Machine Learning and Pattern Recognition

Unsupervised Learning
Sparse Coding

Remember K-Means

- Find K Prototype vectors (M^k)that best represent the data $Y^1 \ldots Y^P$

$$L = \sum_{i=1}^{P} \min_{k=1}^{K} ||Y^i - M^k||^2$$

- Minimize L wrt M^k

$$\frac{\partial L}{\partial M^k} = 2 \sum_{i \in S^k} (M^k - Y^i)$$

$$M^k = \frac{1}{|S^k|} \sum_{i \in S^k} Y^i$$

How to use K-Means?

* For a new sample *Y*, find *k* such that

$$k = \arg\min_k ||M^k - Y||_2^2$$

```
For k = 1..K
    zk = (Mk - Y)T (Mk - Y)
End
k = index of min(z)
```

* Representation (1 of K)

z = [0 0 0 0 ... . z_k .... 0 0 0]

Sparse Coding

* Represent an input vector using an ***overcomplete*** dictionary

$$\begin{pmatrix} \cdot \\ \cdot \\ y_i \\ \cdot \\ \cdot \end{pmatrix} \approx \begin{pmatrix} \cdot & \cdot & \cdot & D_0^j & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & D_i^j & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ z_j \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}$$

# of dictionary elements **>** size of input

# of zero elements **>>>** # of non-zero

Y

Input

D

Dictionary

z Representation (***sparse***)

* Each ***Y*** is represented using a linear combination of columns of ***D***
* How do we calculate **z** for a given **Y**?
* How do we learn **D**?

Sparse Coding - L0

1) Find the sparsest solution that satisfies a given reconstruction error

$$min||z||_0 \;\; s.t. \;\; ||Y - \sum_i D^i z_i||_2^2 \leq \epsilon$$

2) Find the best k-sparse representation that minimizes reconstruction error

$$min||Y - \sum_i D^i z_i||_2^2 \;\; s.t. \;\; ||z||_0 = k$$

* L0 minimization requires search
    * not tractable

Sparse Coding - L0

- Matching Pursuit Algorithms offer greedy solution [Mallat and Zhang '93]
- Greedily pick the dictionary element that reduces residual most
  - very fast, but unstable

```
Function MP (Y,D,n)
 R=Y,z=0
 for k=1..n
  i   = argmax(D^T R)
  z_i = D^i^T R
  R   = R - z_i D^i
 end
```

Sparse Coding - L1

- Relax L0 into closest convex penalty
- Equality of minimum for L0 and L1 is proven under certain conditions [Donoho and Elad '03]

$$\frac{1}{2}||Y - \sum_i D^i z_i||_2^2 + \lambda \sum_i |z_i|_1$$

Input Dictionary Representation

- Convex in **z** and **D** separately, not both
- Fast algorithms exist for solving wrt **z**

Sparse Coding - L1

- Iterative Shrinkage-Thresholding Algorithm (ISTA)
  - First order method
  - Formulation for a general family of objectives

$$\min_z F(z) + G(z)$$

Convex and smooth with Lipschitz constant L

Convex and non-smooth

- Quadratic Approximation at z'

$$Q(z)|_{z'} = F(z') + \langle z - z', \nabla F(z) \rangle + \frac{L}{2}||z - z'||^2 + G(z)$$

- Solution

$$z^{k+1} \leftarrow \arg\min_z G(z) + \frac{L}{2}||z - (z^k - \frac{1}{L}\nabla F(z^k))||^2$$

Sparse Coding - L1

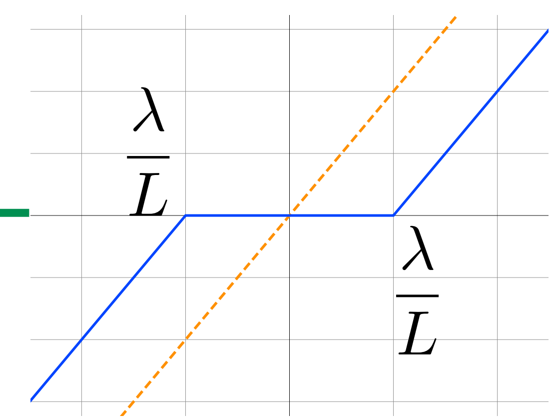
$$z^{k+1} \leftarrow \arg\min_z G(z) + \frac{L}{2}||z - \underbrace{(z^k - \frac{1}{L}\nabla F(z^k))}_{\mathbf{t}}||^2$$

$$\sum_i |z_i|_1$$

$$\frac{1}{2}||Y - \sum_i D^i z_i||_2^2$$

$$z^{k+1} \leftarrow \arg\min_z \ \lambda||z||_1 + \frac{L}{2}||z - t||^2$$

$$z^{k+1} \leftarrow sh_{\lambda/L}\left(z^k - \frac{1}{L}\nabla F(z^k)\right)$$



$$z^{k+1} \leftarrow sh_{\lambda/L}\left(z^k - \frac{1}{L}D^T(Dz - Y)\right)$$

- Loop until some convergence criterion is satisfied
- How do we get L?

Sparse Coding - L1

* **L** is the step size for gradient step
  * It is the smallest Lipschitz constant of the smooth function F(z) and is equal to largest eigenvalue of $D^T D$
  * In practice, one does a line search

```
Function ISTA (Y,D)
 L>0,c>1,z=0
 repeat
  Search L s.t. Q(z)>F(z)+G(z)
  z^{k+1} = sh(z^k-1/L D^T(Dz-Y))
 until convergence
```

Sparse Coding - Learning

- How about **D**?
  - We want to learn it
  - Adapt to data
  - Use online learning for **D**

