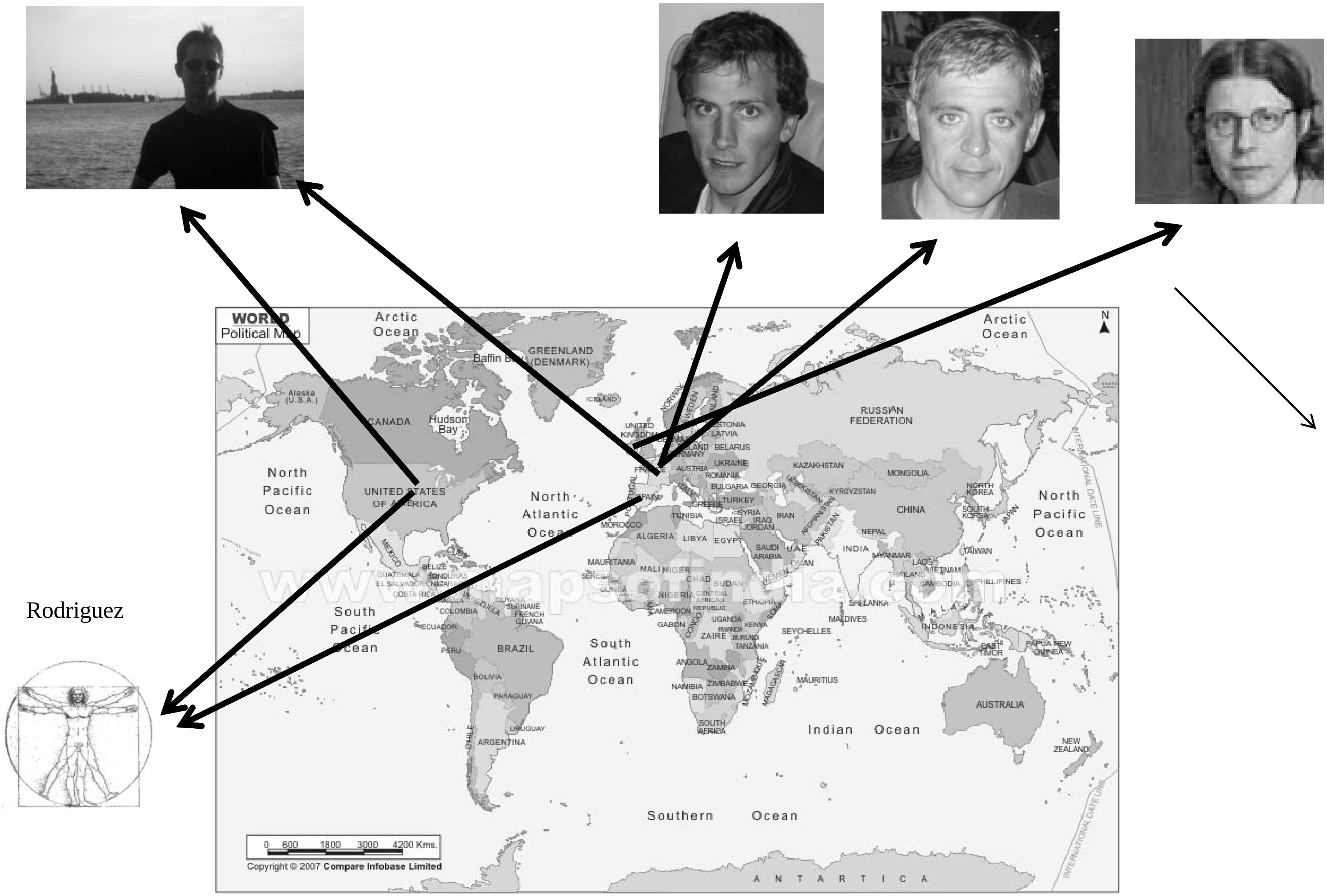
# *Learning to classify*

**Guillermo Sapiro**

University of Minnesota

***Supported by NSF, NGA, NIH, ONR, DARPA, ARO, McKnight Foundation***

***(Some slides adapted from M. Elad)***


Rodriguez
WORLD
Political Map
Arctic Ocean
North Pacific Ocean
North Atlantic Ocean
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Southern Ocean
A N T A R T I C A
GREENLAND (DENMARK)
CANADA
UNITED STATES OF AMERICA
BRAZIL
RUSSIAN FEDERATION
CHINA
INDIA
AUSTRALIA
0 600 1800 3000 4200 Kms.
Copyright © 2007 Compare Infobase Limited

# Goal and Outline

- Introduce and Extend "Learning Sparse Representations"
  - *Mairal, Elad, Sapiro, IEEE-TIP and SIAM-MMS, 2008*


- Learning to classify
  - *Mairal, Bach, Ponce, Sapiro, Zisserman, CVPR 2008*
  - *Rodriguez and Sapiro, IMA pre-print, 2008.*


# Introduction: Sparse and Redundant Representations

***Webster Dictionary:*** **Of few and scattered elements**

# *Restoration* by Energy Minimization

Restoration/representation algorithms are often related to the minimization of an energy function of the form

$$f(\underline{x}) = \frac{1}{2}\|\underline{x} - \underline{y}\|_2^2 + \Pr(\underline{x})$$

Relation to measurements

Prior or regularization

$\underline{y}$ : Given measurements

$\underline{x}$ : Unknown to be recovered

- Bayesian type of approach
- What is the prior? What is the image model?

