# Extended Dictionary Learning : Convolutional and Multiple Feature Spaces

**Konstantina Fotiadou**, Greg Tsagkatakis & Panagiotis Tsakalides

kfot@ics.forth.gr, greg@ics.forth.gr, tsakalid@ics.forth.gr

**ICS-FORTH**

**website**: http://spl.edu.gr/

**twitter**: https://twitter.com/spl_icsforth

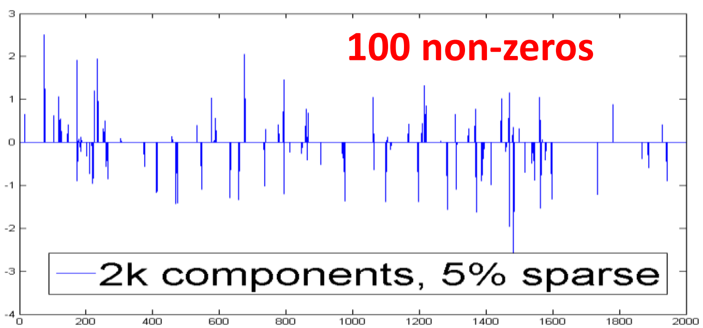
# Agenda

- **Sparse Coding for Image Processing Applications**
- **Coupled Dictionary Training**
- **Learning Deep Features**

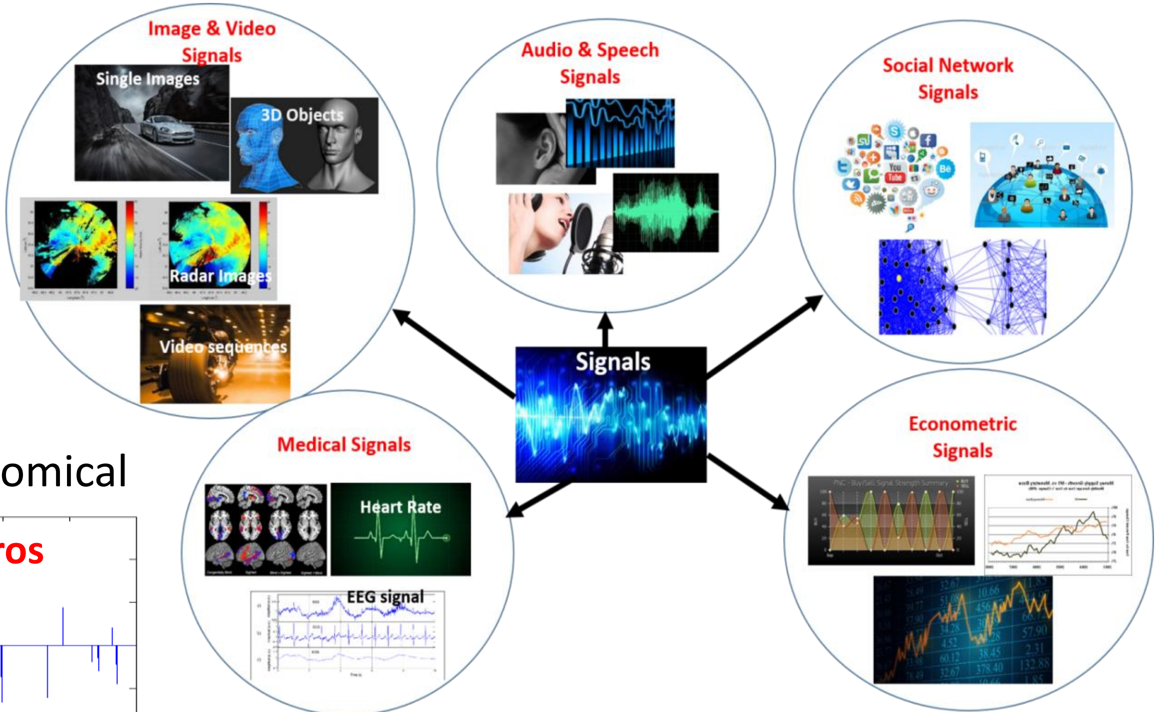
# Sparse Representations (SR) Framework

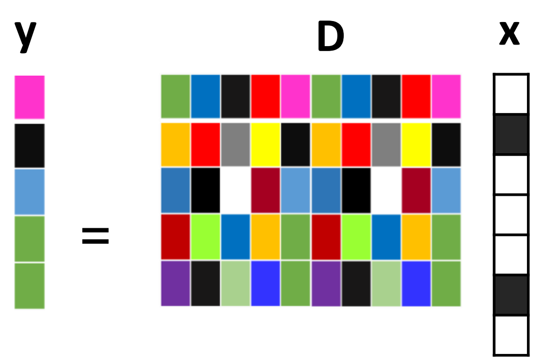
- ❑ **Why SR?**
  - ✓ Natural signal description
  - ✓ High dimensional signals
  - ✓ Efficient prior

- ❑ **Signal sparsity**
  - ➢ **Naturally Sparse**
    - Radar, Seismic, Astronomical




  - ➢ **Sparse in a Dictionary**
    - Images
    - Videos



# Sparse Signal Modeling

- **Under-determined system of linear equations**

$$\boxed{y = Dx}$$

$y \in \mathbb{R}^n$ **Measurement** vector

$D \in \mathbb{R}^{n \times m}$ **Dictionary** matrix

$x \in \mathbb{R}^m$ **sparse** vector

- **Prior knowledge** → $\ell_0$**-norm**

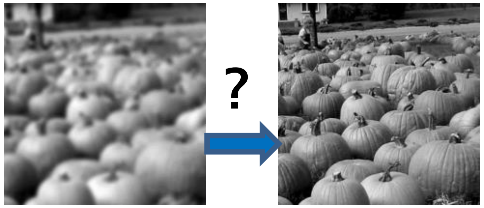
**NP-Hard!!!**

$$\hat{x} = \underset{x}{\operatorname{argmin}} ||x||_0^0 \,, s.t \;\; Dx = y$$

**Relaxation approaches:**
**Smooth the $\ell_0$-norm by the convex $\ell_1$-norm**

**Greedy approaches:**
**Sequence of incremental approximations.**

- **Motivation:** Inverse problems



# Approximation: Greedy vs. Relaxation Techniques

## Greedy-Matching Pursuit

- **MP**: greedy algorithm
- **Step 1**: find **best matching atom**.
- **Next steps**: Find the next **best fitting**, given the previously found atoms
- **Orthogonal MP (OMP):** improved version with re-evaluation