- Per sample energy

$$E(Y, z, D) = \frac{1}{2}||Y - \sum_i D^i z_i||_2^2 + \lambda \sum_i |z_i|_1$$

- Loss

$$L(Y, D) = \frac{1}{|\mathcal{Y}|} \sum_{Y \in \mathcal{Y}} E(Y, z, D)$$

Sparse Coding - Learning

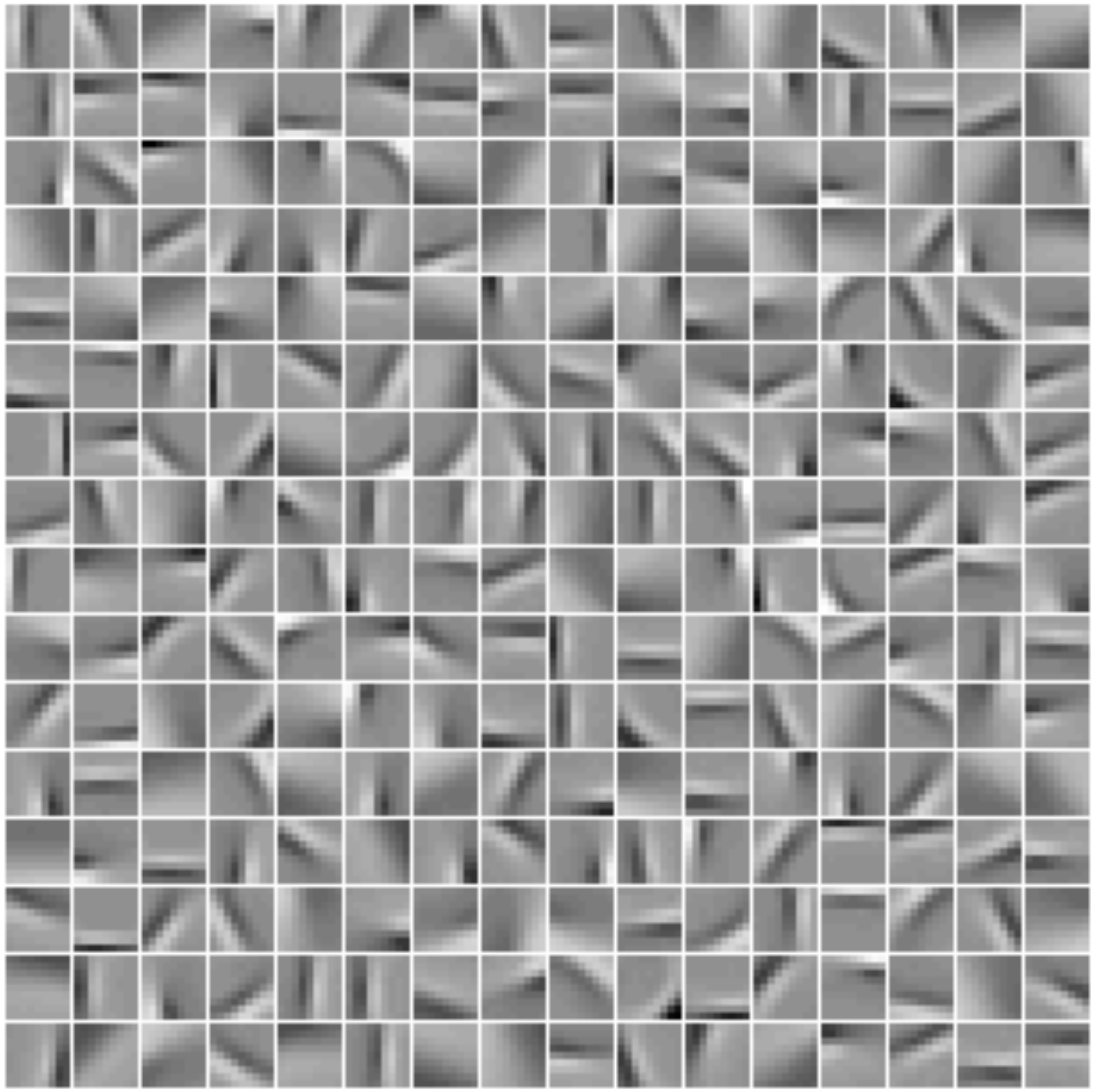
- For each sample, $Y \in \mathcal{Y}$
  1. do inference

     minimize ***E(Y,z,D)*** wrt **z** (use any SC algo)
  2. update parameters

     $$D \leftarrow D - \eta \frac{\partial E}{\partial D}$$
  3. Constrain elements of D to be unit norm
     - dictionary elements grow, z gets smaller, sparsity term gets discarded

