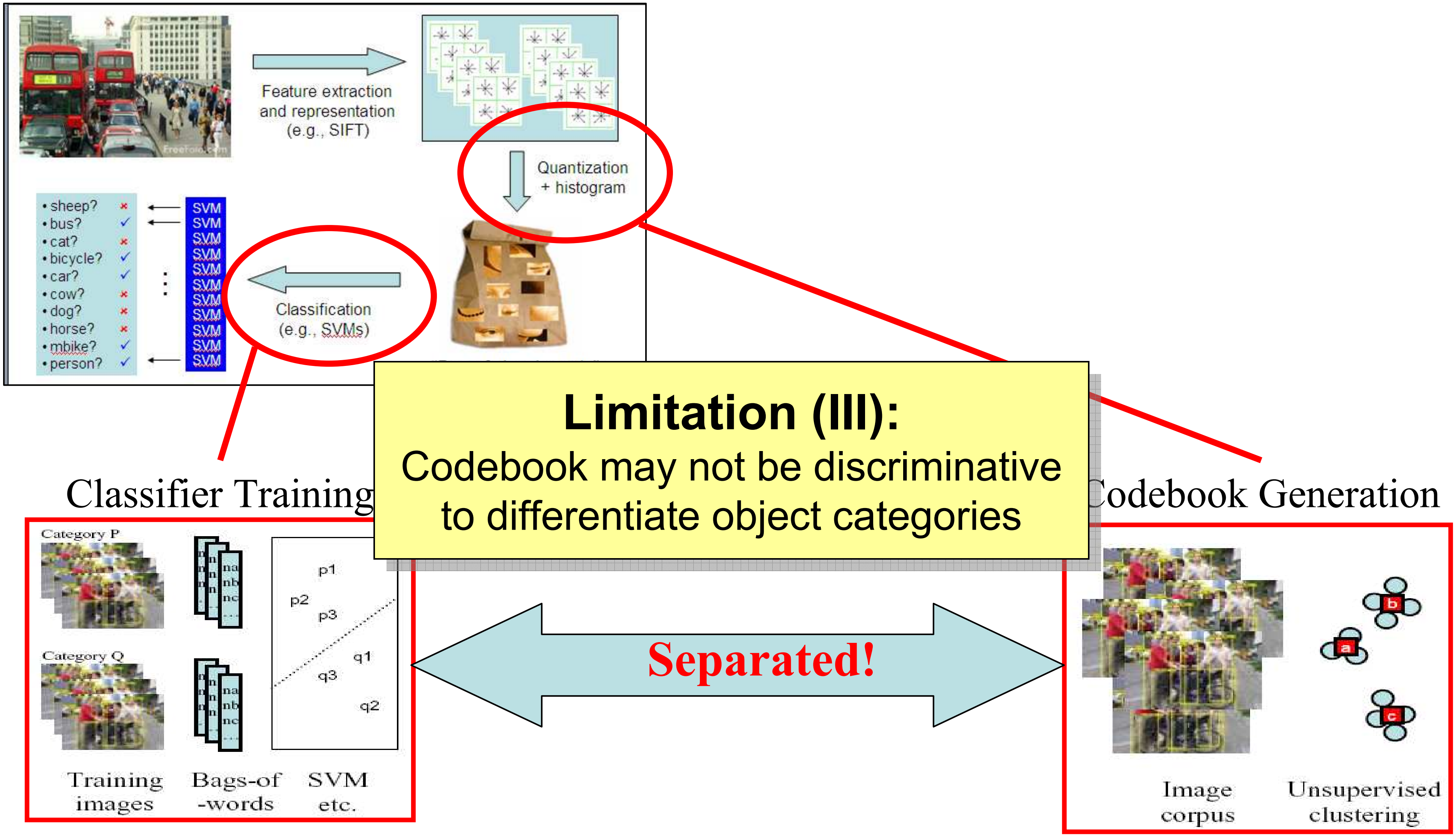
[Liu *et al*., 2007]
discriminative cluster refinement

Limitation (II):

Every SIFT feature forced into one cluster
→ failure to capture partial similarity

Difficulty in deciding the number of clusters
→ wrong choice leads to poor dictionaries

Codebook Construction by Clustering

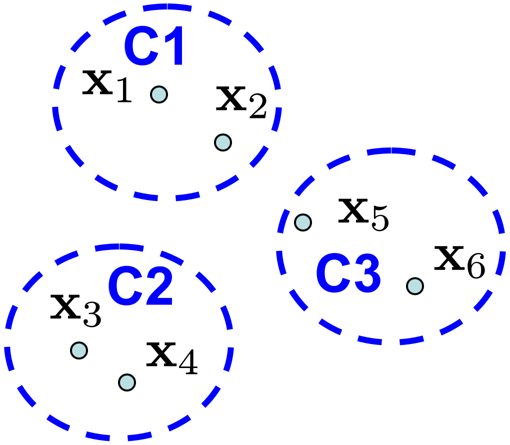


Limitation (III):

