Beyond bag of visual word model for image representation

ERMITES 2013

Matthieu Cord

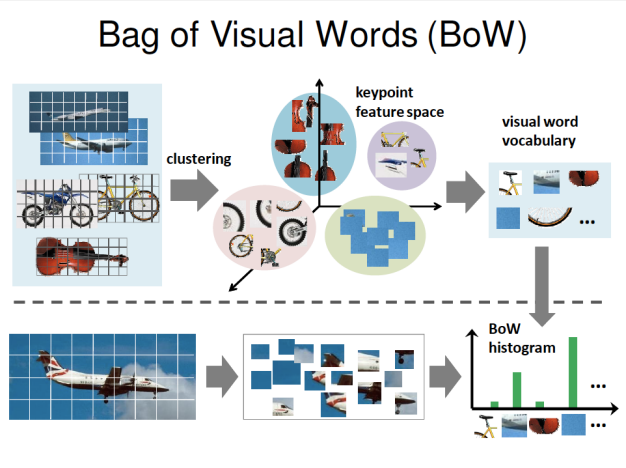
Computer Science dept. (LIP6), UPMC Sorbonne Univ., Paris, France

Outline

1. BoW model
   - BoW representation
   - BoW parametrization
2. Pooling
   - Pooling extension
   - BossaNova
3. Coding
   - Dictionary learning and sparse coding
4. Deep Learning with RBM
   - Deep Learning & Visual Representations
   - Regularizing Latent Representations
   - Deep Supervised Optimization
   - Learning Hierarchical Visual Codes

Image classification pipeline

Bag-of-Visual-Words (BoVW) Model



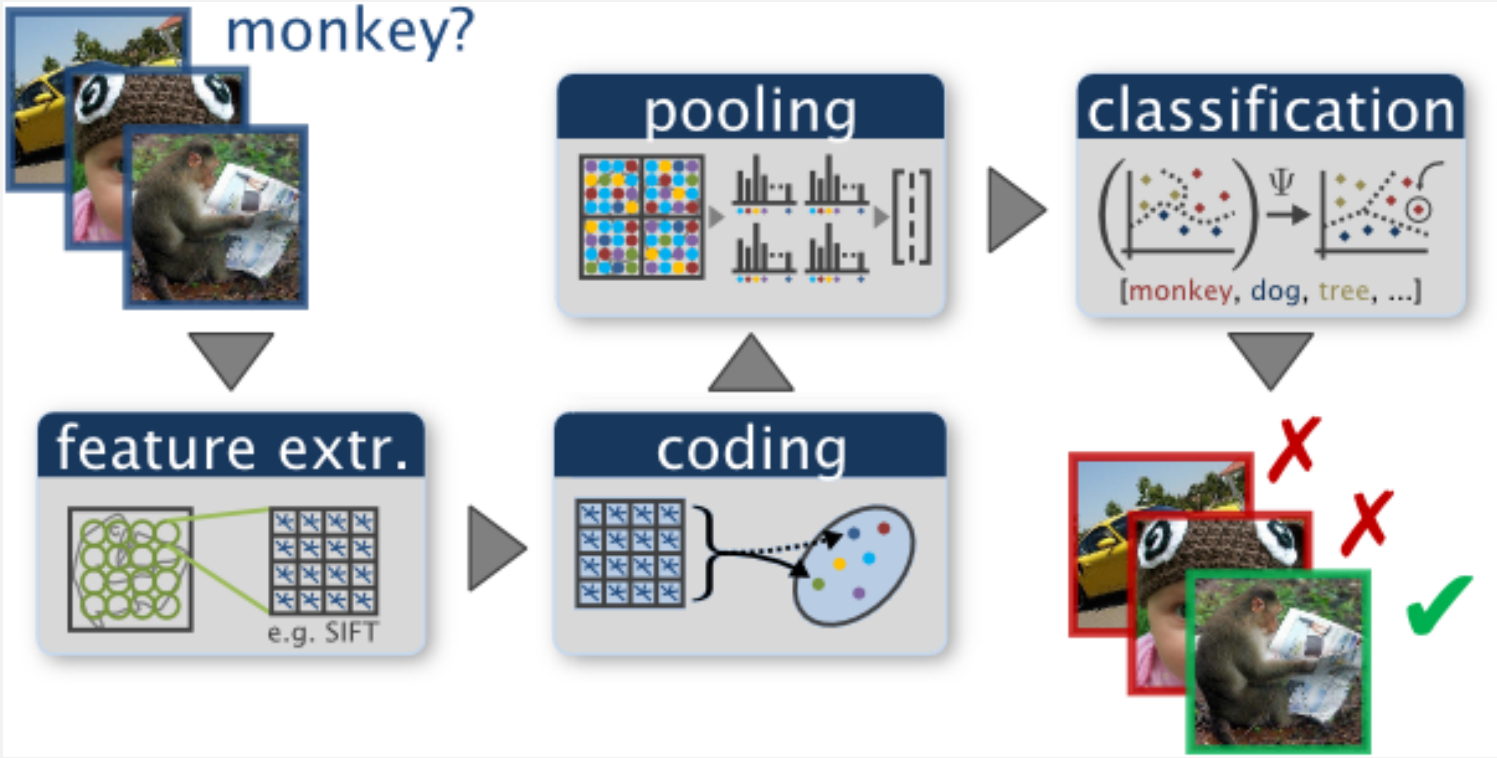
Credit : Prof. Shih-Fu Chang

- 1997 BoW on color features [*Ma*97]
- 2001 BoW on Gabor features [*Fournier*01]
- 2003-4 BoW on SIFT [*Csurka*04]
- 2006 Spatial Information [*Lazebnik*06]
- 2009- Soft-assignement, sparse coding, max pooling [*Wang*10] [*Boureau*10]

[*Ma*97] WY. Ma, BS Manjunath. Netra : A toolbox for navigating large image databases, IEEE ICIP97
[*Fournier*01] J. Fournier, M. Cord, S Philipp. Retin : A content-based image indexing and retrieval system, PAA01
[*Lazebnik*06] P.Lazebnik.S, Schmid.C. Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories CVPR2006.
[*Boureau*10]Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition CVPR2010.
[*Wang*10]J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Locality-constrained linear coding for image classification CVPR10.

Image classification : BoW structure

Coding/Pooling



Credit : K. Chatfield

BoW Model

$\mathbf{X} = (x_1, \ldots, x_j, \ldots, x_N)$ the set of local descriptors (SIFT) for the image
$\mathbf{C} = (c_1, \ldots, c_m, \ldots, c_M)$ the visual dictionary

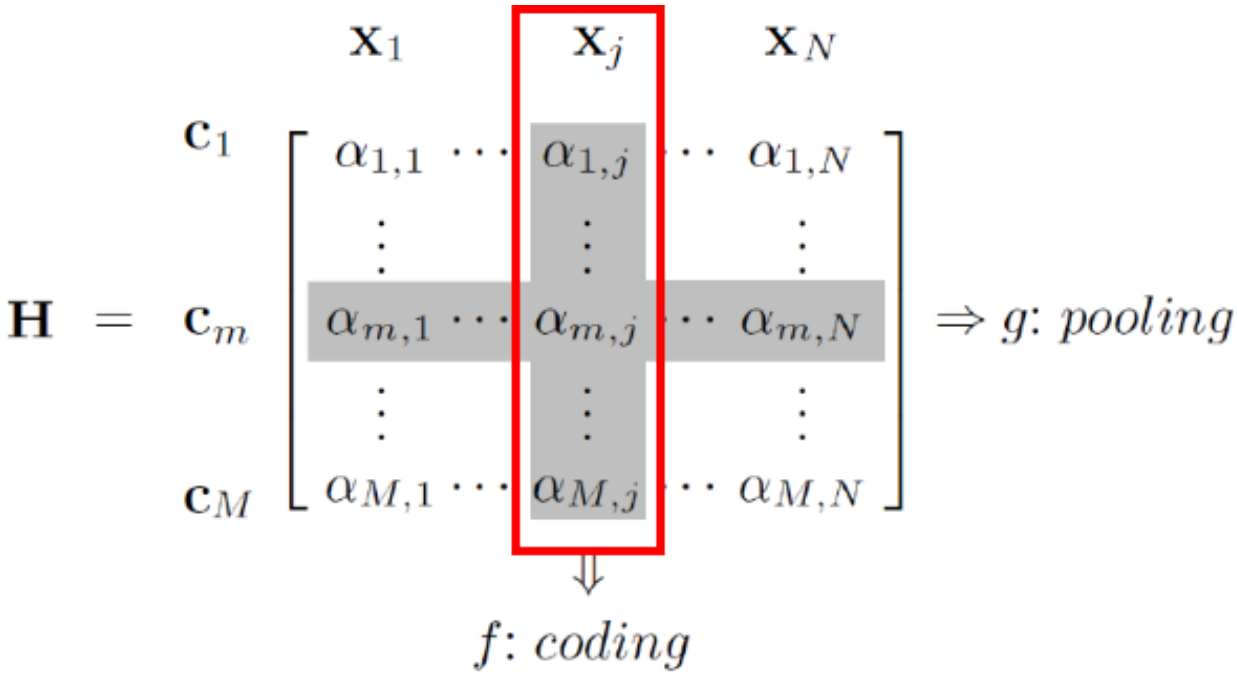
$$\mathbf{H} = \begin{array}{c} \\ \mathbf{c}_1 \\ \\ \mathbf{c}_m \\ \\ \mathbf{c}_M \end{array} \begin{array}{c} \begin{array}{ccccc} \mathbf{x}_1 & & \mathbf{x}_j & & \mathbf{x}_N \end{array} \\ \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{bmatrix} \end{array} \Rightarrow g\text{: } pooling$$

$$\Downarrow$$

$$f\text{: } coding$$

BoW Model

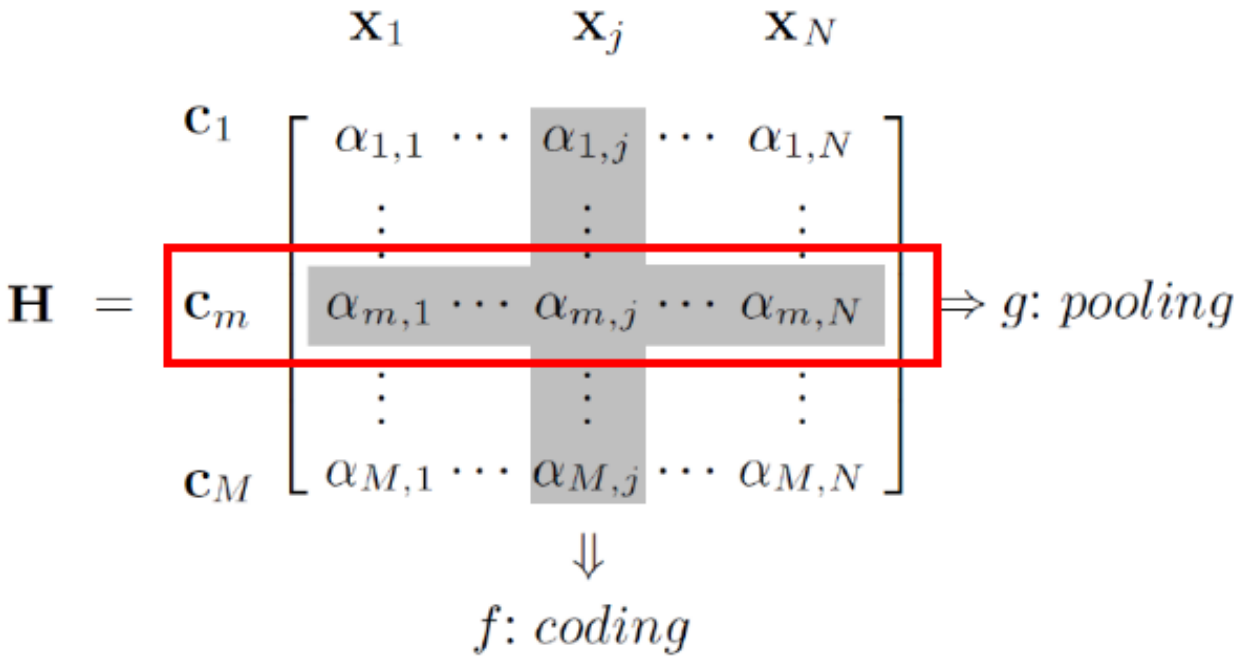
$\mathbf{X} = (x_1, \ldots, x_j, \ldots, x_N)$ the set of local descriptors (SIFT) for the image
$\mathbf{C} = (c_1, \ldots, c_m, \ldots, c_M)$ the visual dictionary



Coding : $\mathbf{x}_j \rightarrow f(\mathbf{x}_j) = \{\alpha_{m,j}\}, \quad \alpha_{m,j} = 1$ iff $m = \underset{k \in \{1,\ldots,M\}}{\arg\min} \|\mathbf{x}_j - \mathbf{c}_k\|_2^2$

BoW Model

$\mathbf{X} = (x_1, \ldots, x_j, \ldots, x_N)$ the set of local descriptors (SIFT) for the image
$\mathbf{C} = (c_1, \ldots, c_m, \ldots, c_M)$ the visual dictionary



Coding : $\mathbf{x}_j \rightarrow f(\mathbf{x}_j) = \{\alpha_{m,j}\}, \quad \alpha_{m,j} = 1$ iff $m = \underset{k \in \{1,\ldots,M\}}{\arg\min} \|\mathbf{x}_j - \mathbf{c}_k\|_2^2$

Pooling : $g(\{\alpha_j\}) = \mathbf{z} : \ \forall m, \ z_m = \sum_{j=1}^{N} \alpha_{m,j}$

BoW Model

$\mathbf{X} = (x_1, \ldots, x_j, \ldots, x_N)$ the set of local descriptors (SIFT) for the image
$\mathbf{C} = (c_1, \ldots, c_m, \ldots, c_M)$ the visual dictionary

$$\mathbf{H} = \begin{array}{c} \\ \mathbf{c}_1 \\ \\ \mathbf{c}_m \\ \\ \mathbf{c}_M \end{array} \begin{array}{c} \begin{array}{ccccc} \mathbf{x}_1 & & \mathbf{x}_j & & \mathbf{x}_N \end{array} \\ \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{bmatrix} \end{array} \Rightarrow g\text{: } pooling$$

$$\Downarrow$$

$$f\text{: } coding$$

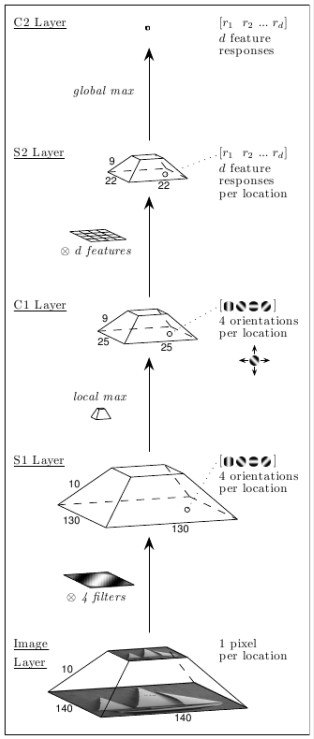
Coding : $\mathbf{x}_j \rightarrow f(\mathbf{x}_j) = \{\alpha_{m,j}\}, \quad \alpha_{m,j} = 1 \text{ iff } m = \underset{k \in \{1,\ldots,M\}}{\arg\min} \|\mathbf{x}_j - \mathbf{c}_k\|_2^2$

Pooling : $g(\{\alpha_j\}) = \mathbf{z} : \ \forall m, \ z_m = \sum_{j=1}^{N} \alpha_{m,j}$

BoW representation : $\mathbf{z} = [z_1, z_2, \cdots, z_M]^{\mathsf{T}}$

Image classification : other approaches

Biologically-inspired Methods [*Fidler*08, *Serre*07, *Mutch*08]



- Mimics feedforward properties of primate visual cortex V1 simple cells
- Based on the HMAX model [*Serre*07, *Mutch*08]
  - $\oplus$ Deep models
  - $\oplus$ Trainable with real images

[*Fidler*08]S. Fidler, B. Boben, and A. Leonardis. Similarity-based cross-layered hierarchical representation for object categorization CVPR2008.
[*Serre*07] T. Serre, *et.al*, Robust object recognition with cortex-like mechanisms, PAMI, 2007.
[*Mutch*08] Mutch.J and Lowe.D.G, Object class recognition and localization using sparse features with limited receptive fields, IJCV, 2008