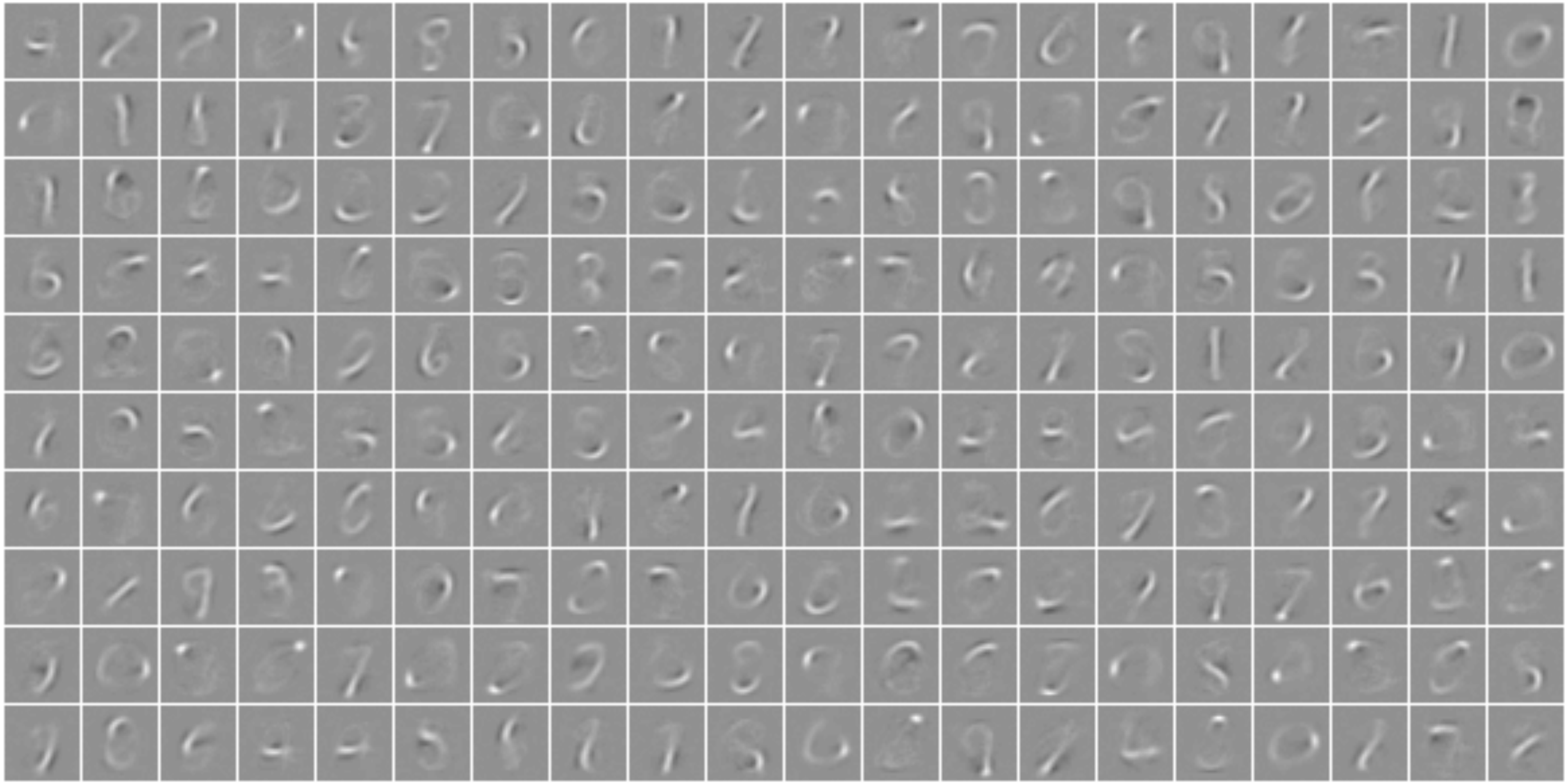
Sparse Coding - Learning

* Learned dictionary **D** by training in natural image patches



Sparse Coding - Learning

- Learned dictionary D by training on MNIST digits



Sparse Coding

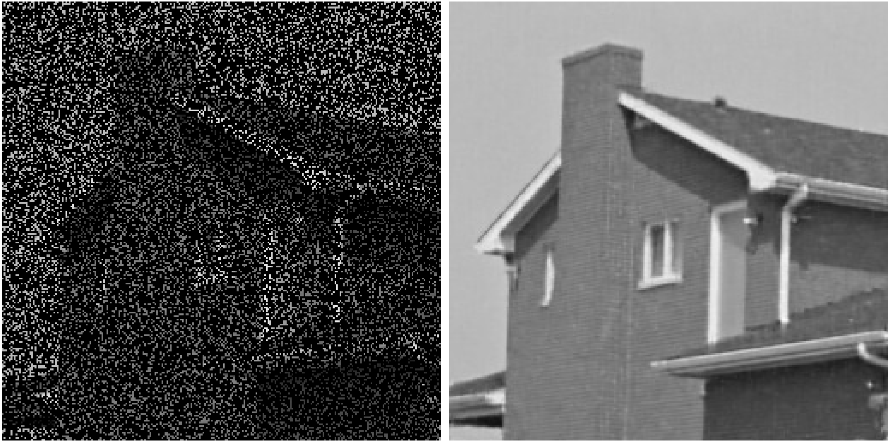
* Cool, available software?
  * http://www.di.ens.fr/willow/SPAMS/
  * http://cs.nyu.edu/~koray
* Applications
  * Image denoising
  * Inpainting
  * Classification
  * Recognition
  * ...

Image Processing Applications

- Slides from Julien Mairal
- http://www.di.ens.fr/~mairal/resources/pdf/ERMITES10.pdf

Sparse representations for image restoration

[Mairal, Sapiro, and Elad, 2008d]



Sparse representations for image restoration

Inpainting, [Mairal, Elad, and Sapiro, 2008b]

Sparse representations for image restoration

Inpainting, [Mairal, Elad, and Sapiro, 2008b]

Sparse representations for video restoration

Key ideas for video processing

[Protter and Elad, 2009]

- Using a 3D dictionary.
- Processing of many frames at the same time.
- Dictionary propagation.

Sparse representations for image restoration
Inpainting, [Mairal, Sapiro, and Elad, 2008d]

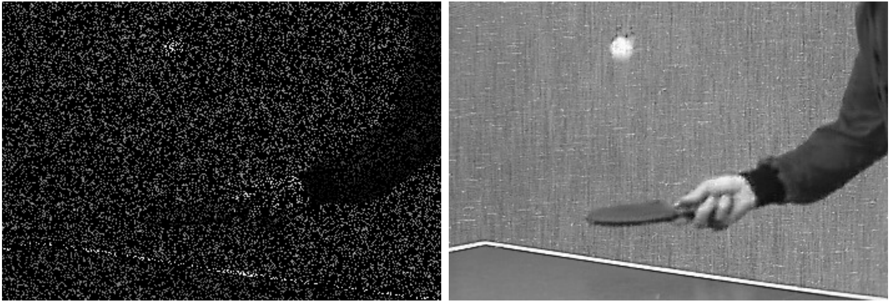


Figure: Inpainting results.

Sparse representations for image restoration
Inpainting, [Mairal, Sapiro, and Elad, 2008d]

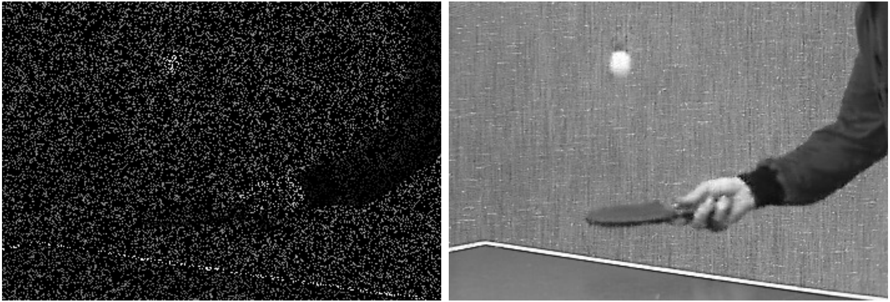
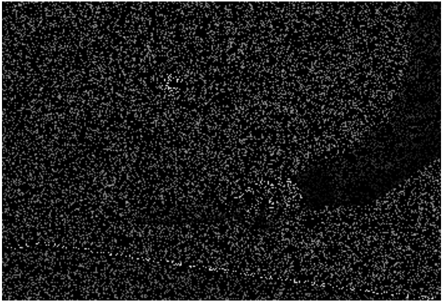


Figure: Inpainting results.

Sparse representations for image restoration
Inpainting, [Mairal, Sapiro, and Elad, 2008d]



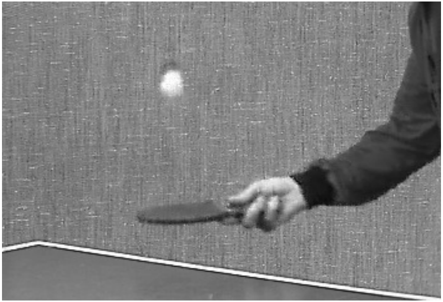


Figure: Inpainting results.

Sparse representations for image restoration

Inpainting, [Mairal, Sapiro, and Elad, 2008d]

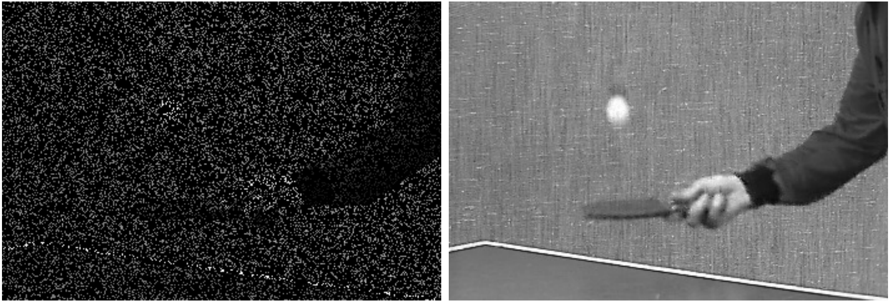


Figure: Inpainting results.

Sparse representations for image restoration

Inpainting, [Mairal, Sapiro, and Elad, 2008d]

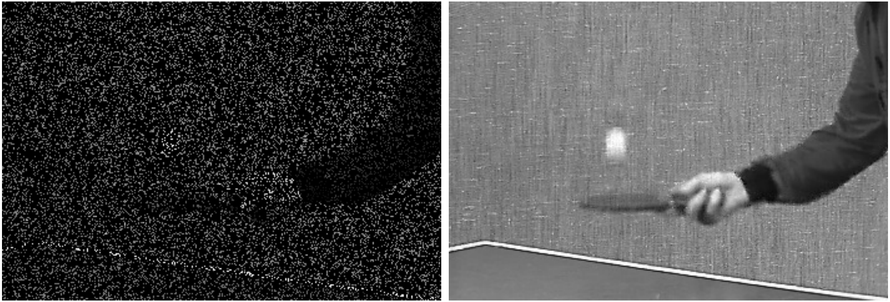


Figure: Inpainting results.