Codebook may not be discriminative to differentiate object categories

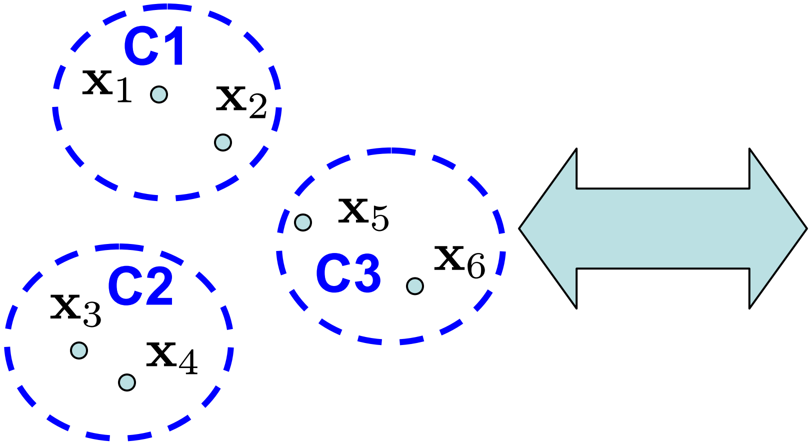
Understanding Clustering

- Clustering is a special coding



Understanding Clustering

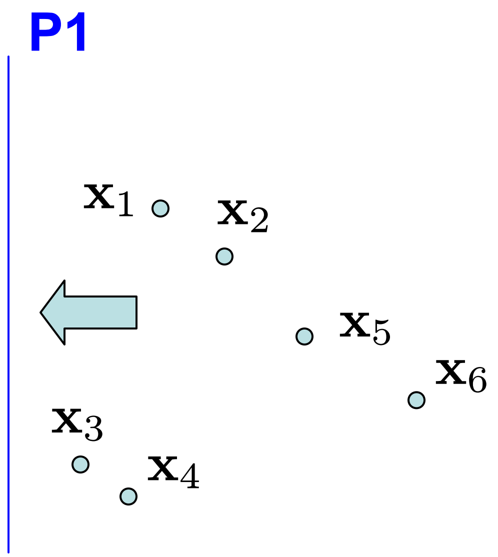
- Clustering is a special coding
  - Two coding vectors are either identical or orthogonal
  - Two coding vectors differ by at most two bits
- More general coding
  - Error Correcting Output Code (ECOC)



	C1	C2	C3
$\mathbf{x}_1$	1	0	0
$\mathbf{x}_2$	1	0	0
$\mathbf{x}_3$	0	1	0
$\mathbf{x}_4$	0	1	0
$\mathbf{x}_5$	0	0	1
$\mathbf{x}_6$	0	0	1

Understanding Clustering

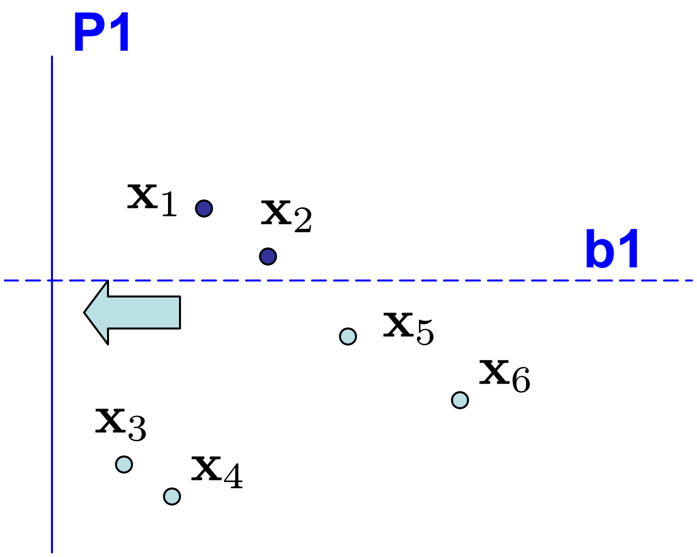
- Clustering is a special coding
  - Two coding vectors are either identical or orthogonal
  - Two coding vectors differ by at most two bits
- Our approach: coding by thresholded projections



	P1	P2	P3	P4
$\mathbf{x}_1$				
$\mathbf{x}_2$				
$\mathbf{x}_3$				
$\mathbf{x}_4$				
$\mathbf{x}_5$				
$\mathbf{x}_6$				

Understanding Clustering

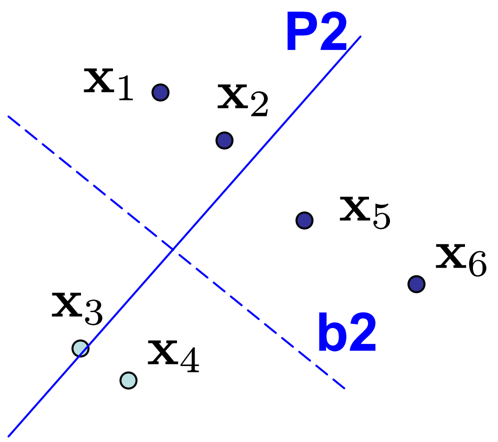
- Clustering is a special coding
  - Two coding vectors are either identical or orthogonal
  - Two coding vectors differ by at most two bits
- Our approach: coding by thresholded projections



	P1	P2	P3	P4
$\mathbf{x}_1$	1			
$\mathbf{x}_2$	1			
$\mathbf{x}_3$	0			
$\mathbf{x}_4$	0			
$\mathbf{x}_5$	0			
$\mathbf{x}_6$	0			

Understanding Clustering

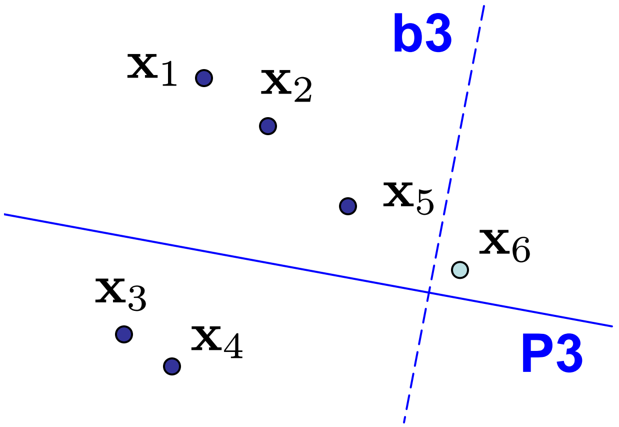
- Clustering is a special coding
  - Two coding vectors are either identical or orthogonal
  - Two coding vectors differ by at most two bits
- Our approach: coding by thresholded projections



	P1	P2	P3	P4
$\mathbf{x}_1$	1	1		
$\mathbf{x}_2$	1	1		
$\mathbf{x}_3$	0	0		
$\mathbf{x}_4$	0	0		
$\mathbf{x}_5$	0	1		
$\mathbf{x}_6$	0	1		

Understanding Clustering

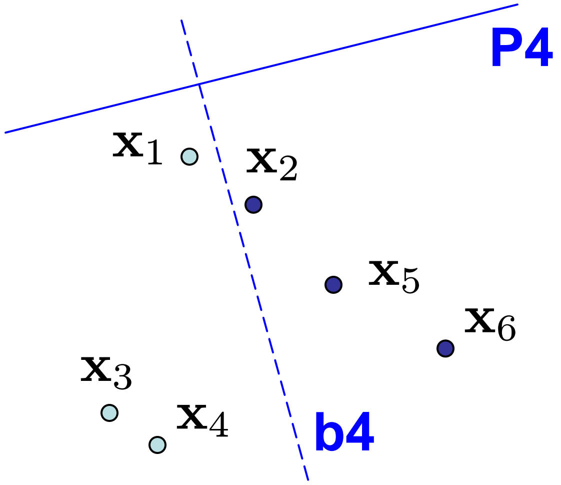
- Clustering is a special coding
  - Two coding vectors are either identical or orthogonal
  - Two coding vectors differ by at most two bits
- Our approach: coding by thresholded projections