Thomas Bayes
1702 - 1761

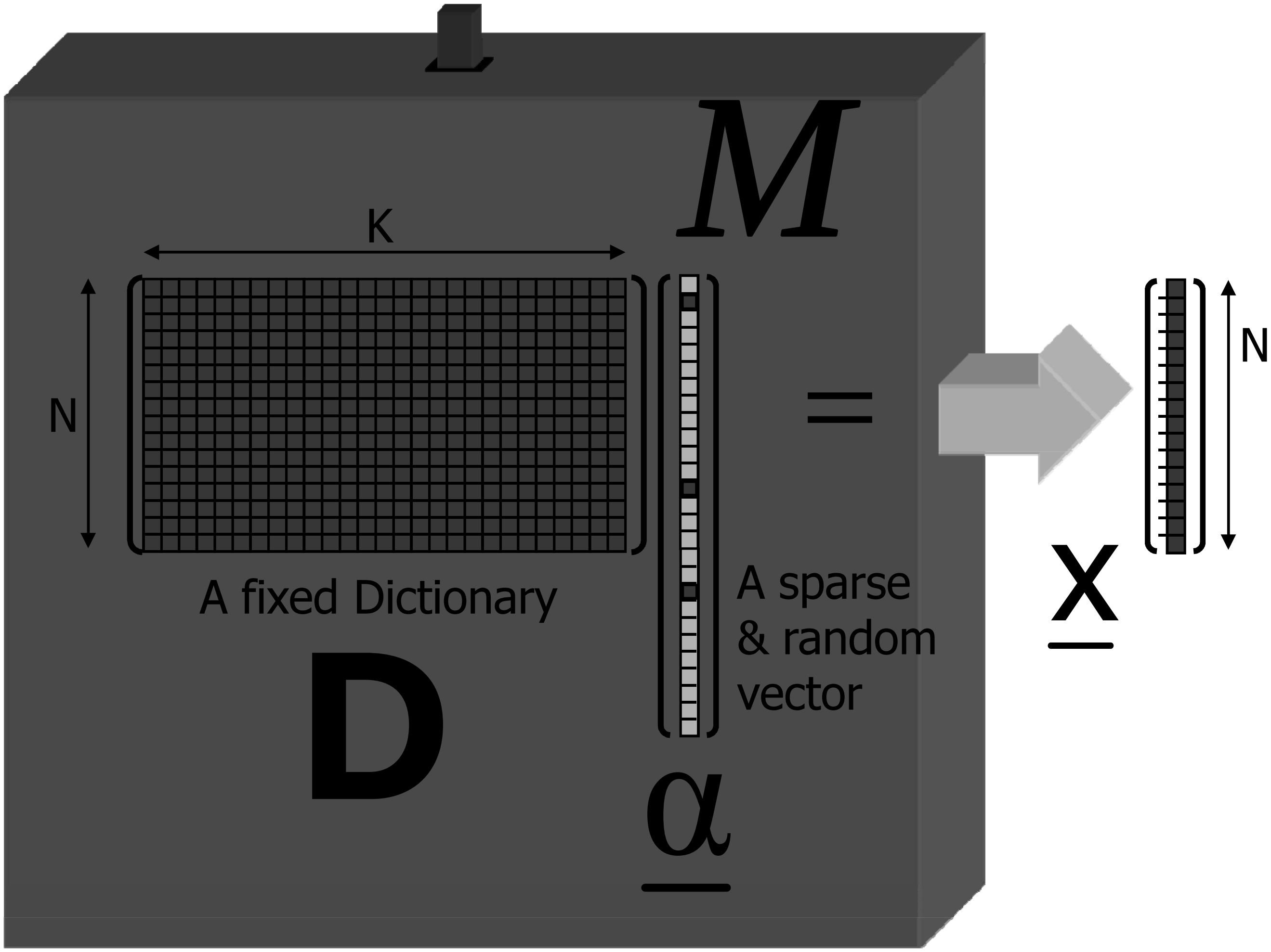
# A *Sparse* Prior Pr($\underline{x}$)

$$\Pr(\underline{x}) = \lambda \|\underline{\alpha}\|_0^0$$

$$\text{for} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$

**Sparse &**
**Redundant**

# The *Sparseland* Model for Images




- Every column in **D** (dictionary) is a prototype signal (Atom).
- The vector $\underline{\alpha}$ contains very few (say L) non-zeros.

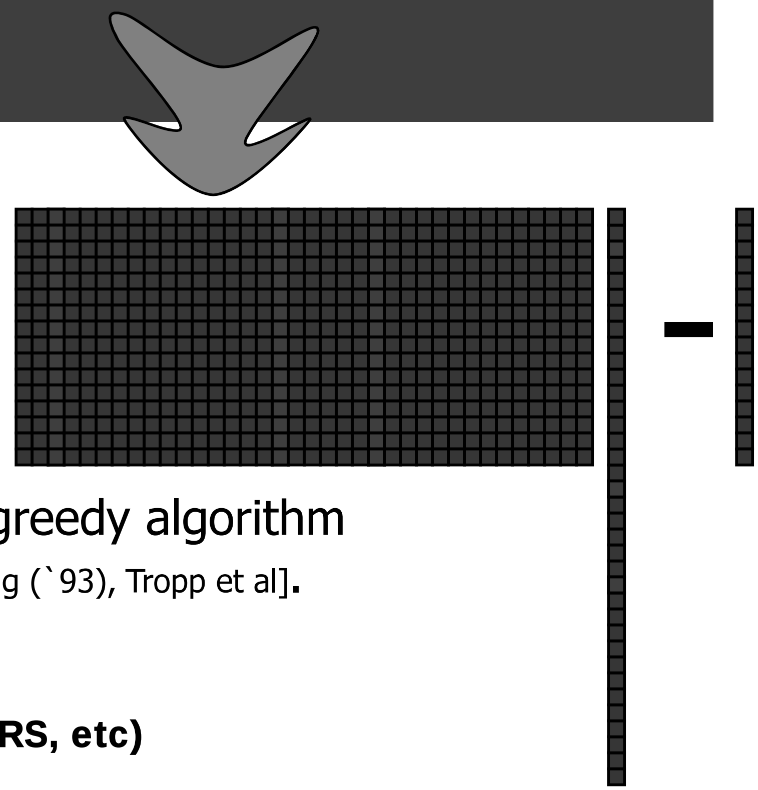
# The Initial Energy Function

- $L_0$ "pseudo-norm" is counting the number of non-zeros in $\underline{\alpha}$.

[redacted]

- The vector $\underline{\alpha}$ is the representation (**sparse**/**redundant**).

$$\mathbf{D}\underline{\alpha}-\underline{y}=$$



- The above is solved (approximated!) using a greedy algorithm - The Matching Pursuit [Classical Statistics, Mallat & Zhang (`93), Tropp et al].

- **L1 optimization can be used as well (Lasso, LARS, etc)**

# What Should D Be?

$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\arg\min} \frac{1}{2} \| \mathbf{D}\underline{\alpha} - \underline{y} \|_2^2 \ \text{ s.t. } \|\underline{\alpha}\|_0^0 \leq L \ \Rightarrow \ \underline{\hat{x}} = \mathbf{D}\underline{\hat{\alpha}}$$

Assumption: Good-behaved Images have a sparse representation

**D** should be chosen such that it sparsifies the representations **(for a given task!)**

One approach to choose **D** is from a known set of transforms (Steerable wavelet, Curvelet, Contourlets, Bandlets, …)