Sparse representations for image restoration
Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

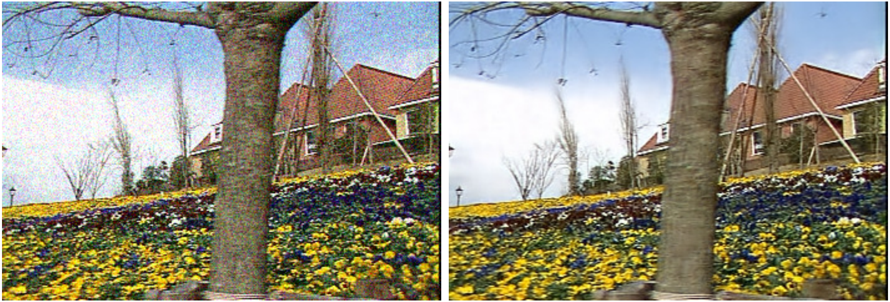


Figure: Denoising results. $\sigma = 25$

Sparse representations for image restoration

Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

Figure: Denoising results. $\sigma = 25$

Sparse representations for image restoration

Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

Figure: Denoising results. $\sigma = 25$

Sparse representations for image restoration
Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

Figure: Denoising results. $\sigma = 25$

Sparse representations for image restoration

Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

Figure: Denoising results. $\sigma = 25$

Digital Zooming

Couzinie-Devy, 2010, Original

Digital Zooming

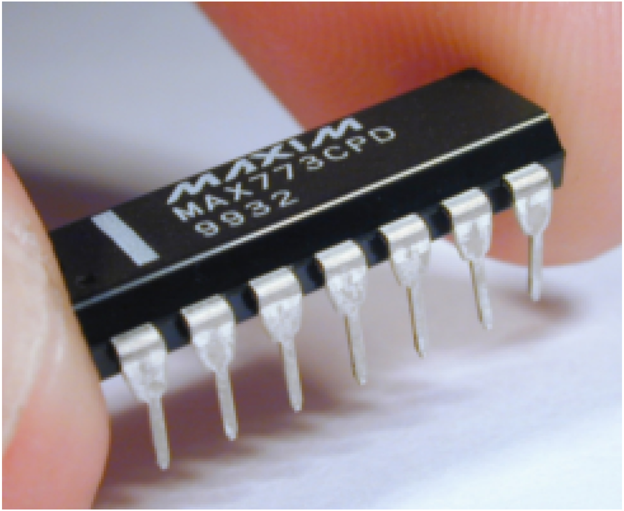
Couzinie-Devy, 2010, Bicubic

Digital Zooming

Couzinie-Devy, 2010, Proposed method

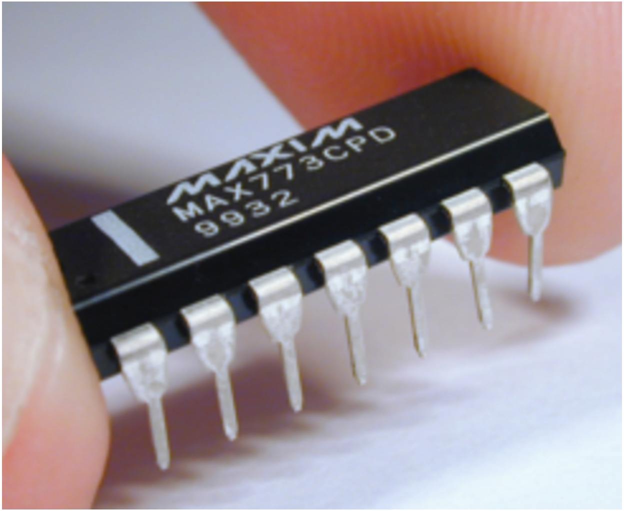
Digital Zooming

Couzinie-Devy, 2010, Original



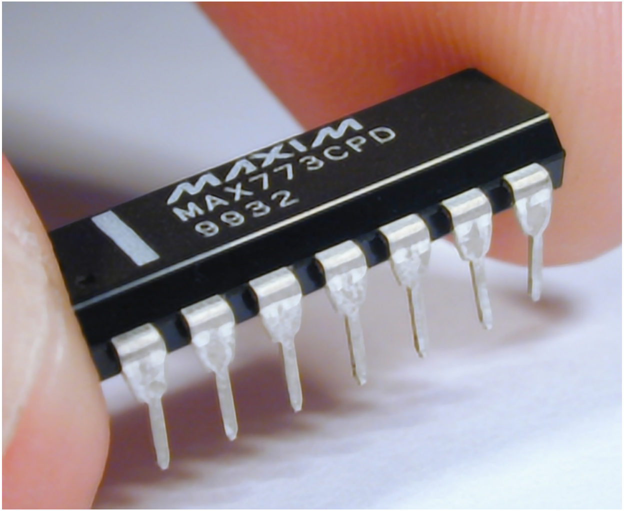
Digital Zooming

Couzinie-Devy, 2010, Bicubic



Digital Zooming

Couzinie-Devy, 2010, Proposed approach



Inverse half-toning

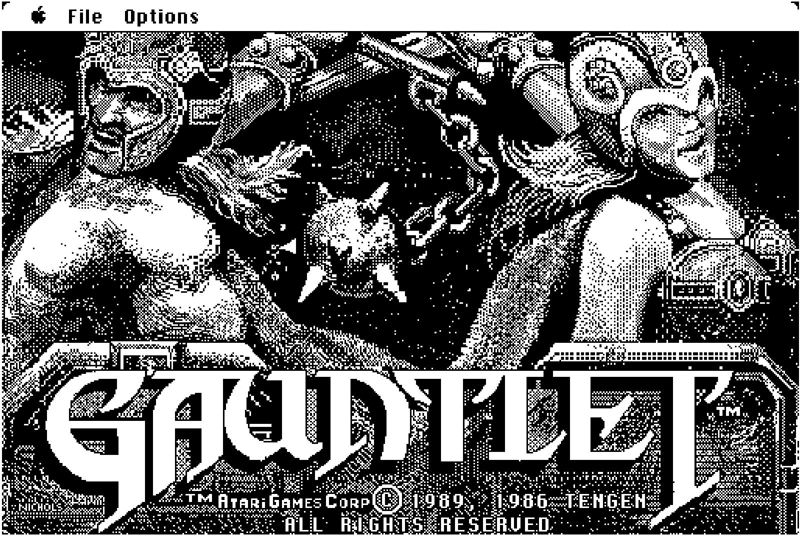
Original

Inverse half-toning

Reconstructed image

Inverse half-toning

Original



Inverse half-toning

Reconstructed image

Inverse half-toning

Original

Inverse half-toning

Reconstructed image

Inverse half-toning

Original

Inverse half-toning

Reconstructed image

Inverse half-toning

Original

Inverse half-toning

Reconstructed image

Optimization for Dictionary Learning
Inpainting a 12-Mpixel photograph

Optimization for Dictionary Learning
Inpainting a 12-Mpixel photograph

Optimization for Dictionary Learning
Inpainting a 12-Mpixel photograph

Optimization for Dictionary Learning
Inpainting a 12-Mpixel photograph

Sparse Coding for Recognition

* Recognition requires two basic operations
  * Feature extraction
  * Classification
* Feature extraction

$$\mathcal{F}(Y) : Y \mapsto z$$

* SIFT, HoG, ...
* Convolutional Nets (w/o the last layer)
* Use sparse coding inference as feature extractor
  * MP(Y) or ISTA(Y)