Image classification : other approaches

Biologically-inspired Methods

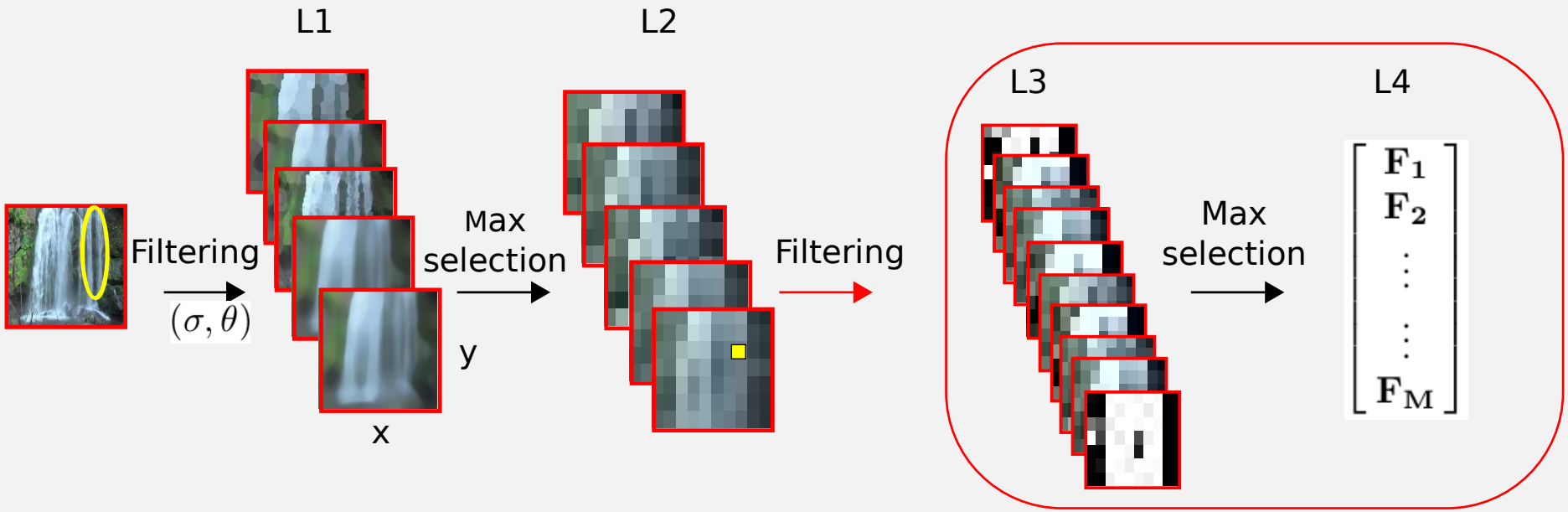


Image classification : other approaches

Biologically-inspired Methods

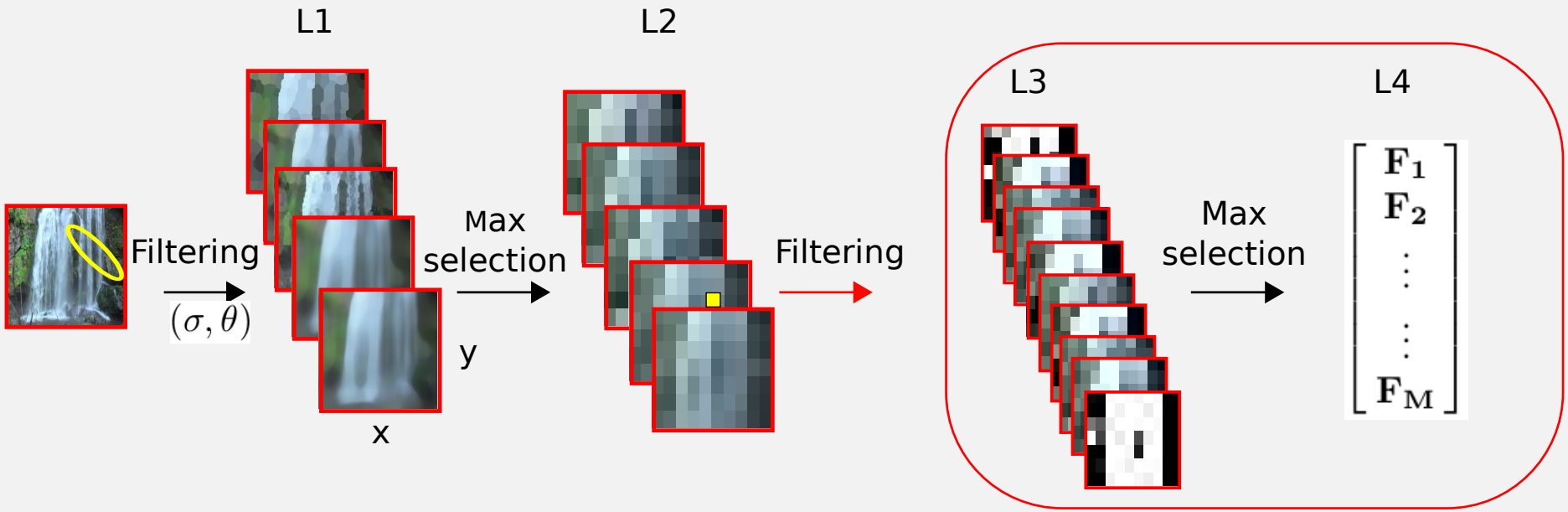


Image classification : other approaches

Biologically-inspired Methods

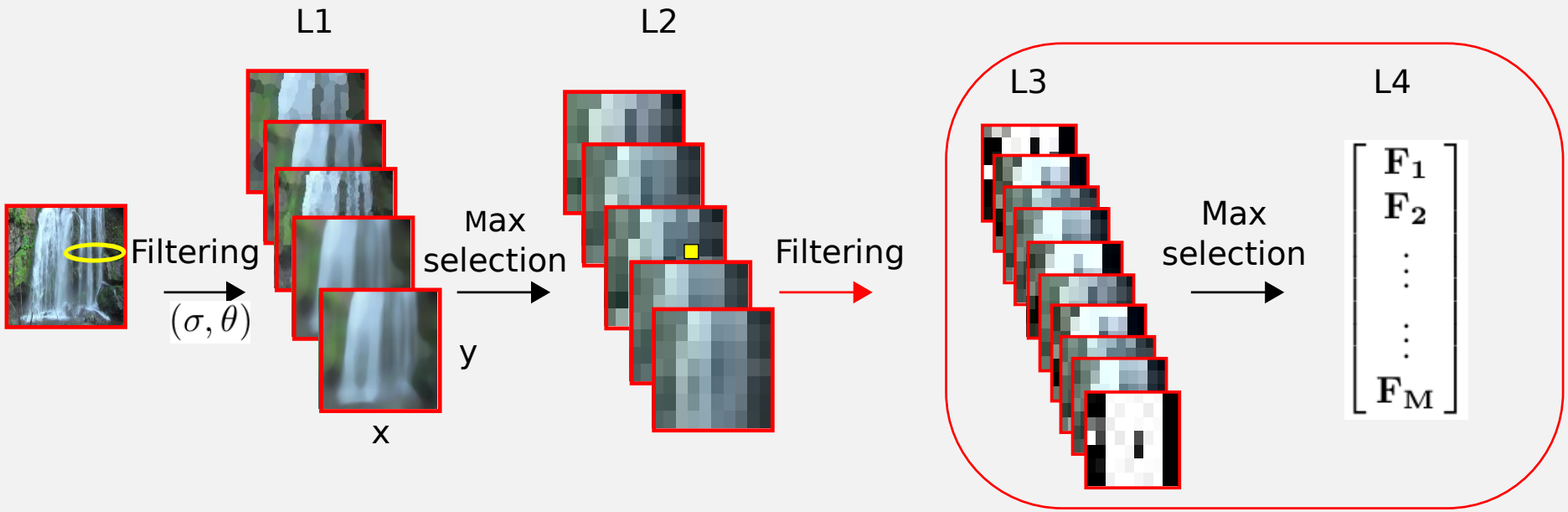


Image classification : other approaches

Biologically-inspired Methods

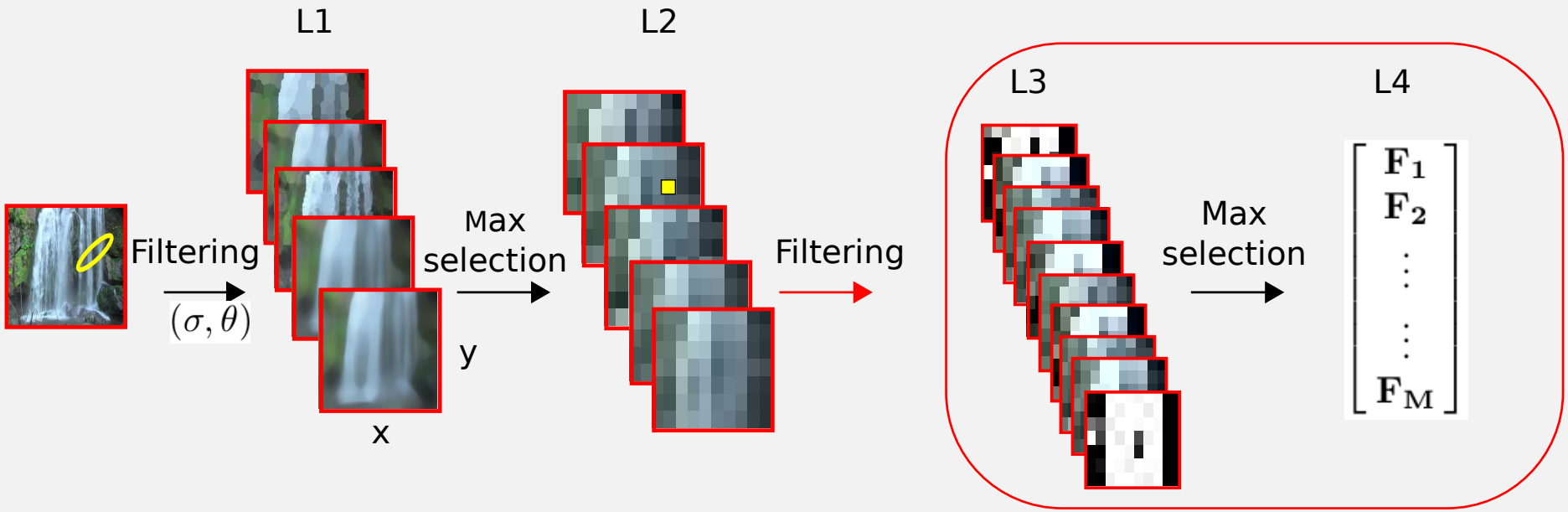


Image classification : other approaches

Biologically-inspired Methods

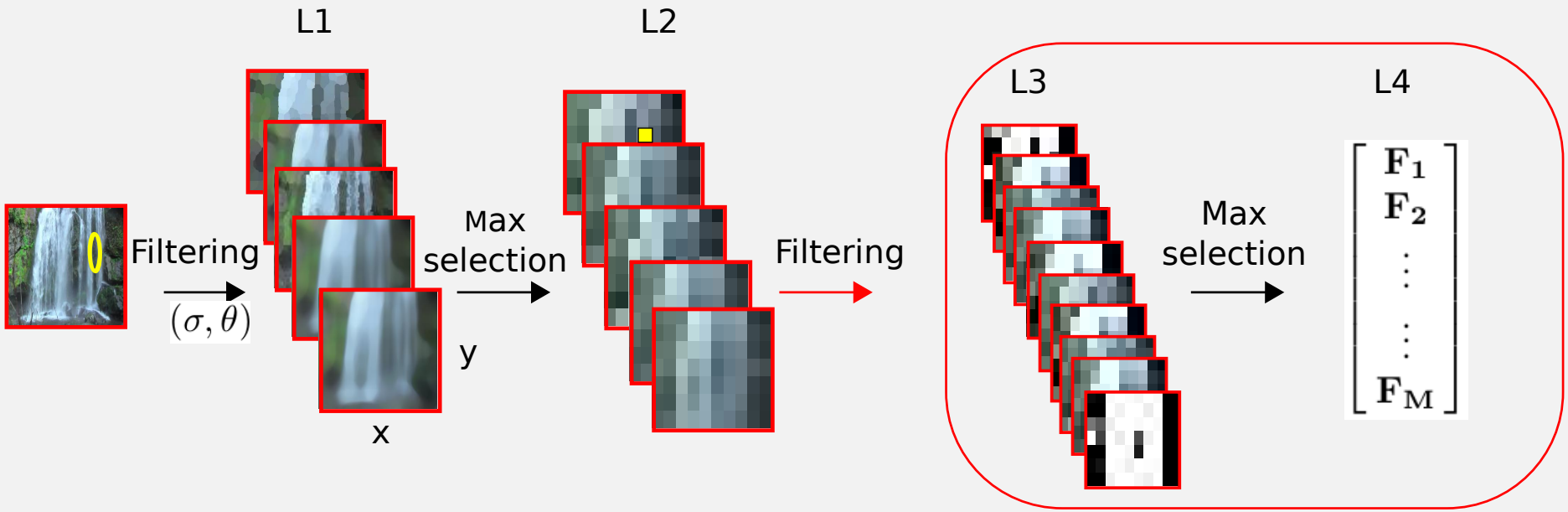


Image classification : other approaches

Biologically-inspired Methods

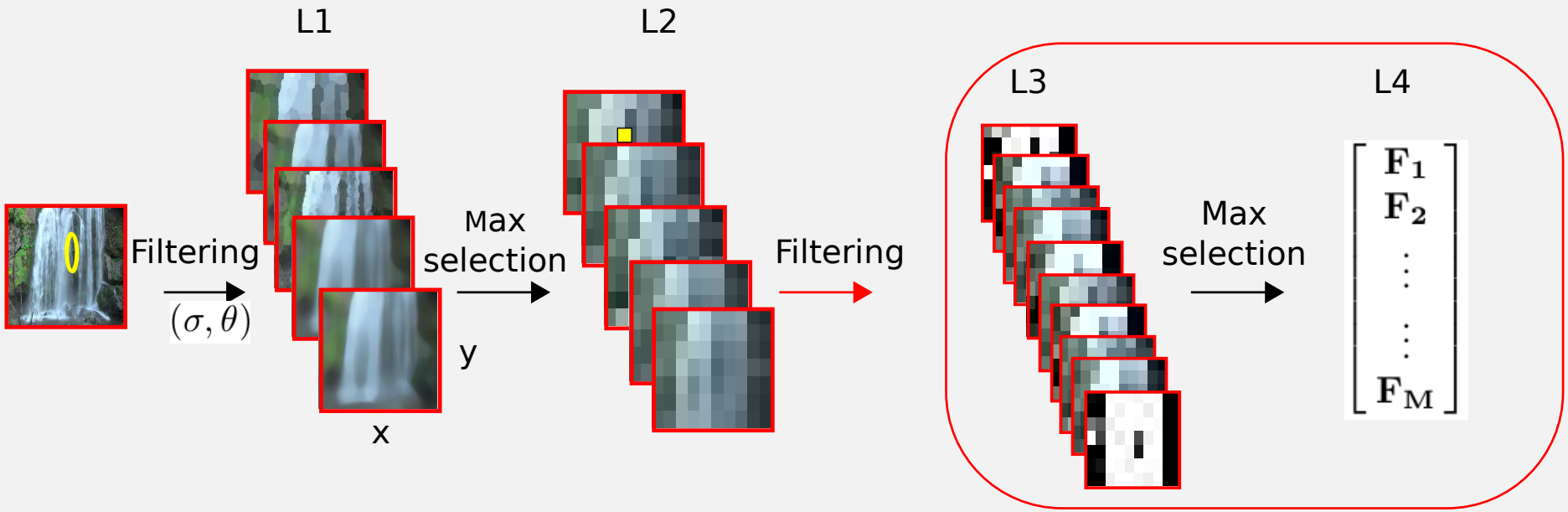


Image classification

Image classification : other approaches

Biologically-inspired Methods

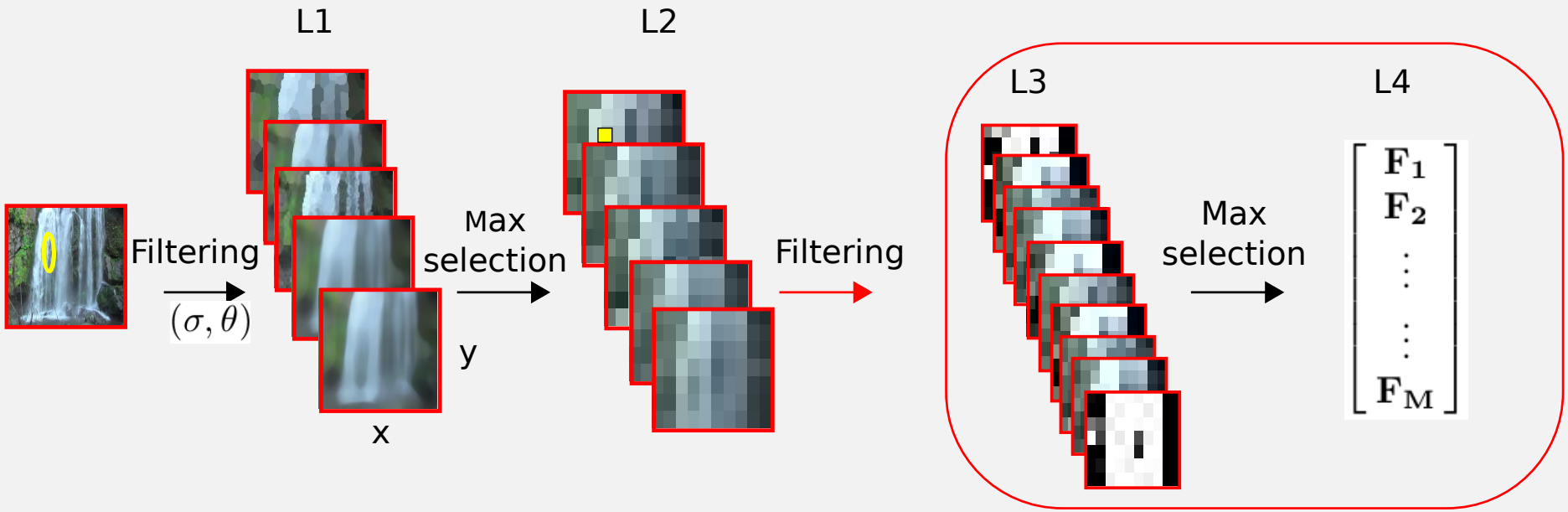


Image classification : other approaches

Biologically-inspired Methods

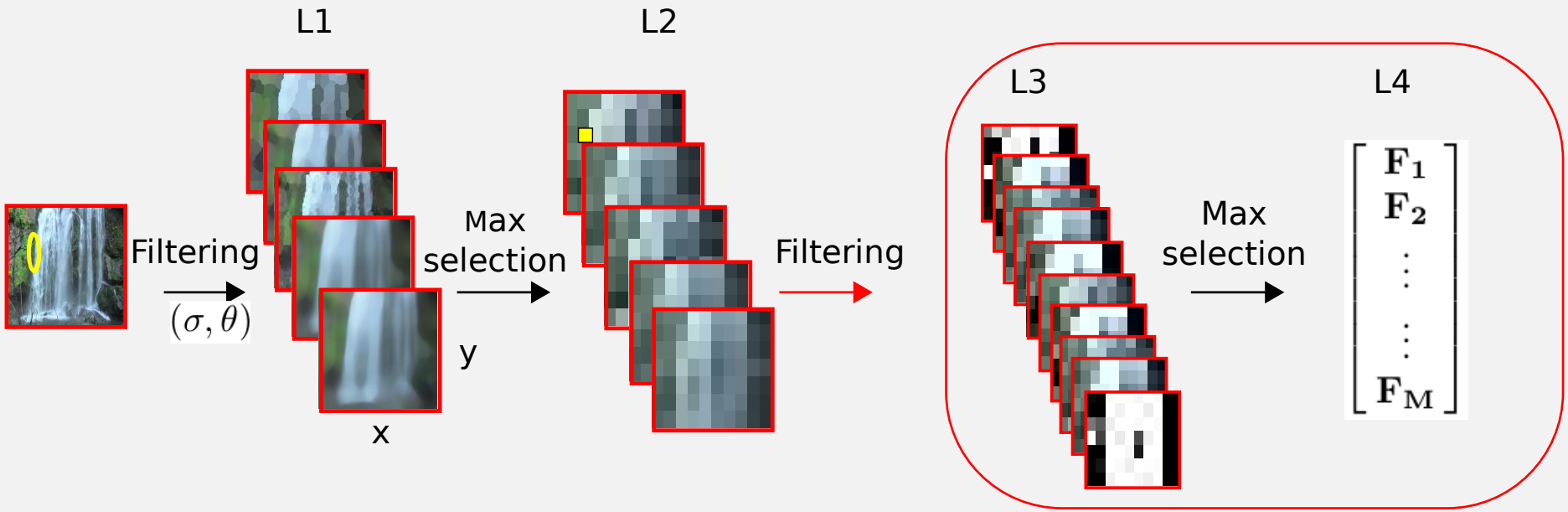


Image classification : other approaches

Biologically-inspired Methods

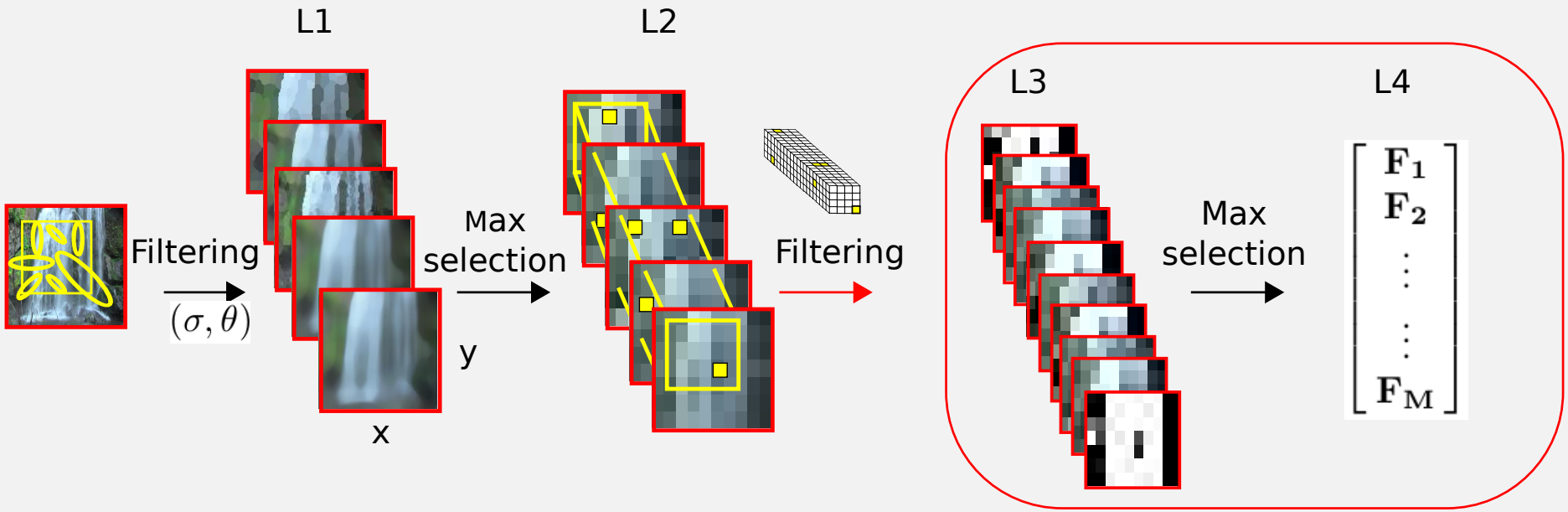


Image classification : other approaches

Biologically-inspired Methods

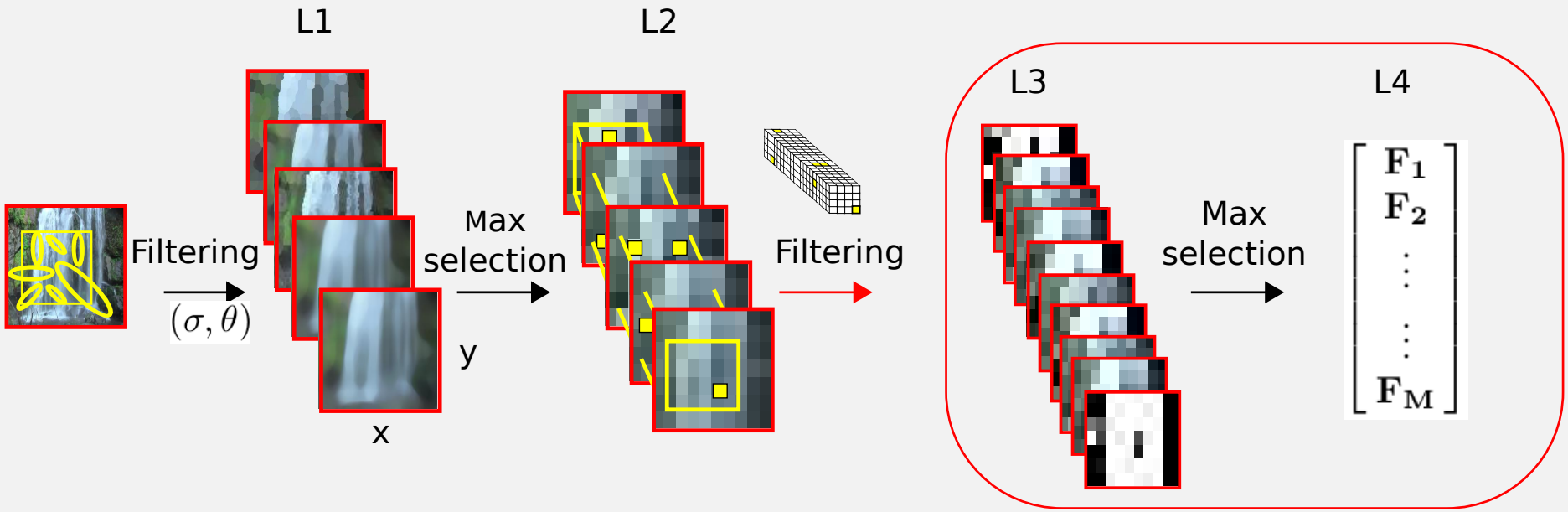


Image classification : other approaches

Biologically-inspired Methods

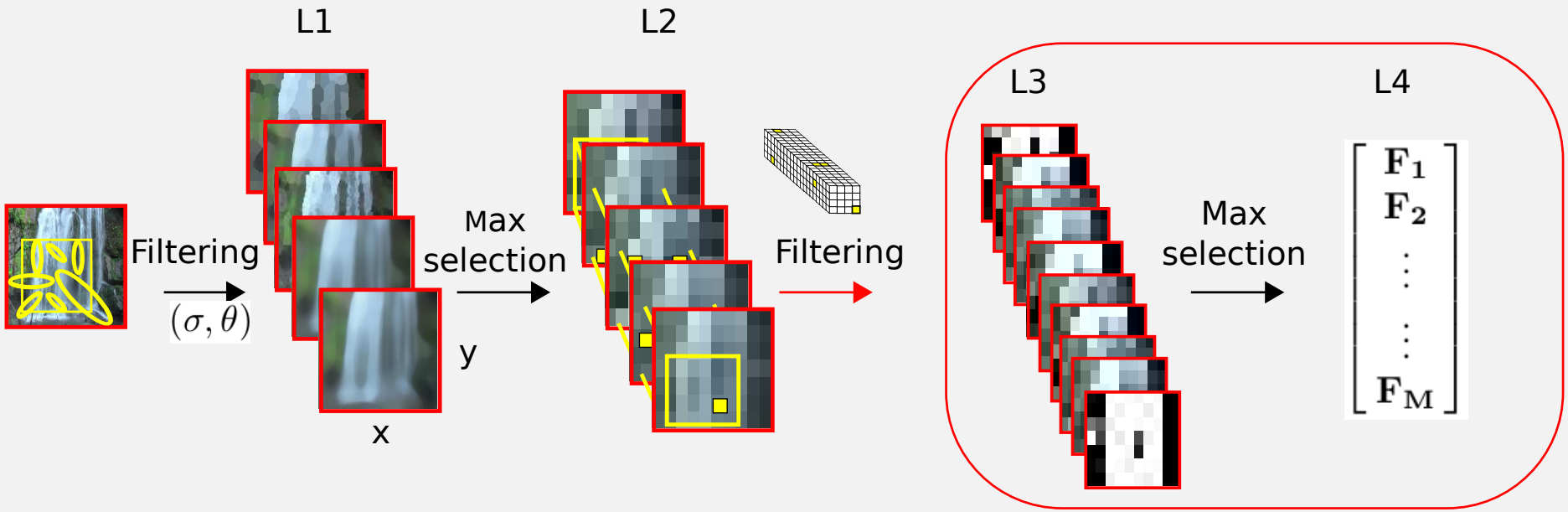


Image classification : other approaches

Biologically-inspired Methods

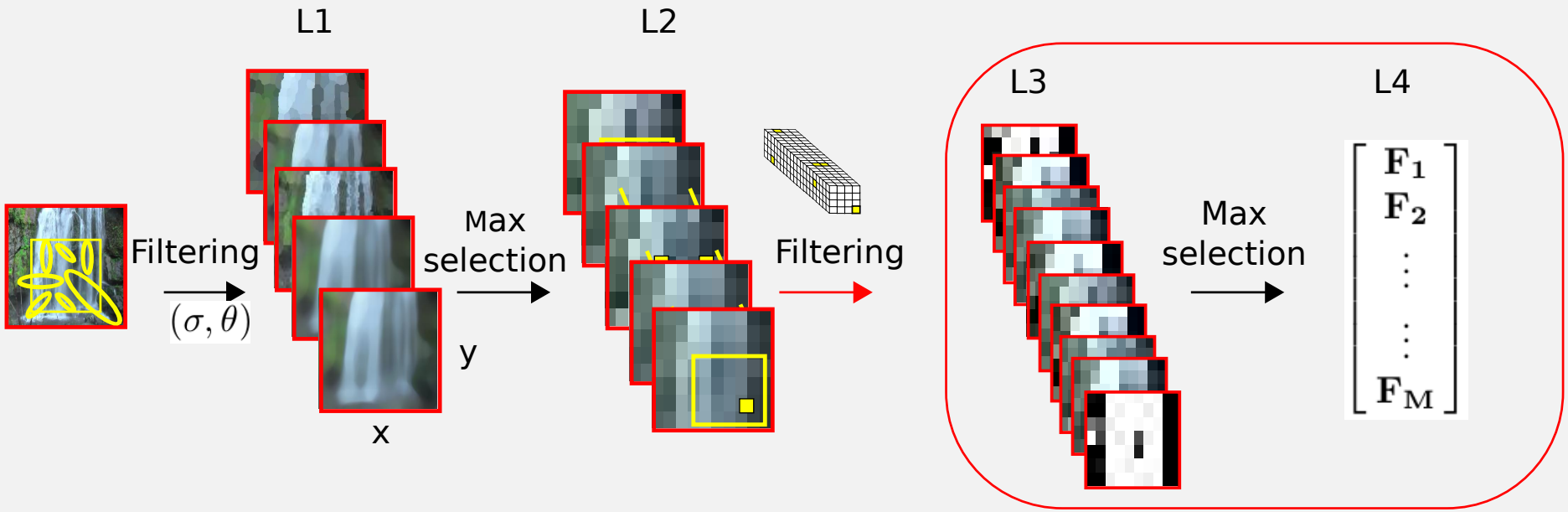


Image classification : other approaches

Biologically-inspired Methods

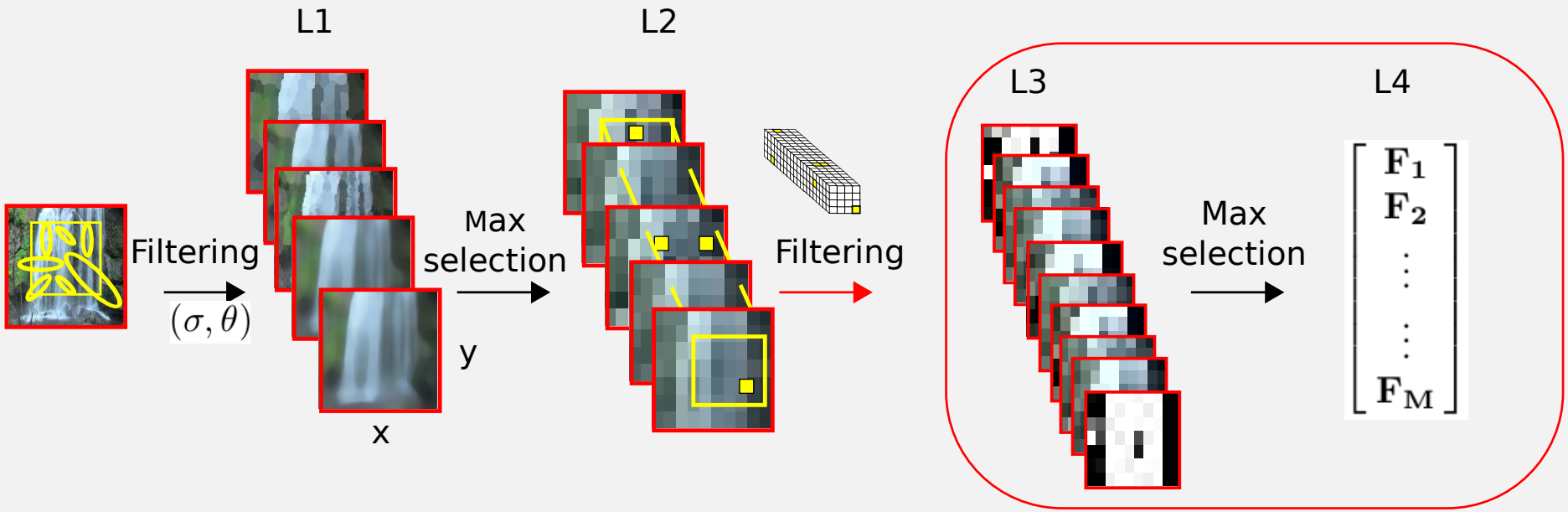


Image classification : other approaches

Biologically-inspired Methods

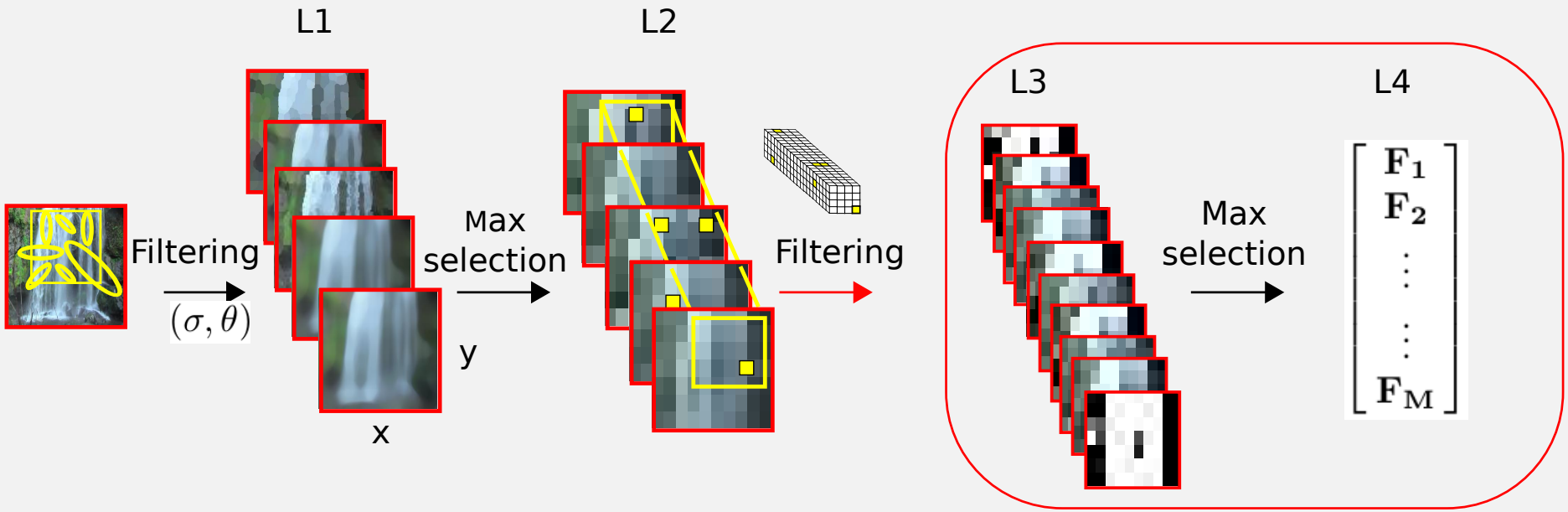
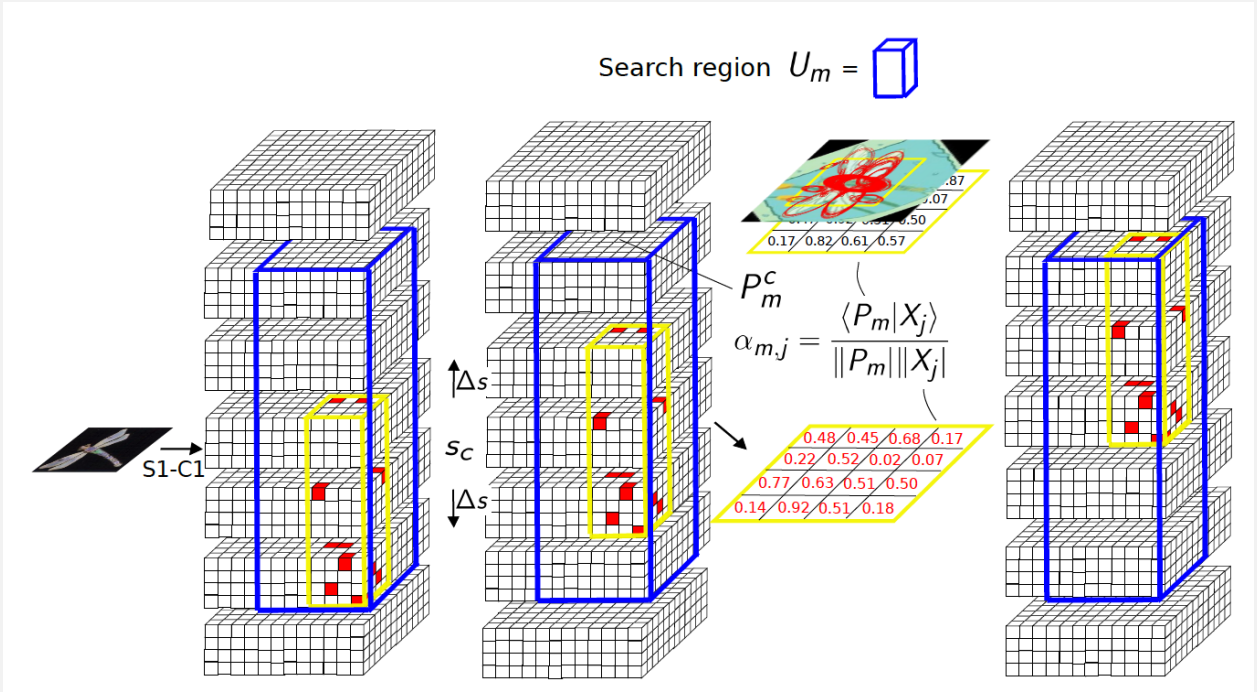


Image classification : other approaches

HMAX model extensions [Theriault11,13]

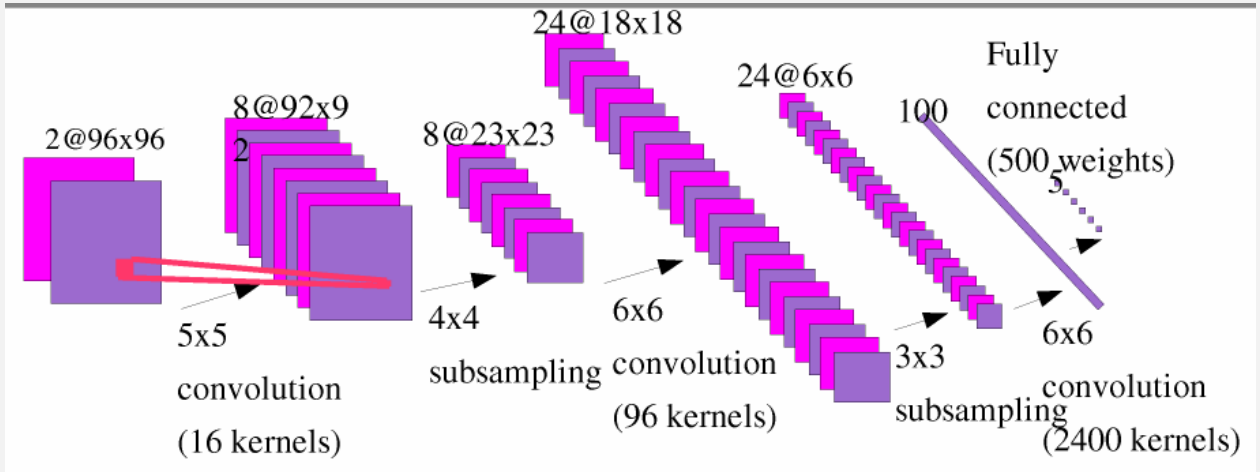


[*Theriault*11] C.Theriault, N. Thome, M. Cord. HMAX-S : Deep scale representation for biologically inspired image categorization Theriault et al. IEEE ICIP11
[*Theriault*13] Extended Coding and Pooling in the HMAX Model, IEEE trans. on Image Processing, 2013

Image classification : other approaches

Deep Networks

- Convolutional networks : [LeCun PhD], improvements [Jarrett09, Lee09]



- Deep Convolutional Neural Networks for large dataset : ImageNet 2012 challenge winner

[*Jarrett*09]K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition ? In Proc ICCV2009.
[*Lee*09]H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. ICML2009
Krizhevsky, A., Sutskever, I. and Hinton, G. E. NIPS 2012

Performance evaluation on Caltech101

Average accuracy results

	15 images	30 images
Bow-like architectures		
[Lazebnik&al CVPR06]	56.4	64.6
[Yang&al CVPR09] sparse coding	67.0	73.2

Hierarchical and biologically inspired architectures		
[Mutch&al IJCV08]	51	56
[Kavukcuoglu&al NIPS10]	-	66.3
[Zeiler&al CVPR2010]	58.6	66.9
[Theriault&al ICIP11]	**60.1 ± 0.2 %**	**69.0 ± 0.8%**

BoW extensions :

- Parametrization, pipeline optimization
- Extended coding, pooling

Optimization of the BoW pipeline

[Chatfield BMVC11] [Law workshop ECCV12]

- Parametrization : find the Winner Cocktail
  - SR : Sampling Rate = gap between centers of patches (pixels)
  - Mono/Multi scale SIFT detection
  - Dictionary size
  - Normalization

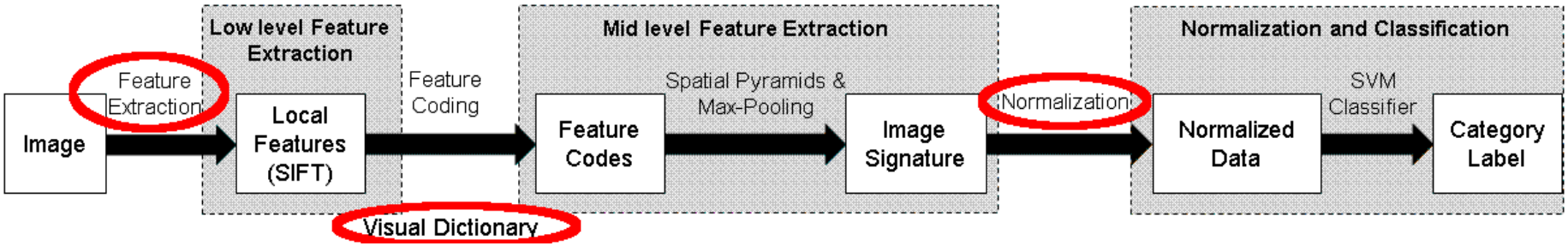


FIGURE: BoW pipeline for classification

SR	Scaling	Codebook Size	Accuracy (no norm)	Acc. (ℓ_2-norm)
8	mono	800	70.07 ± 0.96	70.46 ± 1.04