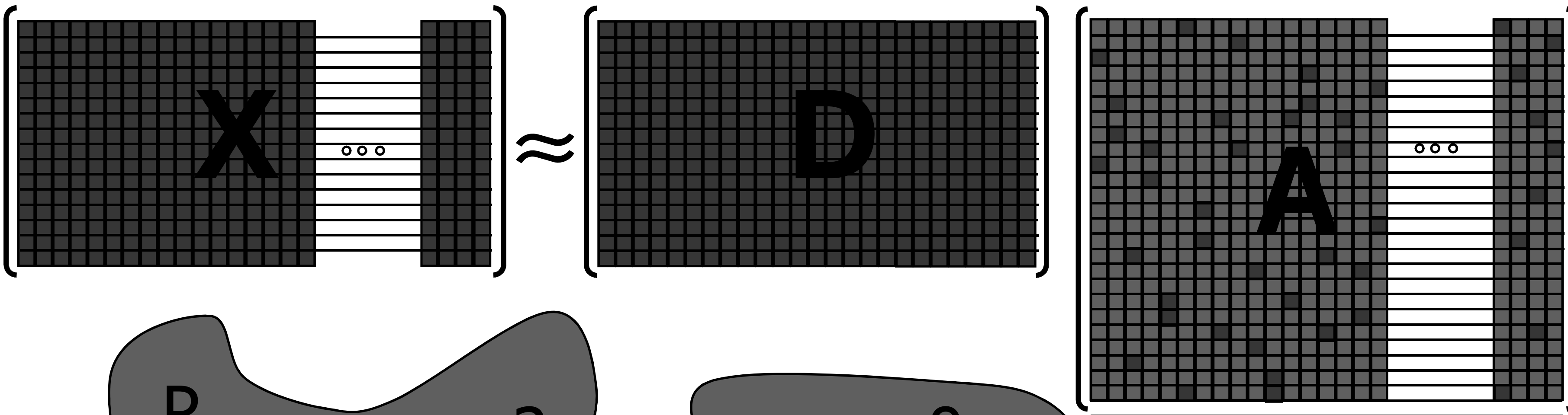
Learn **D** :

**Multiscale Learning**

**Color Image Examples**

**Task adapted**

# Learning D




$$\min_{\mathbf{D},\mathbf{A}} \sum_{j=1}^{P} \left\| \mathbf{D}\underline{\alpha}_j - \underline{x}_j \right\|_2^2 \quad \text{s.t.} \quad \forall j, \ \left\| \underline{\alpha}_j \right\|_0^0 \le L$$

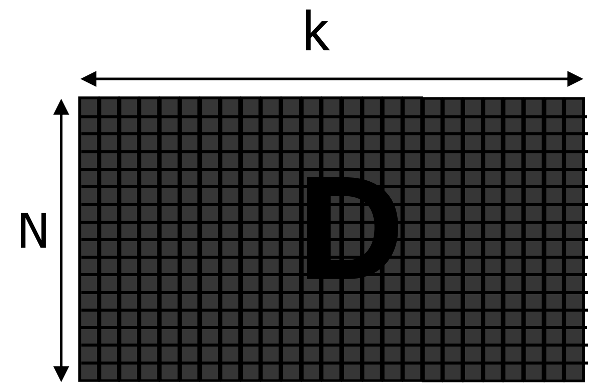
Each example is a linear combination of atoms from **D**

Each example has a sparse representation with no more than L atoms

Field & Olshausen ('96)
Engan et. al. ('99)
Lewicki & Sejnowski ('00)
Cotter et. al. ('03)
Gribonval et. al. ('04)
Aharon, Elad, & Bruckstein ('04)
Aharon, Elad, & Bruckstein ('05)
Ng et al. (2007)

# From Local to Global Treatment

- Algorithm are reasonable for low-dimension signals (N in the range 10-400). As N grows, the complexity and the memory requirements become prohibitive.
- So, how should large images be handled?




- The solution: Force shift-invariant sparsity - on each patch of size N-by-N (N=8) in the image, including overlaps [Buades et al., Seroussi et al., Roth & Black].

$$\underline{\hat{x}} = \underset{\underline{x},\{\underline{\alpha}_{ij}\}_{ij},D}{ArgMin} \ \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \mu\sum_{ij}\left\|\mathbf{R}_{ij}\underline{x}-\mathbf{D}\underline{\alpha}_{ij}\right\|_2^2$$

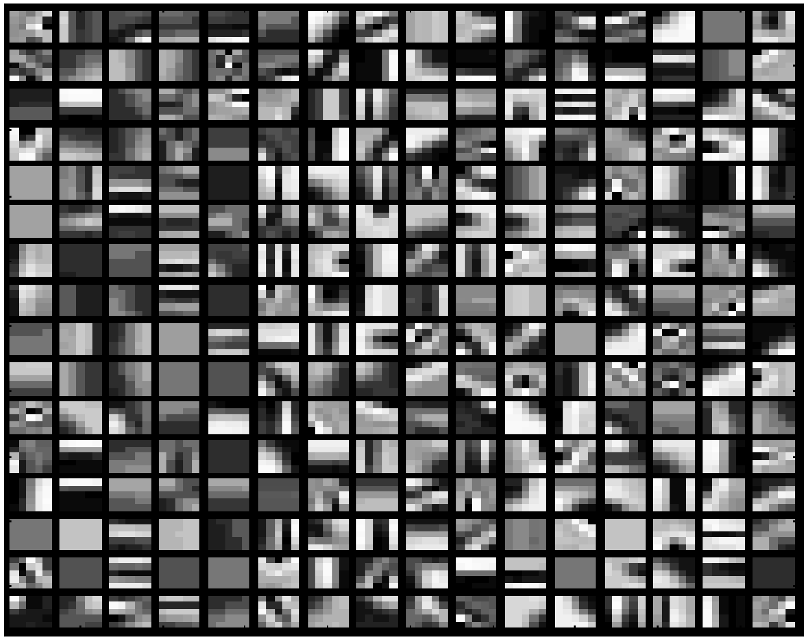
$$s.t. \quad \left\|\underline{\alpha}_{ij}\right\|_0^0 \le L$$

