

Discriminative Visual



Unifying Discriminative Visual Codebook Generation
with
Classifier Training for Object Category Recognition

Liu Yang Michigan State University

Rong Jin Michigan State University

Rahul Sukthankar Intel Research & Carnegie Mellon

Frederic Jurie INRIA

Object Category Recognition



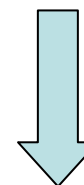
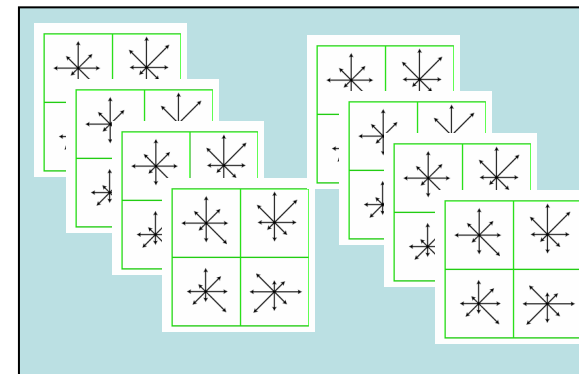
- sheep? ✗
- bus? ✓
- cat? ✗
- bicycle? ✓
- car? ✓
- cow? ✗
- dog? ✗
- horse? ✗
- mbike? ✓
- person? ✓

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



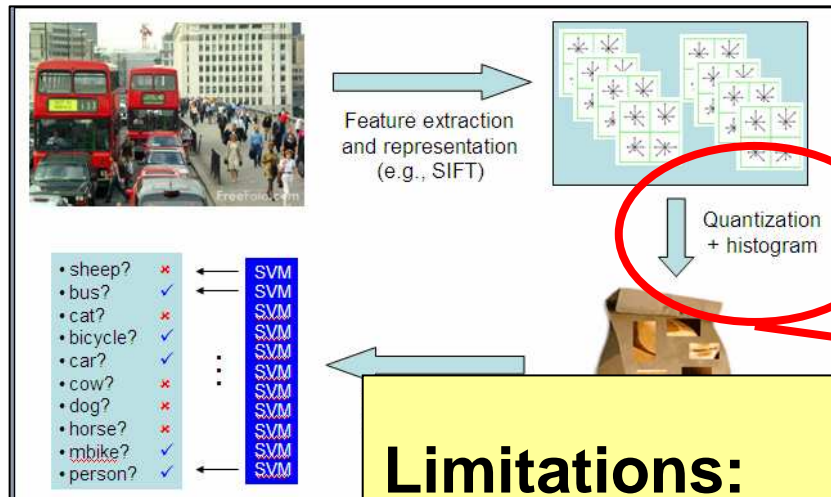
“Bag of visual words”



Classification
(e.g., SVMs)

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

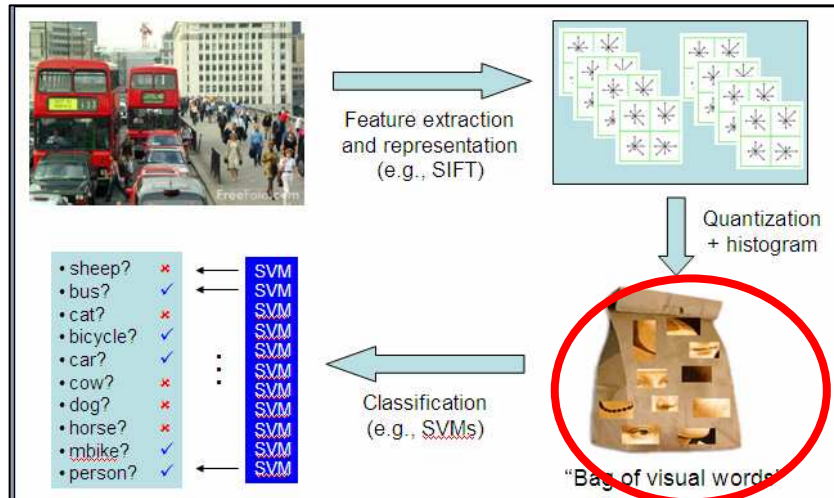
Codebook Construction by Clustering



Limitations:

- Universal dictionary → category independent
- Unsupervised clustering → ignores labeling information
- 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]

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

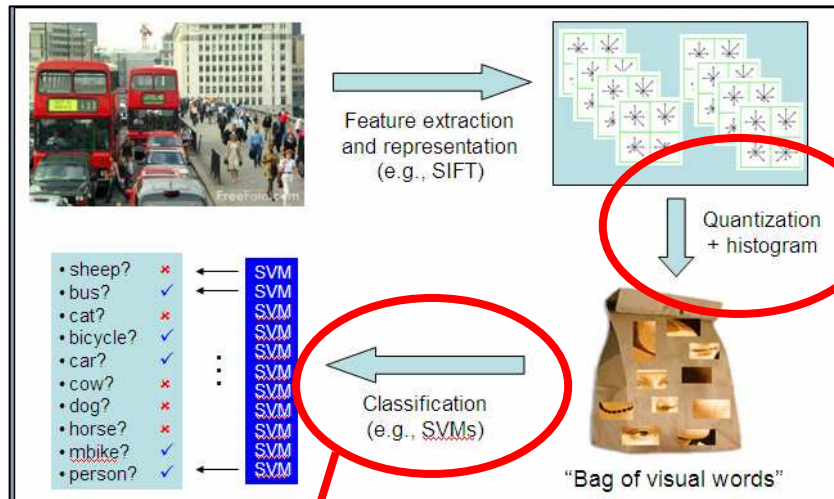
[Liu *et al.*, 2007]
discriminative cluster refinement