## Relaxation-Basis Pursuit

- Instead of $\min_x ||x||_0 \,, s.t.\; Dx = y$

**Noisy Version**

$$\min_x ||x||_1 \,, s.t.\; Dx = y \Longrightarrow \min_x ||x||_1, s.t.\; ||Dx - y||_2 \leq \varepsilon$$

- **Convex & Efficient!**

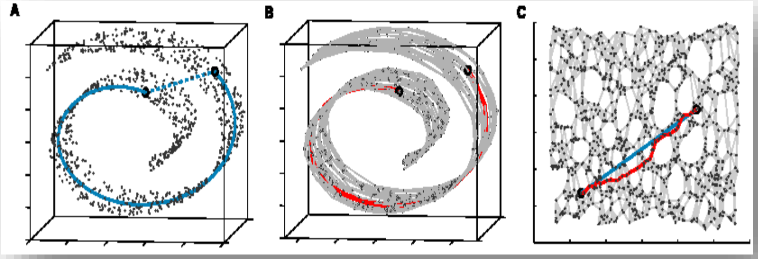
$$\min_x \lambda ||x||_1 + \; ||Dx - y||_2$$

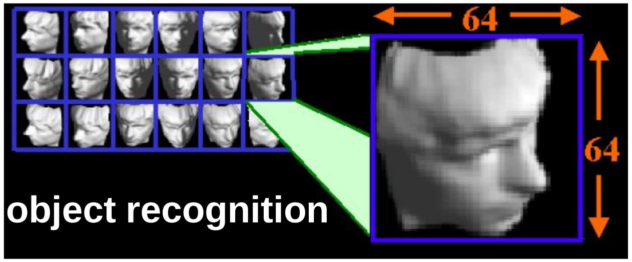
**The LASSO problem**

# Non-linear Sparsity Models

- **Generic Formulation** → $\min \mathcal{F}(\mathbf{y},\mathbf{s}) \quad \text{s.t.} \quad \mathbf{s} \in \mathcal{S}$

- **Motivation**
  - **Non-linear feature spaces!**
  - **Non-linear Dimensionality Reduction**




- **State-of-the-art**
  - **Quantization**[1] → $\mathcal{F}(\mathbf{y},\mathbf{s}) = \mathcal{Q}(\mathbf{y},\mathbf{s})$ → $\min \|\mathbf{y} - \mathcal{Q}(\Phi \mathbf{D}\mathbf{x})\|_2 \quad \text{s.t.} \quad \|\mathbf{x}\|_0 \leq K$
  - **Multipath SC**
  - **Convolutional SC**

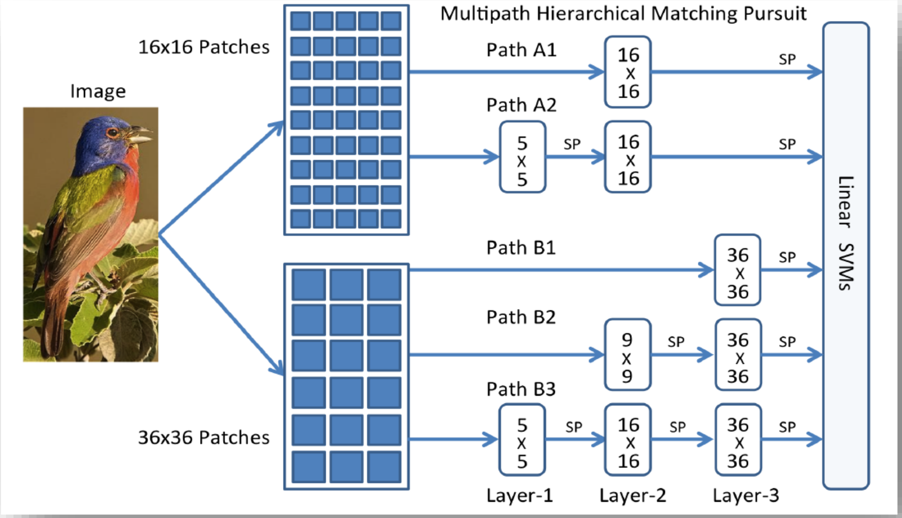
[1] Recovery of quantized compressed sensing measurements, G Tsagkatakis, P Tsalakides, IS&T/SPIE EI 2015

# Multipath sparse coding

- **Task**: *learn & combine recursive sparse coding through many pathways on multiple bags of patches of varying size to learn characteristic features*
- **Application: Image Classification**
- **Algorithm**:
  - Multipath Hierarchical Matching Pursuit
  - Mutual Incoherence Dictionary Learning

$$\min_{D,X} \|\mathbf{Y} - \mathbf{DX}\|_F + \lambda \sum_{i=1}^{M} \sum_{j=1, i\neq j}^{M} |\mathbf{d}_i^T \mathbf{d}_j|$$

$$\text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K \qquad \|\mathbf{d}_i\|_2 = 1 \quad \forall i$$

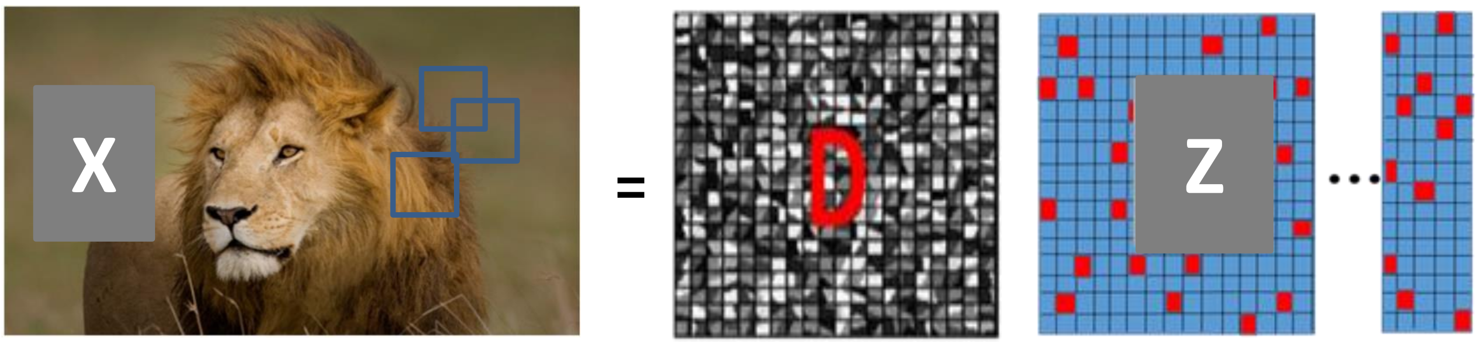



Bo, Liefeng, Xiaofeng Ren, and Dieter Fox. "Multipath sparse coding using hierarchical matching pursuit." *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013.