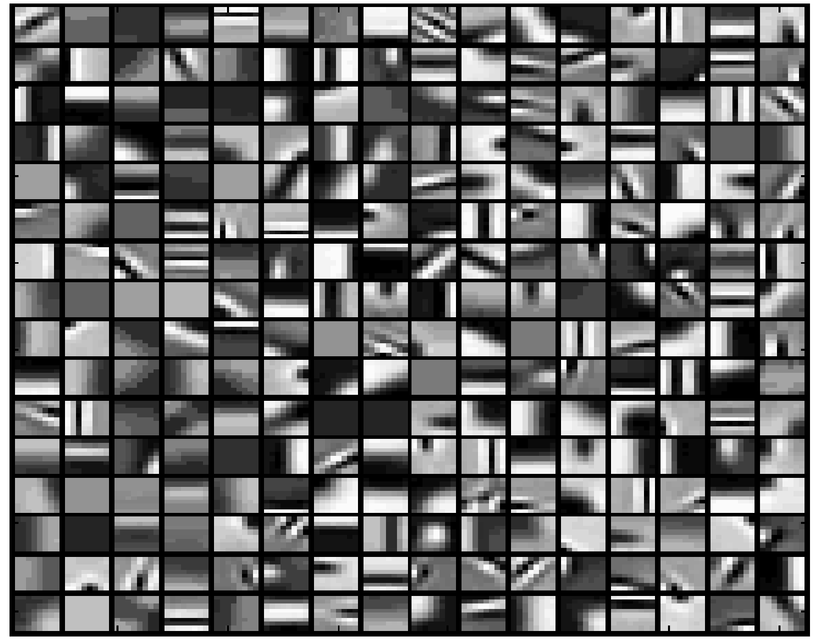
Extracts a patch in the ij location

The prior

# Show me the pictures

# Change the Metric in the OMP

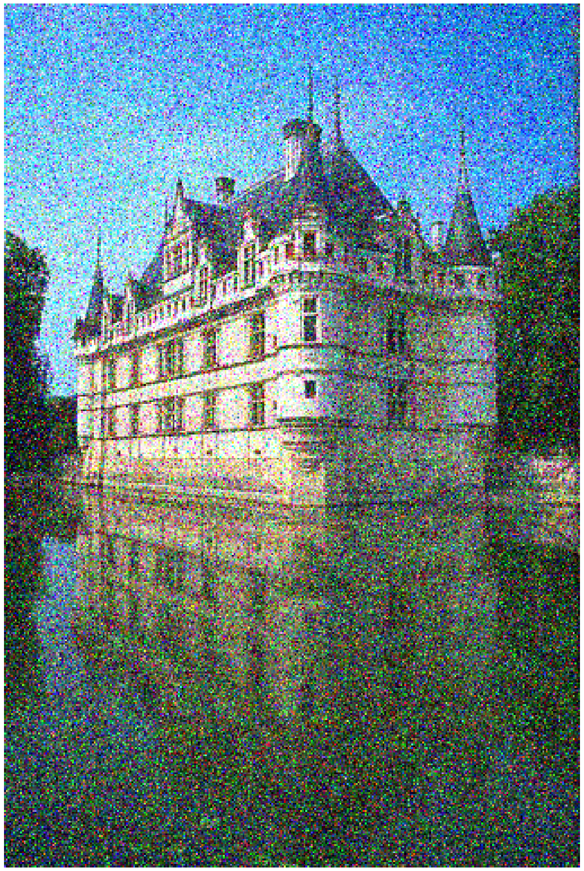




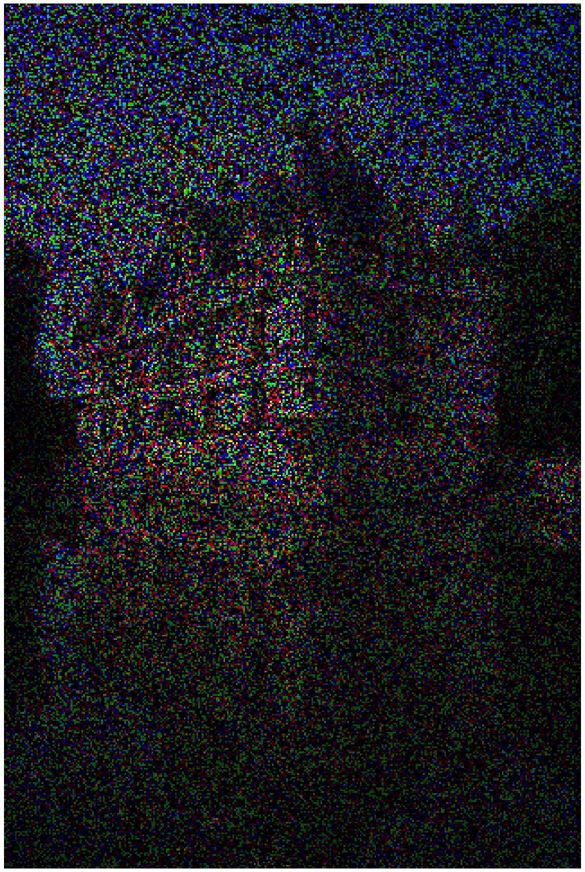
$$< \mathbf{y}, \mathbf{x} >_\gamma = \mathbf{y}^T\mathbf{x} + \frac{\gamma}{n^2}\mathbf{y}^T\mathbf{K}^T\mathbf{K}\mathbf{x} = \mathbf{y}^T(\mathbf{I} + \frac{\gamma}{n}\mathbf{K})\mathbf{x},$$

$$\mathbf{K} = \begin{pmatrix} \mathbf{J}_n & 0 & 0 \\ 0 & \mathbf{J}_n & 0 \\ 0 & 0 & \mathbf{J}_n \end{pmatrix}.$$