6	mono	800	71.64 ± 0.99	72.01 ± 0.96
3	mono	800	72.45 ± 1.05	72.73 ± 0.99
8	mono	1700	71.67 ± 0.93	71.95 ± 0.90
8	mono	3300	72.13 ± 0.99	72.50 ± 0.97
8	multi	800	73.35 ± 0.89	73.83 ± 0.96
8	multi	1700	75.34 ± 0.92	75.97 ± 0.86
8	multi	3300	76.91 ± 0.98	77.02 ± 0.94
3	multi	800	73.81 ± 0.95	73.99 ± 0.86
3	multi	1700	75.72 ± 1.13	76.00 ± 0.94
3	multi	3300	77.23 ± 1.02	77.47 ± 0.99
3	multi	6500	78.00 ± 1.05	78.46 ± 0.95

TABLE: Classification results on Caltech-101 with 30 training images per class

SR : Sampling Rate = gap between centers of patches (pixels)

Optimization of the BoW pipeline

- Impact of BoW parameters on PASCAL VOC 2007 and Caltech-101

Method		*codebook size* 256	600	1500	2000	4000	8000
(a) FK	Lin	77.78 ± 0.56	–	–	–	–	–
(b) LLC	Lin	–	73.10 ± 1.09	74.84 ± 0.67	75.75 ± 0.71	76.15 ± 0.59	76.95 ± 0.39
(c) LLC	Chi	–	72.30 ± 1.08	74.23 ± 0.62	75.24 ± 0.71	75.95 ± 0.57	76.62 ± 0.61
(d) VQ	Chi	–	72.65 ± 0.77	73.62 ± 0.51	73.93 ± 0.79	74.41 ± 1.04	74.23 ± 0.65
(e) KCB	Chi	–	73.38 ± 0.65	75.24 ± 0.63	75.50 ± 0.65	75.92 ± 0.63	75.93 ± 0.57

FIGURE: Classification results on the Caltech-101 Dataset. **FK** : Fisher Kernel, **LLC** : Locality-constrained linear Coding, **KCB** : Kernel Codebook, **VQ** : baseline Vector Quantiz. method

- Ultradense sampling ($\sim$ 50,000 features/image)
- Fisher Vector Size : $21 \times 2DK \simeq 21 \times 40,000 \simeq 850,000$ elements
- VLAD [Jégou&al CVPR 2010] and VLAT [Picard&Gosselin ICIP 2011]

Conclusion

	[Law ECCVw 2012]	
	Cal-101	Sc-15
Sampling Rate	X	X
Scaling	XXX	X
Codebook Size	XXX	XXX
Normalization	X	X

TABLE: Importance of parameters

- Huge performance difference according to the chain parameter tuning
  - the devil is in the (parameter) details ... (Chatfield's title)
- Fair comparisons : implementation details
- Sampling rate more important in mono-scale setup

BoW Extensions/Improvements

- Spatial Pyramid Matching [Lazebnik et al., 2006]
- Max pooling [Yang et al., 2009]
- Soft assignment [Gemert et al., 2010]
- LLC [Wang et al., 2010]
- VLAD [Jégou et al., 2010]
- Super-Vector Coding [Zhou et al., 2010]
- Fisher Vector [Perronnin et al., 2010]
- Spatial Fisher Vector [Krapac et al., 2011]
- VLAT [Picard et al., 2011]
- Compact VLAT [Negrel et al., 2012]

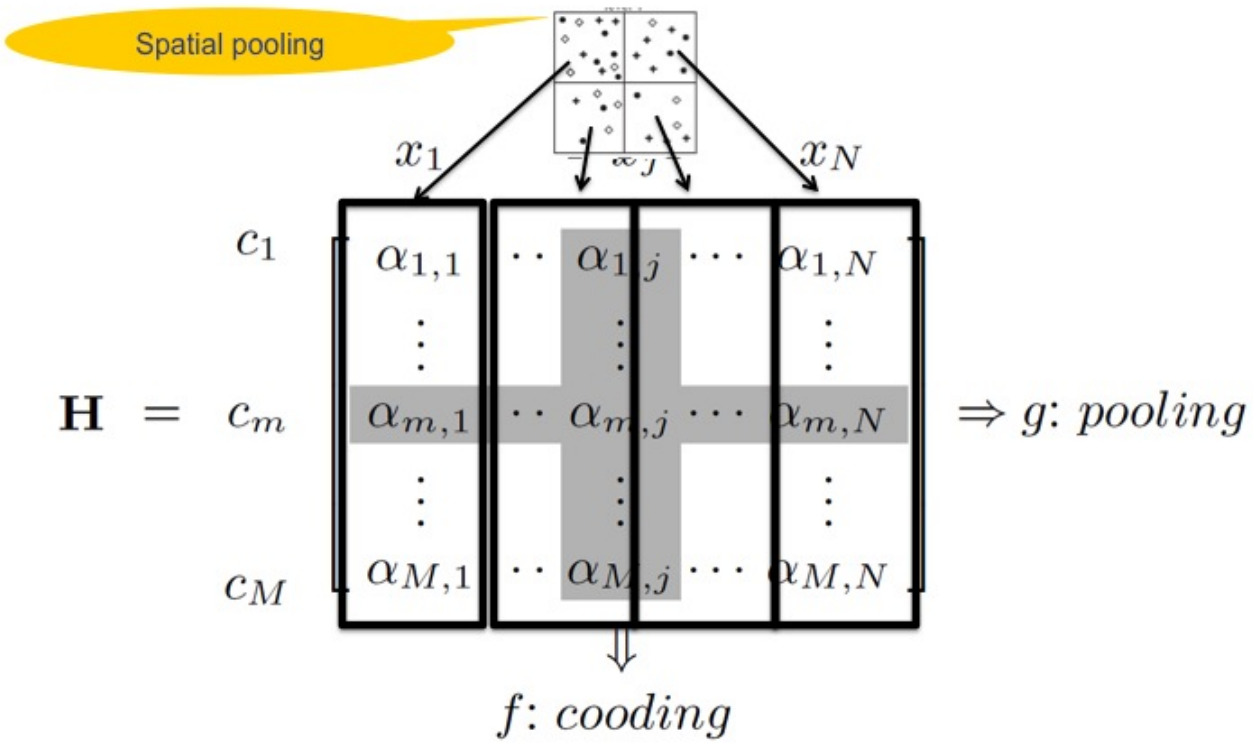
BoW Extensions/Improvements

- Spatial Pyramid Matching [Lazebnik et al., 2006]
- Max pooling [Yang et al., 2009]
- Soft assignment [Gemert et al., 2010]
- LLC [Wang et al., 2010]
- VLAD [Jégou et al., 2010]
- Super-Vector Coding [Zhou et al., 2010]
- **Fisher Vector [Perronnin et al., 2010]**
- Spatial Fisher Vector [Krapac et al., 2011]
- VLAT [Picard et al., 2011]
- Compact VLAT [Negrel et al., 2012]

Outline

1. BoW model
2. Pooling
   - Pooling extension
   - BossaNova
3. Coding
4. Deep Learning with RBM

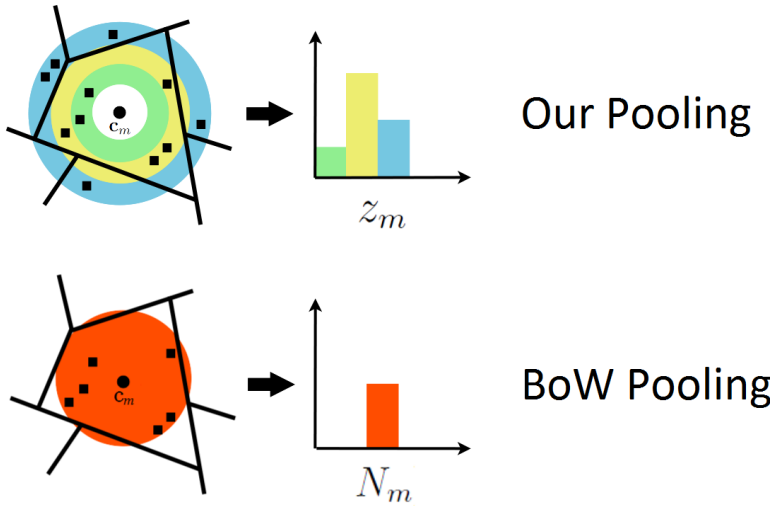
Pooling extension : Spatial Pyramid [Lazebnik et al., 2006]



Clusters for pooling in feature space [Boureau et al. ICCV 2011]

Pooling extension

- Pooling operator : averaging, max, Lp norm
- Learning in spatial pooling : spatial weight learned per visual word [Feng CVPR 2011] => supervised techniques (to learn classifiers and parts of the representation)
- A more information-preserving pooling operation : a distance-to-codeword distribution (BossaNova model)

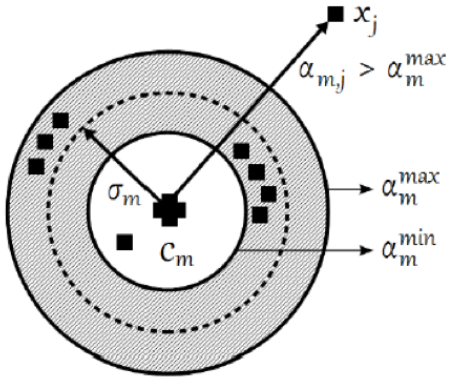


BossaNova Model (PhD's work of Sandra Avila)