	P1	P2	P3	P4
$\mathbf{x}_1$	1	1	1	
$\mathbf{x}_2$	1	1	1	
$\mathbf{x}_3$	0	0	1	
$\mathbf{x}_4$	0	0	1	
$\mathbf{x}_5$	0	1	1	
$\mathbf{x}_6$	0	1	0	

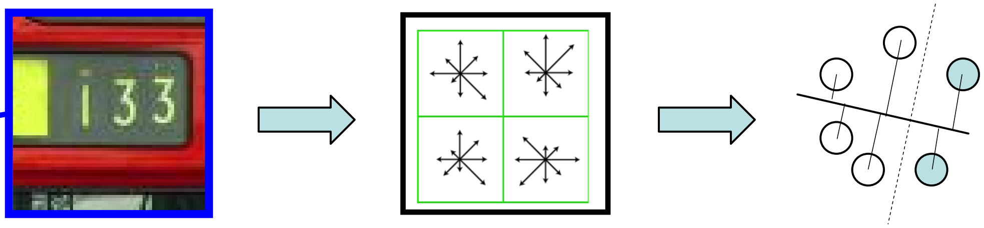
Understanding Clustering

- Clustering is a special coding
  - Two coding vectors are either identical or orthogonal
  - Two coding vectors differ by at most two bits
- Our approach: coding by thresholded projections
  - Non-orthogonal codes – chosen for maximal class separation
  - Key questions: how to select the projections P and thresholds b?



	P1	P2	P3	P4
$\mathbf{x}_1$	1	1	1	0
$\mathbf{x}_2$	1	1	1	1
$\mathbf{x}_3$	0	0	1	0
$\mathbf{x}_4$	0	0	1	0
$\mathbf{x}_5$	0	1	1	1
$\mathbf{x}_6$	0	1	0	1

Anatomy of a Visual Bit



$$g_k(\mathbf{x}, y) = I(\mathbf{x}^\top \mathbf{w}_k^y - b_k^y) = \begin{cases} 1 & \mathbf{x}^\top \mathbf{w}_k^y > b_k^y \\ 0 & \mathbf{x}^\top \mathbf{w}_k^y \leq b_k^y \end{cases}$$

(learned)

“Is this feature relevant to the `bus’ category?”

- Weakly-supervised learning of visual bits
- Applying visual bits to object category recognition

Image Classification using Visual Bits

Category *a*

	$g_1(x,a)$	$g_2(x,a)$	…	$g_T(x,a)$
$\mathbf{x}_1$	1	1	..	…
$\mathbf{x}_2$	1	0	..	…
…	…	…	…	…
$\mathbf{x}_n$	0	0	…	…

feature-level representation

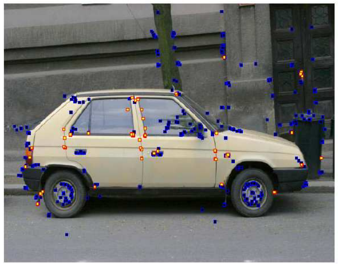


Image Classification using Visual Bits

Category *a*

	$g_1(x,a)$	$g_2(x,a)$	…	$g_T(x,a)$
$\mathbf{x}_1$	1	1	..	…
$\mathbf{x}_2$	1	0	..	…
…	…	…	…	…
$\mathbf{x}_n$	0	0	…	…
X	2	1	…	…

+

image
representation

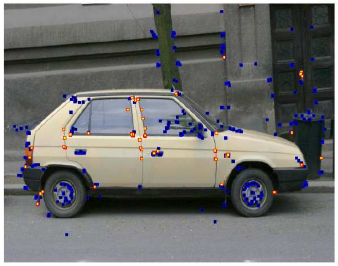


Image Classification using Visual Bits

Category *a*

	$g_1(X,a)$	$g_2(X,a)$	…	$g_T(X,a)$
X	2	1	…	…

Classifier for Cat. *a*

$$f_a(X) = \sum_{k=1}^{T} \alpha_k g_k(X,a)$$

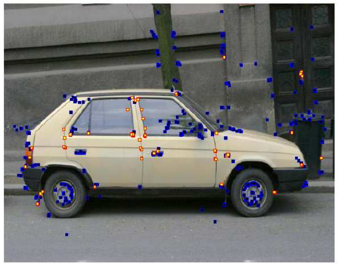
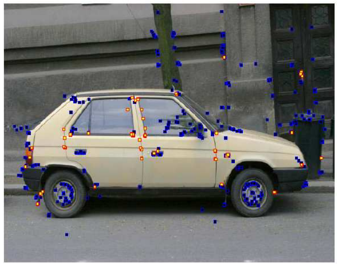