# Example: Non-uniform noise



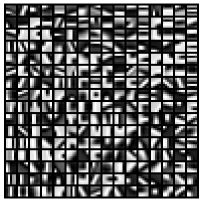
# Example: Inpainting

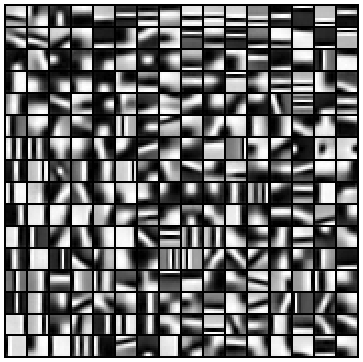


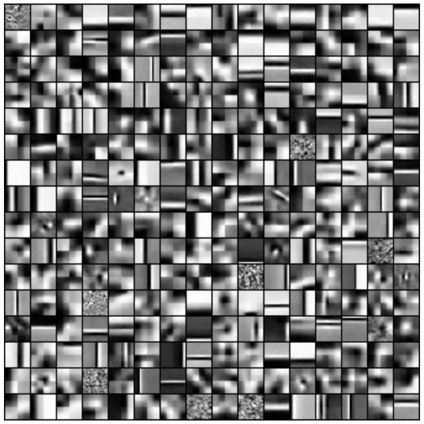
# Example: Inpainting


# Learning to Classify

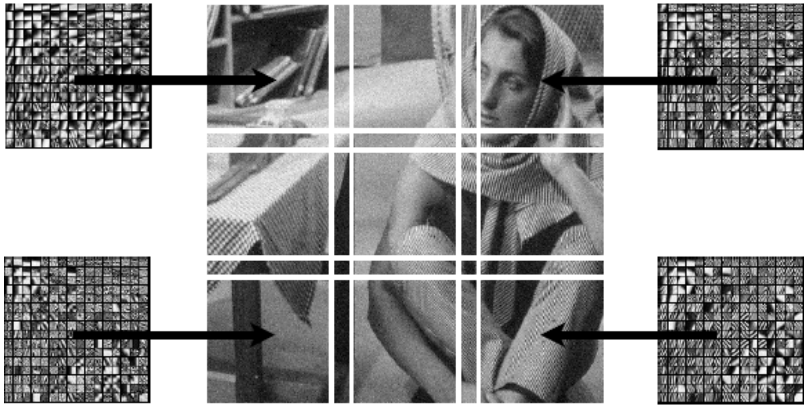
# Global Dictionary

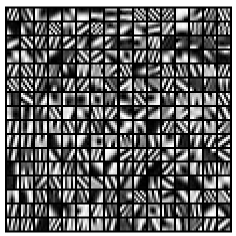


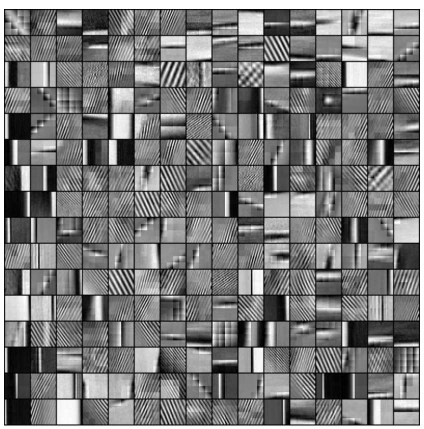


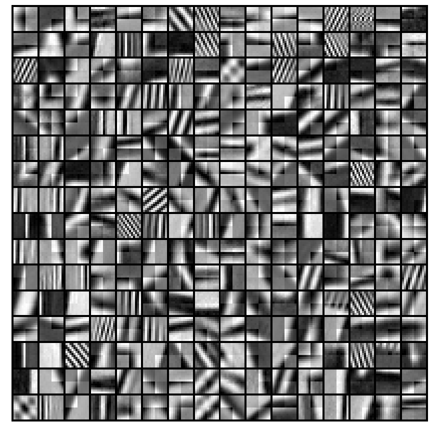


# Barbara

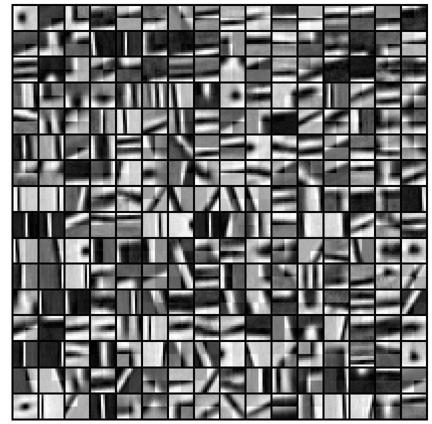


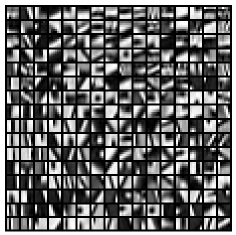


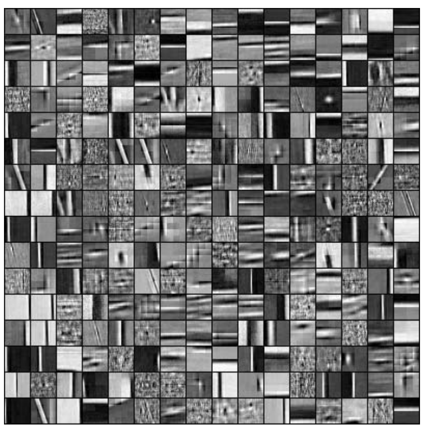




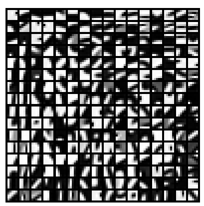
# Boat

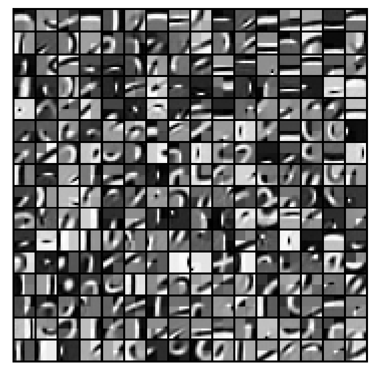






# Digits





# Which dictionary? How to learn them?

- Multiple reconstructive dictionary? (Payre)
- Single reconstructive dictionary? (Ng et al, LeCunn et al.)
- **Dictionaries for classification!**
- **See also Winn et al., Holub et al., Lasserre et al., Hinton et al. for joint discriminative/generative probabilistic approaches**

# Learning *multiple* reconstructive and *discriminative* dictionaries

- Learn dictionaries with a task in mind
- Move beyond ad-hoc features for recognition

- Learn one dictionary per-class
  - Good for the appropriate class
  - Bad for the other classes

***With J. Mairal, F. Bach, J. Ponce, and A. Zisserman, CVPR 2008***

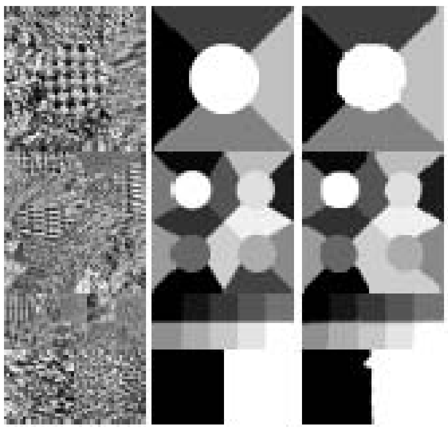
# Learning *multiple* reconstructive and *discriminative* dictionaries

$$
\begin{aligned}
\alpha^\star(\mathrm{x}, \mathrm{D}) &\equiv \arg\min_{\alpha \in \mathbb{R}^k} \|\mathrm{x} - \mathrm{D}\alpha\|_2^2, \ \text{s.t.} \ \|\alpha\|_0 \leq L, \\
\mathcal{R}(\mathrm{x}, \mathrm{D}, \alpha) &\equiv \|\mathrm{x} - \mathrm{D}\alpha\|_2^2, \\
\mathcal{R}^\star(\mathrm{x}, \mathrm{D}) &\equiv \|\mathrm{x} - \mathrm{D}\alpha^\star(\mathrm{x}, \mathrm{D})\|_2^2.
\end{aligned}
$$

$$
\mathcal{C}_i^\lambda(y_1, y_2, ..., y_N) := \log\Big(\sum_{j=1}^{N} e^{-\lambda(y_j - y_i)}\Big)
$$