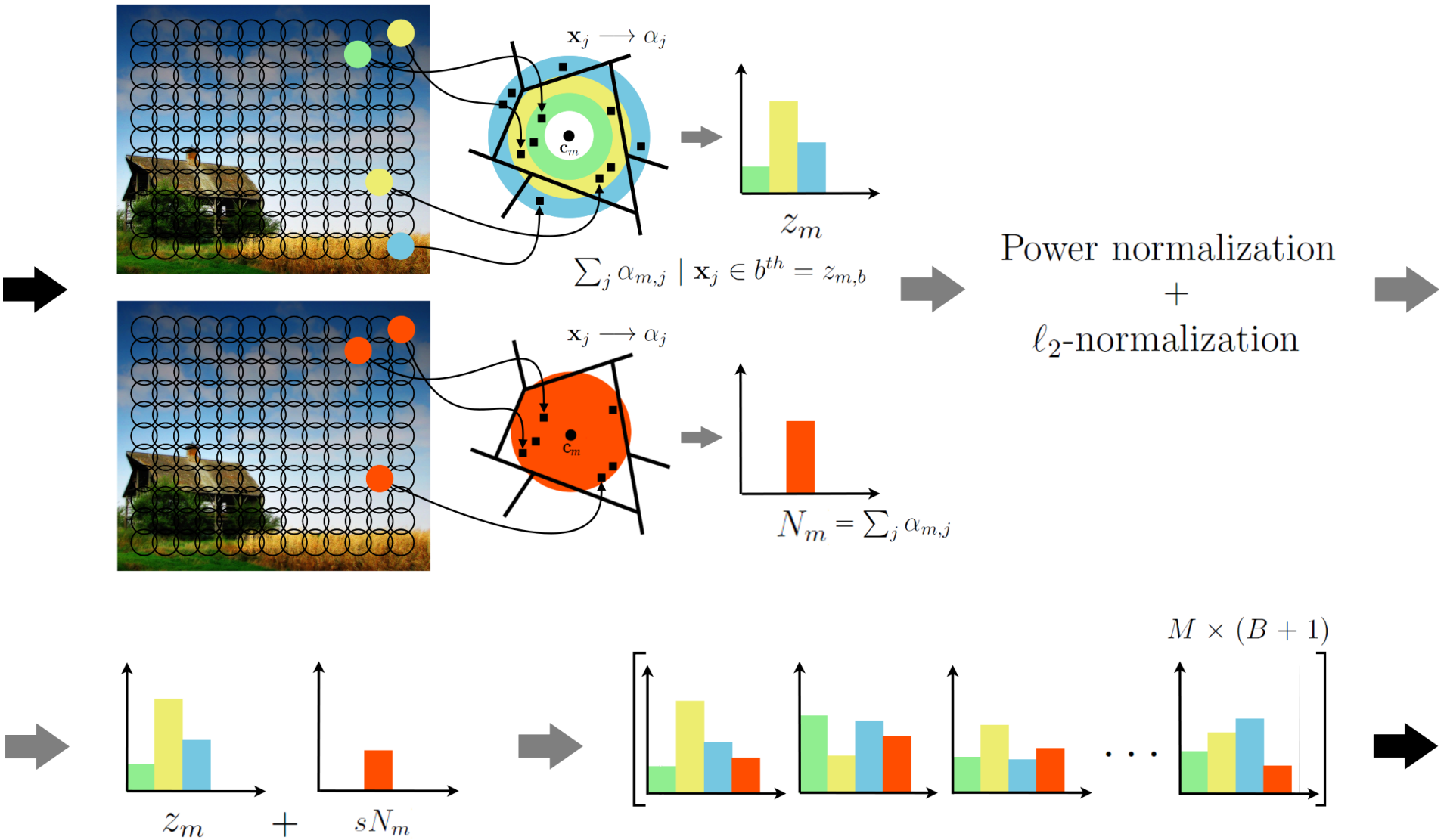
Pooling Formalism

$$
\begin{aligned}
g : \mathbb{R}^N &\longrightarrow \mathbb{R}^B \\
\alpha_{\mathbf{m}} &\longrightarrow g(\alpha_m) = z_m \\
z_{m,b} &= \text{card}\left(\mathbf{x}_j \mid \alpha_{m,j} \in \left[\frac{b}{B}; \frac{b+1}{B}\right]\right) \\
&\quad \frac{b}{B} \geq \alpha_m^{min} \quad \text{and} \quad \frac{b+1}{B} \leq \alpha_m^{max}
\end{aligned}
$$

B : number of bins of each histogram z_m, and $[\alpha_m^{min}; \alpha_m^{max}]$ distance range



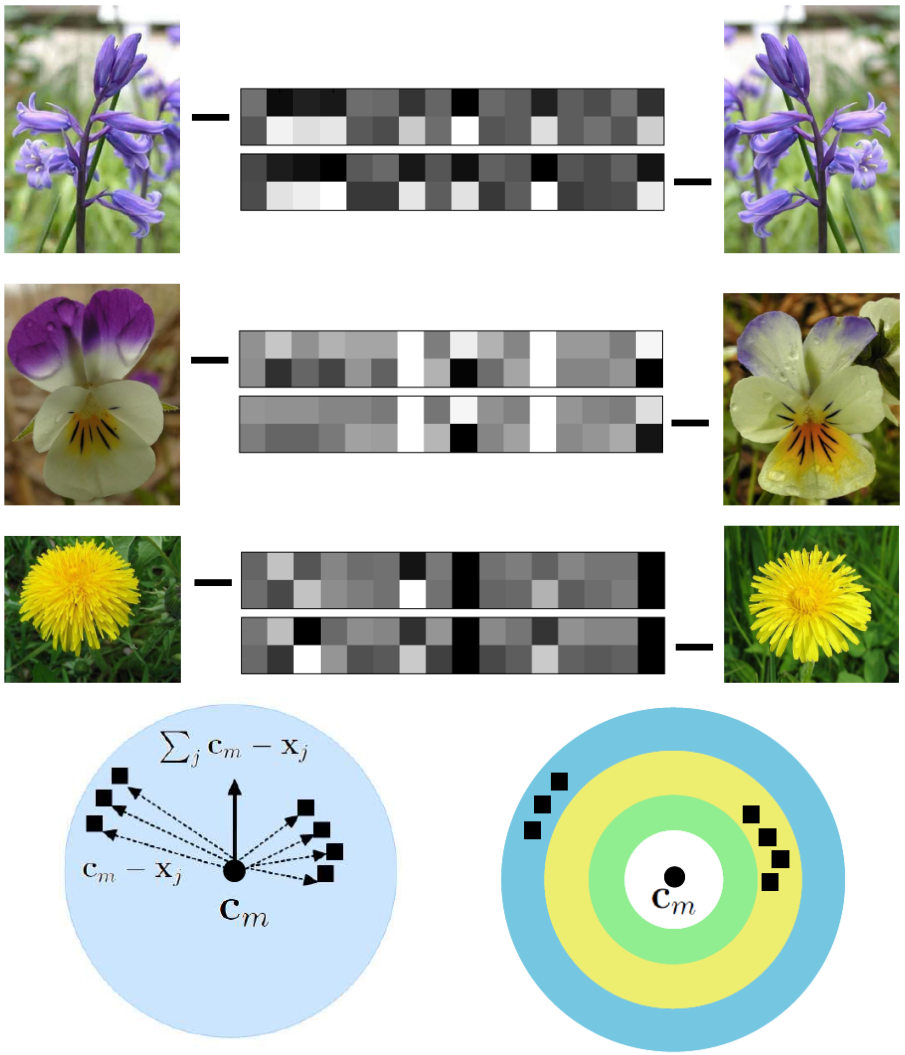
BossaNova Representation



BossaNova Representation

BossaNova (BN) Parameters

- **B** (number of bins) : {2, 4, 6, 8, 10}
- α_{min} : {0, 0.6}
- α_{max} : {1.5, 2.0}
- **s** (cross weight) : $\{10^{-4}; 1\}$
- **M** (codebook) : {128 ; 8192}



Fisher Vector / BossaNova

Experimental Results

- Implemented methods : Bag-of-Words (BoW), Fisher Vector (FV), BOSSA, BossaNova (BN), BN + FV
- Datasets : PASCAL VOC 2007, 15-Scenes, MIRFLICKR, ImageCLEF 2011
- MIRFLICKR : 25000 images, manually annotated for 38 concepts.
- ImageCLEF 2011 Photo Annotation : 18000 images, 99 concepts

Experimental Results – MIRFLICKR

	MAP (%)
Our methods	
BossaNova [Avila et al., 2012]	54.4
BossaNova + FV [Avila et al., 2012]	**56.0**
Implemented methods	
BoW [Sivic and Zisserman, 2003]	51.5
FV [Perronnin et al., 2010]	54.3
Published results	
[Huiskes et al., 2010]	37.5
[Guillaumin et al., 2010]	53.0

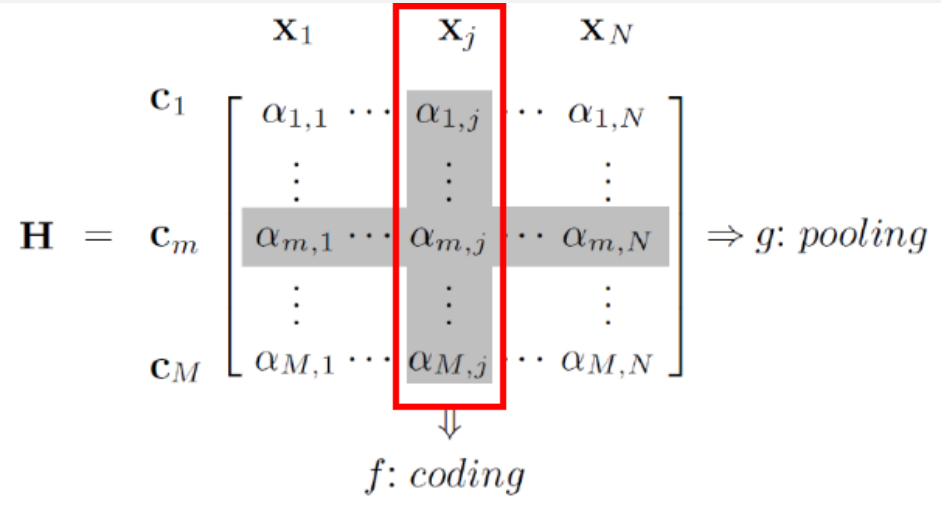
- Project Web page with codes available
  `https://sites.google.com/site/bossanovasite/`
- Publication : CVIU'12 *Pooling in Image Representation : the Visual Codeword Point of View*, S. Avila, N. Thome, M. Cord, E. Valle, A. araujo

Outline

1. BoW model
2. Pooling
3. Coding
   - Dictionary learning and sparse coding
4. Deep Learning with RBM

Advanced coding : Localized Soft Coding [Liu ICCV 2011]

LSC principle



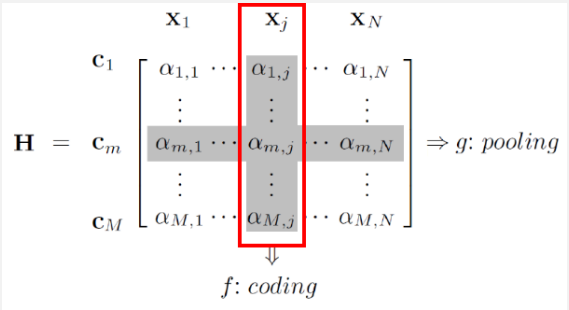
$$\alpha_{m,j} = \frac{e^{-\beta \hat{d}(x_j, c_m)}}{\sum_{l=1}^{M} e^{-\beta \hat{d}(x_j, c_l)}} \qquad \hat{d}(x_j, c_m) = \begin{cases} d(x_j, c_m) & \text{if } c_m \in \mathcal{N}_k(x_j)\ ^a \\ \infty & \text{otherwise} \end{cases}$$

followed by max pooling, no normalization of the BoW, and Linear SVM

a. $\mathcal{N}_k(x_i)$ the k-nearest neighbors

Dictionary learning and sparse coding

Principle



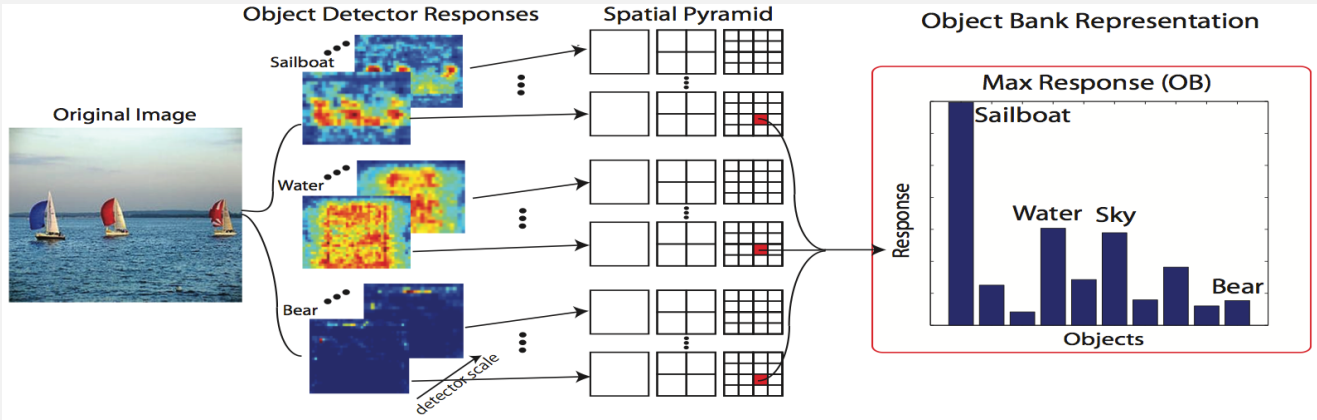
- Sparse coding (with matrix C of codewords) for local feature x_j :

$$\alpha_j = Argmin_\alpha L(\alpha, C) = \|x_j - C\alpha\|_2^2 + \lambda\|\alpha\|_1$$

- Dictionary learning : alternate optimization over code α and matrix C over a set of local features (with constraints on vector norms)
- Discussion : one scheme for optimizing C, another for coding (most important) [Coates Ng ICML 2011]
- Sparse Auto encoders [Ranzato 2006, Bengio 2006] and RBM for dictionary learning [Hinton 2006]

Dictionary learning : unsupervised/supervised/tranfered

Object Bank [Fei-Fei NIPS 2010] ... As a BoW strategy



- Similar to BoW where visual words are object detectors
- Dictionary learned with supervised schemes (=> transfer of knowledge)
- Very good perf when combined with BossaNova = 69% on Pascal VOC 2007 [ICIP 2013]

$$\mathbf{H} = \begin{array}{c} \\ \mathbf{c}_1 \\ \\ \mathbf{c}_m \\ \\ \mathbf{c}_M \end{array} \begin{array}{c} \begin{array}{ccccc} \mathbf{x}_1 & & \mathbf{x}_j & & \mathbf{x}_N \end{array} \\ \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{bmatrix} \end{array} \Rightarrow g\text{: } pooling$$

$$\Downarrow$$

$$f\text{: } coding$$

Outline

1. BoW model
2. Pooling
3. Coding
4. Deep Learning with RBM
   - Deep Learning & Visual Representations
   - Regularizing Latent Representations
   - Deep Supervised Optimization
   - Learning Hierarchical Visual Codes

4 Deep Learning with RBM

1. Deep Learning & Visual Representations

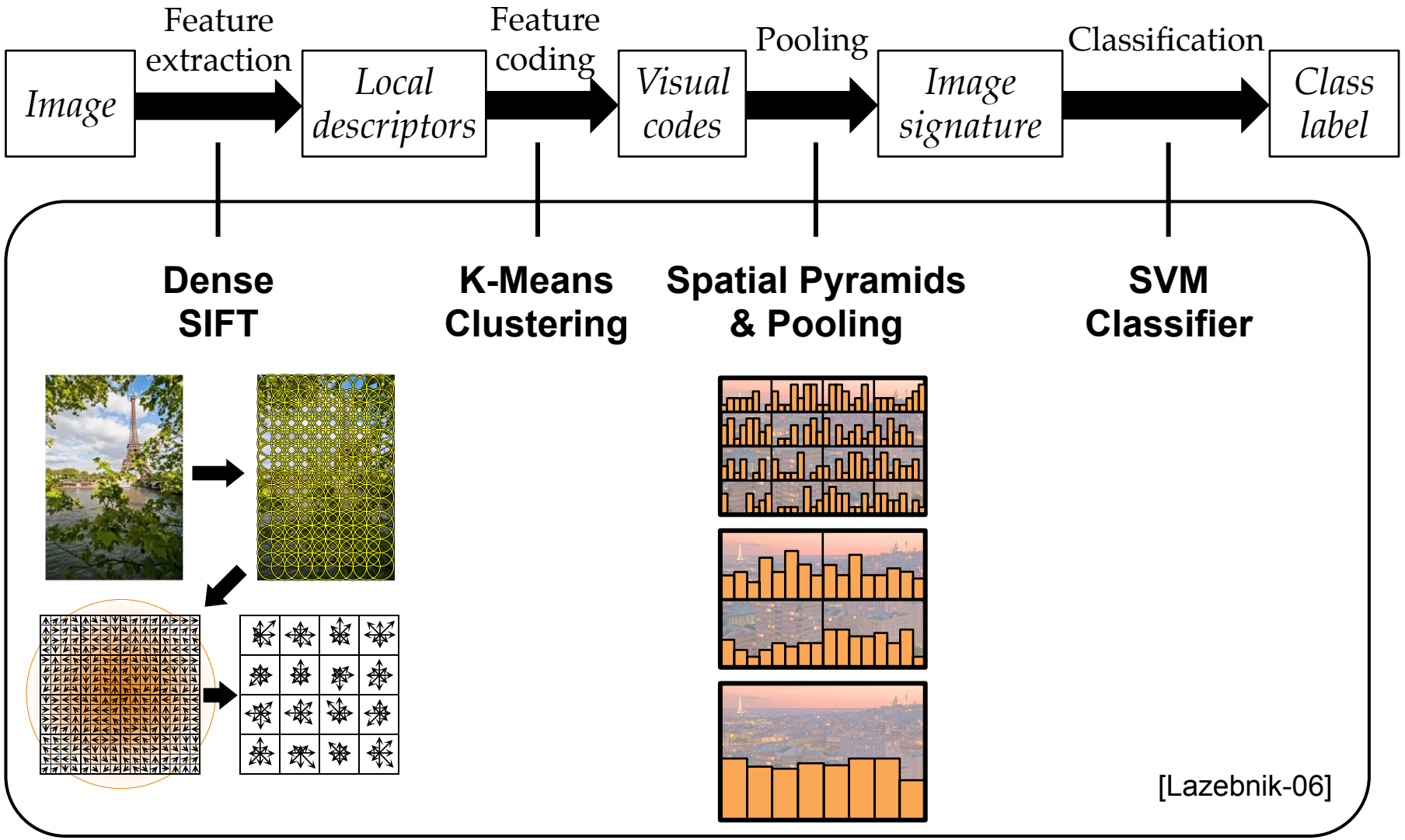
Layer-Wise Learning Scheme

PhD thesis of Hanlin Goh (July 2013)

Unsupervised and supervised visual codes with restricted Boltzmann machines
H. Goh, N. Thome, M. Cord & J.-H. Lim, *European Conference on Computer Vision (ECCV)*, 2012.

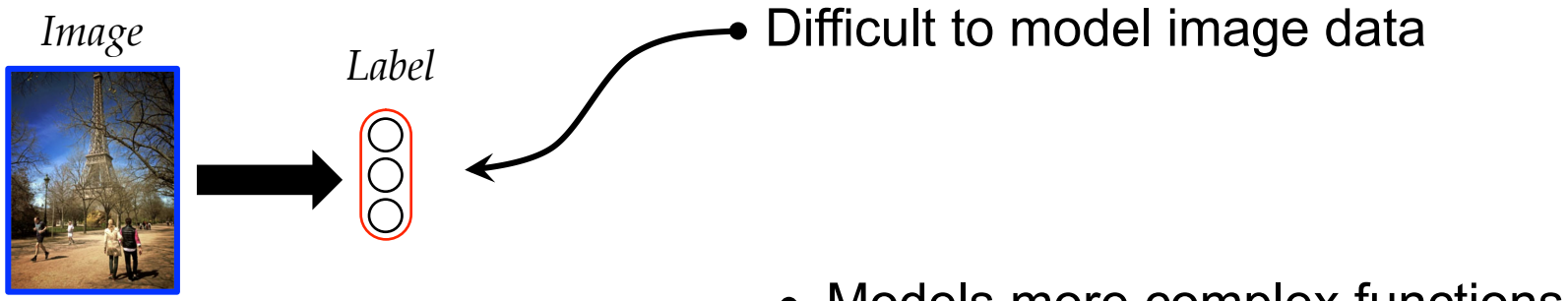
Top-down regularization of deep belief networks
H. Goh, N. Thome, M. Cord & J.-H. Lim, *Neural Information Processing Systems (NIPS)*, 2013.