# Patch Based vs Convolutional Sparse Modeling

- **Patch-based Sparse Modeling**

$$\arg\min_{\mathbf{d},\mathbf{z}} \quad \frac{1}{2}\sum_{n=1}^{N} ||\mathbf{x}_n - \mathbf{D}\mathbf{z}_n||_2^2 + \beta||\mathbf{z}_n||_1 \quad \text{subject to} \quad ||\mathbf{d}_k||_2^2 \leq 1 \text{ for } k = 1\dots K$$




- **Convolutional Sparse Modeling**

$$\arg\min_{\mathbf{d},\mathbf{z}} \quad \frac{1}{2}||\mathbf{x} - \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k||_2^2 + \beta\sum_{k=1}^{K}||\mathbf{z}_k||_1$$
$$\text{subject to} \quad ||\mathbf{d}_k||_2^2 \leq 1 \text{ for } k = 1\dots K.$$

**Convolution operator**

- ✓ Models local structures that appear anywhere in the image
- ✓ Translation Invariance
  - Orientations
  - Frequencies

# Convolutional Sparse Coding

- **Optimization**
  - ADMM,
  - Proximal Gradient,
  - Block-Toeplitz
- **Applications**
  - Compression
  - Super-resolution
  - Inpainting
  - HDR synthesis
  - Deconvolution

$$\begin{aligned} \underset{\mathbf{d},\mathbf{z}}{\arg\min} \quad & \frac{1}{2}\|\mathbf{x} - \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2^2 \leq 1 \text{ for } k = 1 \ldots K. \end{aligned}$$

Hilton, and Lucey. "Optimization methods for convolutional sparse coding." *arXiv preprint 2014*.
Felix, Heidrich, and Wetzstein. "Fast and Flexible Convolutional Sparse Coding." *IEEE CVPR 2015*

# Convolutional Sparse Coding: ADMM Optimization

- **Initial Optimization Problem**

$$\begin{aligned}\arg\min_{\mathbf{d},\mathbf{z}} \quad & \frac{1}{2}\|\mathbf{x}-\sum_{k=1}^{K}\mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta\sum_{k=1}^{K}\|\mathbf{z}_k\|_1 \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2^2 \leq 1 \text{ for } k = 1\ldots K.\end{aligned}$$

- **General Formulation**

$$\begin{aligned}\underset{\mathbf{d},\mathbf{z}}{\operatorname{argmin}} \ & \frac{1}{2}\|\mathbf{x}-\mathbf{M}\sum_{k=1}^{K}\mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta\sum_{k=1}^{K}\|\mathbf{z}_k\|_1 \\ \text{subject to} \ & \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1,\ldots,K\}\end{aligned}$$

**M**: diagonal matrix that masks out the boundaries of the padded estimation

- **Include the constraints in the objective function**

$$(\mathbf{M}^T\mathbf{M}+\mathbb{I})\mathbf{x} = \mathbf{b}$$

$$\underset{\mathbf{d},\mathbf{z}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x}-\mathbf{M}\sum_{k=1}^{K}\mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta\sum_{k=1}^{K}\|\mathbf{z}_k\|_1 + \sum_{k=1}^{K}\operatorname{ind}_C(\mathbf{d}_k),$$

- **Auxiliary form**

$$\underset{\mathbf{d},\mathbf{z}}{\operatorname{argmin}} \ f_1(\mathbf{D}\mathbf{z}) + \sum_{k=1}^{K}\left(f_2(\mathbf{z}_k) + f_3(\mathbf{d}_k)\right)$$

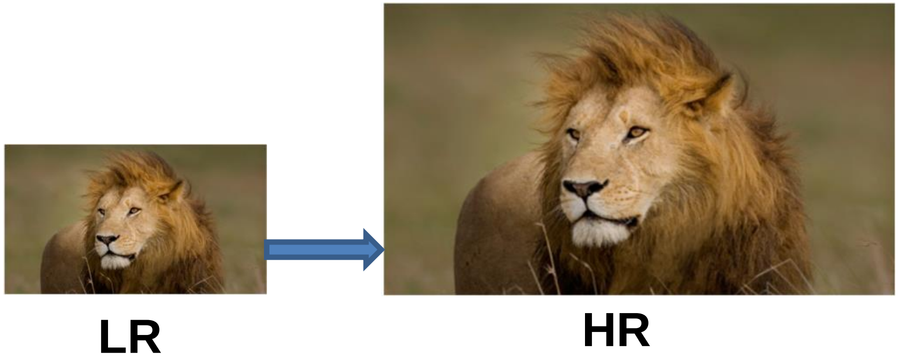
where:

$$f_1(\mathbf{v}) = \frac{1}{2}\|\mathbf{x}-\mathbf{M}\mathbf{v}\|_2^2, \ f_2(\mathbf{v}) = \beta\|\mathbf{v}\|_1, \ f_3(\mathbf{v}) = \operatorname{ind}_C(\mathbf{v})$$

- **ADMM formulation →**

$$\underset{\mathbf{z}}{\operatorname{argmin}} \ h(\mathbf{y}) + g(\mathbf{z}) \quad \text{subject to} \quad \mathbf{A}\mathbf{y} = \mathbf{z}$$