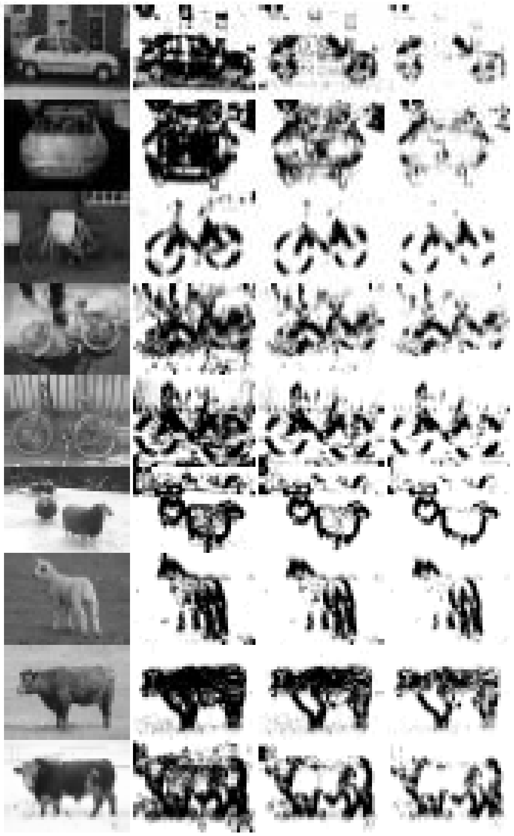
$$
\min_{\{D_j\}_{j=1}^N} \sum_{i=1...N, l \in S_i} \mathcal{C}_i^\lambda(\{\mathcal{R}^\star(x_l, D_j)\}_{j=1}^N) + \lambda\gamma\mathcal{R}^\star(x_l, D_i)
$$

# Texture classification



| ♯ | Prior 1 | Prior 2 | Prior 3 | Prior 4 | R1 | R2 | D1 | D2 |
|---|---|---|---|---|---|---|---|---|
| 1 | 7.2 | 6.7 | 5.5 | 3.37 | 2.22 | 1.69 | 1.89 | **1.61** |
| 2 | 18.9 | 14.3 | **7.3** | 16.05 | 24.66 | 36.5 | 16.38 | 16.42 |
| 3 | 20.6 | 10.2 | 13.2 | 13.03 | 10.20 | 5.49 | 9.11 | **4.15** |
| 4 | 16.8 | 9.1 | 5.6 | 6.62 | 6.66 | 4.60 | 3.79 | **3.67** |
| 5 | 17.2 | 8.0 | 10.5 | 8.15 | 5.26 | **4.32** | 5.10 | 4.58 |
| 6 | 34.7 | 15.3 | 17.1 | 18.66 | 16.88 | 15.50 | 12.91 | **9.04** |
| 7 | 41.7 | 20.7 | 17.2 | 21.67 | 19.32 | 21.89 | 11.44 | **8.80** |
| 8 | 32.3 | 18.1 | 18.9 | 21.96 | 13.27 | 11.80 | 14.77 | **2.24** |
| 9 | 27.8 | 21.4 | 21.4 | 9.61 | 18.85 | 21.88 | 10.12 | **2.04** |
| 10 | 0.7 | 0.4 | NA | 0.36 | 0.35 | **0.17** | 0.20 | **0.17** |
| 11 | **0.2** | 0.8 | NA | 1.33 | 0.58 | 0.73 | 0.41 | 0.60 |
| 12 | 2.5 | 5.3 | NA | 1.14 | 1.36 | **0.37** | 1.97 | 0.78 |
| **Av.** | 18.4 | 10.9 | NA | 10.16 | 9.97 | 10.41 | 7.34 | **4.50** |

# Natural images classification



# Some dictionaries

Reconstructive

Discriminative

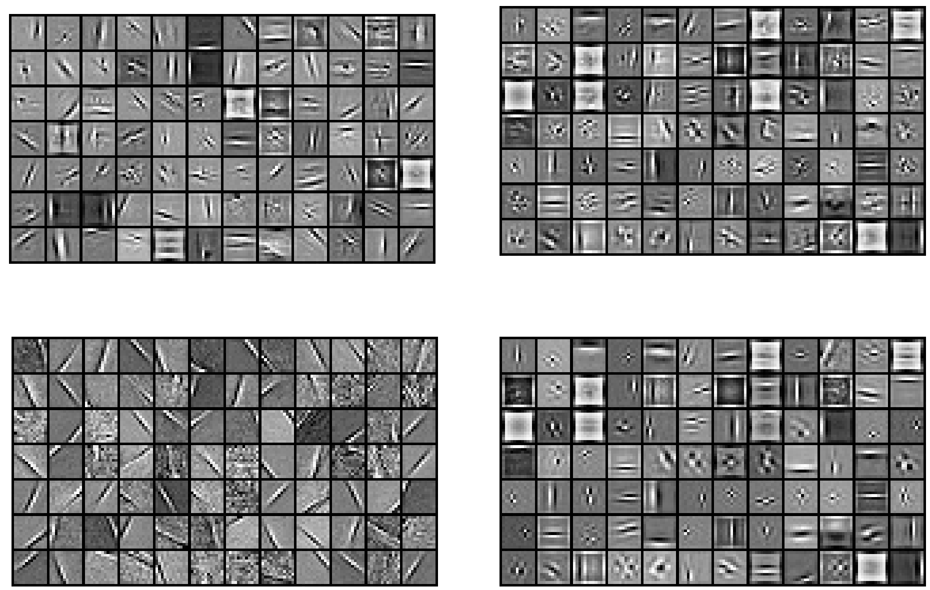
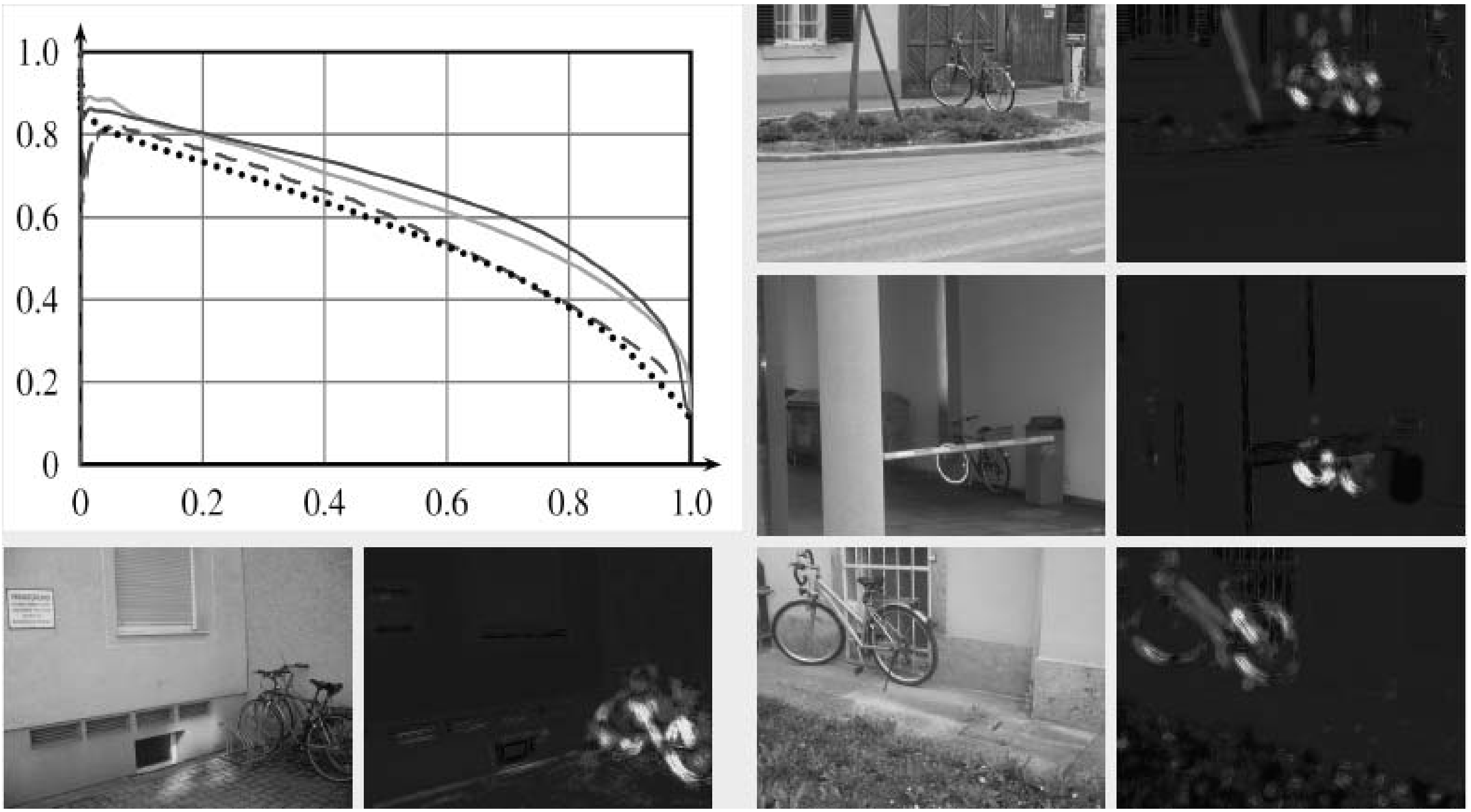