Image Classification Model



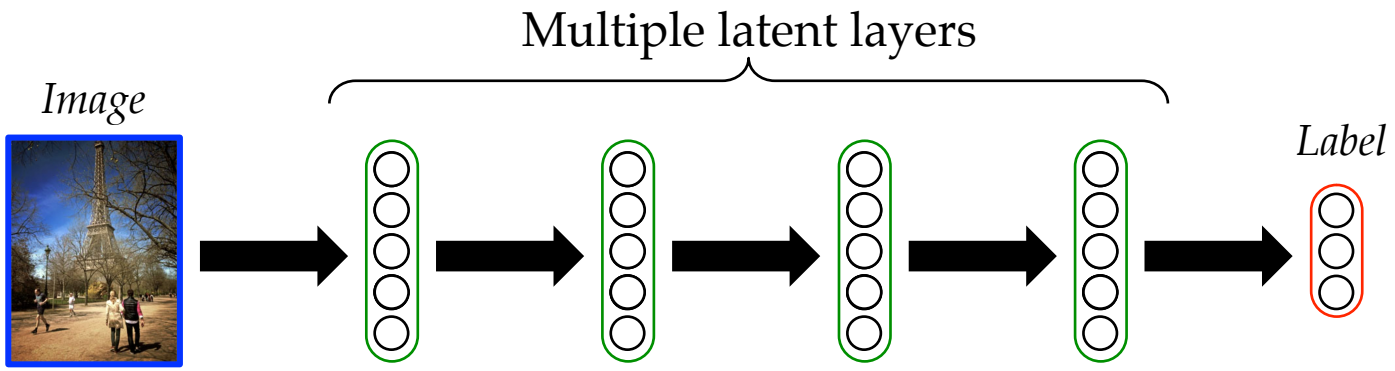
Deep Visual Representations

Shallow Architecture



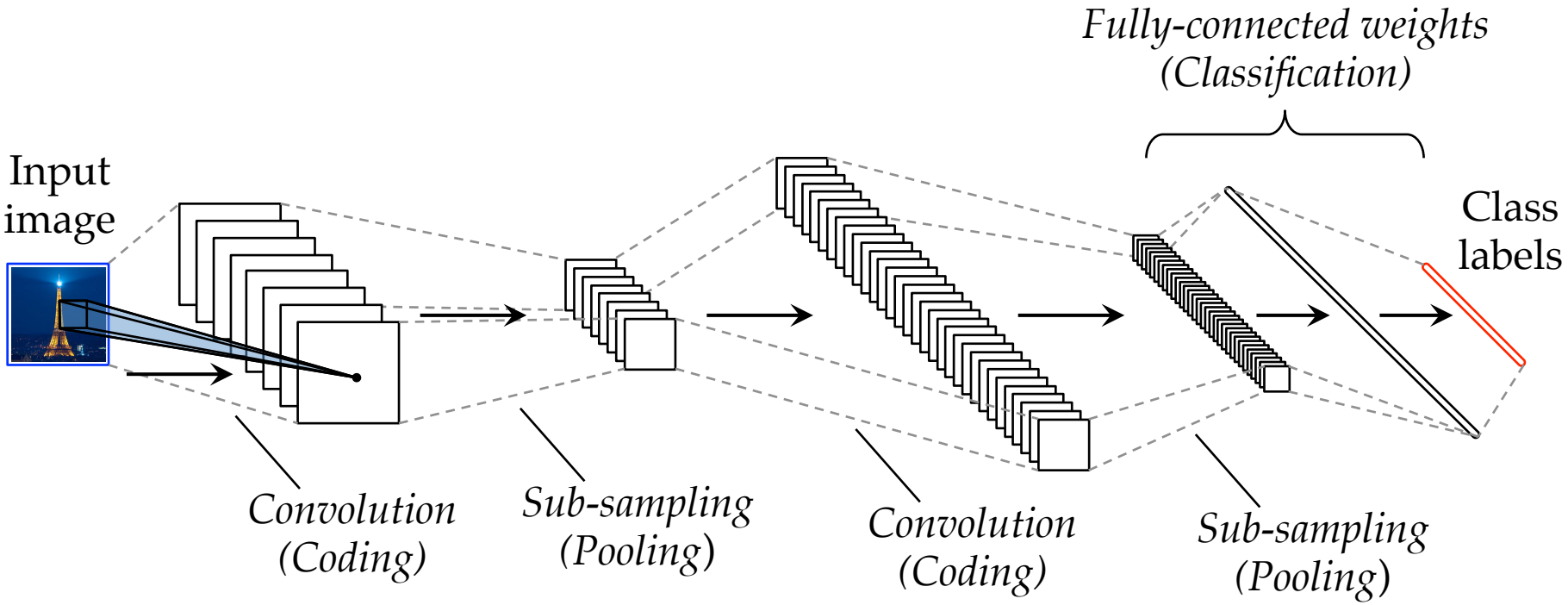
Models more complex functions

Deep Architecture



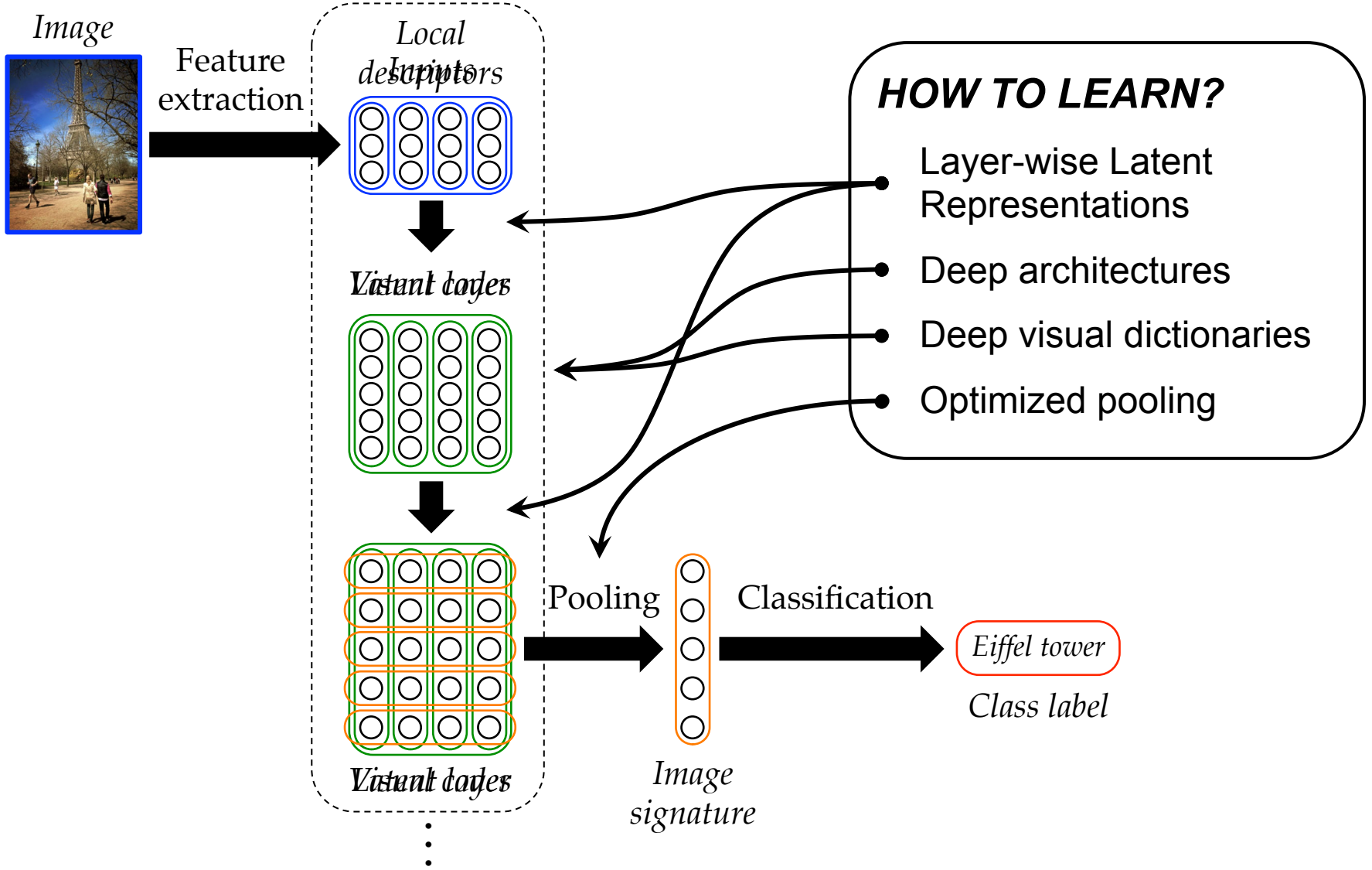
Deep Convolutional Networks

[LeCun-89]



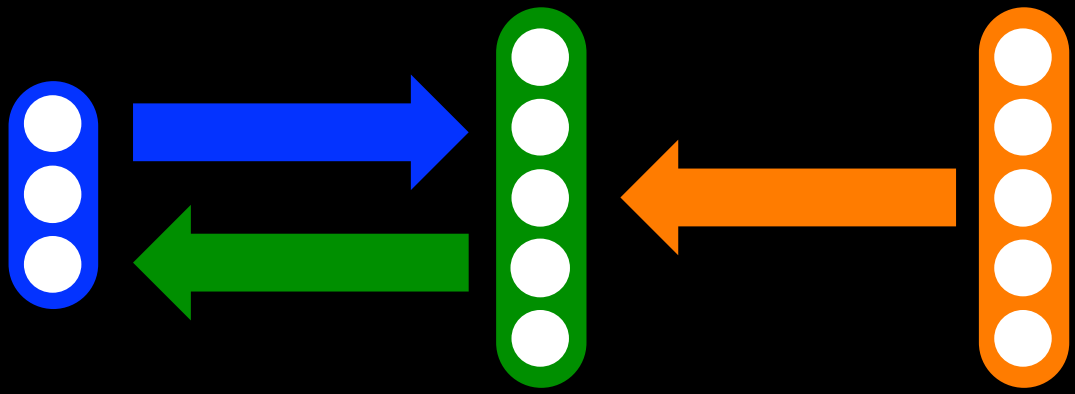
- **Convolution** uses local weights shared across the whole image.
- **Sub-sampling** shrinks the spatial dimensions.

Learning architecture



2. Regularizing Latent Representations

A New Layer-Wise Learning Scheme



Greedy Layer-Wise Learning

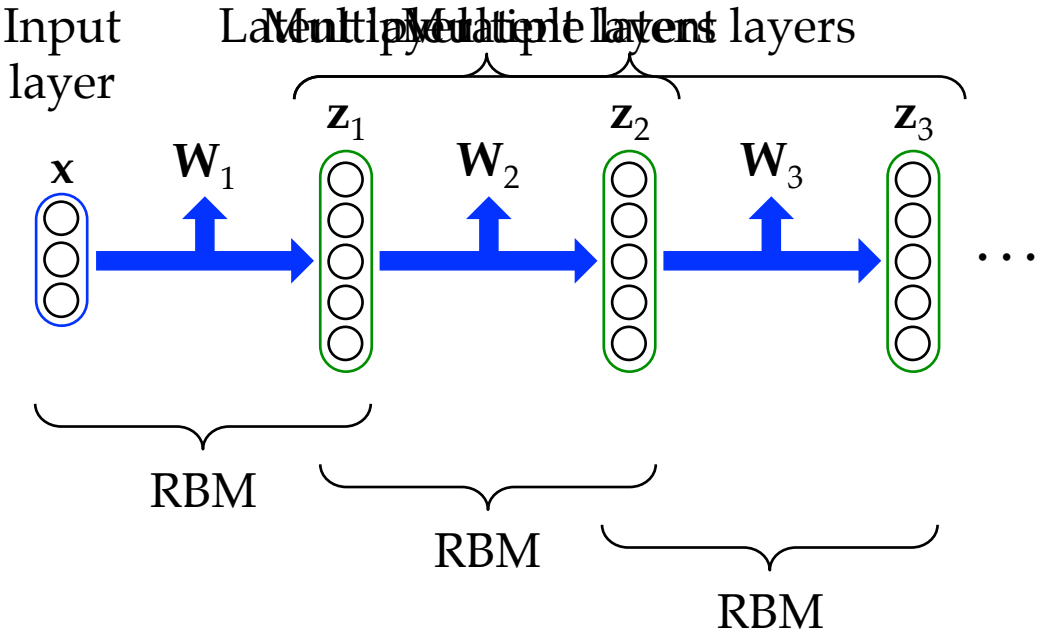
[Hinton-06, Bengio-06, Ranzato & LeCun-06]

- Representations are learned one layer at a time from the bottom-up
- Each new layer models the distribution of the previous layer
- Learning is performed using unsupervised building blocks

Building blocks

- Restricted Boltzmann machines
- Decoder networks
- Auto-encoder networks

Deep Belief Network

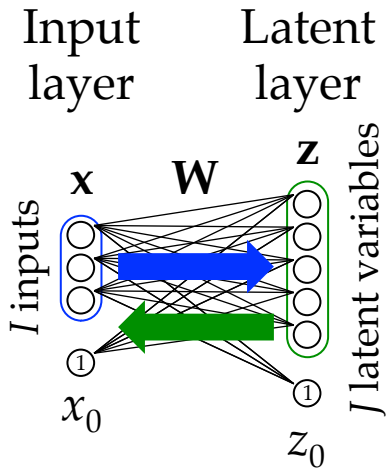


Restricted Boltzmann Machines (RBM)

[Smolensky-86]

Objective

- Learn a projection to a good feature space using unsupervised learning



Energy function: $E(\mathbf{x}, \mathbf{z}) = -\sum_{i=0}^{I}\sum_{j=0}^{J} x_i w_{ij} z_j$

Sampling functions

$P(z_j|\mathbf{x}) = sigm(\mathbf{W}^T\mathbf{x})$

$P(x_i|\mathbf{z}) = sigm(\mathbf{W}\mathbf{z})$

Optimization

- Maximum likelihood approximation
- Contrastive divergence learning algorithm

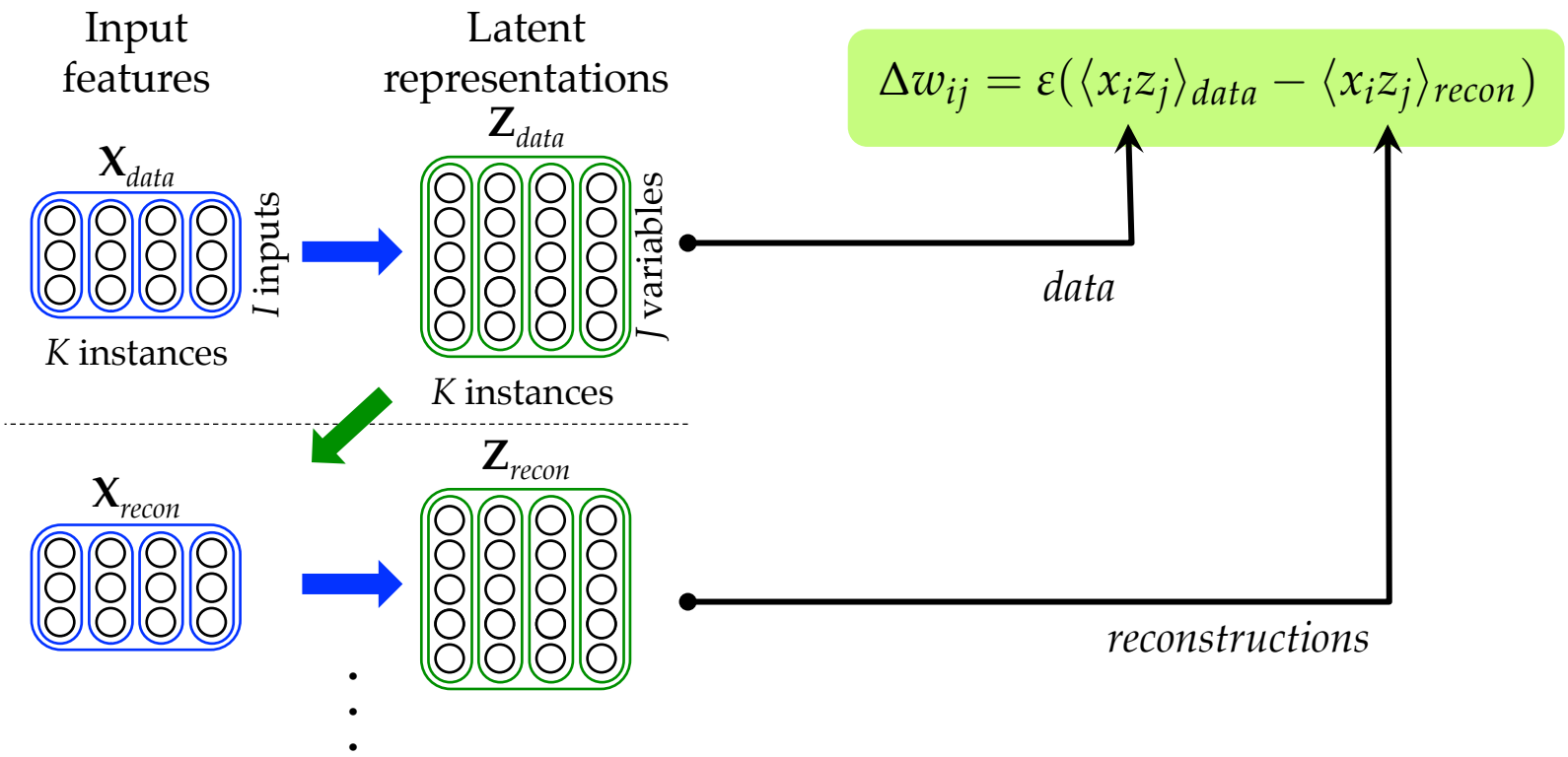
Contrastive Divergence Learning

[Hinton-02]

Maximum likelihood approximation: $\mathcal{L}_{RBM} = -\sum_{k=1}^{|\mathcal{D}_{train}|} \log P(\mathbf{x}_k)$

Step 1: Alternating Gibbs Sampling

Step 2: Update Parameters



Existing "Sparse" Regularization

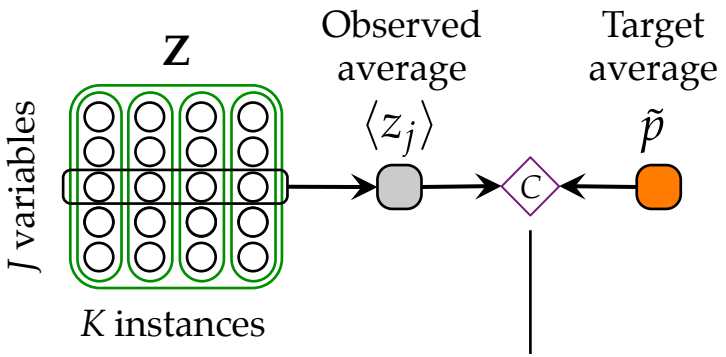
- **Solely using the maximum likelihood criteria may not be the best way to learn latent representations… REGULARIZE!**

$$\mathcal{L}_{RBM+reg} = \underbrace{-\sum_{k=1}^{|\mathcal{D}_{train}|} \log P(\mathbf{x}_k)}_{\textit{Maximum Likelihood}} + \underbrace{\lambda h(\mathbf{z})}_{\textit{Regularization Term}}$$

- **"Sparsity": Low average activation for each latent variable**

[Lee & Ng-08]

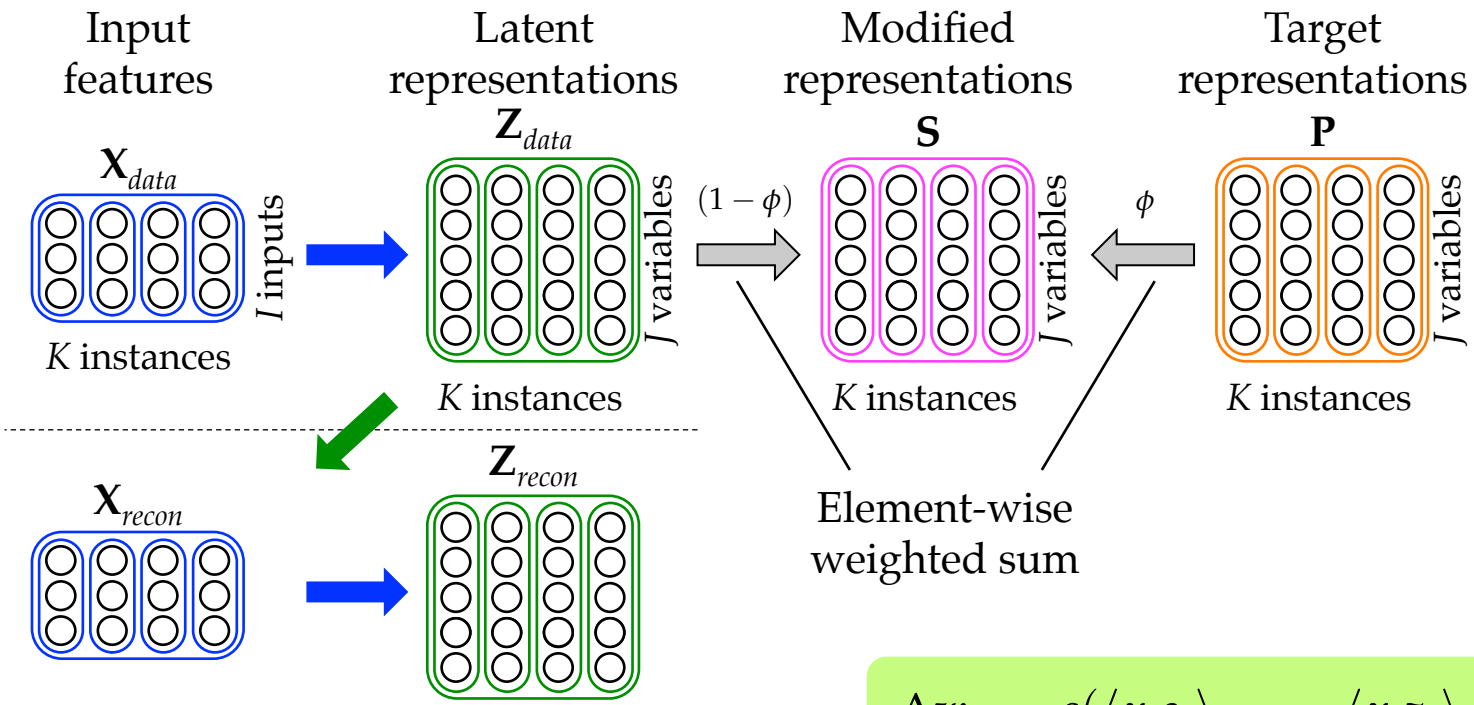
$$h(\mathbf{z}) = \sum_{j=1}^{J} \|\tilde{p} - \langle z_j \rangle\|_2^2$$



- Low average activation $\neq$ "sparsity"

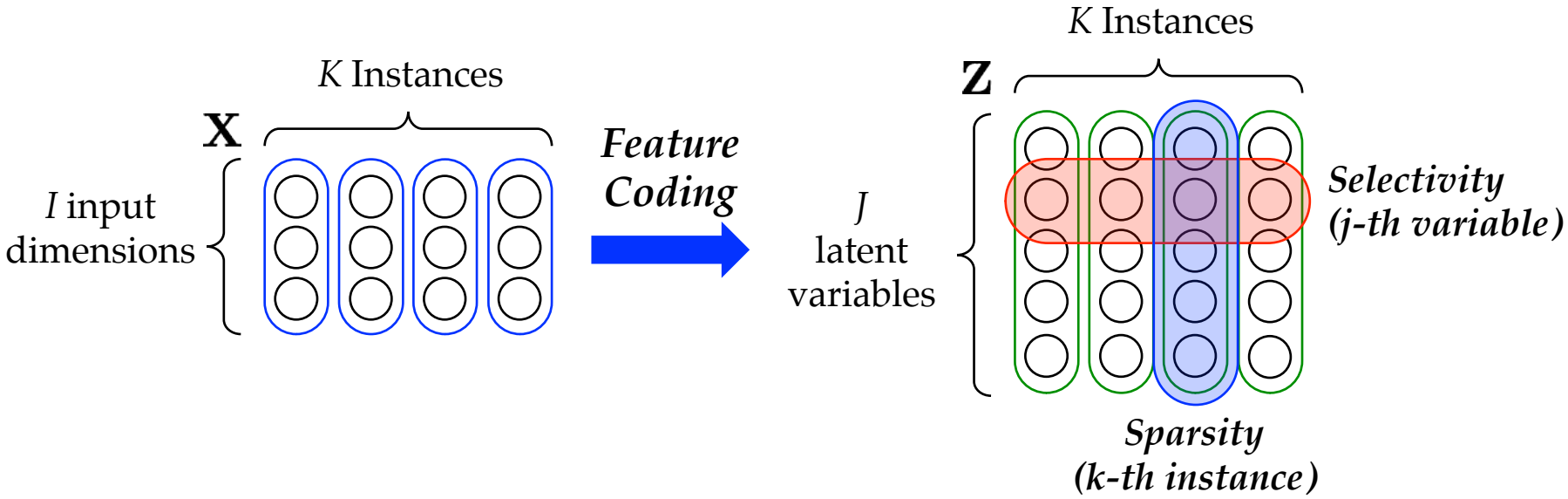
Proposed Generic RBM Regularization

$$\mathcal{L}_{RBM+reg} = \underbrace{-\sum_{k=1}^{|\mathcal{D}_{train}|} \log P(\mathbf{x}_k)}_{\text{Maximum Likelihood}} \underbrace{- \lambda \sum_{k=1}^{|\mathcal{D}_{train}|} \sum_{j=1}^{J} \log P(p_{jk}|z_{jk})}_{\text{Cross-Entropy Penalty}}$$



$$\Delta w_{ij} = \varepsilon(\langle x_i s_j \rangle_{data} - \langle x_i z_j \rangle_{recon})$$