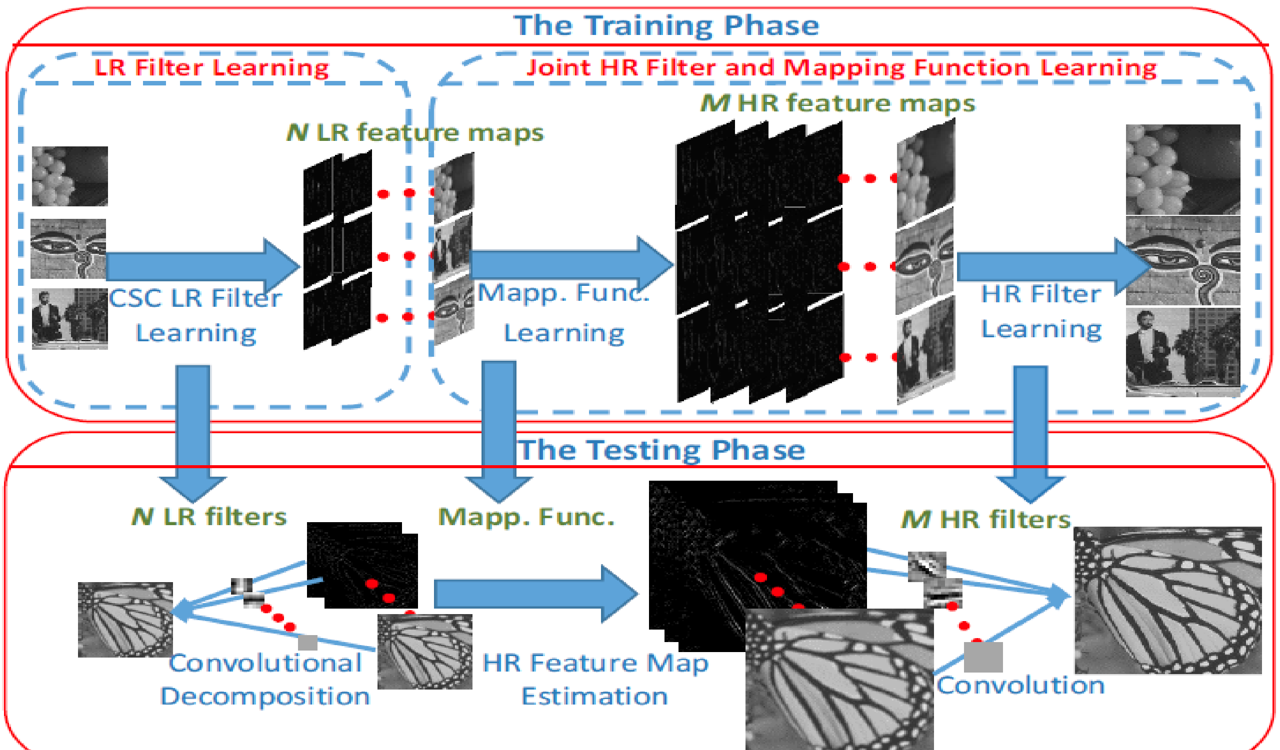
# Application: Convolutional Sparse Coding for Image Super-Resolution




$$\min_{Z} \| X - \sum_{i=1}^{N} f_i \otimes Z_i \|_F^2 + \lambda \sum_{i=1}^{N} \| Z_i \|_1$$

**Input image** **filters** **Feature Maps**

- **Optimization:** ADMM




Gu, Shuhang, et al. "Convolutional Sparse Coding for Image Super-resolution." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

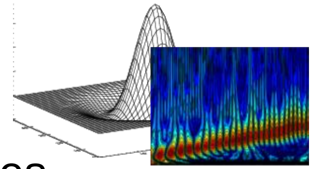
# Agenda

- Sparse Coding for Image Processing Applications
- **Coupled Dictionary Training**
- Learning Deep Features

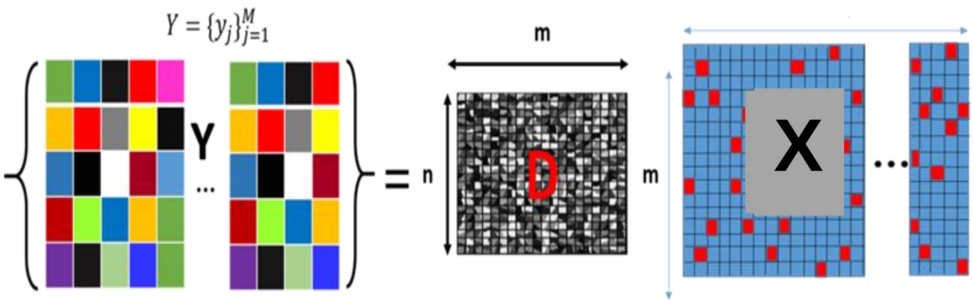
# How to select the proper dictionaries

- Goal: ***Find the proper D in order to sparsify the input data***

- **What kind of D?**
  - Parametric: Fourier signals, wavelets, curvelets, etc.
  - Data driven: Randomly selected image examples
  - **Trained**: **Learning** from randomly selected image examples



**Dictionary Training:** *Given a set of training signals Y, and a fixed size dictionary D, how can we find D?*




$$\min_{D,X} ||Y - DX||_F^2 \;\; s.t \; \forall i \;\; ||x_i||_0 \leq L, i \in (1, N)$$

**State-of-the-art: K-SVD**

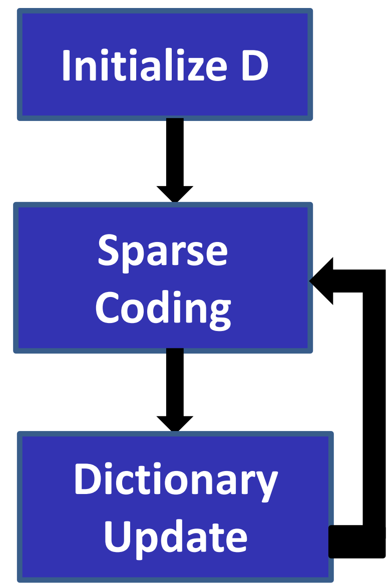
# Dictionary Training: The K-SVD algorithm[1]

- **Step 1 - Sparse Coding Stage:** Find the best representation coefficient matrix **X**.

$$\min_{D,X} ||Y - DX||_F^2 \;\; s.t \;\; \forall i \;\; ||x_i||_0 \leq L, i \in (1, N)$$

$$\Downarrow$$

$$\min_{x_i} ||y_i - Dx_i||_2^2 \;\; , s.t. \;\; ||x||_0 \leq L$$




- **Step 2- Dictionary Update Stage:** Update one column at a time.