Image Classification using Visual Bits



Category *a*

	$g_1(X,a)$	$g_2(X,a)$	…	$g_T(X,a)$
X	2	1	…	…

Classifier for Cat. *a*

$$f_a(X) = \sum_{k=1}^{T} \alpha_k g_k(X, a)$$

Category *b*

	$g_1(X,b)$	$g_2(X,b)$	…	$g_T(X,b)$
X	0	1	…	…

Classifier for Cat. *b*

$$f_b(X) = \sum_{k=1}^{T} \alpha_k g_k(X, b)$$

………

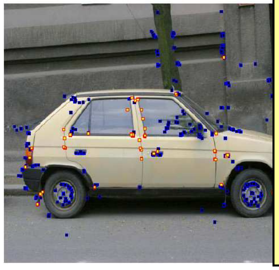
Category *z*

	$g_1(X,z)$	$g_2(X,z)$	…	$g_T(X,z)$
X	…	…	…	…

Classifier for Cat. *z*

$$f_z(X) = \sum_{k=1}^{T} \alpha_k g_k(X, z)$$

Image Classification using Visual Bits



Category *a*

	$g_1(X,a)$	$g_2(X,a)$	…	$g_T(X,a)$
X	2	1	…	…

Classifier for Cat. *a*

$$f_a(X) = \sum_{k=1}^{T} \alpha_k g_k(X,a)$$

Category *b*

	$g_1(X,b)$	$g_2(X,b)$	…	$g_T(X,b)$

Classifier for Cat. *b*

$$T \quad g_k(X,b)$$

Learn visual bit functions *g*(*x*, *a*) and weights α together

↕

Unify code generation with discriminative classifier

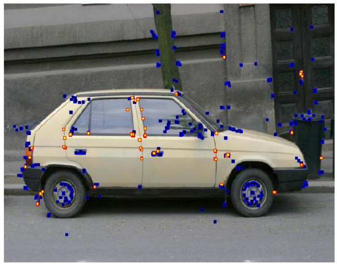
Category *z*

	$g_1(X,z)$	$g_2(X,z)$	…	$g_T(X,z)$
X	…	…	…	…

Classifier for Cat. *z*

$$f_z(X) = \sum_{k=1}^{T} \alpha_k g_k(X,z)$$

Image Classification using Visual Bits



Category a

	$g_1(x,a)$	$g_2(x,a)$	…	$g_T(x,a)$
X	2	1	…	…

Classifier for Cat. *a*

$$f_a(X) = \sum_{k=1}^{T} \alpha_k g_k(X,a)$$

Category *b*

	$g_1(X,b)$	$g_2(X,b)$	…	$g_T(X,b)$
X	0	1	…	…

Classifier for Cat. *b*

$$f_b(X) = \sum_{k=1}^{T} \alpha_k g_k(X,b)$$

$$f_a(X) = \sum_{i=1}^{N} \alpha_i k(\vec{g}(X,a), \vec{g}(X_i,a))$$

Category *z*

	$g_1(x,z)$	$g_2(x,z)$	…	$g_T(x,z)$
X	…	…	…	…

Classifier for Cat. *z*

$$f_z(X) = \sum_{k=1}^{T} \alpha_k g_k(X,z)$$

$k(\mathbf{x},\mathbf{x}') : \mathbb{R}^T \times \mathbb{R}^T \to \mathbb{R}$: kernel function

$\vec{g}(X,a) \equiv (g_1(X,a),\ldots,g_T(X,a))$: visual bit vector

Image Classification using Visual Bits

Category a

	$g_1(x,a)$	$g_2(x,a)$	…	$g_T(x,a)$
X	2	1	…	…

Classifier for Cat. *a*

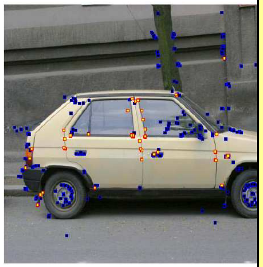
$$f_a(X) = \sum_{k=1}^{T} \alpha_k g_k(X,a)$$

Category *b*

Classifier for Cat. *b*

$$T$$

$$g_k(X,b)$$



Cat. *z*

$$g_k(X,z)$$

Generalizes to nonlinear classifier

(can be implemented using standard SVM)

$$f_a(X) = \sum_{i=1}^{N} \alpha_i k(\vec{g}(X,a), \vec{g}(X_i,a))$$

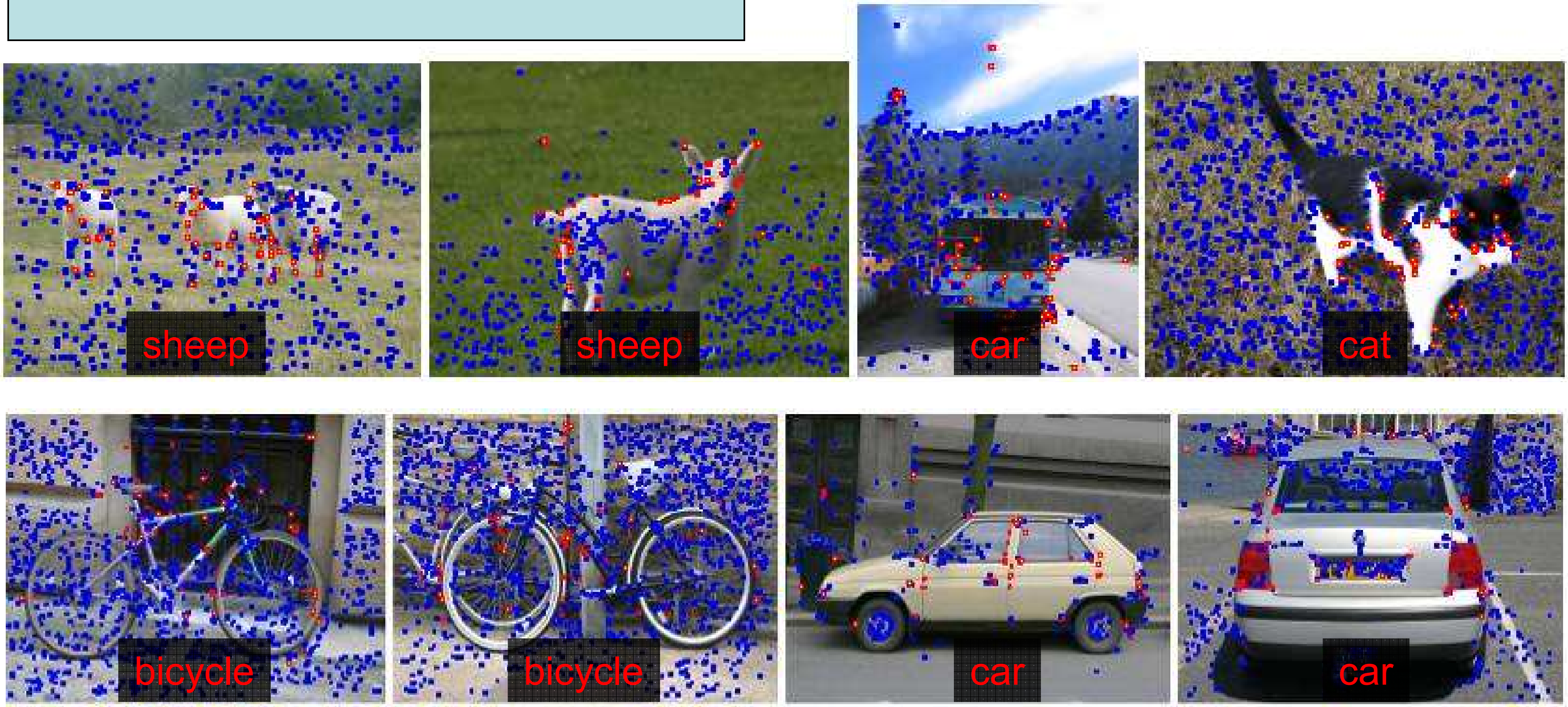
$$k(\mathbf{x},\mathbf{x}) : \mathbb{R}^T \times \mathbb{R}^T \to \mathbb{R}: \text{ kernel function}$$