Figure Background

# Semi-supervised detection learning



# Learning a *Single* Discriminative and Reconstructive Dictionary

- Learn dictionaries with a task in mind
- Move beyond ad-hoc features for recognition

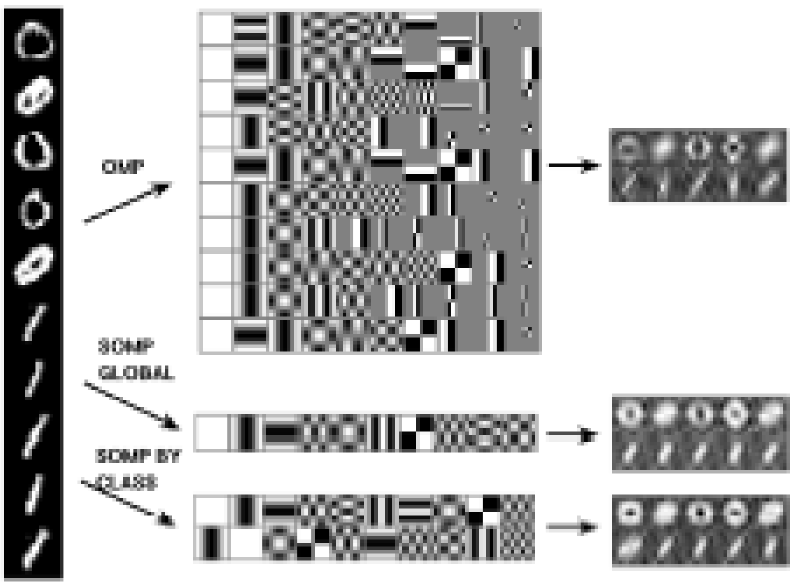
- Exploit the representation coefficients for classification
  - Include this in the optimization
  - *Class supervised simultaneous OMP*

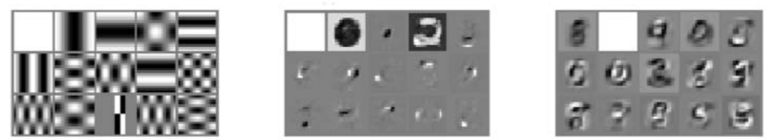
***With F. Rodriguez, IMA Pre-print***

# Learning a *Single* Discriminative and Reconstructive Dictionary

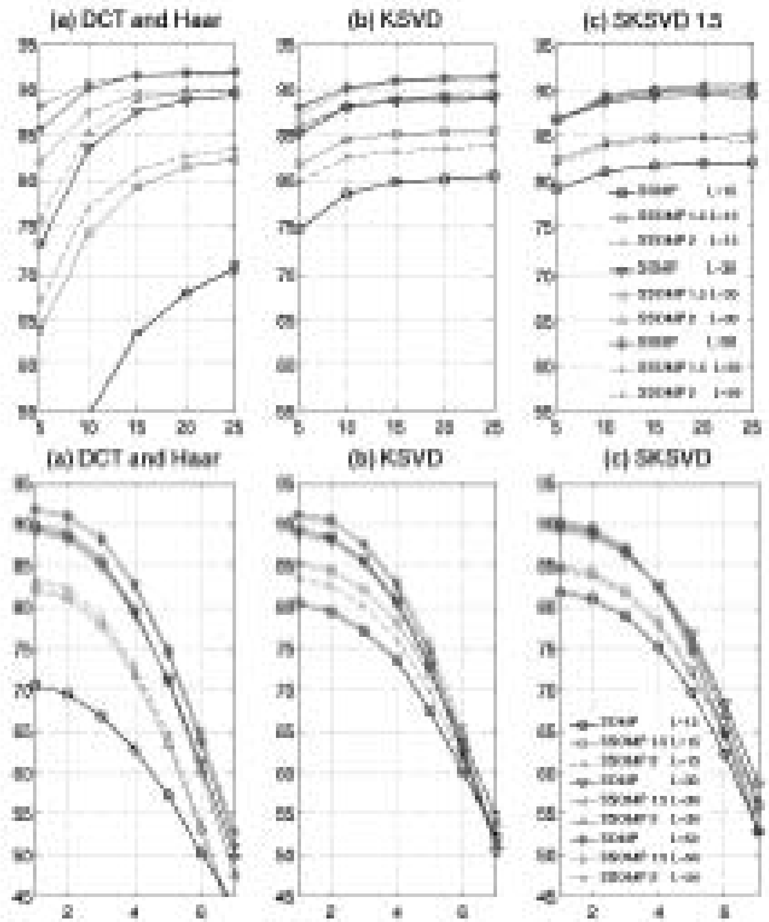
$$\max_{\mathbf{D},\alpha} \left\{ \theta \cdot J(\{\{\alpha_i^j\}_{i=1}^{n_j}\}_{j=1}^{c}) - \sum_{j=1}^{c} \sum_{i=1}^{n_j} \|\mathbf{x_i^j} - \mathbf{D}\alpha_\mathbf{i}^\mathbf{j}\|_2^2 \right\}$$