  **For each column** k={1,…,M}

  - **Solve**:

$$\min_{d_k, x_k} ||E_k - d_k x_k^T||_F^2$$

**Representation error:**

$$E_k = Y - \sum_{j \neq k}^{M} d_j \, x_j^T$$

  - **Define the group of indices that use the atom $d_k$** :

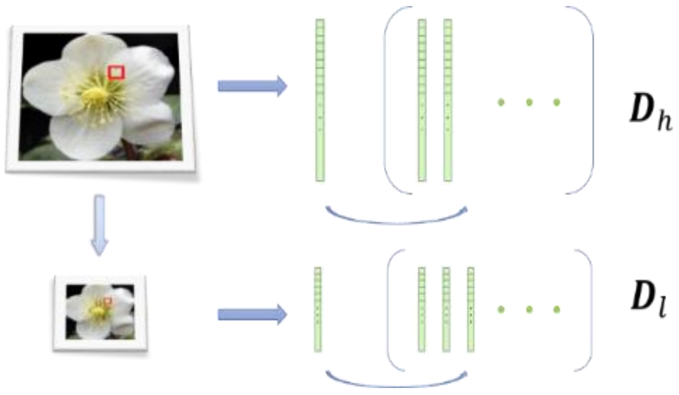
$$\omega_k = \{i | 1 \leq i \leq M, x_k^T(i) \neq 0\}$$

  - Restrict $E_k$ by choosing columns that correspond to $\omega_k$
  - **SVD** decomposition: $E_k = U\Delta V^T$. The first column of $U$ is the updated column $\widetilde{d}_k$. Update the first column of $V$ to be the updated coefficients $x_R^k$

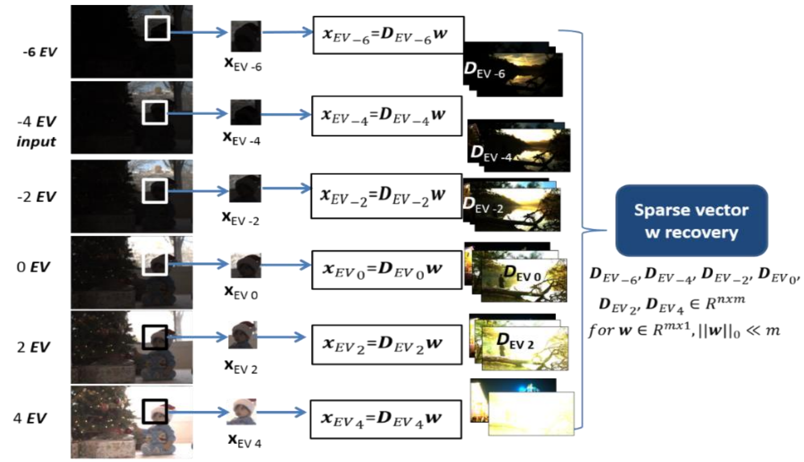
  **end**

[1] M. Aharon et al: *K-SVD : An algorithm for designing over-complete dictionaries for sparse representation. IEEE Transactions on Image Processing. 2006*

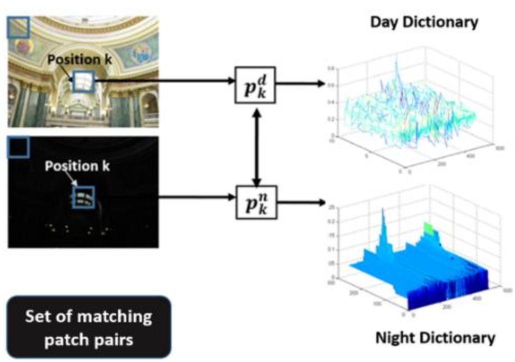
# Motivation for Joint Dictionaries



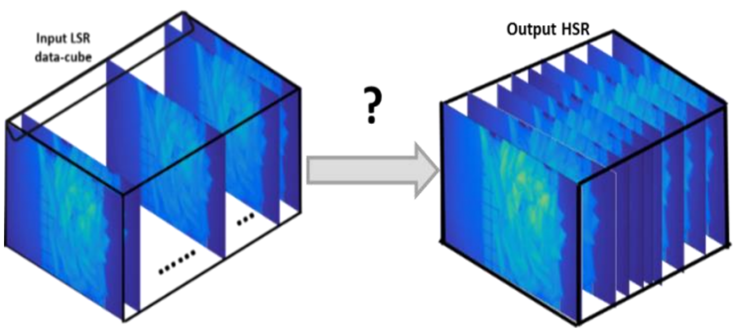

**Super-resolution**




**Snapshot HDR**




**De-nighting**




**Spectral Super-Resolution**

# Coupled Dictionary Learning

- **Goal**: ***Learn jointly two dictionary matrices***: $\mathbf{D}_h$, $\mathbf{D}_\ell$

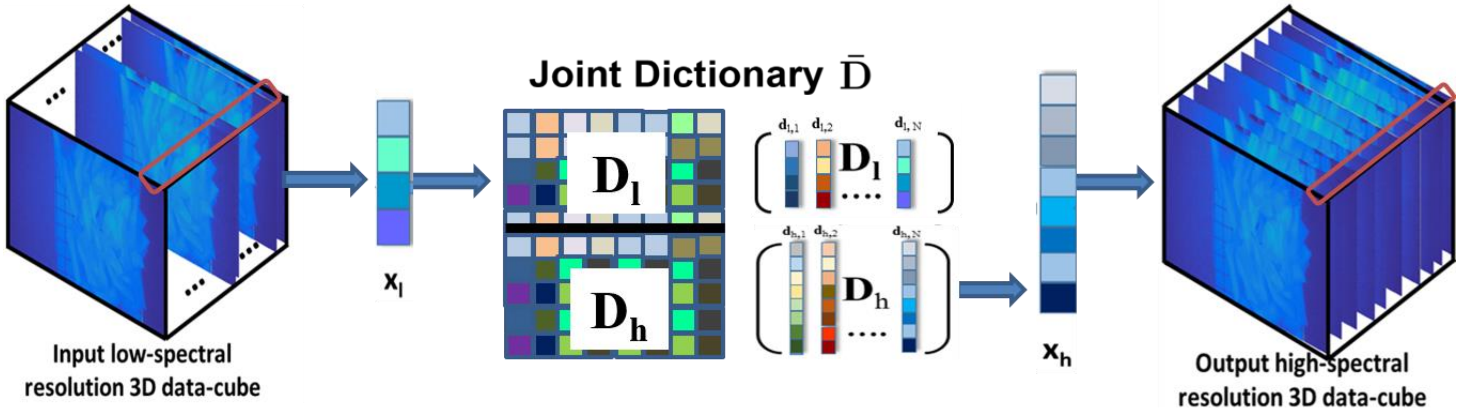
- **Concatenated feature space:**

$$\underset{\mathbf{D},\mathbf{W}}{\operatorname{argmin}} ||\bar{\mathbf{S}} - \bar{\mathbf{D}}\mathbf{W}||_F + \lambda||\mathbf{W}||_1,$$