$$\vec{g}(X,a) = (g_1(X,a), \ldots, g_T(X,a)): \text{ visual bit vector}$$

Relevant Visual Bits Localize Concepts

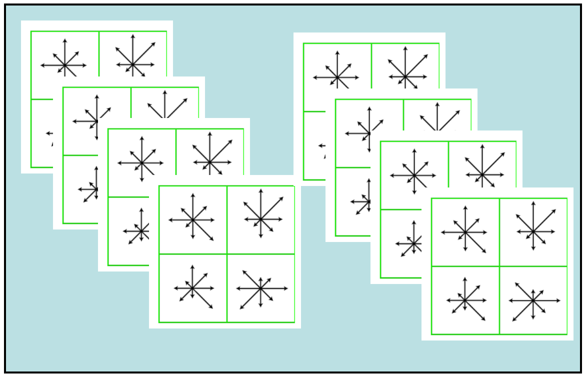
Relevance of feature **x** to category *y*

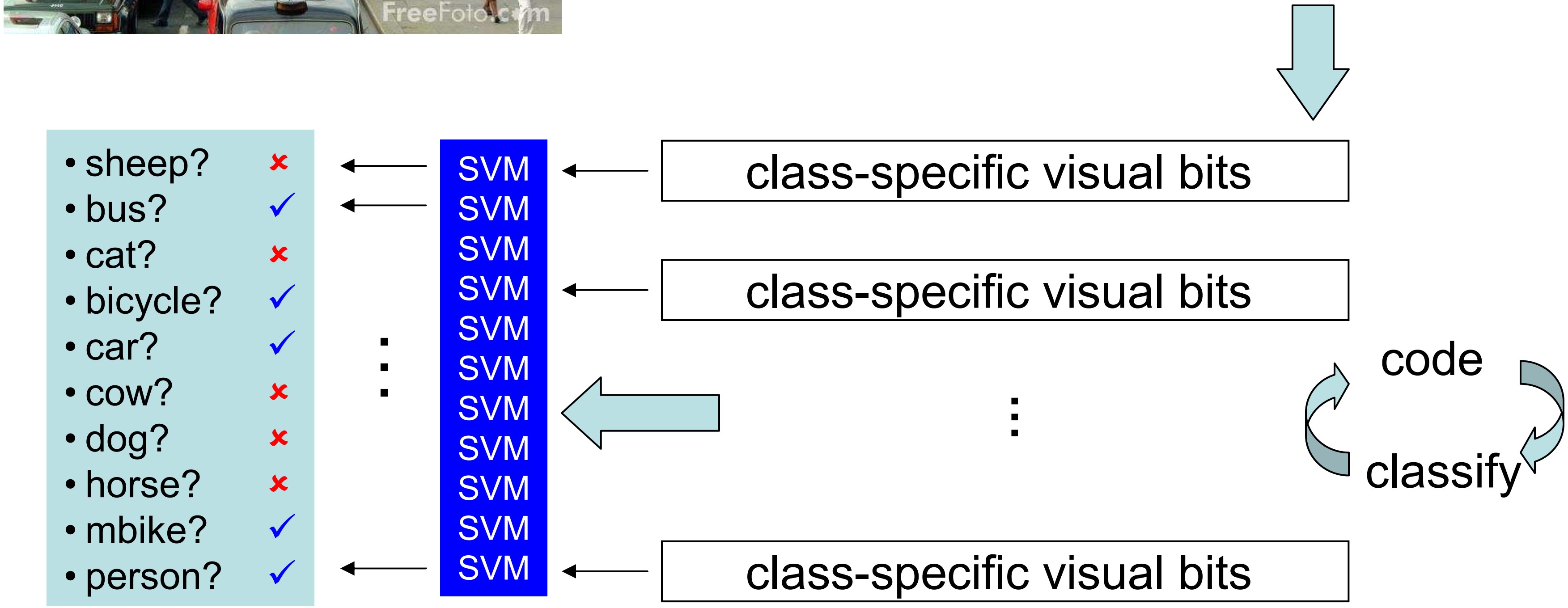
$$\sum_{k=1}^{T} \alpha_k g_k(\mathbf{x}, y)$$



Unified Approach

Feature extraction and representation (e.g., SIFT)





- sheep? ✗
- bus? ✓
- cat? ✗
- bicycle? ✓
- car? ✓
- cow? ✗
- dog? ✗
- horse? ✗
- mbike? ✓
- person? ✓