Limitation (II):

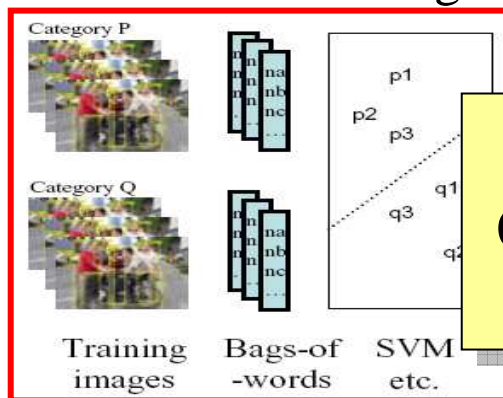
Every SIFT feature forced into one cluster
→ failure to capture partial similarity

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

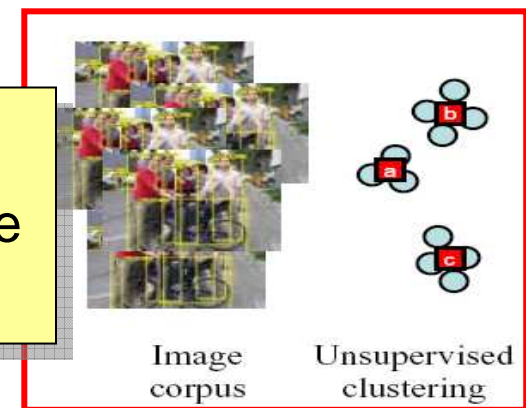
Codebook Construction by Clustering



Classifier Training



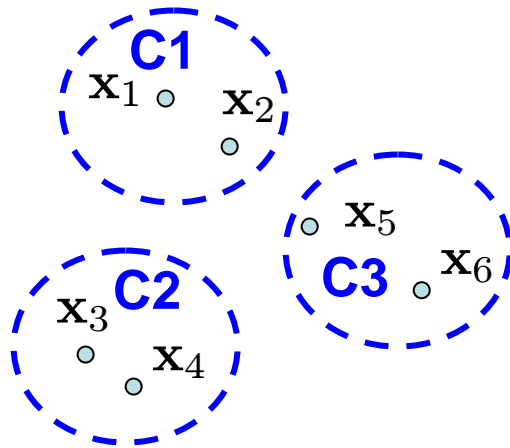
Codebook Generation



Limitations (cont.):
Codebook may not be discriminative to differentiate object categories