Sparse Coding for Recognition

- Mid-level feature extraction

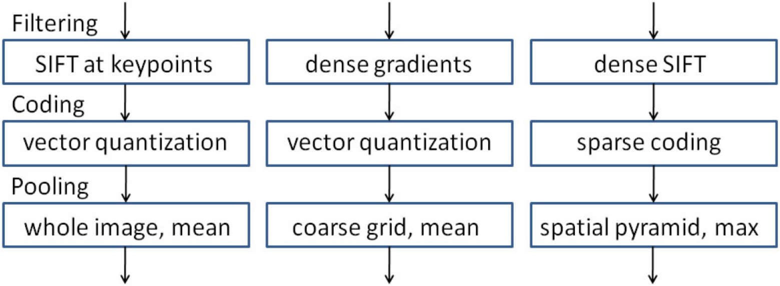
  $$\mathcal{F}(z) : z \mapsto z'$$

- First layer
  - SIFT, HoG, ...
- Second layer
  - Quantize into a visual wordbook
  - Do Sparse coding to learn wordbook

Learning Mid-Level Features

- Y-Lan Boureau
  - Learning Mid-Level Features for Object Recognition, *CVPR'2010*
- Slides from Julien Mairal
- http://www.di.ens.fr/~mairal/resources/pdf/ERMITES10.pdf

Learning Codebooks for Image Classification



Idea

Replacing Vector Quantization by Learned Dictionaries!

- unsupervised: [Yang et al., 2009]
- supervised: [Boureau et al., 2010, Yang et al., 2010]

Learning Codebooks for Image Classification

Let an image be represented by a set of low-level descriptors $\mathbf{y}_i$ at N locations identified with their indices $i = 1, \ldots, N$.

- hard-quantization:

$$\mathbf{y}_i \approx \mathbf{D}\boldsymbol{\alpha}_i, \quad \boldsymbol{\alpha}_i \in \{0,1\}^p \text{ and } \sum_{j=1}^{p} \boldsymbol{\alpha}_i[j] = 1$$

- soft-quantization:

$$\boldsymbol{\alpha}_i[j] = \frac{e^{-\beta\|\mathbf{y}_i - \mathbf{d}_j\|_2^2}}{\sum_{k=1}^{p} e^{-\beta\|\mathbf{y}_i - \mathbf{d}_k\|_2^2}}$$

- sparse coding:

$$\mathbf{y}_i \approx \mathbf{D}\boldsymbol{\alpha}_i, \quad \boldsymbol{\alpha}_i = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

Learning Codebooks for Image Classification

Table from Boureau et al. [2010]

Method	Caltech-101, 30 training examples: Average Pool	Caltech-101, 30 training examples: Max Pool	15 Scenes, 100 training examples: Average Pool	15 Scenes, 100 training examples: Max Pool
	Results with basic features, SIFT extracted each 8 pixels			
Hard quantization, linear kernel	51.4 ± 0.9 [256]	64.3 ± 0.9 [256]	73.9 ± 0.9 [1024]	80.1 ± 0.6 [1024]
Hard quantization, intersection kernel	64.2 ± 1.0 [256] **(1)**	64.3 ± 0.9 [256]	80.8 ± 0.4 [256] **(1)**	80.1 ± 0.6 [1024]
Soft quantization, linear kernel	57.9 ± 1.5 [1024]	69.0 ± 0.8 [256]	75.6 ± 0.5 [1024]	81.4 ± 0.6 [1024]
Soft quantization, intersection kernel	66.1 ± 1.2 [512] **(2)**	70.6 ± 1.0 [1024]	81.2 ± 0.4 [1024] **(2)**	83.0 ± 0.7 [1024]
Sparse codes, linear kernel	61.3 ± 1.3 [1024]	$\mathbf{71.5 \pm 1.1}$ [1024] **(3)**	76.9 ± 0.6 [1024]	83.1 ± 0.6 [1024] **(3)**
Sparse codes, intersection kernel	70.3 ± 1.3 [1024]	$\mathbf{71.8 \pm 1.0}$ [1024] **(4)**	83.2 ± 0.4 [1024]	$\mathbf{84.1 \pm 0.5}$ [1024] **(4)**
	Results with macrofeatures and denser SIFT sampling			
Hard quantization, linear kernel	55.6 ± 1.6 [256]	70.9 ± 1.0 [1024]	74.0 ± 0.5 [1024]	80.1 ± 0.5 [1024]
Hard quantization, intersection kernel	68.8 ± 1.4 [512]	70.9 ± 1.0 [1024]	81.0 ± 0.5 [1024]	80.1 ± 0.5 [1024]
Soft quantization, linear kernel	61.6 ± 1.6 [1024]	71.5 ± 1.0 [1024]	76.4 ± 0.7 [1024]	81.5 ± 0.4 [1024]
Soft quantization, intersection kernel	70.1 ± 1.3 [1024]	73.2 ± 1.0 [1024]	81.8 ± 0.4 [1024]	83.0 ± 0.4 [1024]
Sparse codes, linear kernel	65.7 ± 1.4 [1024]	$\mathbf{75.1 \pm 0.9}$ [1024]	78.2 ± 0.7 [1024]	83.6 ± 0.4 [1024]
Sparse codes, intersection kernel	73.7 ± 1.3 [1024]	$\mathbf{75.7 \pm 1.1}$ [1024]	83.5 ± 0.4 [1024]	$\mathbf{84.3 \pm 0.5}$ [1024]

	Unsup	Discr
Linear	83.6 ± 0.4	$\mathbf{84.9 \pm 0.3}$
Intersect	84.3 ± 0.5	$\mathbf{84.7 \pm 0.4}$

Yang et al. [2009] have won the PASCAL VOC'09 challenge using this kind of techniques.

Training Predictors

- Sparse coding solution requires optimization : L0 or L1
- Many efficient algorithms, but still slow
- Can we train a feed-forward predictor function for feature extraction?
  - Predictive Sparse Decomposition (PSD)