SVM
SVM
SVM
SVM
SVM
SVM
SVM
SVM
SVM
SVM
SVM

class-specific visual bits

class-specific visual bits

⋮

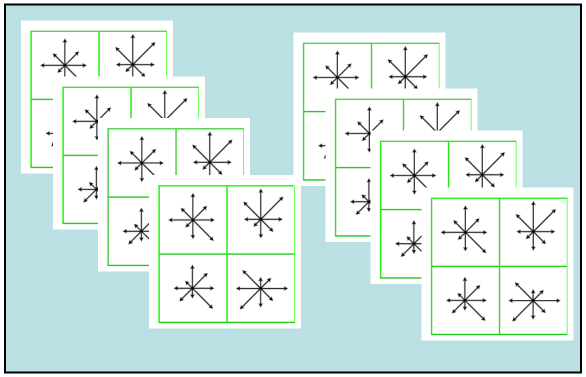
class-specific visual bits

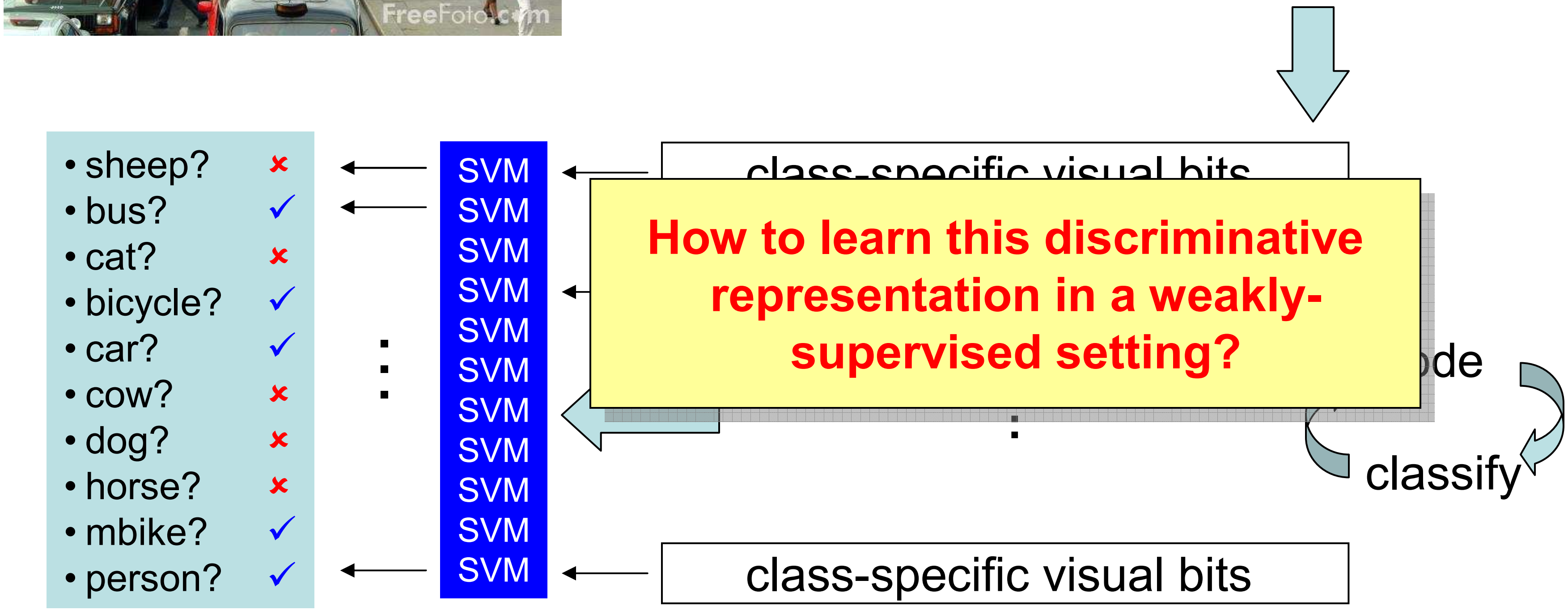
code

classify

Unified Approach

Feature extraction and representation (e.g., SIFT)





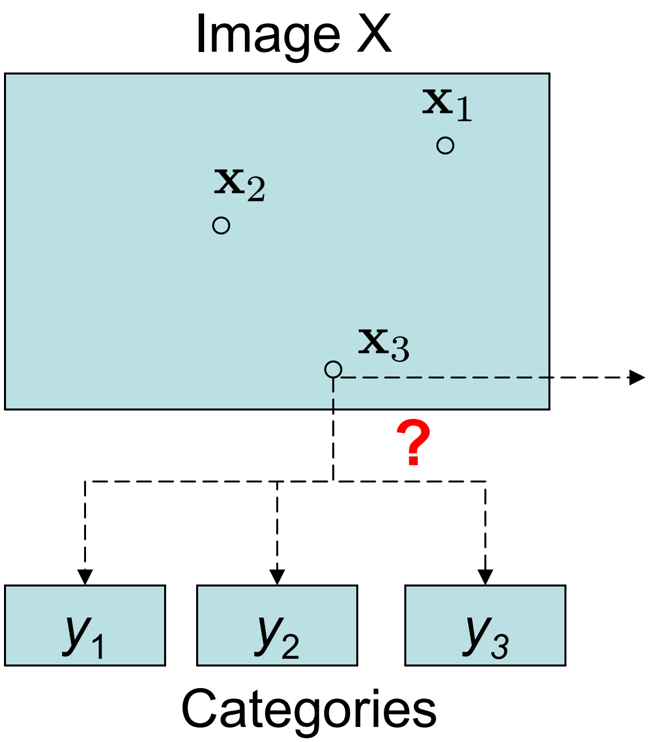
- sheep? ✗
- bus? ✓
- cat? ✗
- bicycle? ✓
- car? ✓
- cow? ✗
- dog? ✗
- horse? ✗
- mbike? ✓
- person? ✓

How to learn this discriminative representation in a weakly-supervised setting?

Learning Visual Bits

Optimization Framework

- Given visual bit functions $g(x, a)$ and weights α, how to measure if they are able to classify image X=($\mathbf{x}_1$,…,$\mathbf{x}_n$) into cat. (y_1, y_2, …, y_K)



Challenge

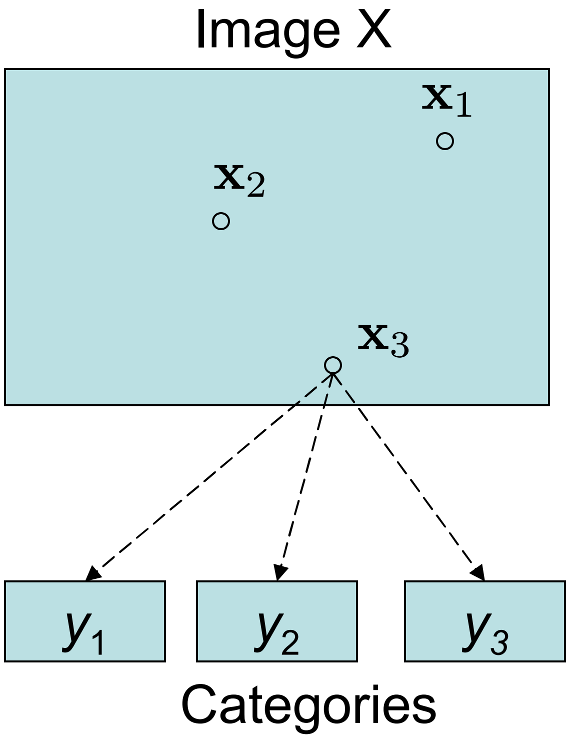
Which features correspond to which categories, or do not correspond to any category of interest at all ?