Feature Coding: Sparsity & Selectivity



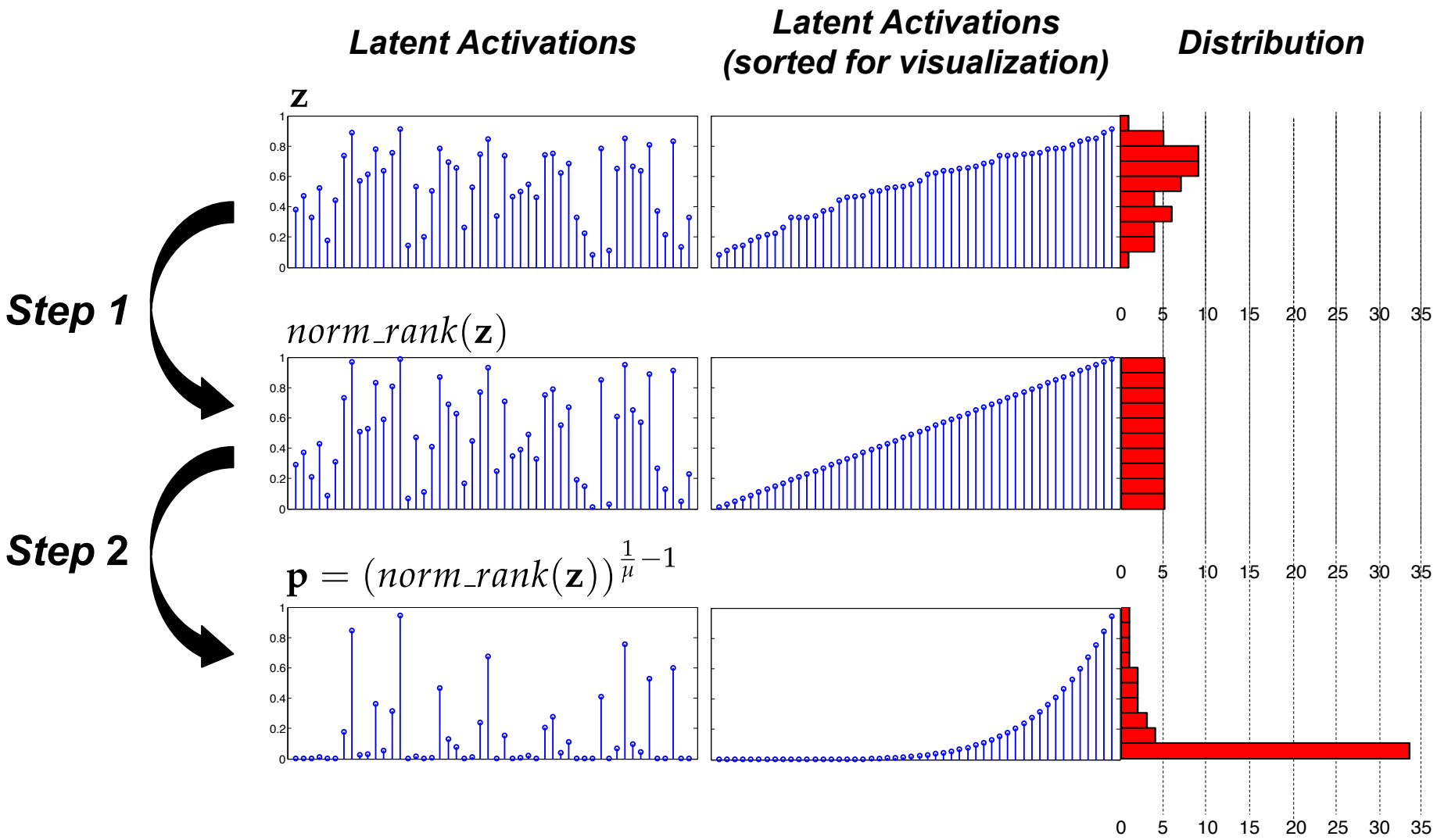
High Selectivity

- Each latent variable responds to a small subset of instances

High Sparsity

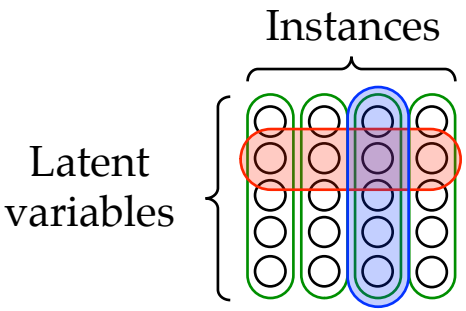
- Each instance invokes response from small subset of latent variables

Sparse & Selective Data Transform

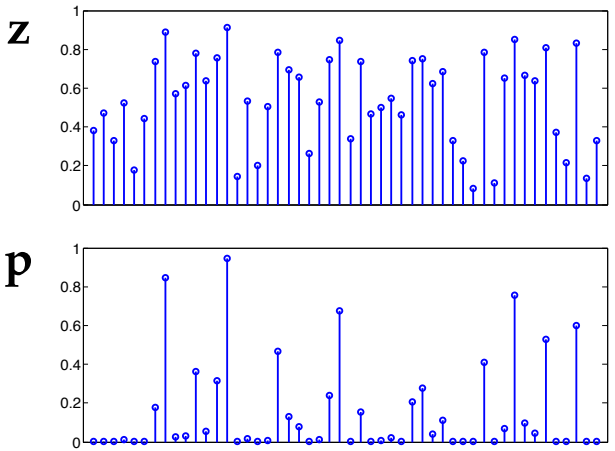


Sparse & Selective RBM Regularization

Step 1 – Compute Target Representations



Data transform

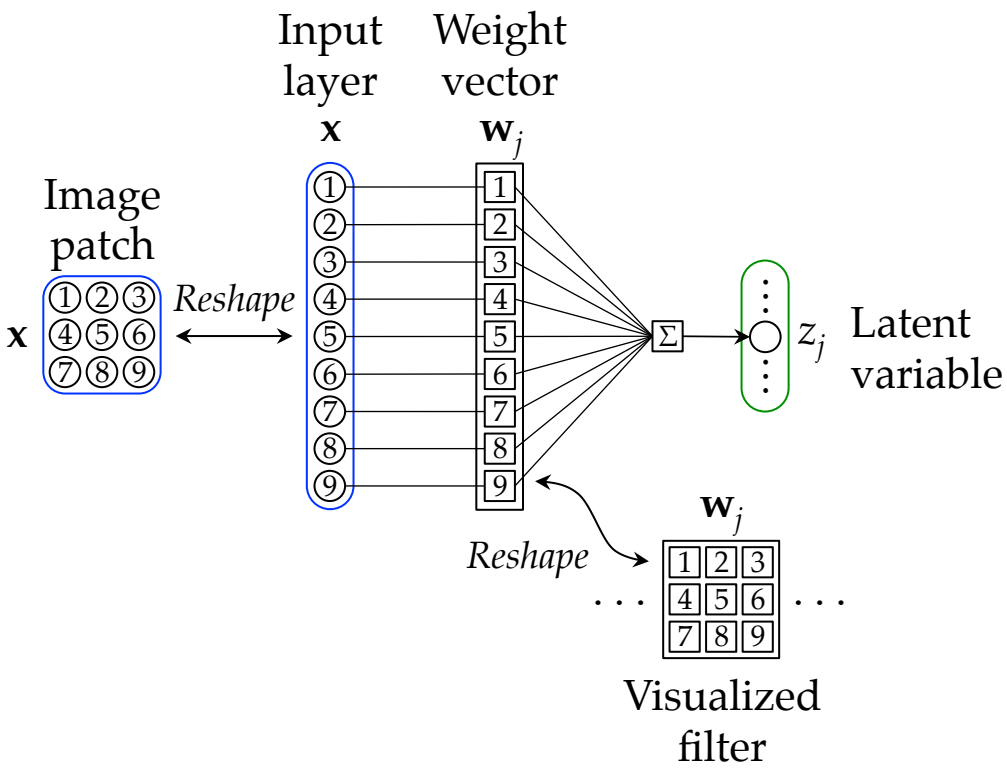


Step 1a – Transform every column
Step 1b – Retransform every row

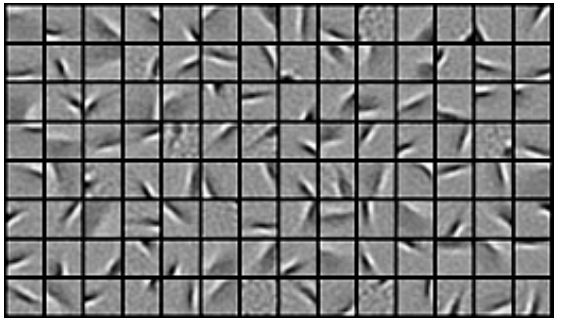
Step 2 – Regularize RBM Learning

$$\mathcal{L}_{RBM+reg} = -\sum_{k=1}^{|\mathcal{D}_{train}|} \log P(\mathbf{x}_k) - \lambda \sum_{k=1}^{|\mathcal{D}_{train}|} \sum_{j=1}^{J} \log P(p_{jk}|z_{jk})$$

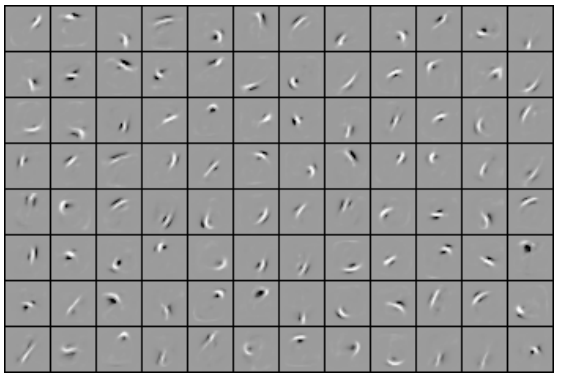
Visualization of Learned Weights



From Natural Images

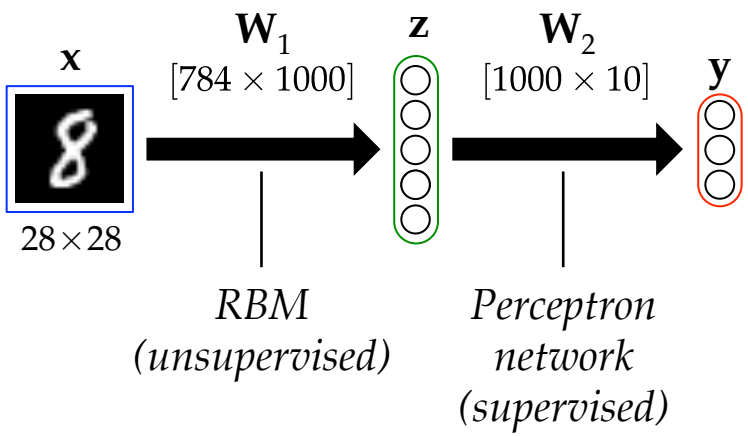


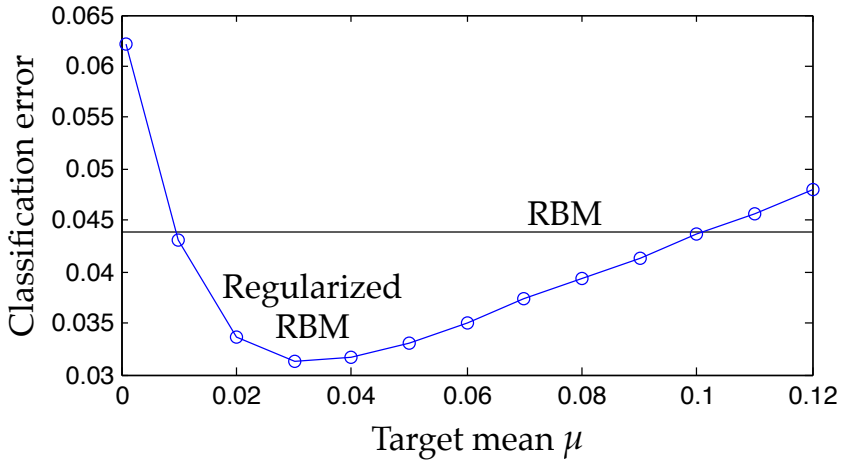
From Handwritten Digits



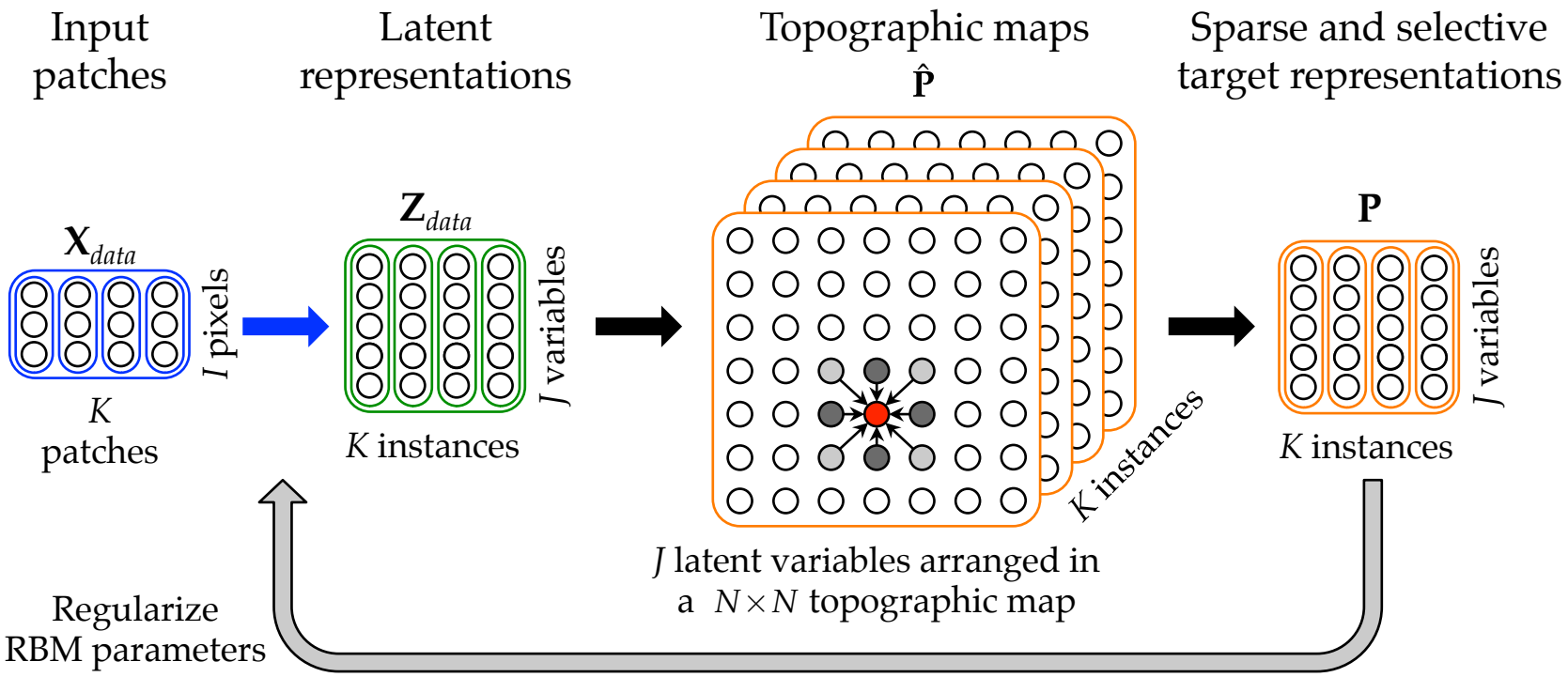
Initial Experiments – Single RBM

- Sparsity and selectivity are successfully transferred to the latent representation.
- Classification error is minimum when the activation level is low, but not at the lowest.

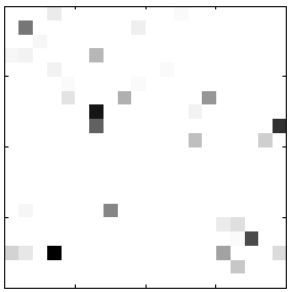




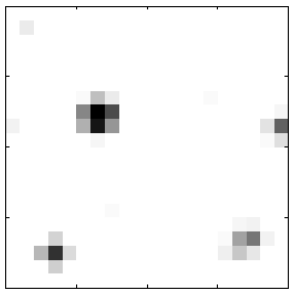
Topographic Organization



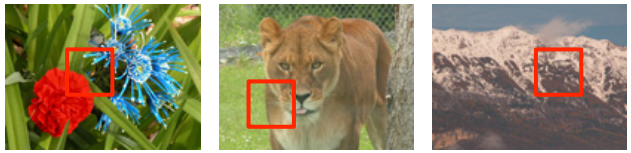
Before:



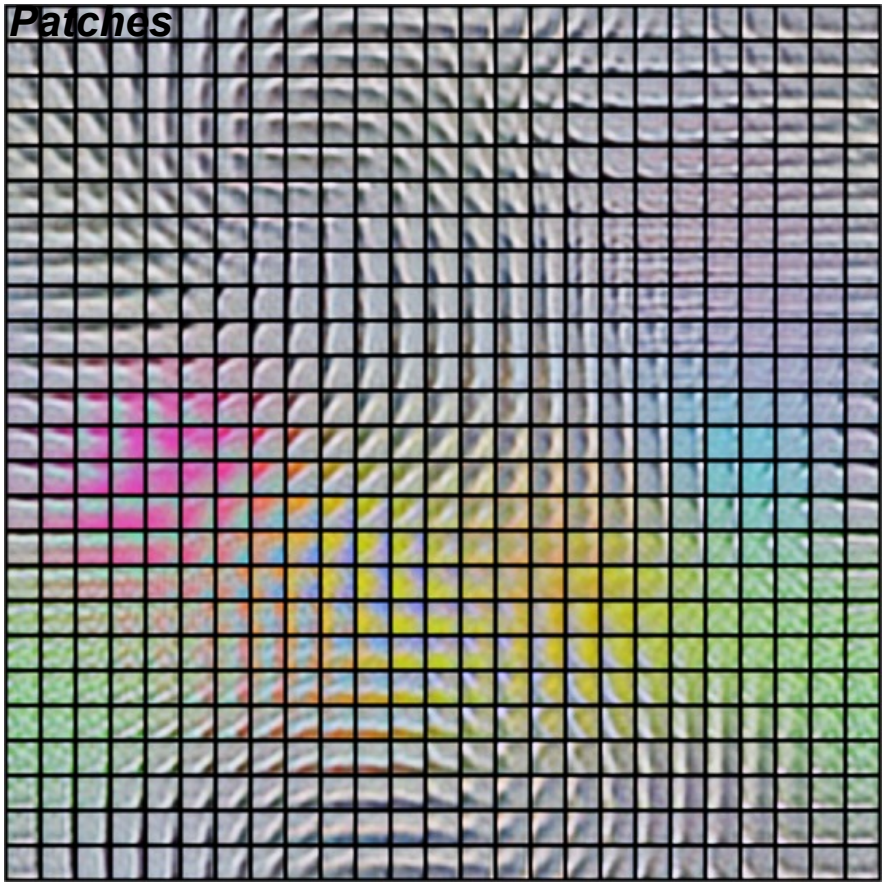
After:



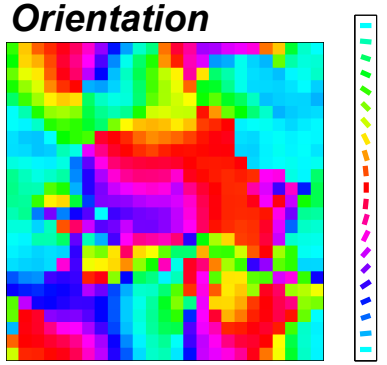
Smooth Topographic Map



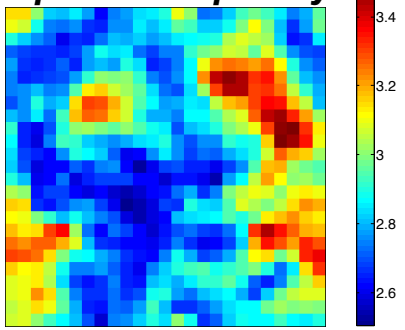
From Color Natural Image Patches



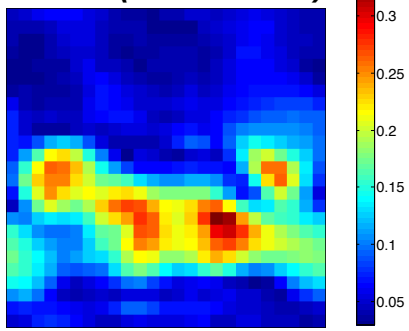
Orientation



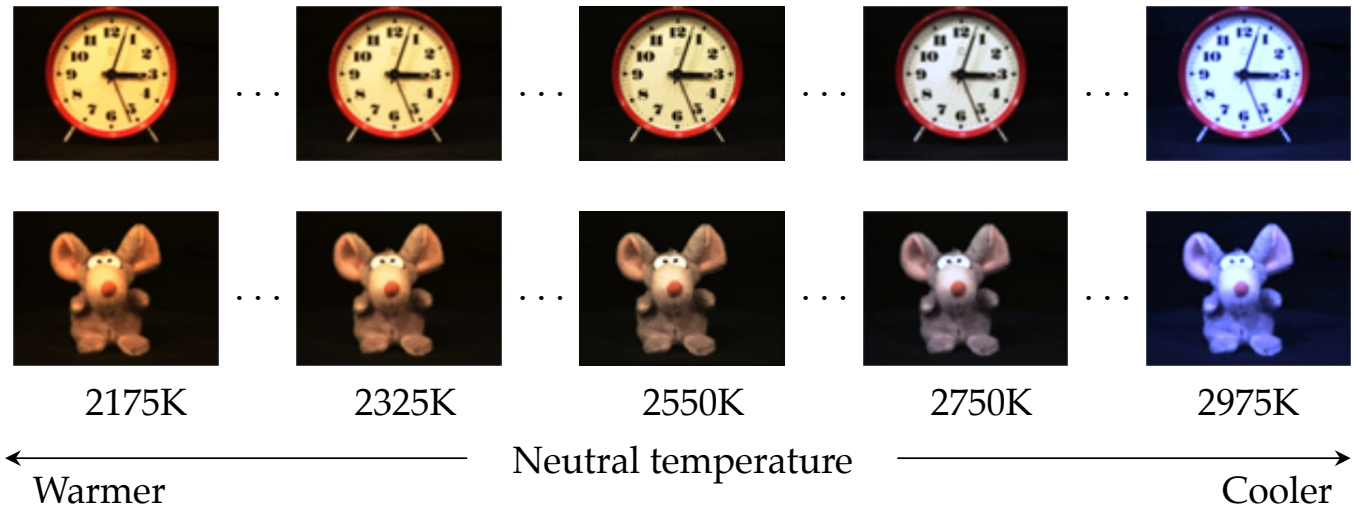
Spatial Frequency



Color (Saturation)



Latent Representation Invariance



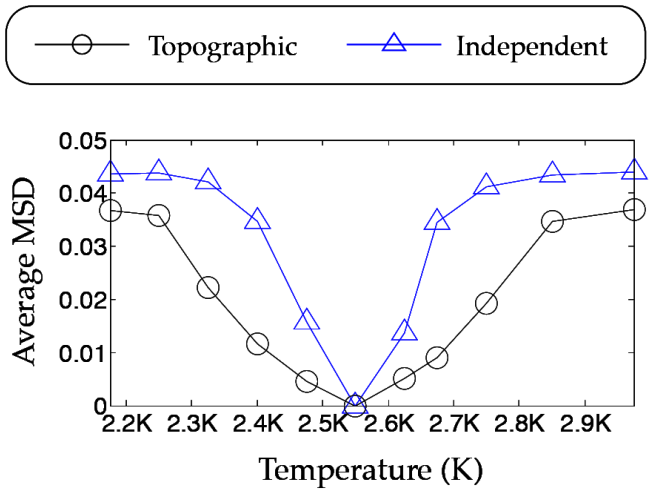


Image transformations

- Color temperature
- Rotation
- Scale
- Translation

Summary – Latent Representations

Regularizing Latent Representations

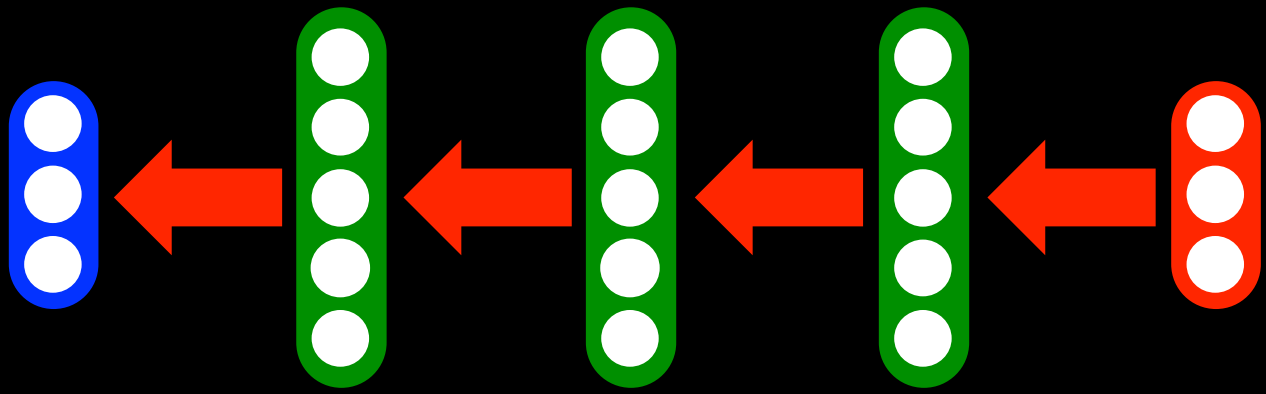
- Regularizing restricted Boltzmann machines
- Generic and able to take in any structured representation as “priors”