$$\text{s. t. } ||\bar{\mathbf{D}}(:,j)||_2^2 \leq 1, \quad j = \{1, ..., K\},$$

**where**

$$\bar{\mathbf{S}} = \begin{bmatrix} \mathbf{S}_h \\ \mathbf{S}_l \end{bmatrix}, \bar{\mathbf{D}} = \begin{bmatrix} \mathbf{D}_h \\ \mathbf{D}_l \end{bmatrix}$$




**Limitation !!**

- **Sub-Optimal Coding Scheme → Individual Feature Spaces** $\mathbf{W}_h \neq \mathbf{W}_\ell$

# Alternating Direction Method of Multipliers (ADMM) for Coupled Dictionary Learning (CDL)

- ***Optimization problem:***

$$\underset{\mathbf{D}_h,\mathbf{D}_\ell,\mathbf{W}_h,\mathbf{W}_\ell}{\operatorname{argmin}} ||\mathbf{D}_h\mathbf{W}_h - \mathbf{S}_h||_F + ||\mathbf{D}_\ell\mathbf{W}_\ell - \mathbf{S}_\ell||_F + \lambda_h||\mathbf{W}_h||_1 + \lambda_\ell||\mathbf{W}_\ell||_1,$$

$$\text{s.t.} \quad \mathbf{W}_h = \mathbf{W}_\ell, \quad ||\mathbf{D}_h(:,i)||_2 \leq 1, \quad ||\mathbf{D}_\ell(:,i)||_2 \leq 1$$

- **Setting**: $\mathrm{P} = \mathbf{D}_h$ **and** $\mathrm{Q} = \mathbf{D}_\ell$

$$\min_{\mathbf{D}_h,\mathbf{W}_h,\mathbf{D}_l,\mathbf{W}_l} ||\mathbf{S}_h - \mathbf{D}_h\mathbf{W}_h||_F^2 + ||\mathbf{S}_l - \mathbf{D}_l\mathbf{W}_l||_F^2 + \lambda_l||\mathbf{Q}||_1 + \lambda_h||\mathbf{P}||_1$$

$$\text{s. t. } \mathbf{P} = \mathbf{W}_h, \mathbf{Q} = \mathbf{W}_l, \mathbf{W}_h = \mathbf{W}_l, ||\mathbf{D}_h(:,i)||_2 \leq 1, ||\mathbf{D}_l(:,i)||_2 \leq 1$$

- ***Augmented Lagrangian Function:***

$$L(\mathbf{D}_h,\mathbf{D}_l,\mathbf{W}_h,\mathbf{W}_l,\mathbf{P},\mathbf{Q},Y_1,Y_2,Y_3) = \frac{1}{2}||\mathbf{D}_h\mathbf{W}_h - \mathbf{S}_h||_F^2 + \frac{1}{2}||\mathbf{D}_\ell\mathbf{W}_\ell - \mathbf{S}_\ell||_F^2 +$$
$$\lambda_h||\mathbf{P}||_1 + \lambda_\ell||\mathbf{Q}||_1 + < Y_1, \mathbf{P} - \mathbf{W}_h > + < Y_2, \mathbf{Q} - \mathbf{W}_\ell > + < Y_3, \mathbf{W}_h - \mathbf{W}_\ell > +$$
$$\frac{c_1}{2}||\mathbf{P} - \mathbf{W}_h||_F^2 + \frac{c_2}{2}||\mathbf{Q} - \mathbf{W}_\ell||_F^2 + \frac{c_3}{2}||\mathbf{W}_h - \mathbf{W}_\ell||_F^2$$

# ADMM for CDL - Algorithm

1. **Input**: Training examples $\mathbf{S}_h$ and $\mathbf{S}_l$, numb. of iterations: **K** and step size params. **c1, c2, c3**.
2. **Initialization**:
   - **Dictionaries** → random selection of the columns of $\mathbf{S}_h$ and $\mathbf{S}_l$
   - **Lagrangian matrices** → $\mathbf{Y}_1 = \mathbf{Y}_2 = \mathbf{Y}_3 = 0$.
3. **for k = 1,…,K do**

➢ Update $\mathbf{W}_h$ and $\mathbf{W}_l$

$$\begin{cases} \mathrm{W}_h = (\mathrm{D}_h^T \cdot \mathrm{D}_h + c_1 \cdot I + c_3 \cdot I)^{-1} \cdot (\mathrm{D}_h^T \cdot \mathrm{S}_h + Y_1 - Y_3 + c_1 \cdot \mathbf{P} + c_3 \cdot \mathrm{W}_l) \\ \mathrm{W}_l = (\mathrm{D}_l^T \cdot \mathrm{D}_l + c_2 \cdot I + c_3 \cdot I)^{-1} \cdot (\mathrm{D}_l^T \cdot \mathrm{S}_l + Y_2 + Y_3 + c_2 \cdot \mathbf{Q} + c_3 \cdot \mathrm{W}_h) \end{cases}$$

➢ Update **P** and **Q** → $\mathbf{P} = S_{\lambda_h}\left(\left|\mathbf{W}_h - \frac{Y_1}{c_1}\right|\right)$ **and**: $\mathbf{Q} = S_{\lambda_l}\left(\left|\mathbf{W}_l - \frac{Y_2}{c_2}\right|\right)$

**for j = 1,…,N do**

➢ Update $\boldsymbol{\varphi}_h$ and $\boldsymbol{\varphi}_l$ → $\phi_h = \mathbf{W}_h(j,:) \cdot \mathbf{W}_h(j,:)^T$ **and** $\phi_l = \mathbf{W}_l(j,:) \cdot \mathbf{W}_l(j,:)^T$