Learning Visual Bits

Optimization Framework

- Given visual bit functions *g*(*x*, *a*) and weights α, how to measure if they are able to classify image X=($\mathbf{x}_1,\ldots,\mathbf{x}_n$) into cat. ($y_1, y_2, \ldots, y_K$)

Solution: consider all possibilities



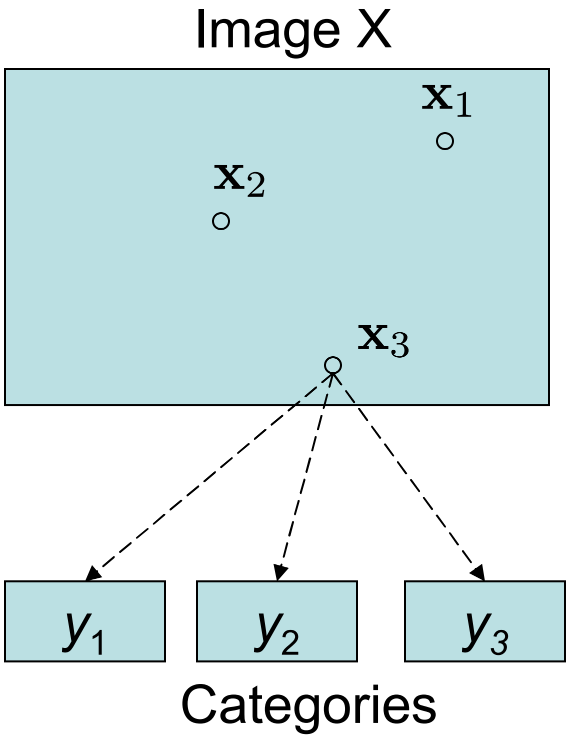
$$f(\mathbf{x}_3, y_1) = \sum_{k=1}^{T} \alpha_k g_k(\mathbf{x}_3, y_1)$$

$$f(\mathbf{x}_3, y_2) = \sum_{k=1}^{T} \alpha_k g_k(\mathbf{x}_3, y_2)$$

$$f(\mathbf{x}_3, y_3) = \sum_{k=1}^{T} \alpha_k g_k(\mathbf{x}_3, y_3)$$

Learning Visual Bits
Optimization Framework

- Given visual bit functions *g*(*x*, *a*) and weights α, how to measure if they are able to classify image X=($\mathbf{x}_1$,…,$\mathbf{x}_n$) into cat. (y_1, y_2, …, y_K)



Solution: consider all possibilities

$$f(\mathbf{x}_3, y_1) = \sum_{k=1}^{T} \alpha_k g_k(\mathbf{x}_3, y_1)$$

$$f(\mathbf{x}_3, y_2) = \sum_{k=1}^{T} \alpha_k g_k(\mathbf{x}_3, y_2)$$

$$f(\mathbf{x}_3, y_3) = \sum_{k=1}^{T} \alpha_k g_k(\mathbf{x}_3, y_3)$$

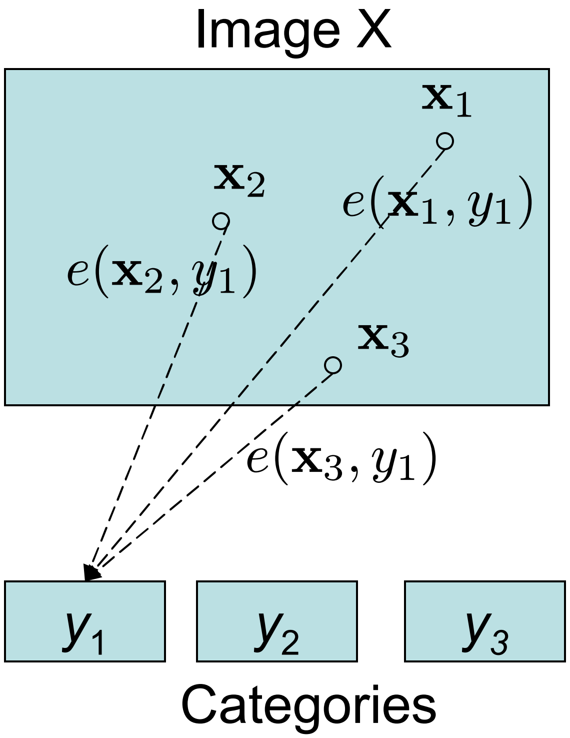
Prob. of associating feature **x** with cat. *y*

$$\boxed{e(\mathbf{x}_3, y_i) = \frac{\exp(f(\mathbf{x}_3, y_i))}{\sum_{z=1}^{m} \exp(f(\mathbf{x}_3, z))}}$$

Learning Visual Bits

Optimization Framework

- Given visual bit functions *g*(*x*, *a*) and weights α, how to measure if they are able to classify image X=($\mathbf{x}_1,\ldots,\mathbf{x}_n$) into cat. ($y_1$, y_2, …, y_K)



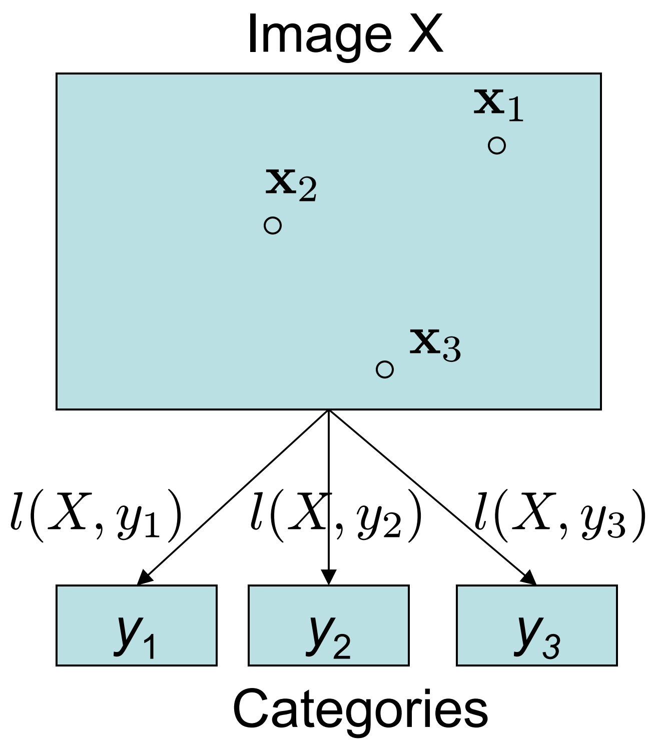
Loss function for image X

$$l(X, y_1) = \frac{n}{\sum_{j=1}^{n} e(\mathbf{x}_j, y_1)}$$

Learning Visual Bits

Optimization Framework

- Given visual bit functions *g*(*x*, *a*) and weights α, how to measure if they are able to classify image X=($\mathbf{x}_1$,…,$\mathbf{x}_n$) into cat. (y_1, y_2, …, y_K)