Designing Interesting Representations

- Inducing code sparsity and selectivity
- Inducing topographic organization

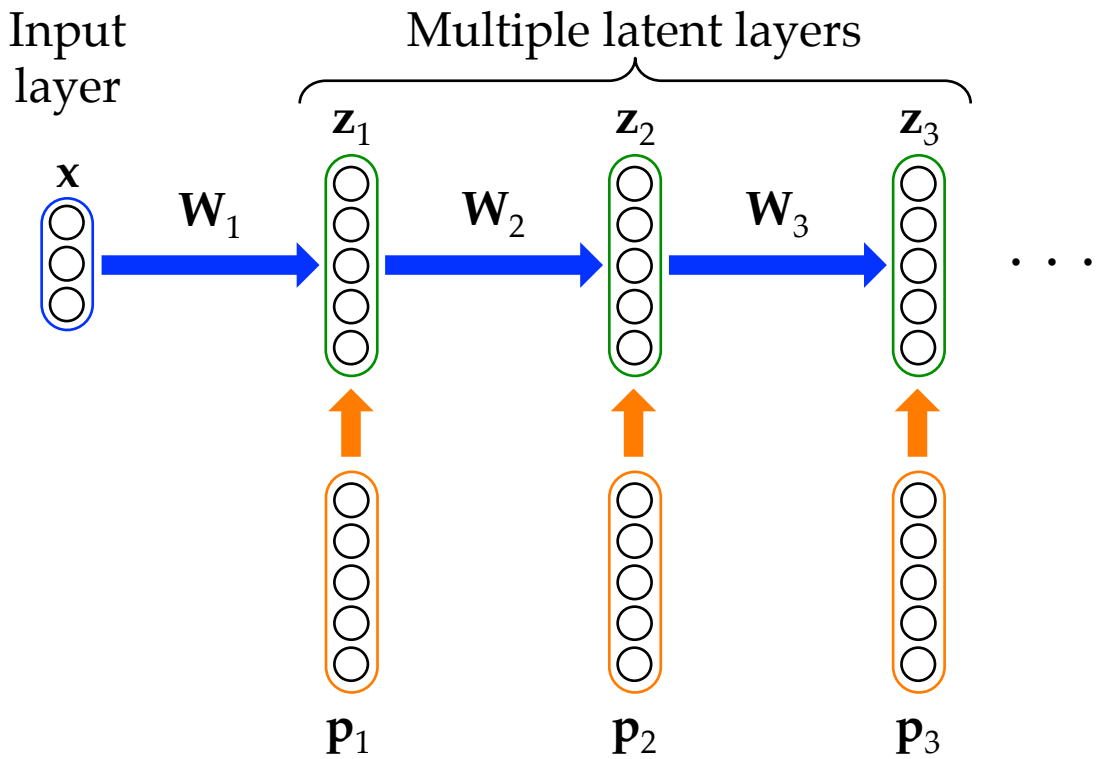
3. Deep Supervised Optimization

Combining Bottom-Up & Top-Down Signals



Layer-Wise Learning & Regularization

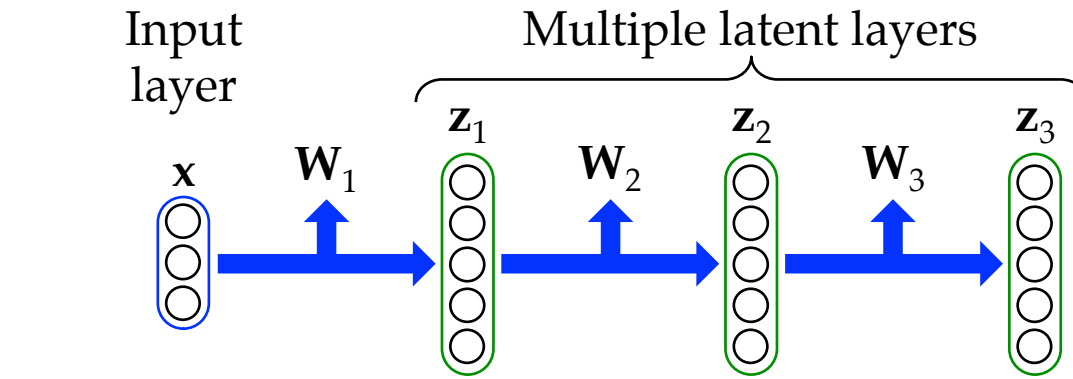
- RBMs are stacked from the bottom-up
- Each RBM models the distribution of the previous layer
- RBMs are regularized to assume some representational property



Proposed Deep Learning Strategy

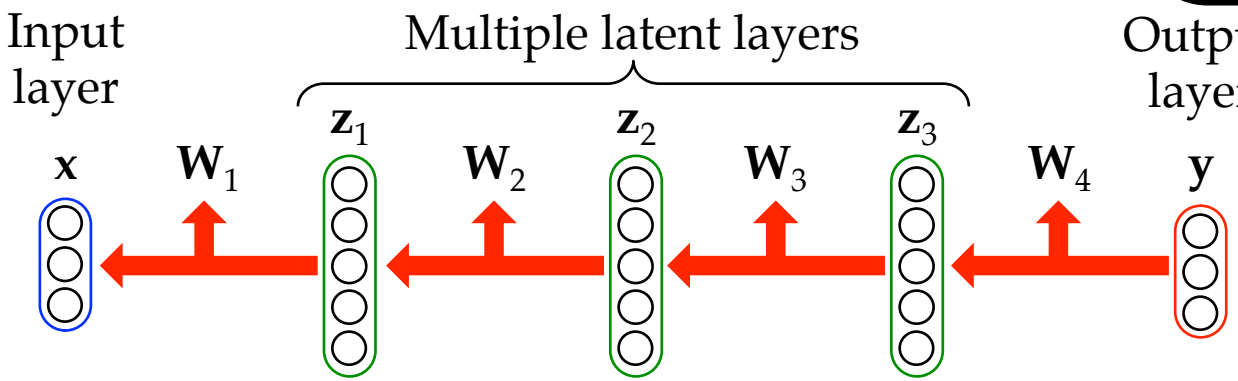
[Hinton-06, Benjio-06, Ranzato & LeCun-06]

Phase 1: Greedy Unsupervised Pre-Training



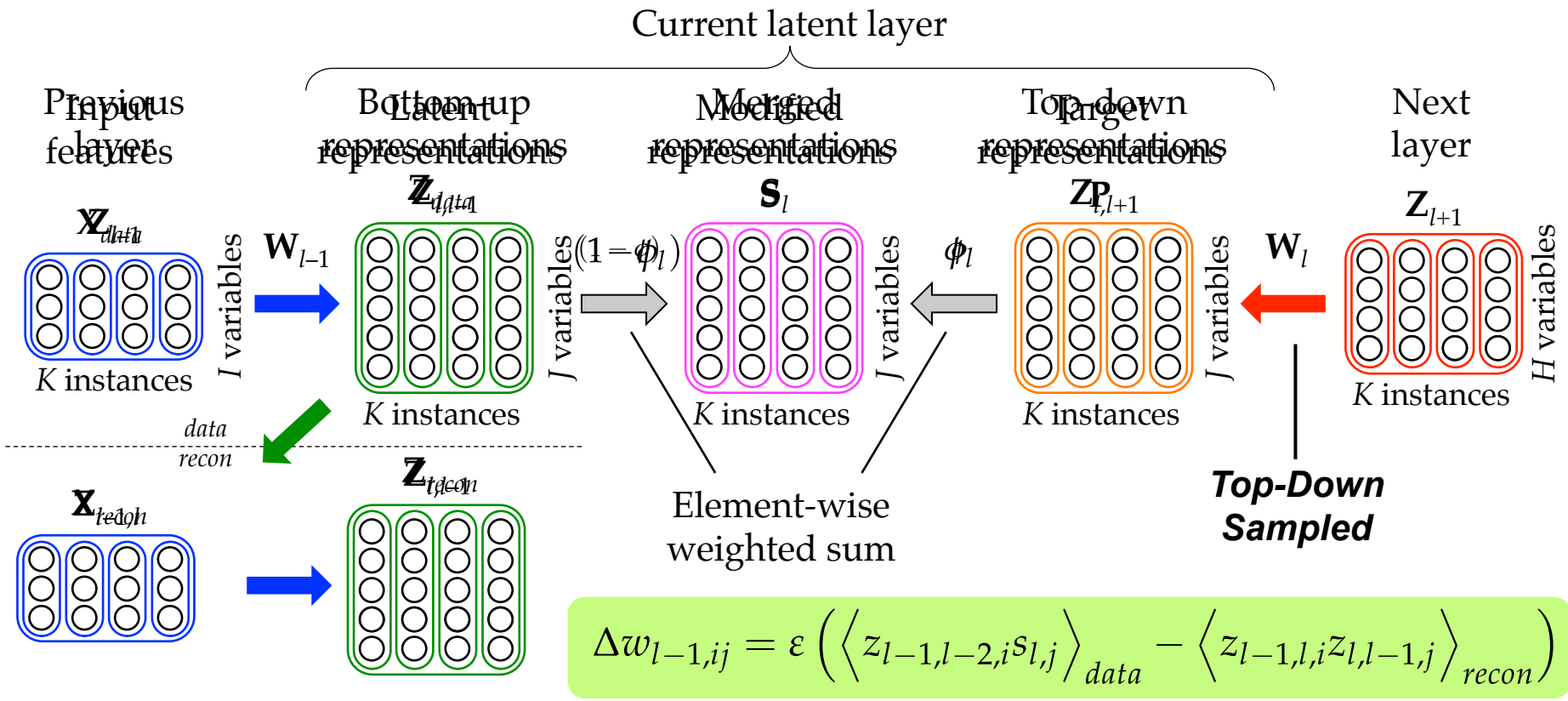
Bottom-Up + Top-Down Deep Learning

Phase 2: Discriminative Fine-Tuning



Top-Down Regularized Building Block

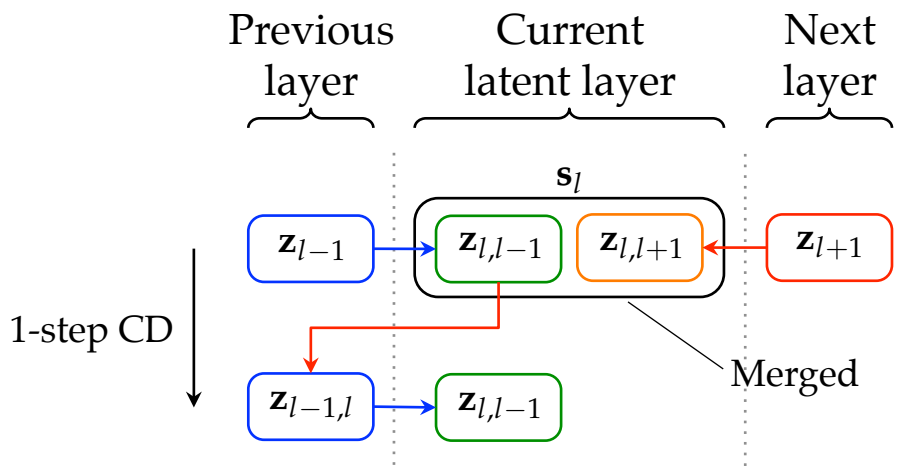
$$\mathcal{L}_{l,RBM+top-down} = \underbrace{-\sum_{k=1}^{|\mathcal{D}_{train}|} \log P(\mathbf{z}_{l-1,k})}_{\textbf{Maximum Likelihood}} \underbrace{-\lambda \sum_{k=1}^{|\mathcal{D}_{train}|} \sum_{j=1}^{J} \log P(z_{l,l+1,jk} | z_{l,l-1,jk})}_{\textbf{Cross-Entropy Penalty}}$$



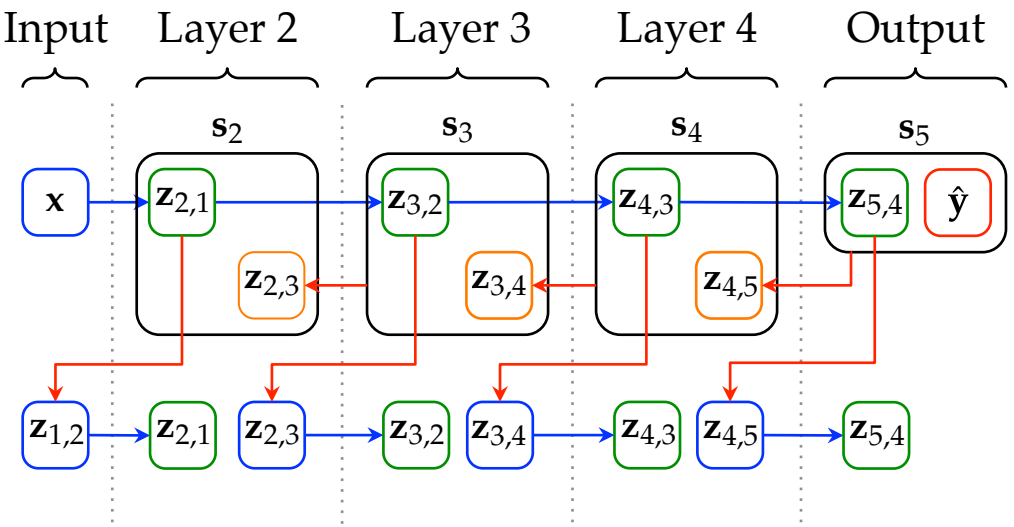
$$\Delta w_{l-1,ij} = \varepsilon \left(\left\langle z_{l-1,l-2,i} s_{l,j} \right\rangle_{data} - \left\langle z_{l-1,l,i} z_{l,l-1,j} \right\rangle_{recon} \right)$$

Top-Down Regularized Deep Network

Basic Building Block:

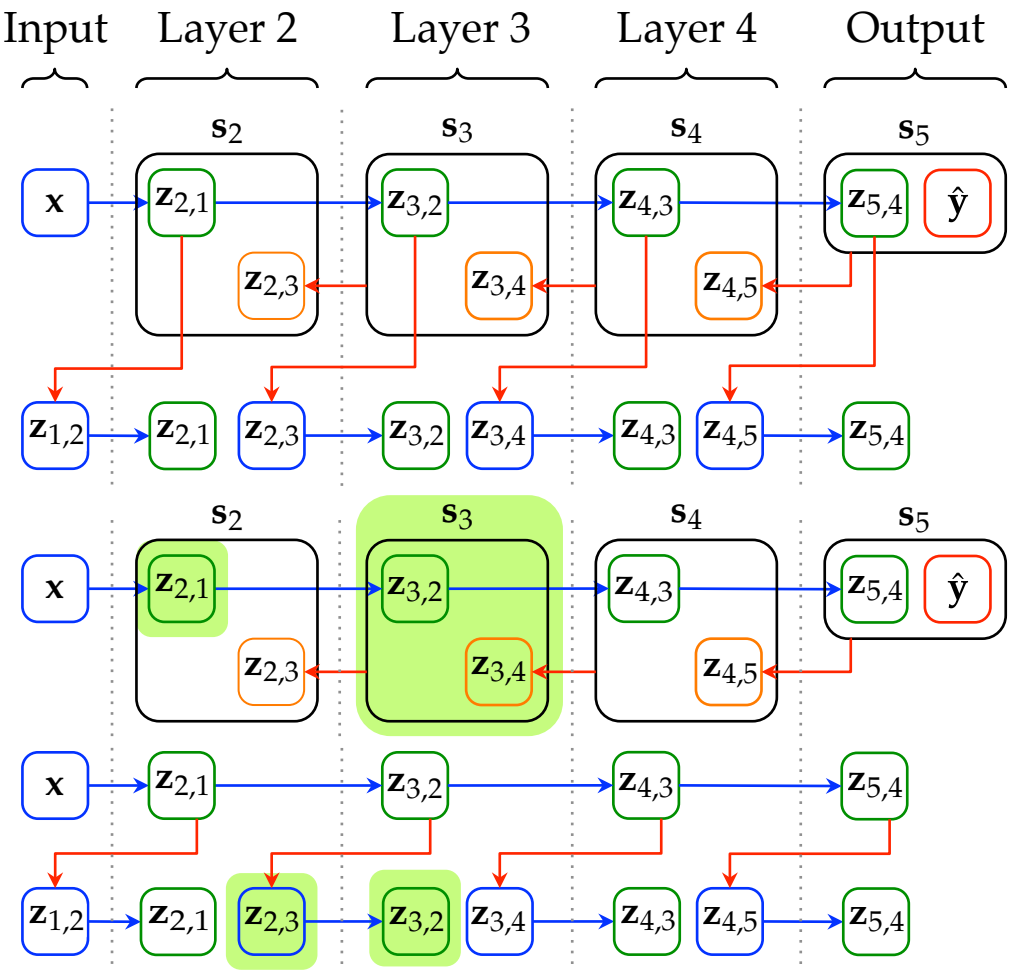


5-Layer Deep Network:



Deep Learning Algorithm

5-Layer Deep Network:



Step 1:
Forward-Backward Sampling

Step 2:
Alternating Gibbs Sampling

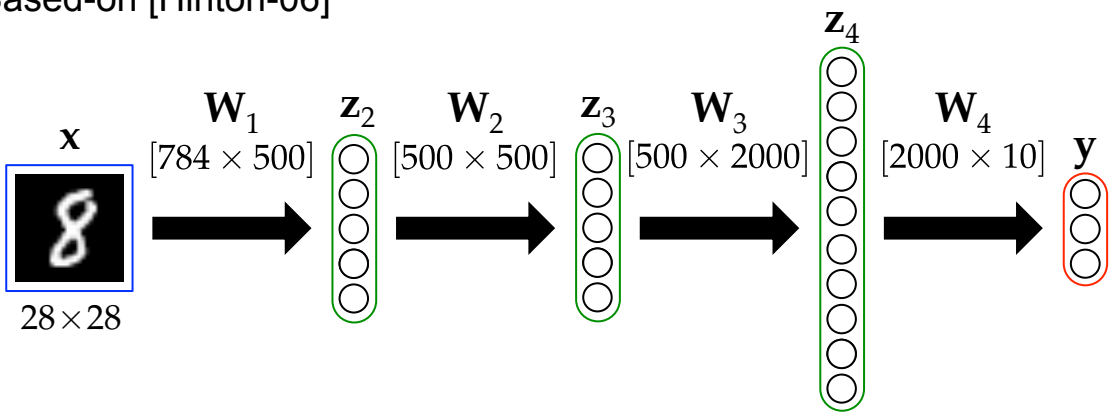
Step 3:
Update all RBMs

$$\Delta w_{l-1,ij} = \varepsilon \left(\left\langle z_{l-1,l-2,i} s_{l,j} \right\rangle_{data} - \left\langle z_{l-1,l,i} z_{l,l-1,j} \right\rangle_{recon} \right)$$

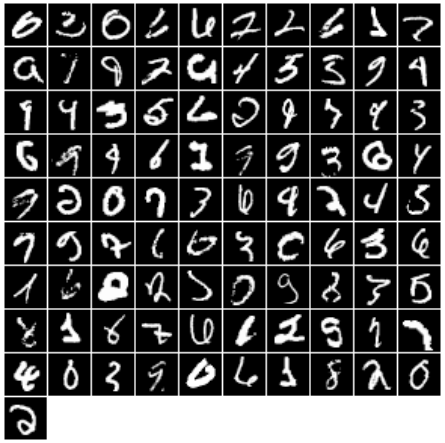
Results – Handwritten Digit Recognition

Deep Network for Handwritten Digit Recognition:

Based-on [Hinton-06]



Wrong Classifications:



Phase 1		*Phase 2*		*Phase 3*	
RBMs [Hinton-06]	2.49%	Up-down [Hinton-06]	1.25%	–	
Sparse & selective RBMs	2.14%	Forward-backward	1.14%	Backpropagation	0.98%
			1.06%		**0.91%**
		–			1.08%
Random weights	–	Forward-backward	1.61%	–	
Encoder-decoders	2.67%	SPREAD	1.25%	Backpropagation	1.03%
Random weights	–		1.58%	–	

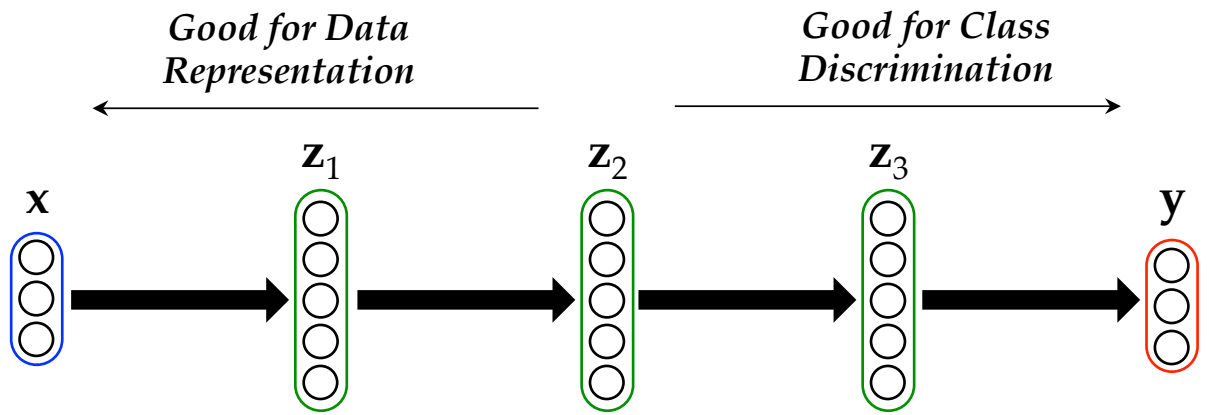
Summary – Deep Supervised Learning

Deep Supervised Optimization

- Three-phase deep learning
- Top-down regularized deep networks (global optimization)
- Simple implementation adapted from previous regularization scheme

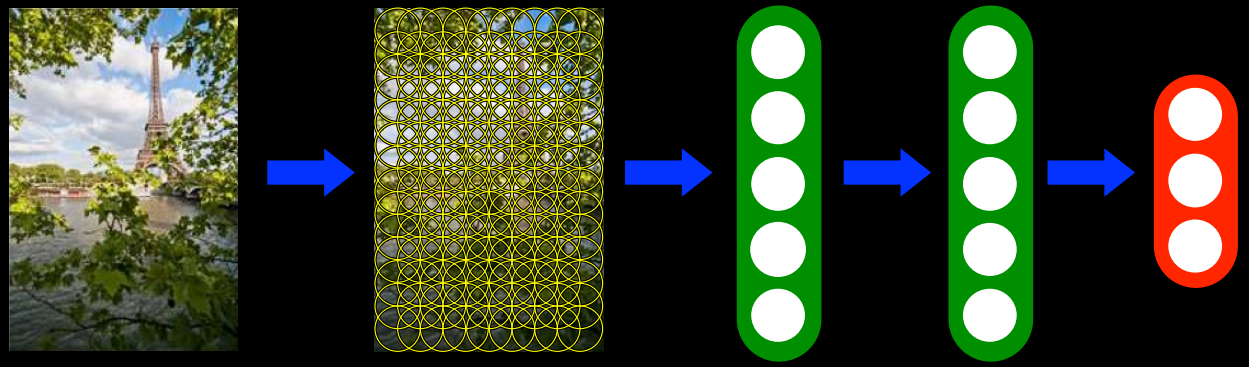
How to do Deep Learning?