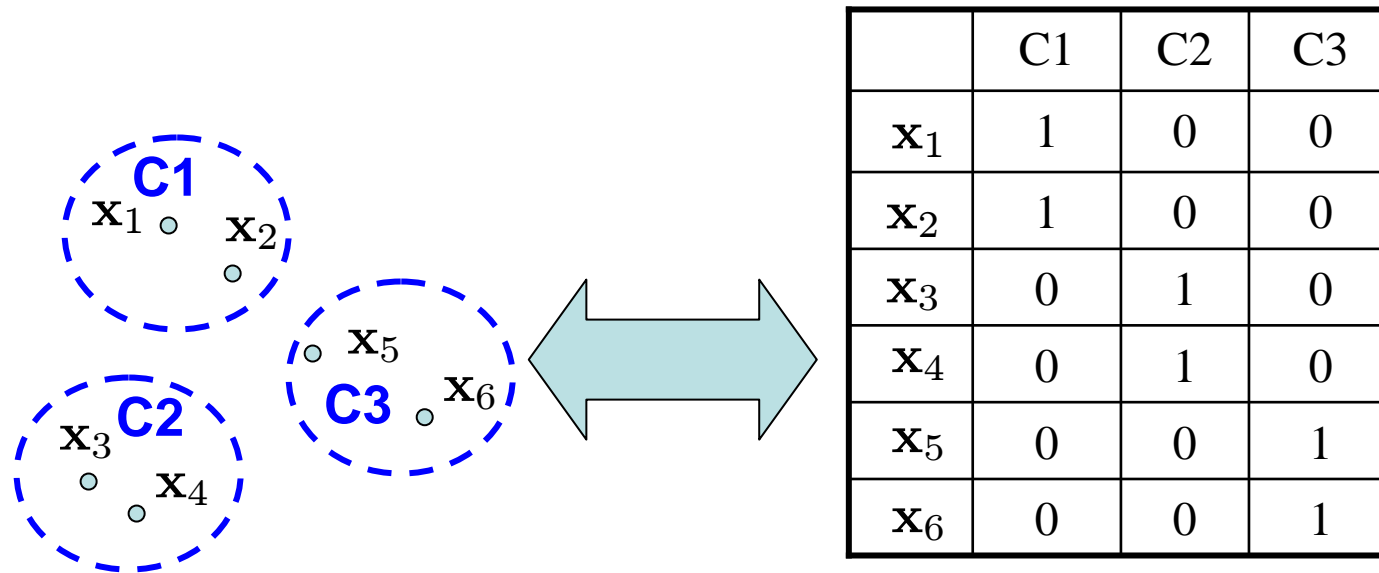
Understanding Clustering

- Clustering is a special coding



Understanding Clustering

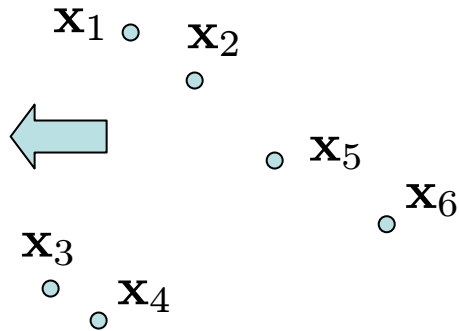
- Clustering is a special coding
 - One and only one bit is on
- More general coding
 - Error Correcting Output Code (ECOC)



Understanding Clustering

- Clustering is a special coding
 - One and only one bit is on
- Our approach: coding by thresholded projections

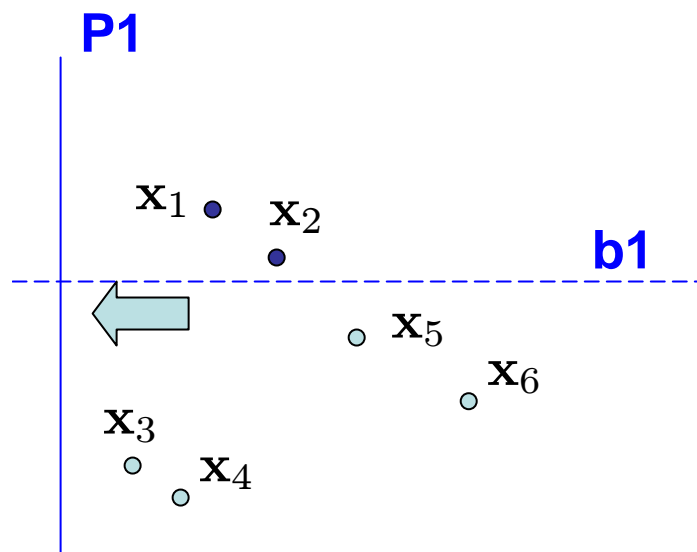
P1



	P1	P2	P3	P4
x_1				
x_2				
x_3				
x_4				
x_5				
x_6				

Understanding Clustering

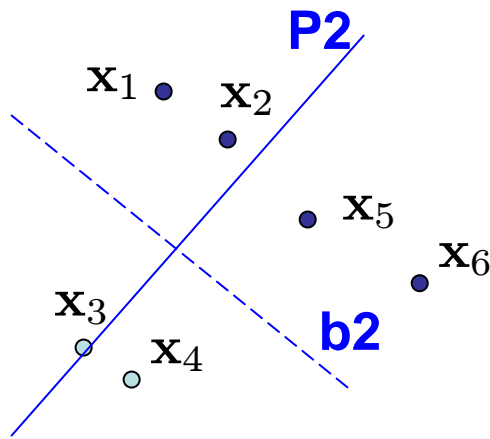
- Clustering is a special coding
 - One and only one bit is on
- Our approach: coding by thresholded projections



	P1	P2	P3	P4
x_1	1			
x_2	1			
x_3	0			
x_4	0			
x_5	0			
x_6	0			

Understanding Clustering

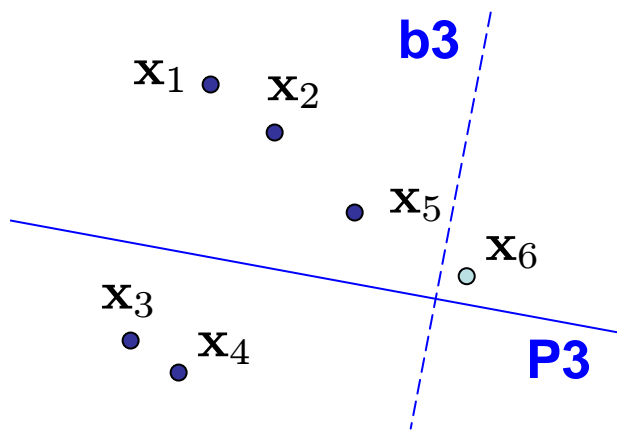
- Clustering is a special coding
 - One and only one bit is on
- Our approach: coding by thresholded projections



	P1	P2	P3	P4
x ₁	1	1		
x ₂	1	1		
x ₃	0	0		
x ₄	0	0		
x ₅	0	1		
x ₆	0	1		

Understanding Clustering

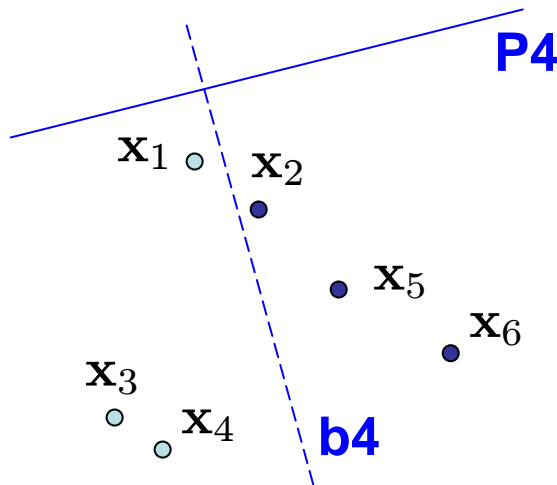
- Clustering is a special coding
 - One and only one bit is on
- Our approach: coding by thresholded projections