Loss function for image X

$$l(X, y_1) = \frac{n}{\sum_{j=1}^{n} e(\mathbf{x}_j, y_1)}$$

$$l(X, \mathbf{y}) = \sum_{y \in \mathbf{y}} l(X, y)$$

Loss function for the image collection

$$\mathcal{L}(\alpha_{1:T}, g_{1:T}) = \sum_{i=1}^{N} l(X_i, \mathbf{y}_i)$$

Learning Visual Bits

Optimization Framework

Given a collection of training images

$$\mathcal{T} = \{(X_i, \mathbf{y}_i), i = 1, \ldots, N\}$$

Find optimal visual bits and combination weights by solving

$$\min_{g_{1:T}, \alpha_{1:T}} \mathcal{L}(\alpha_{1:T}, g_{1:T}) = \sum_{i=1}^{N} l(X_i, \mathbf{y}_i)$$

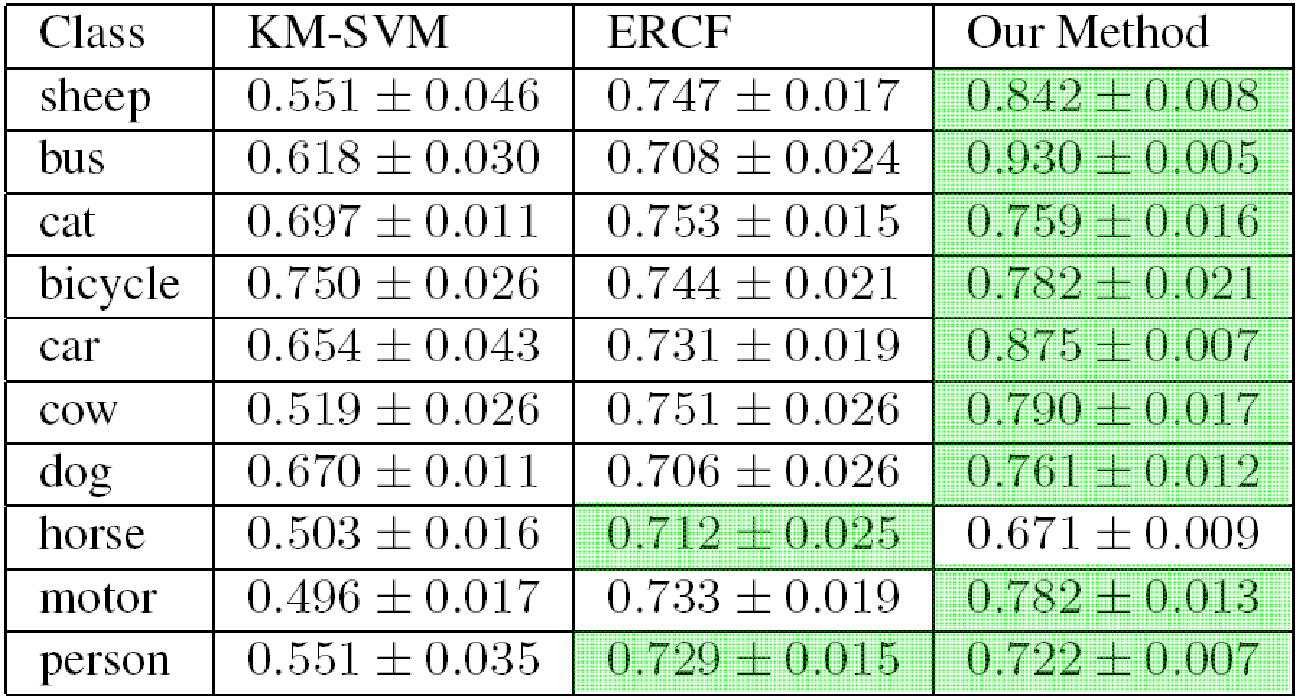
Overview of optimization algorithm (reminiscent of boosting)

- Iterative approach: learn one visual bit (*g*) and weight (*α*) at a time
- Employ bound optimization to decouple *g* and *α*

[details in paper and supplementary material]

Results on PASCAL 2006
(AUR with 100 training examples)

- Follows methodology from [Marszalek & Schmid, 2006]
- Baselines
  - Standard: K-means (k=1000) + SVM (χ^2 kernel)
  - Discriminative: Extremely Randomized Clustering Forests



Class	KM-SVM	ERCF	Our Method
sheep	0.551 ± 0.046	0.747 ± 0.017	0.842 ± 0.008
bus	0.618 ± 0.030	0.708 ± 0.024	0.930 ± 0.005
cat	0.697 ± 0.011	0.753 ± 0.015	0.759 ± 0.016
bicycle	0.750 ± 0.026	0.744 ± 0.021	0.782 ± 0.021
car	0.654 ± 0.043	0.731 ± 0.019	0.875 ± 0.007
cow	0.519 ± 0.026	0.751 ± 0.026	0.790 ± 0.017
dog	0.670 ± 0.011	0.706 ± 0.026	0.761 ± 0.012
horse	0.503 ± 0.016	0.712 ± 0.025	0.671 ± 0.009
motor	0.496 ± 0.017	0.733 ± 0.019	0.782 ± 0.013
person	0.551 ± 0.035	0.729 ± 0.015	0.722 ± 0.007

Conclusion

- Unify codebook construction + classifier training
  - Generate codebooks by iterative projection
  - Efficiently learn projection and weights together
- Impact on object category recognition
  - Learns better representations with limited training data
  - No parameters to tune

intel™

Leap ahead™