➢ Update $\mathbf{D}_h$ and $\mathbf{D}_l$

$$\mathbf{D}_h^{(k+1)}(:,j) = \mathbf{D}_h(:,j)^{(k)}(:,j) + \frac{\mathbf{S}_h \cdot \mathbf{W}_h(j,:)}{\phi_h + \delta}$$

$$\mathbf{D}_l^{(k+1)}(:,j) = \mathbf{D}_l(:,j)^{(k)}(:,j) + \frac{\mathbf{S}_l \cdot \mathbf{W}_l(j,:)}{\phi_l + \delta}$$

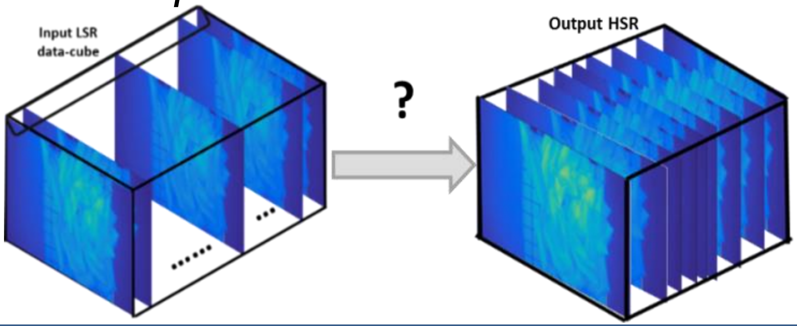
**end**

- ❑ **Normalize** $\mathbf{D}_h$ and $\mathbf{D}_l$ between [0,1]
- ❑ **Update** Lagrange multiplier matrices $Y_1$, $Y_2$ and $Y_3$

**end**

# Application: Spectral-Super Resolution

**Task:** *Given few acquired spectral observations of a hyperspectral scene, synthesize the full spectrum*




***So far…***

**Hardware solutions**

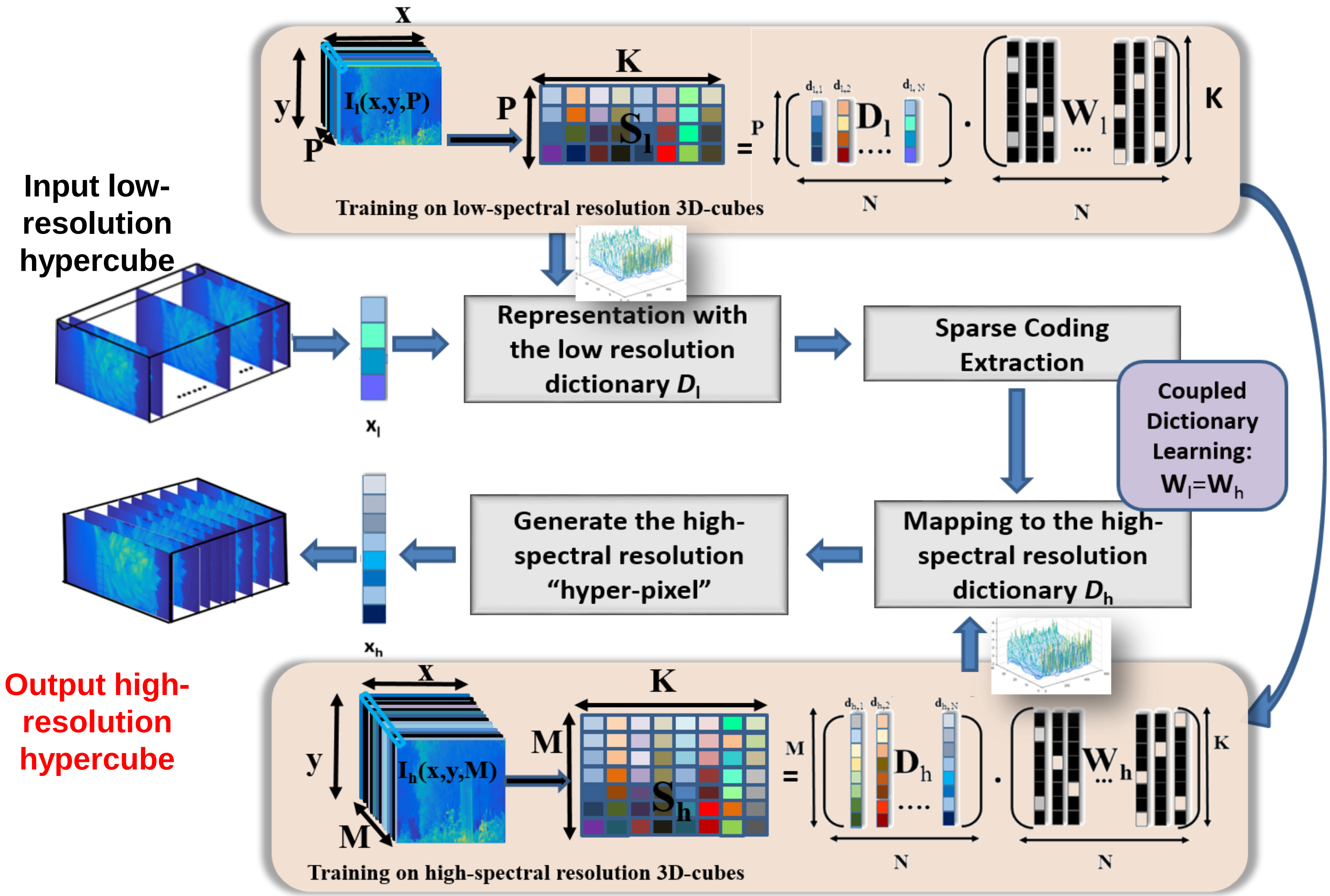
- Modify hyperspectral sensor's characteristics
- Additional optical elements

**Key intuition!**

- **SSR →** *Post – acquisition technique*
- **Inverse Imaging Problem**
  - ✓ **Prior knowledge:** *Sparse signal modeling*

K. Fotiadou, G. Tsagkatakis, and P. Tsakalides: *"Spectral Super-Resolution via Coupled Dictionary Learning."* ***Submitted*** *in IEEE Transactions* Special Issue on Computational Imaging for Earth Sciences.