	P1	P2	P3	P4
x_1	1	1	1	
x_2	1	1	1	
x_3	0	0	1	
x_4	0	0	1	
x_5	0	1	1	
x_6	0	1	0	

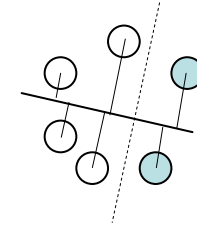
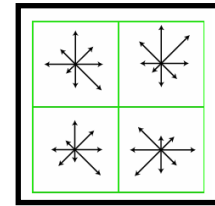
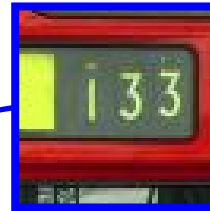
Understanding Clustering

- Clustering is a special coding
 - One and only one bit is on
- 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
x_1	1	1	1	0
x_2	1	1	1	1
x_3	0	0	1	0
x_4	0	0	1	0
x_5	0	1	1	1
x_6	0	1	0	1

Anatomy of a Visual Bit



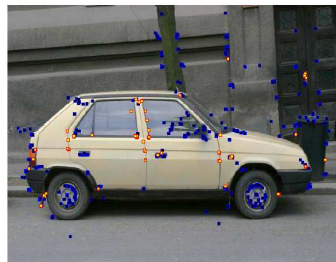
$$g_k(\mathbf{x}, y) = I(\mathbf{x}^\top \mathbf{w}_k^y - b_k^y) = \begin{cases} 1 & \mathbf{x}^\top \mathbf{w}_k^y > b_k^y \\ 0 & \mathbf{x}^\top \mathbf{w}_k^y \leq b_k^y \end{cases}$$