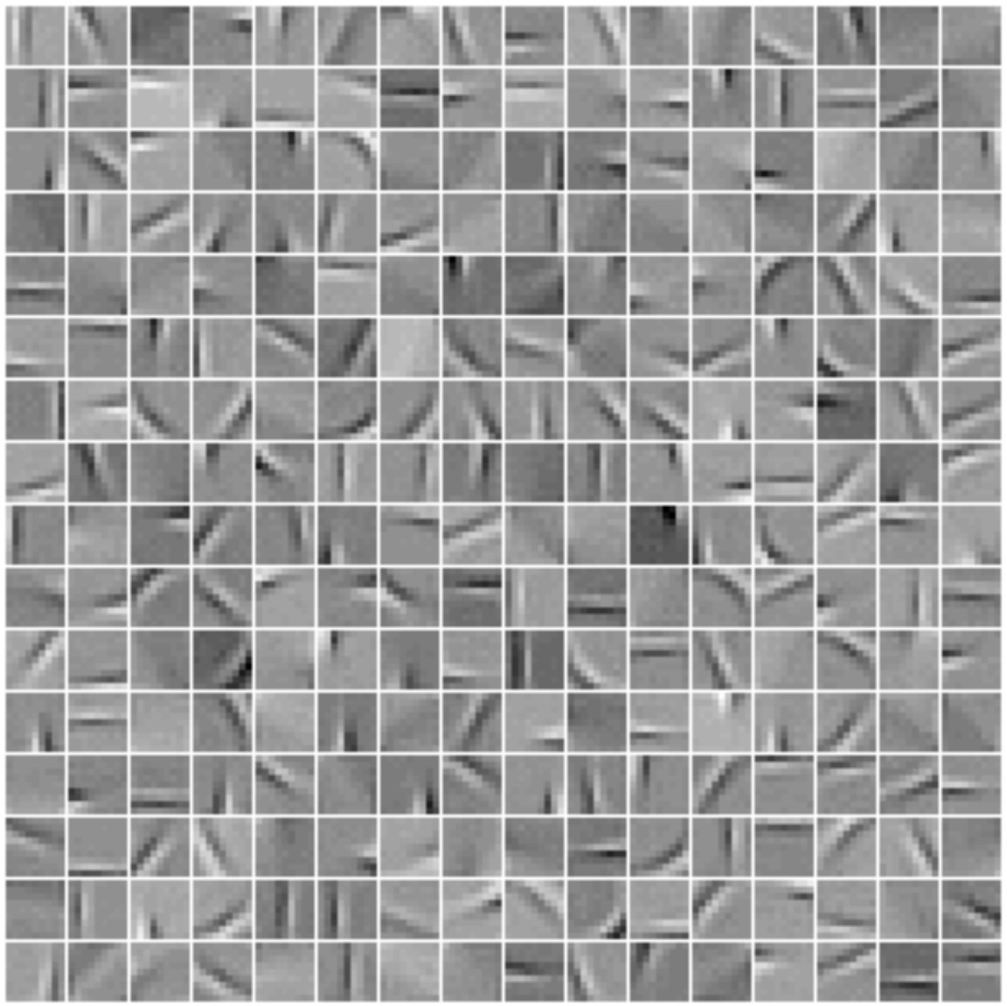
Predictive Sparse Decomposition

$$E(Y, z, D, K) = \underbrace{\frac{1}{2}||Y - \sum_i D^i z_i||_2^2 + \lambda \sum_i |z_i|_1}_{\text{Sparse Coding}} + \underbrace{\beta||z - C(Y; K)||_2^2}_{\text{Prediction}}$$

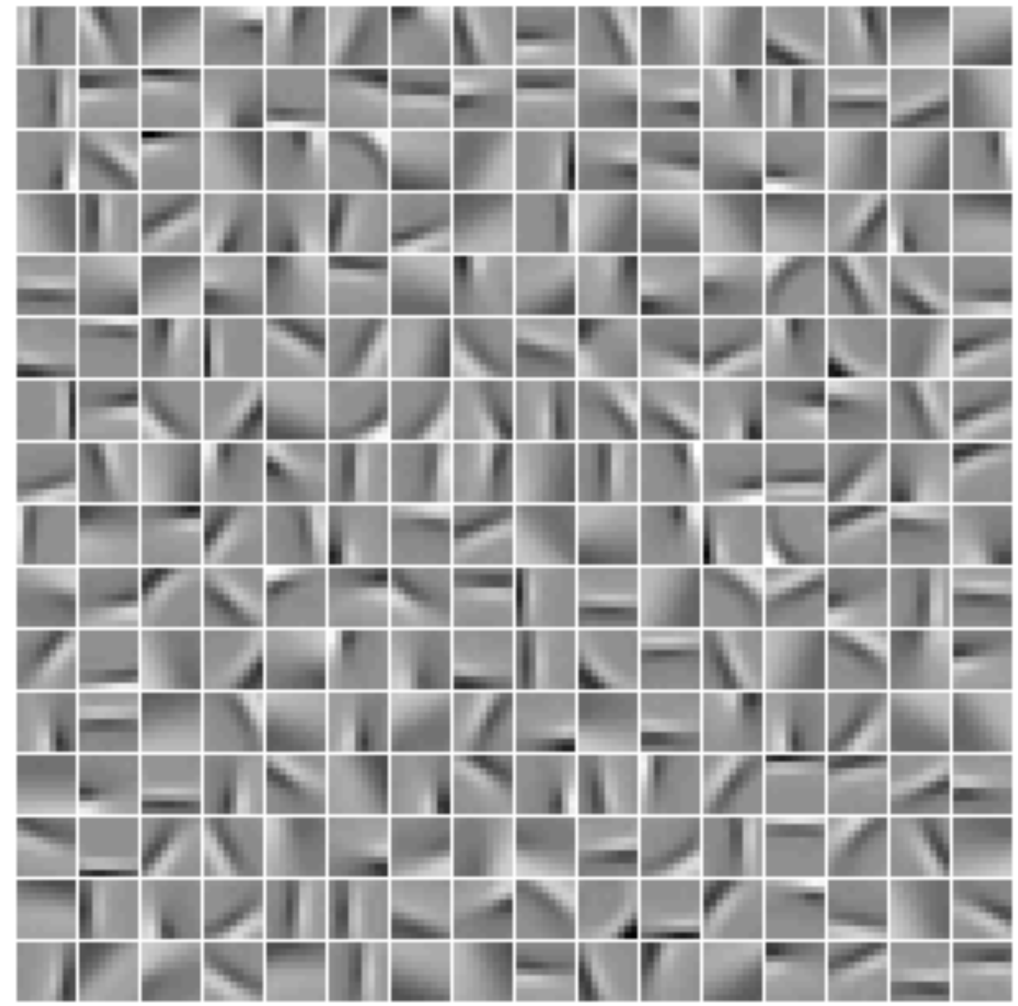
$$C(Y; K) = g \cdot tanh(y * k)$$

* **Learning**
* For each sample from data, do:
  1. Fix *K* and *D*, minimize to get optimal z
  2. Using the optimal value of z update *D* and *K*
  3. Scale elements of *D* to be unit norm.

Predictive Sparse Decomposition



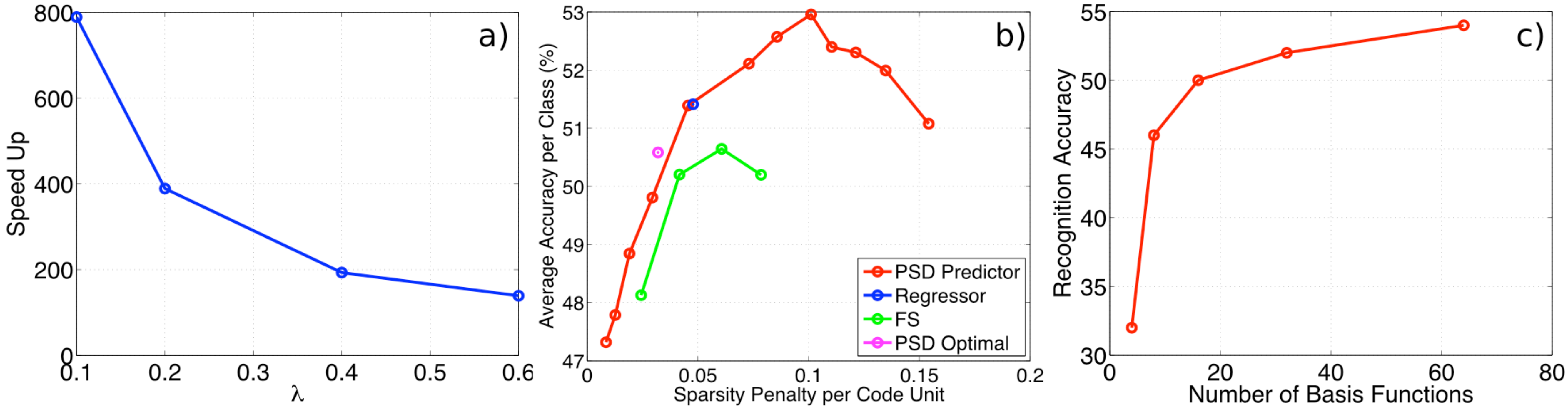
Encoder (*K*)