- Bridge between fully-supervised to strongly discriminative learning
- Gradual transition between modelling the data and modelling the label

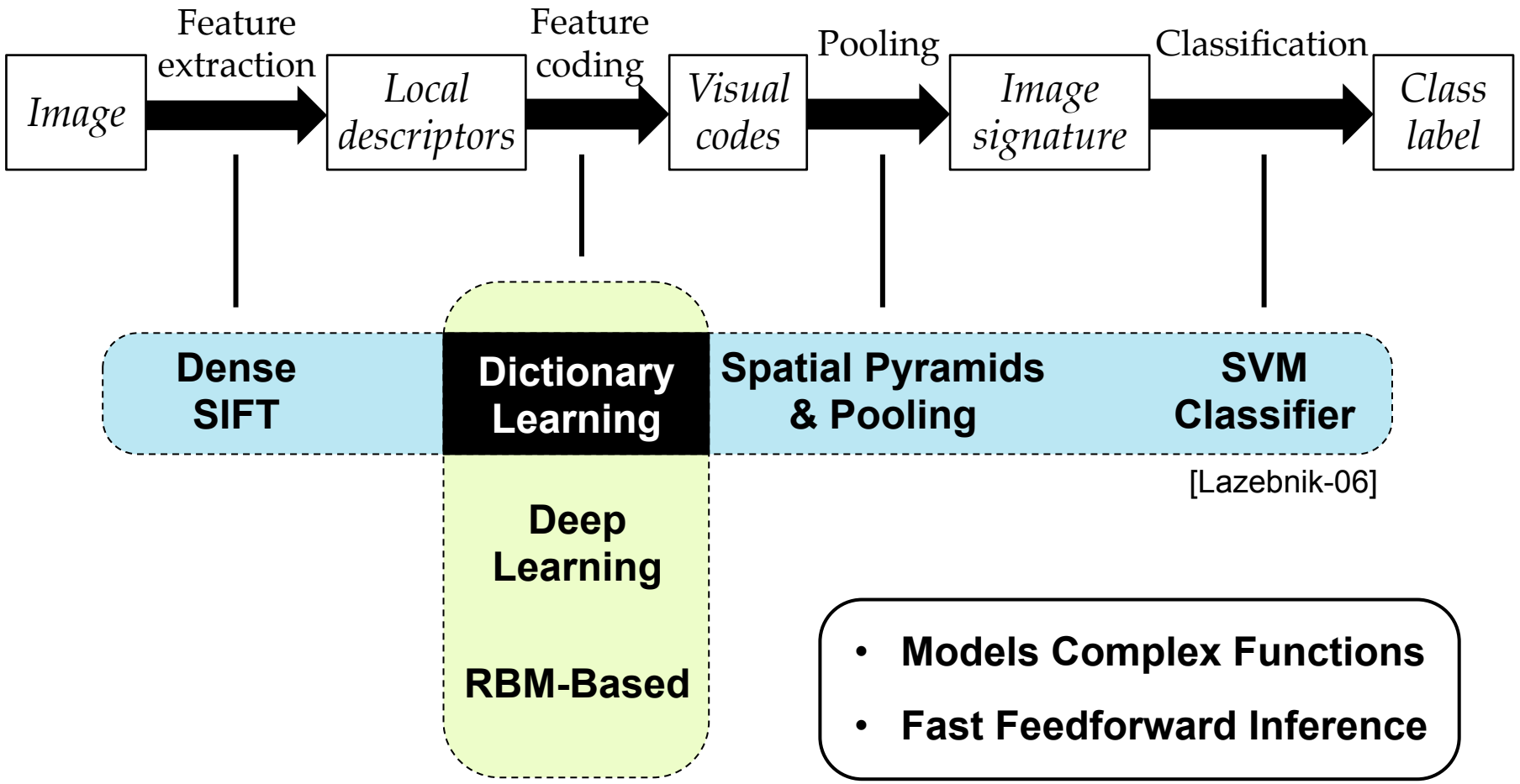


4. Learning Hierarchical Visual Codes

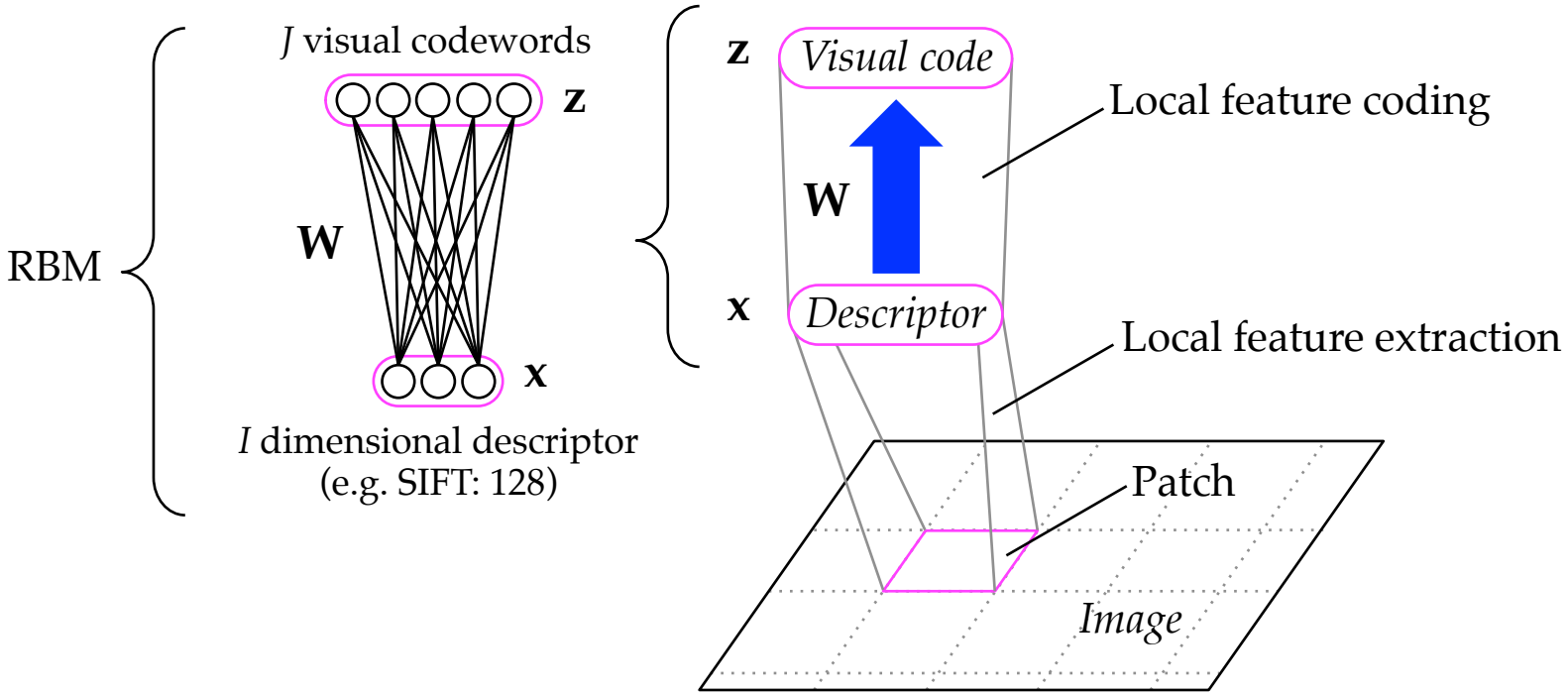
Image Classification from SIFT



Learning Visual Representations

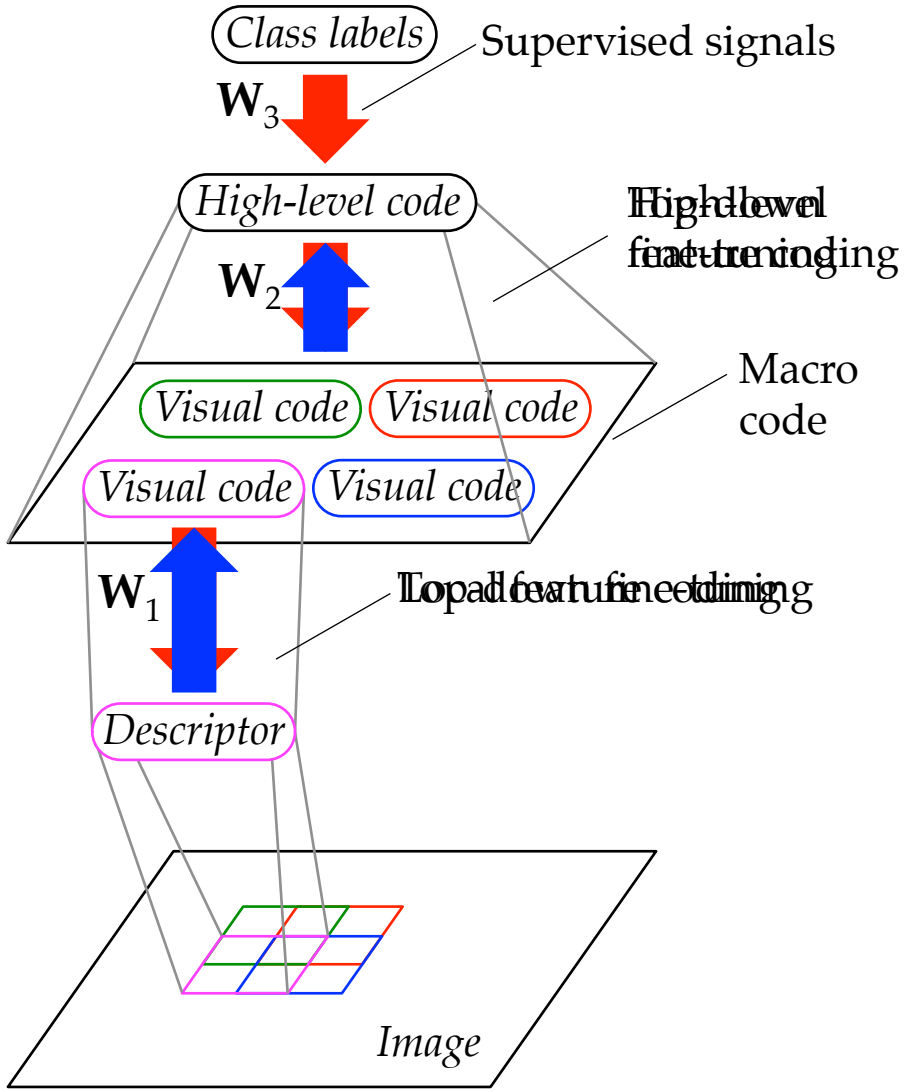


Single Layer Visual Dictionary



- Local descriptors are extracted from densely sampled patches
- Using an RBM to encode a local image descriptor (SIFT)
- RBM has two layers:
  - Input layer – descriptor
  - Latent layer – visual code

Hierarchical Visual Dictionary



- **Spatial Aggregation**
- **Greedy RBM Stacking**
- **Supervised Fine-Tuning**
  - Top-down regularized learning
  - Discriminative backpropagation

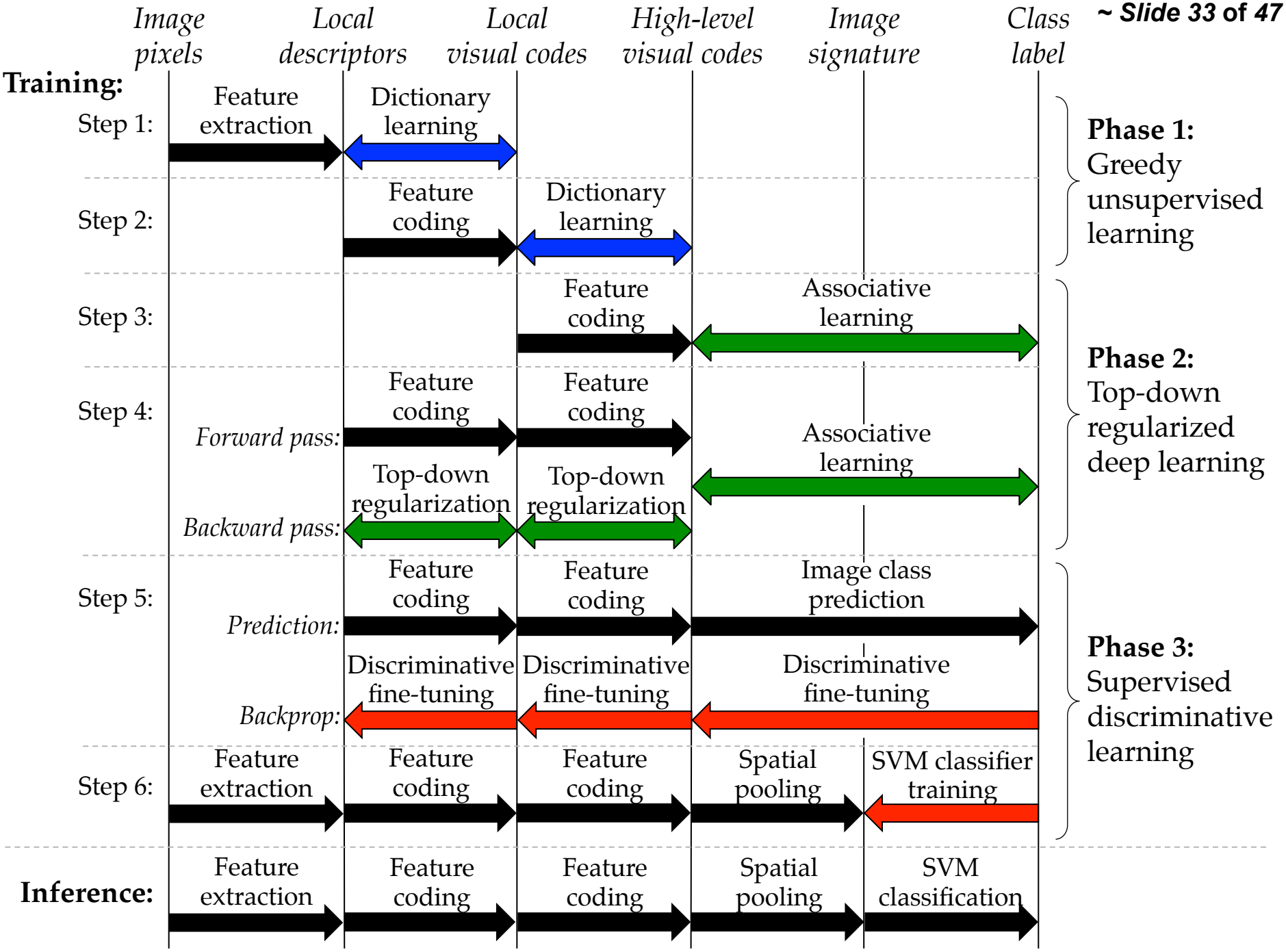
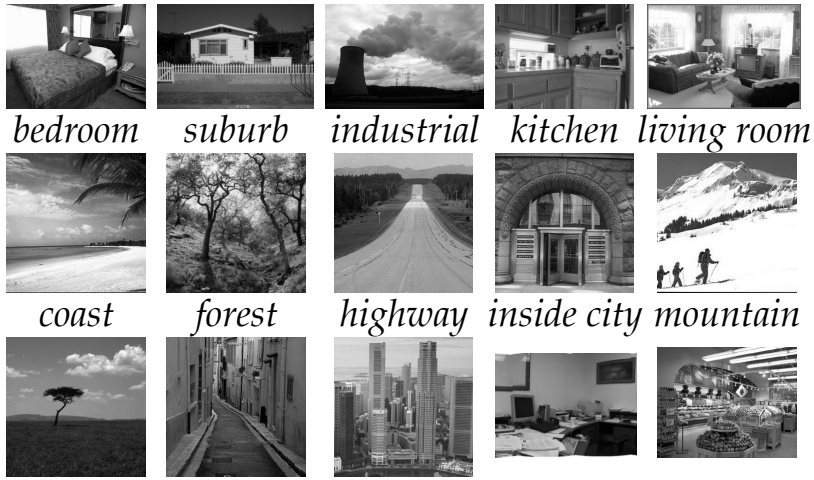


Image pixels
Local descriptors
Local visual codes
High-level visual codes
Image signature
Class label
Training:
Step 1:
Feature extraction
Dictionary learning
Step 2:
Feature coding
Dictionary learning
Step 3:
Feature coding
Associative learning
Step 4:
Forward pass:
Feature coding
Feature coding
Backward pass:
Top-down regularization
Top-down regularization
Associative learning
Step 5:
Prediction:
Feature coding
Feature coding
Image class prediction
Backprop:
Discriminative fine-tuning
Discriminative fine-tuning
Discriminative fine-tuning
Step 6:
Feature extraction
Feature coding
Feature coding
Spatial pooling
SVM classifier training
Inference:
Feature extraction
Feature coding
Feature coding
Spatial pooling
SVM classification
Phase 1: Greedy unsupervised learning
Phase 2: Top-down regularized deep learning
Phase 3: Supervised discriminative learning

Experimental Datasets

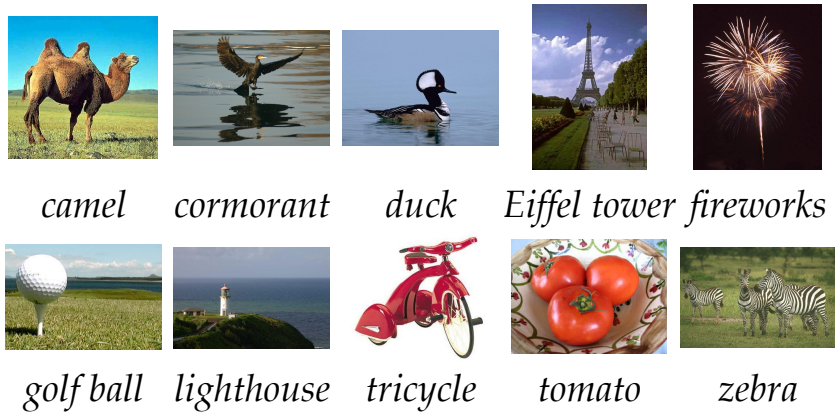
- **Object & Scene Recognition**
  - Single-label problems

15-Scenes

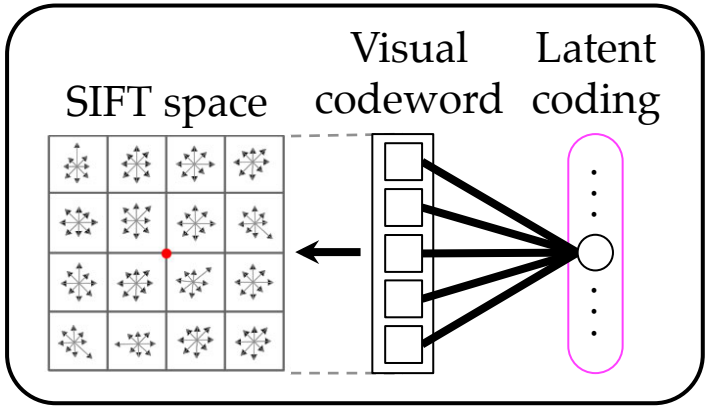


Caltech-101

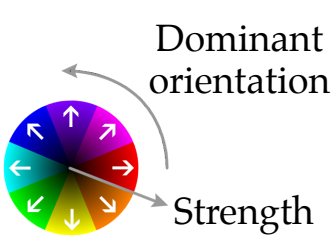
Caltech-256

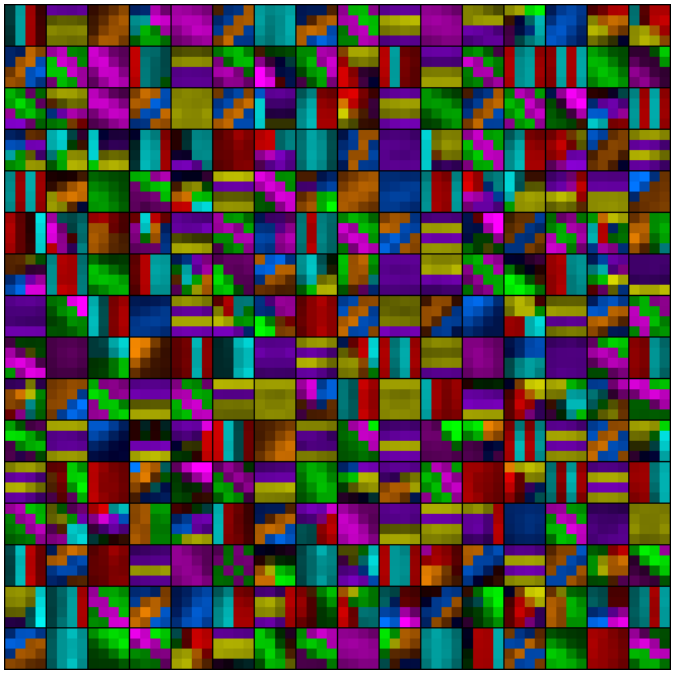


Visualization of Visual Codewords



- Automatically discovered spatially coherent features
- Features are diverse





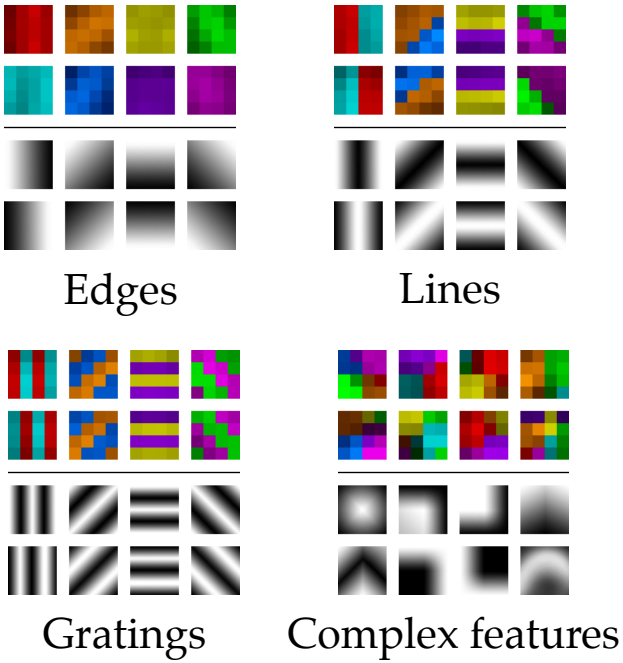


Image Classification Results

Architecture	*Caltech-101 (30 tr.)*	*15-Scenes (100 tr.)*	
Unsupervised Single-layer	78.0%	85.7%	
Supervised Single-layer	78.9%	86.0%	▶
Unsupervised Hierarchical	72.8%	82.5%	▼
Supervised Hierarchical	79.7%	86.4%	▲

- Image classification accuracies are high on a competitive task
- Visual dictionaries are small and concise
- Unsupervised hierarchies do not do as well as single-layer models
- Supervision is crucial for deep architectures;
  Less important for shallow architectures.

Comparison with Other Methods

Architecture	*Authors*	*Caltech-101 (30 tr.)*	*15-Scenes (100 tr.)*
Hard Assignment	[Lazebnik-06]	64.6%	81.1%
Soft Assignment	[Liu-11]	74.2%	82.7%
ScSPM	[Yang-09]	73.2%	80.3%
LLC	[Wang-10]	73.4%	–
Sparse Coding + Max Pooling	[Boureau-10]	75.7%	84.3%
Sparse RBM	[Sohn-11]	74.9%	–
CRBM	[Sohn-11]	77.8%	–
Discriminative Sparse Coding	[Boureau-10]	–	85.6%
LC-KSVD	[Jiang-11]	73.6%	–
Our Proposed Architecture		**79.7%**	**86.4%**

- Competitive results among feature coding methods
- Inference is faster than sparse decoder networks

The End, Thanks!

- Coding => Deep
- Pooling => learning ? Polling in deep
- Unsupervised / Supervised / Other

People involved – LIP6, Univ. UPMC Sorbonne Univ., Paris, France

Matthieu Cord, Nicolas Thome, matthieu.cord@lip6.fr

- PhD students : Sandra Avila, Hanlin Goh, Mar Law, Denis Pitzalis
- Post-Docs : Christian Theriault
- Research Inge. J. Guyomard

BossaNova Project Web page with codes available :
`https://sites.google.com/site/bossanovasite/`
JKernelMachines (Java) with D. Picard :
`https://mloss.org/software/view/409/`

http ://webia.lip6.fr/∼cord/