(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

	$g_1(x,a)$	$g_2(x,a)$...	$g_T(x,a)$
x_1	1	1
x_2	1	0
...
x_n	0	0

feature-level
representation

Image Classification using Visual Bits

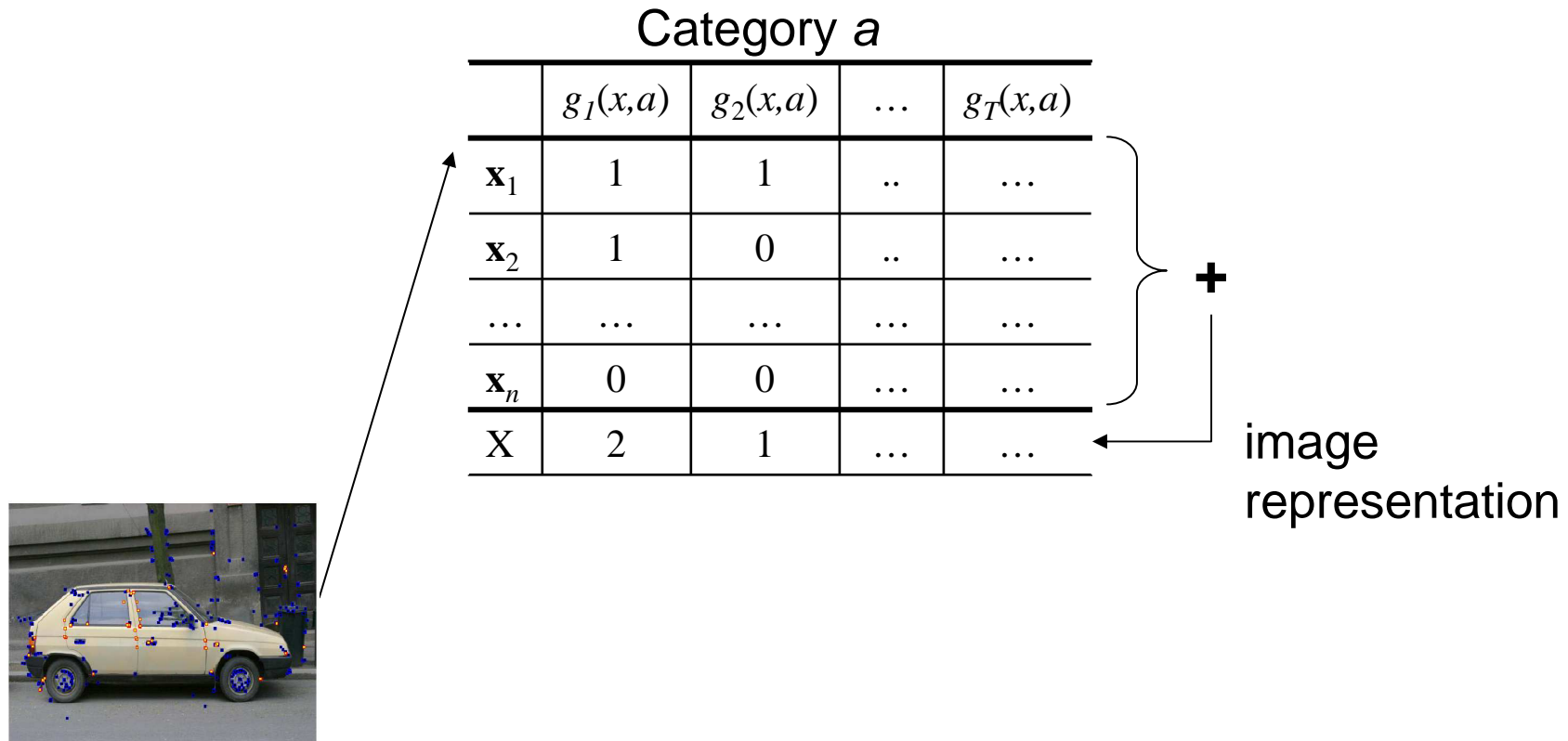


Image Classification using Visual Bits

Category a

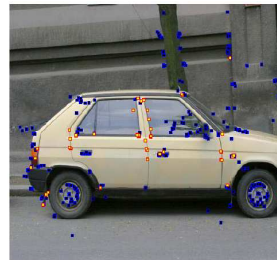
	$g_1(X, a)$	$g_2(X, a)$...	$g_T(X, a)$
X	2	1

Classifier for Cat. a

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



Image Classification using Visual Bits



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

⇕

Unify code generation with discriminative classifier

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

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

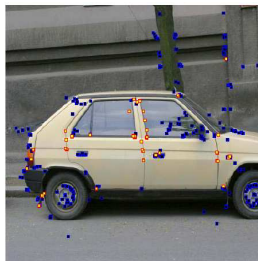
Classifier for Cat. a

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

Category b				

Classifier for Cat. b

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



Generalizes to nonlinear classifier
(can be implemented using standard SVM)

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

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

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

$g_k(X, b)$

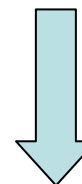
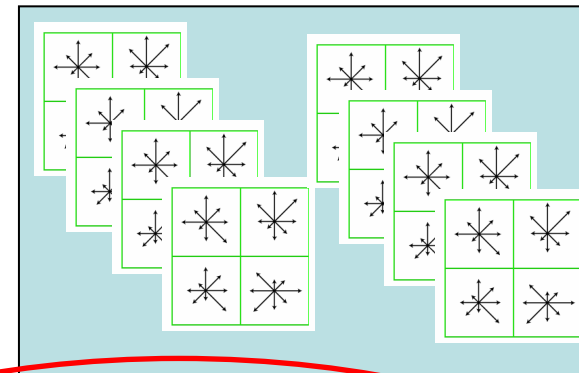
Cat. z

$g_k(X, z)$

Standard Approach



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



Quantization
+ histogram



“Bag of visual words”



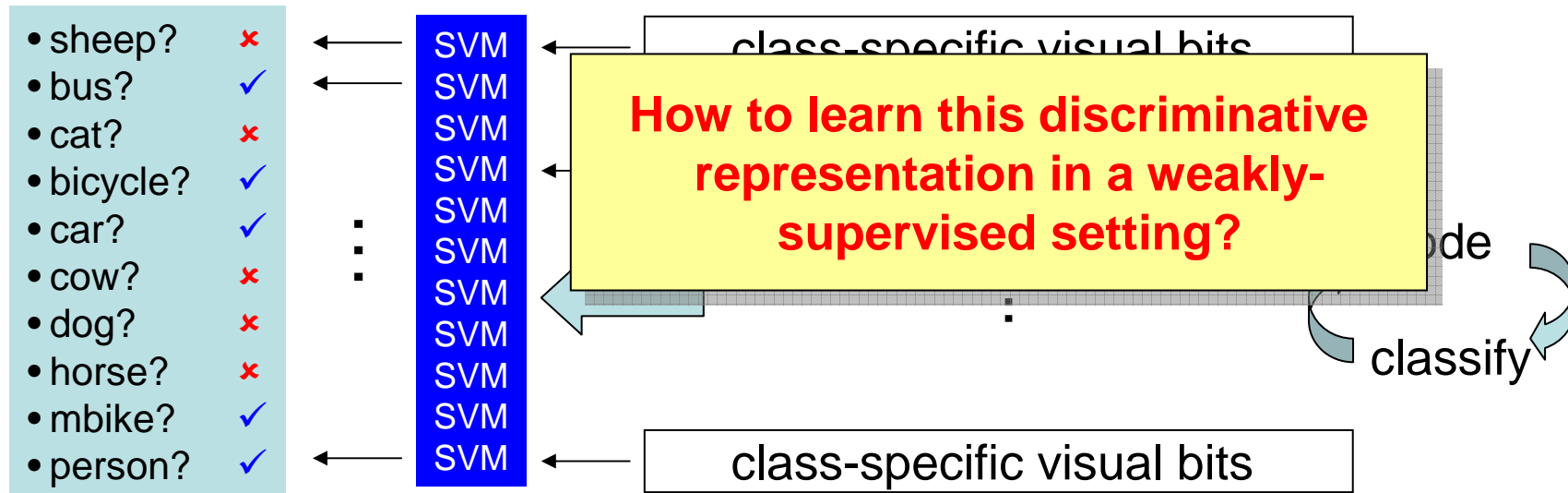
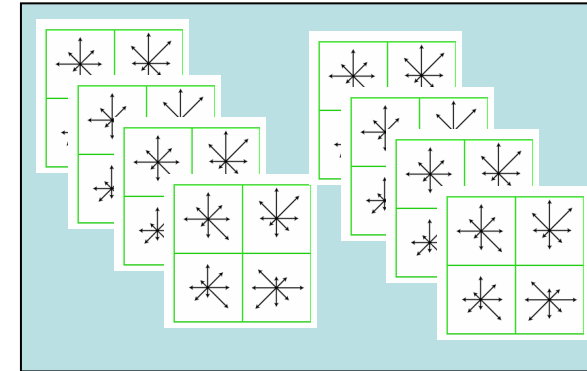
Classification
(e.g., SVMs)