Decoder (*D*)

- 12 x12 natural image patches
- 256 dictionary elements

Recognition - C101

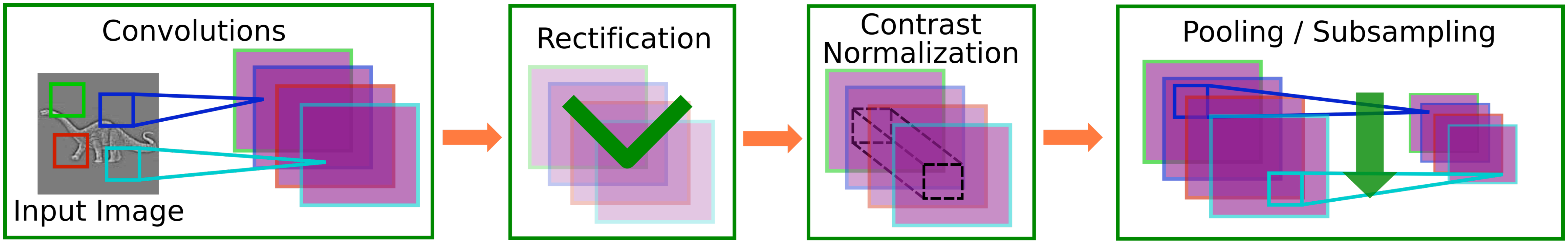


- Optimal (Feature Sign, Lee'07) vs PSD features
- PSD features perform slightly better
- Naturally optimal point of sparsity
- After 64 features not much gain
- PSD features are hundreds of times faster

Training Deep Networks with PSD

- Train layer-wise [Hinton'06]
  - $C(Y, K^1)$
  - $C(f(K^1), K^2)$
  - $C(f(K^2), K^3)$
  - ...
- Each layer is trained on the output $f(\mathbf{z})$ produced from previous layer.
- f is a series of non-linearity and pooling operations

Multi-Stage Object Recognition

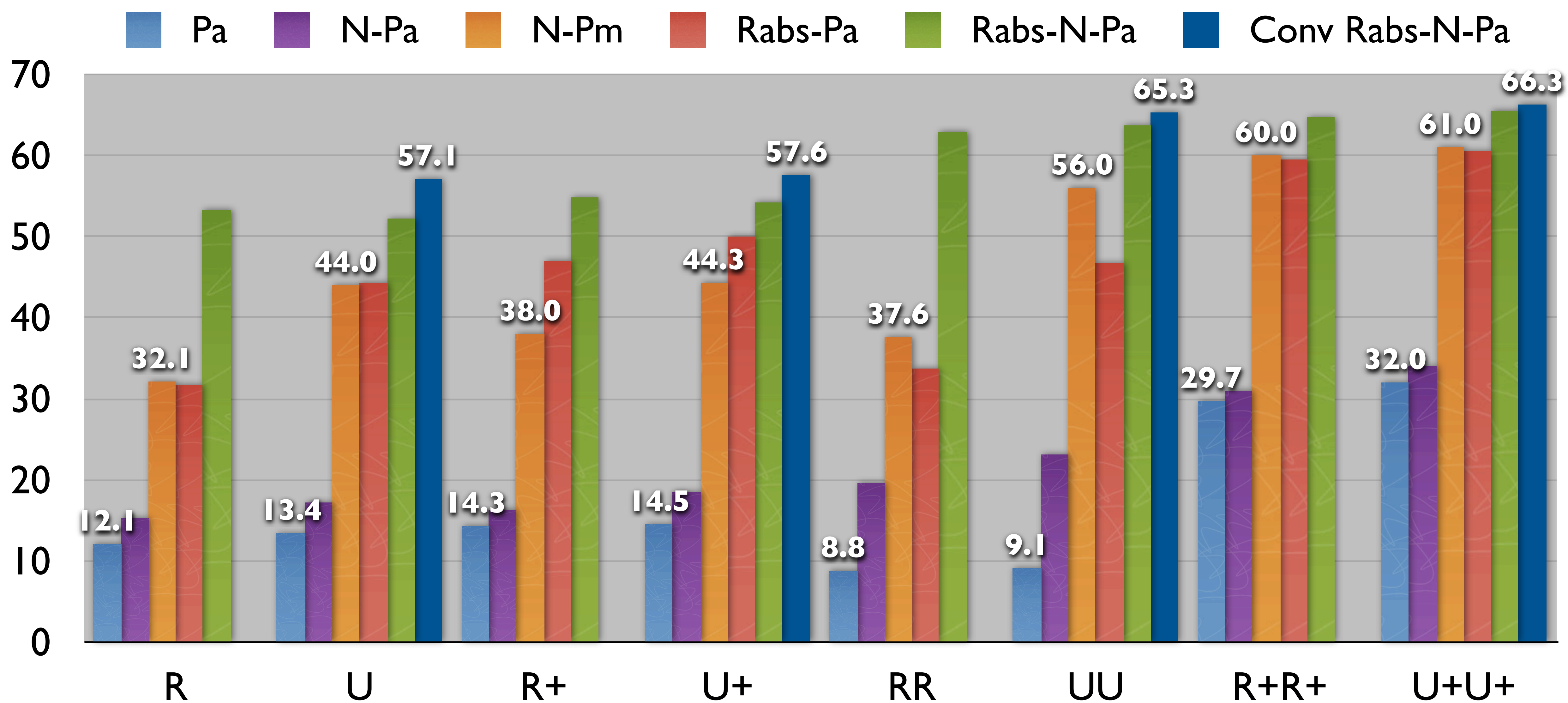


Filterbank - C(x;K) Non-linearities Pooling

	Filterbank	Abs	LCN	Pooling
Conv Net	Learned	✗	✗	Average
HMAX	Gabor	✗	✗	Max

- Building block of a multi-stage architecture
- Only the *Filterbank* is learned
- 0.53% on MNIST using 3 stages