# Digits images:
# Some dictionaries



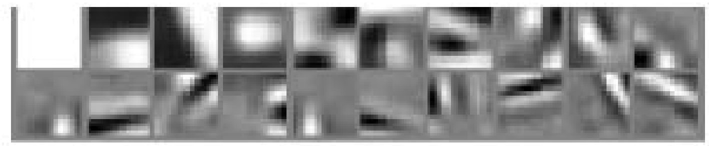



# Digits images: Robust to noise and occlusions



# Natural mages (preliminary)



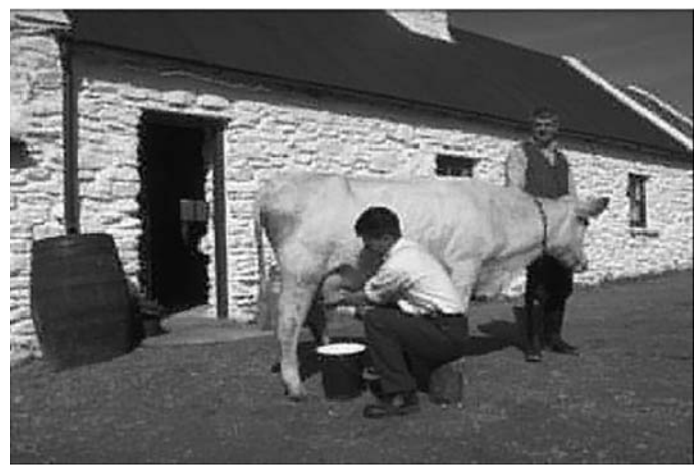



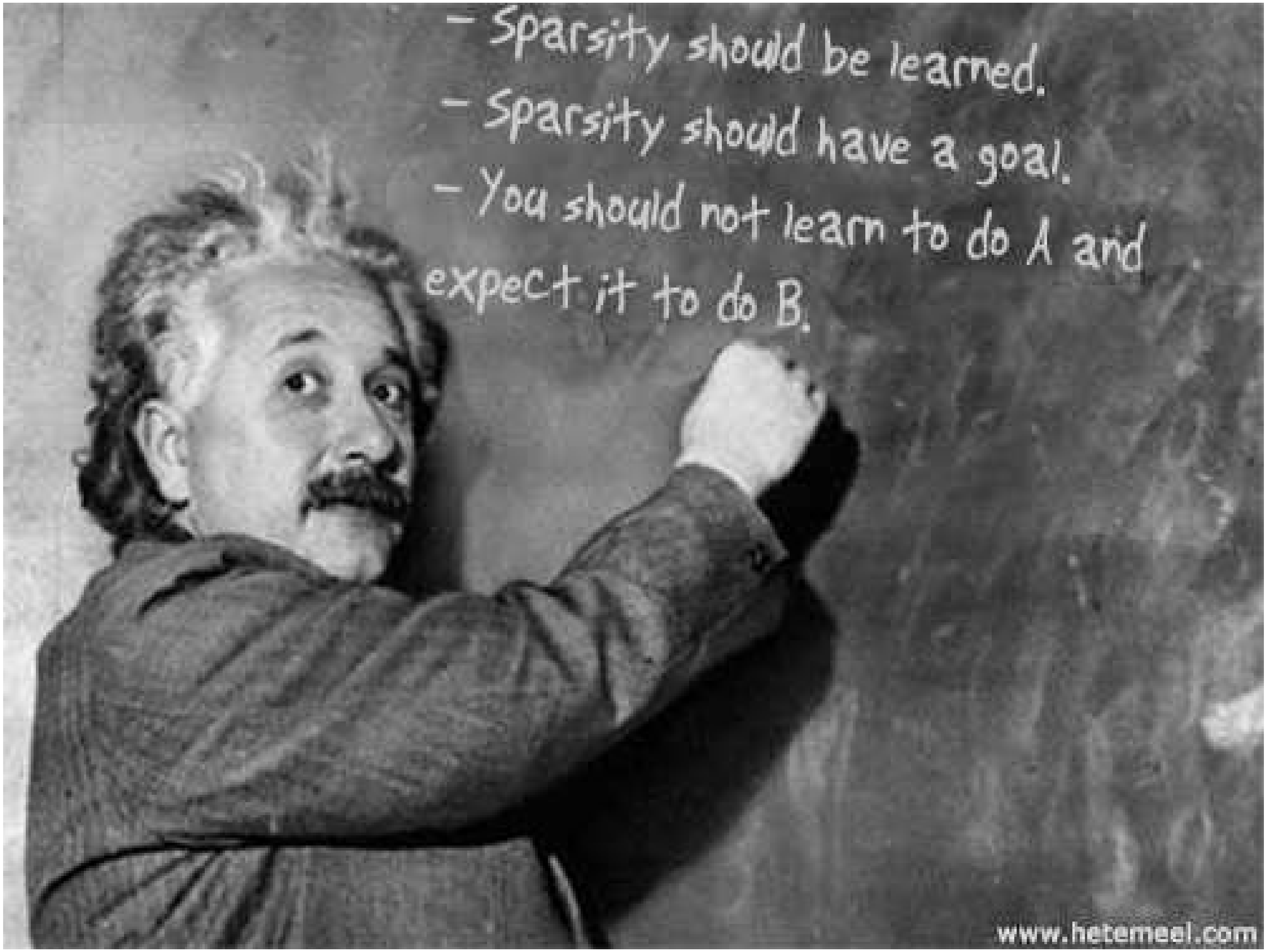
94% recognition for 3 classes

# Conclusions

- Learn for the Task :Classification

- Sensing…




- Sparsity should be learned.
- Sparsity should have a goal.
- You should not learn to do A and expect it to do B.
www.hetemeel.com