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

Unified Approach



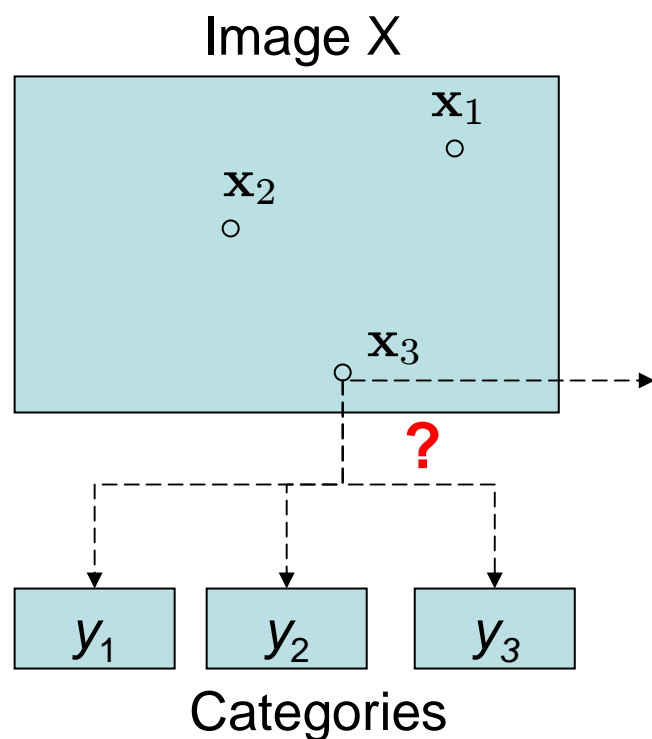
Feature extraction
and representation
(e.g., SIFT)



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 $X=(\mathbf{x}_1, \dots, \mathbf{x}_n)$ into cat. (y_1, y_2, \dots, y_K)



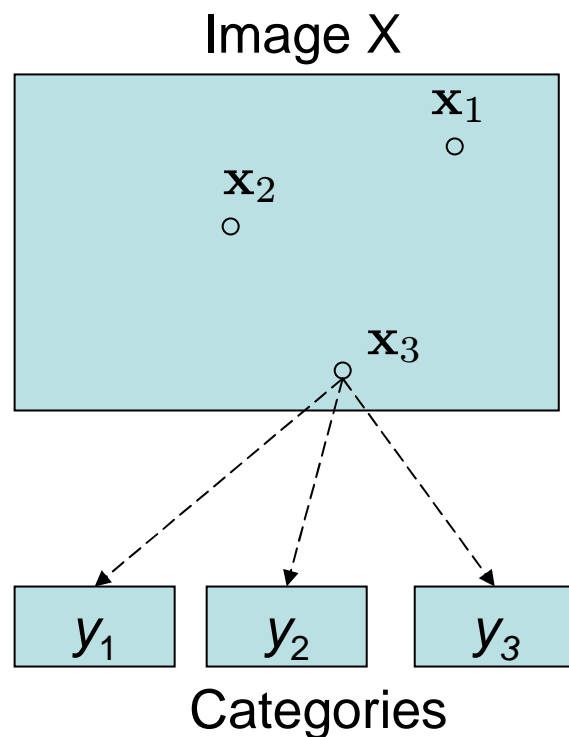
Challenge

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 $X=(\mathbf{x}_1, \dots, \mathbf{x}_n)$ into cat. (y_1, y_2, \dots, y_K)