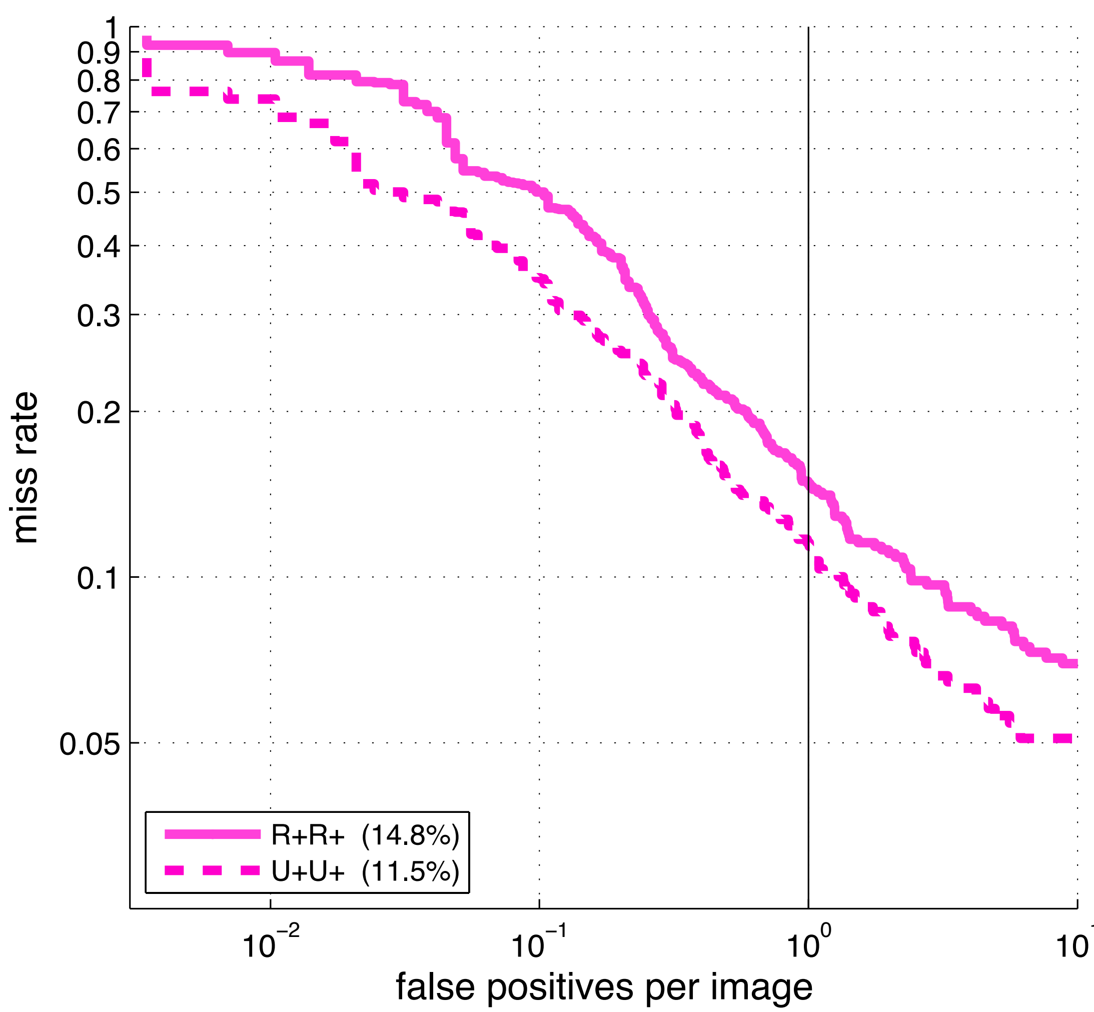
Object Recognition - Caltech 101

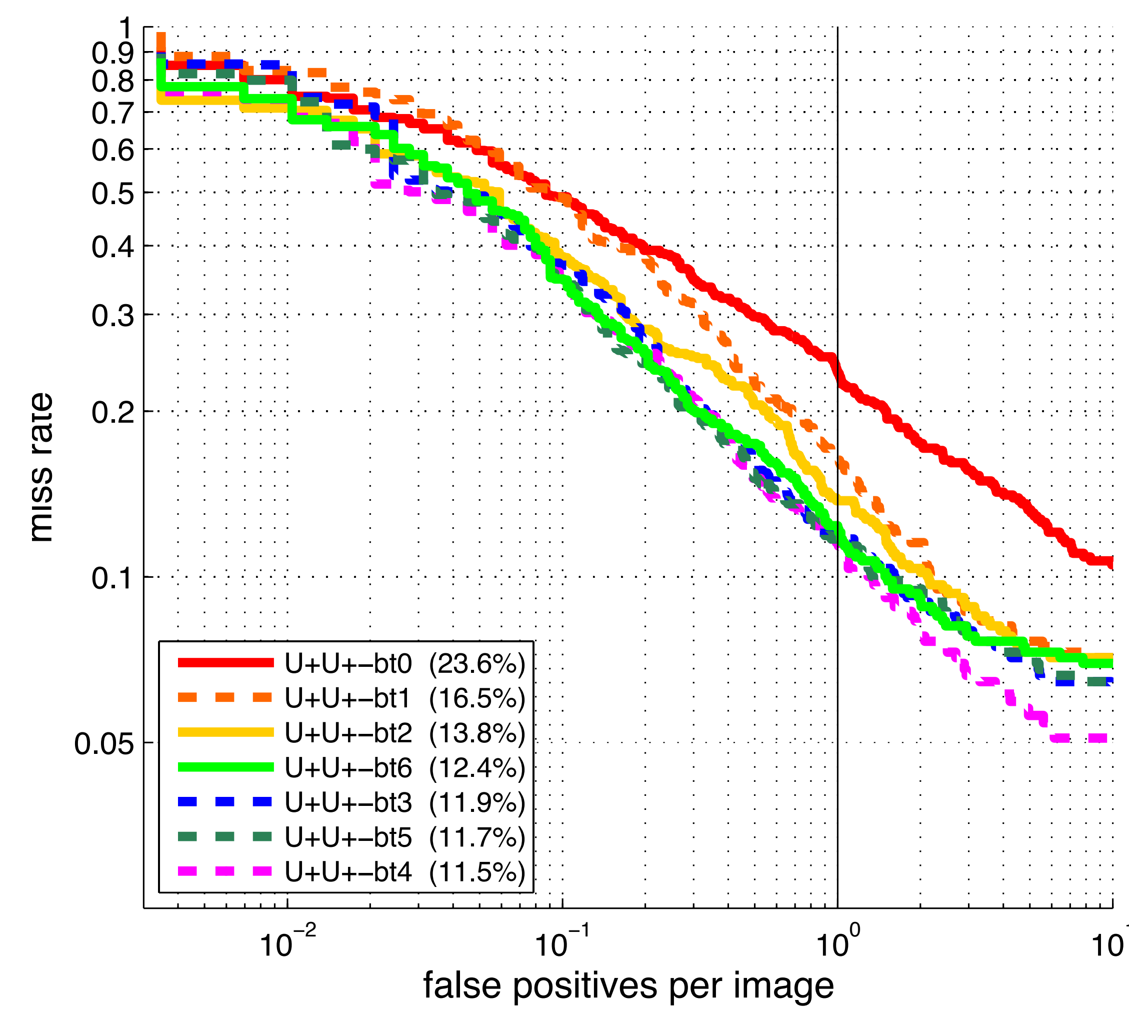


	R	RR	U	UU
Unsupervised	✘	✘	✓	✓
Random	✓	✓	✘	✘
Supervised	R^+	$R^+ R^+$	U^+	$U^+ U^+$

Pedestrian Detection

- Convolutional Predictive Sparse Decomposition
- 2 layer architecture
- INRIA Dataset
- Unsupervised Learning improves by 20%





Pedestrian Detection