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

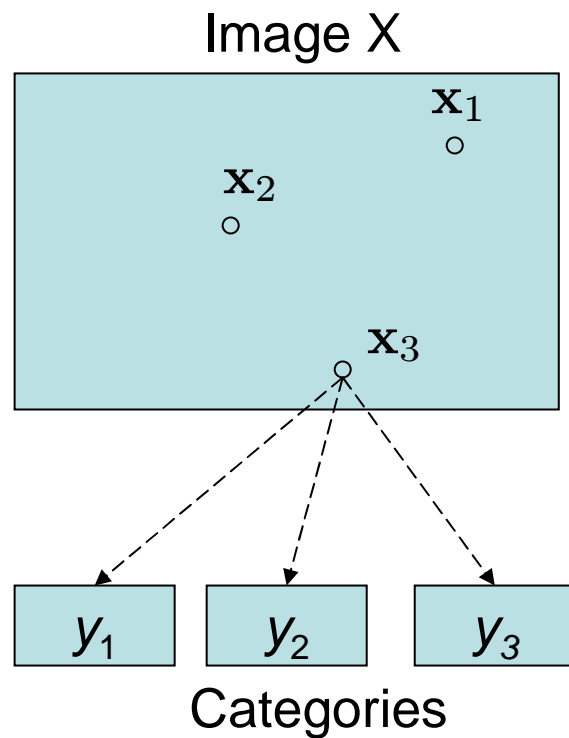
$$f(\mathbf{x}_3, y_2) = \sum_{k=1}^T \alpha_k g_k(\mathbf{x}_3, y_2)$$

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

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 $X=(\mathbf{x}_1, \dots, \mathbf{x}_n)$ into cat. (y_1, y_2, \dots, y_K)



Prob. of associating feature \mathbf{x} with cat. y

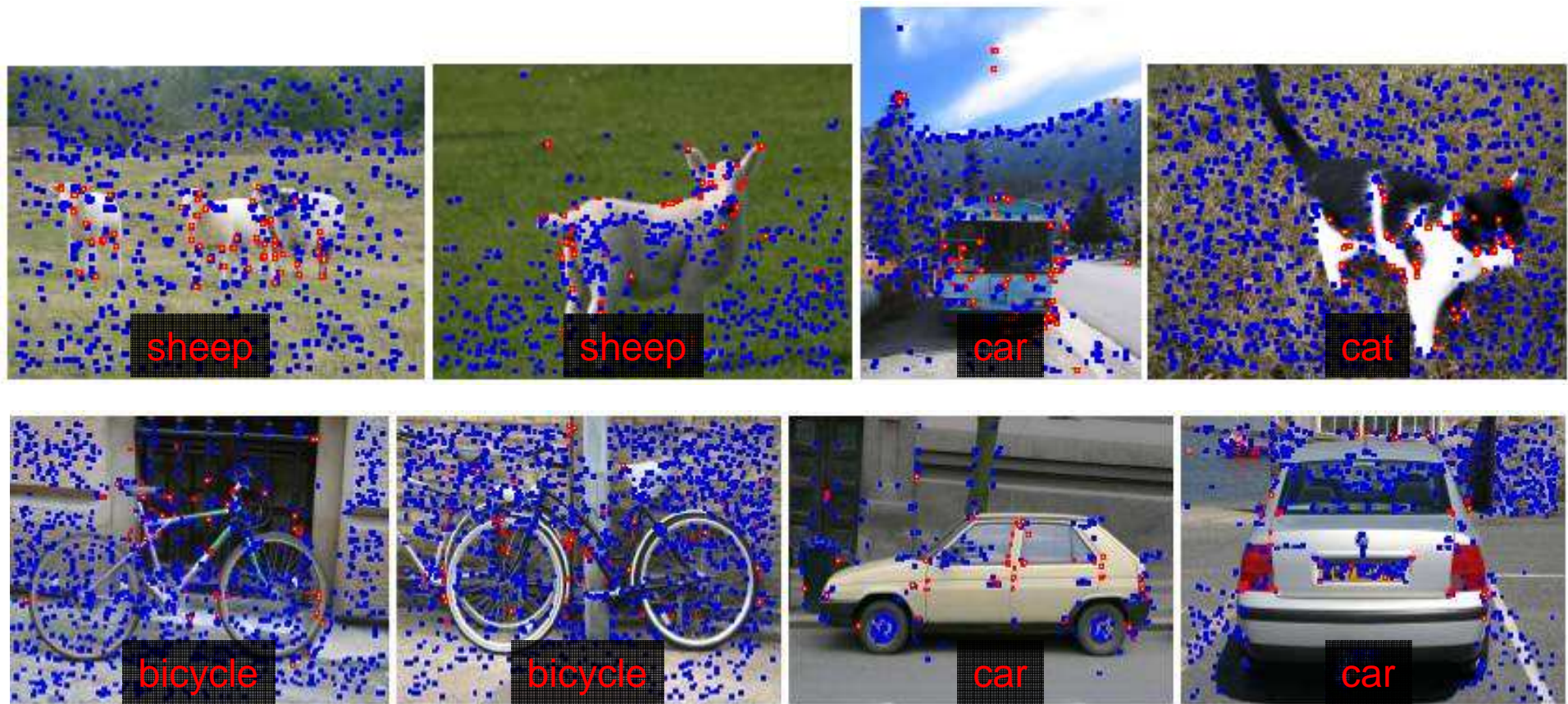
$$f(\mathbf{x}_3, y_1) = \sum_{k=1}^T \alpha_k g_k(\mathbf{x}_3, y_1)$$

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

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

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

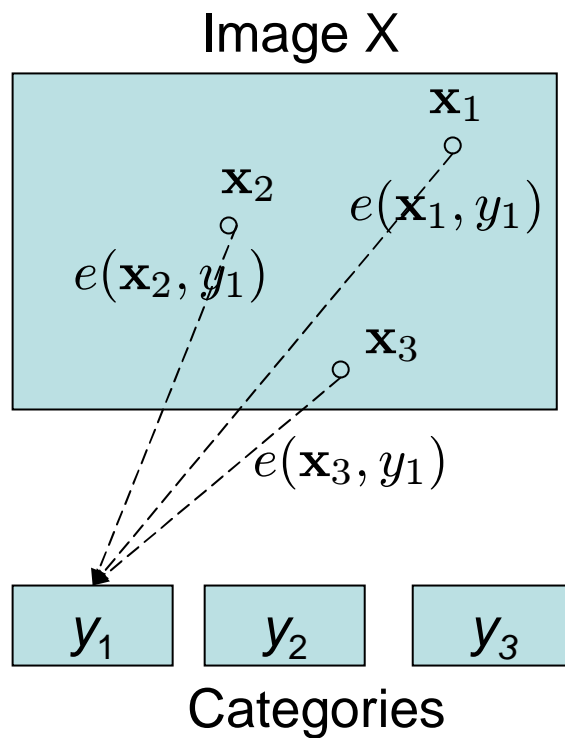
Relevant Visual Bits Localize Concepts



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 $X=(\mathbf{x}_1, \dots, \mathbf{x}_n)$ into cat. (y_1, y_2, \dots, y_K)



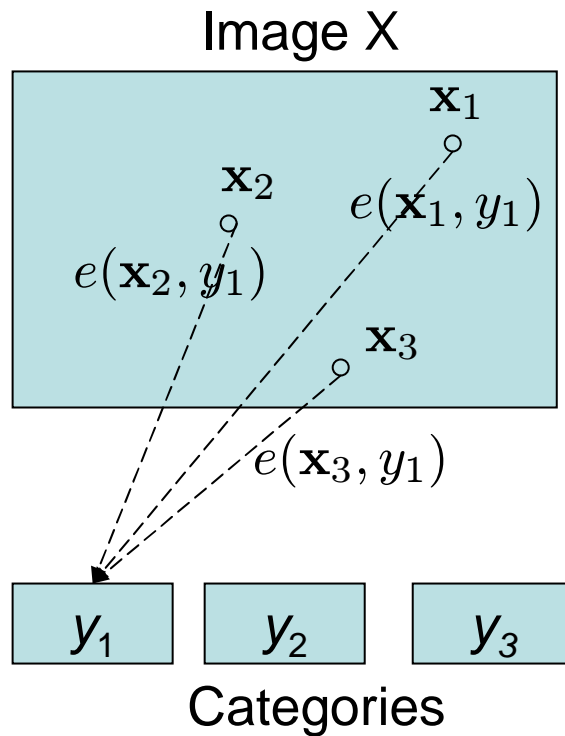
Loss function for image X

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

Learning Visual Bits

Optimization Framework

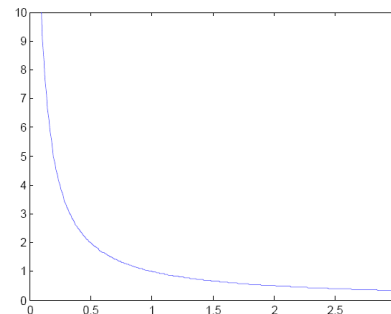
- Given visual bit functions $g(x, a)$ and weights α , how to measure if they are able to classify image $X=(\mathbf{x}_1, \dots, \mathbf{x}_n)$ into cat. (y_1, y_2, \dots, y_K)



Loss function for image X

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

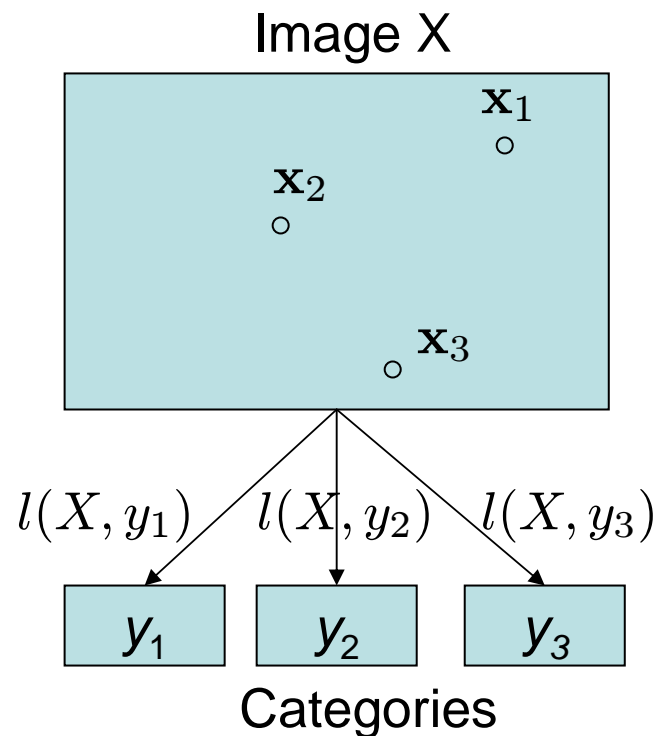
1. Diminishing rewards
2. Relation to exponential loss



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 $X=(\mathbf{x}_1, \dots, \mathbf{x}_n)$ into cat. (y_1, y_2, \dots, 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

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

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