# Proposed Spectral Coupled Dictionary Learning (SCDL) scheme

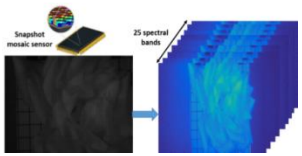


# SSR Experimental Setup

- **Dictionaries Generation:**
  - 100K coupled pairs of **high** & **low-spectral** resolution hyperspectral images

- **Hyperspectral Acquisition:**

1. **IMEC's snapshot mosaic hyperspectral instruments**
   - 5x5 snapshot mosaic hyperspectral sensor
   - Spectral resolution: 25 spectral bands
   - (600 – 875 nm)

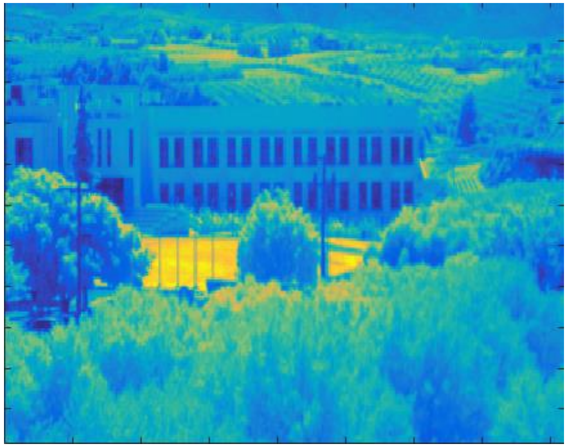



2. **Hyperion's remote sensing hyperspectral data**

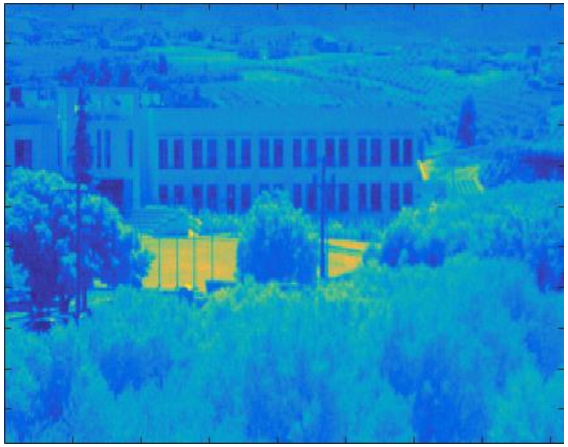
- **Down-sampling factors : (x2), (x3), (x4)**
- Evaluation Metrics: Peak signal to noise ratio (PSNR)

$$PSNR = 10log_{10}[L_{max}^2/MSE(x,y,\lambda)]$$

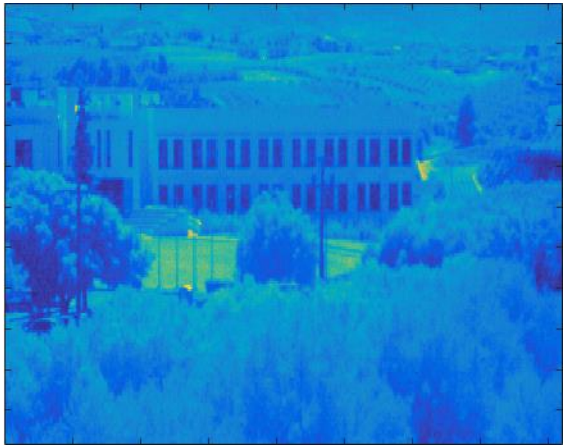
# Simulation Results: FORTH Building (x2)



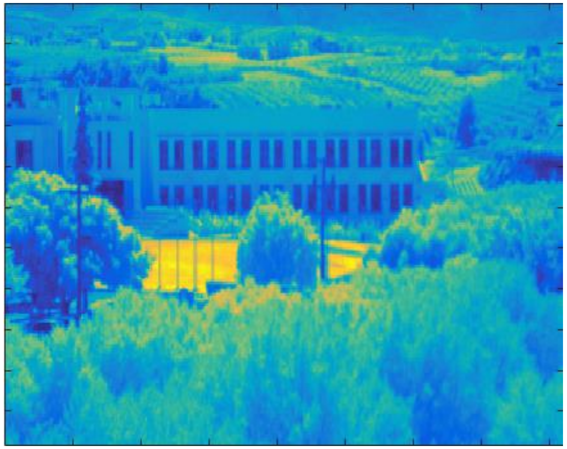
Ground Truth, **636.57** nm



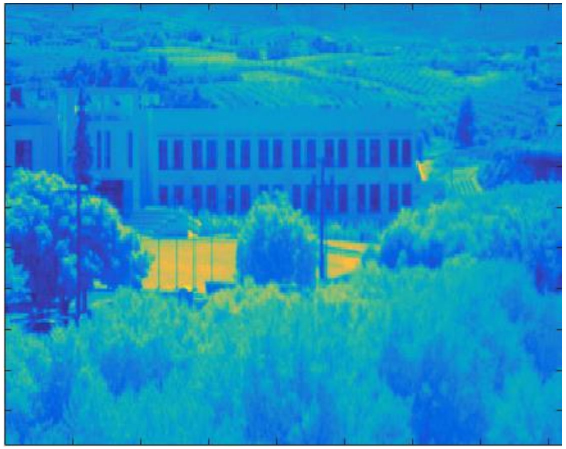
Ground Truth, **684.73** nm



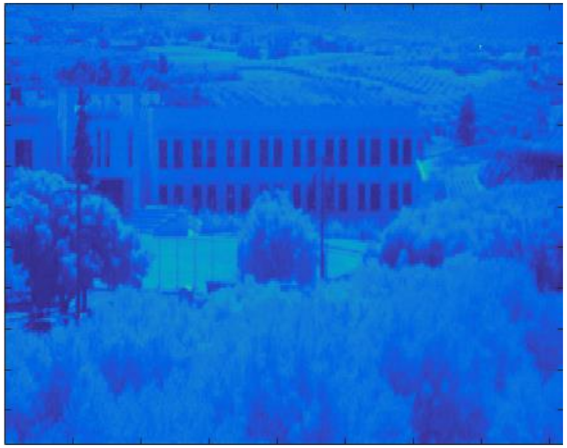
Ground Truth, **724.24** nm



(x2) SCDL, **636.57** nm,
SSIM: 0.97

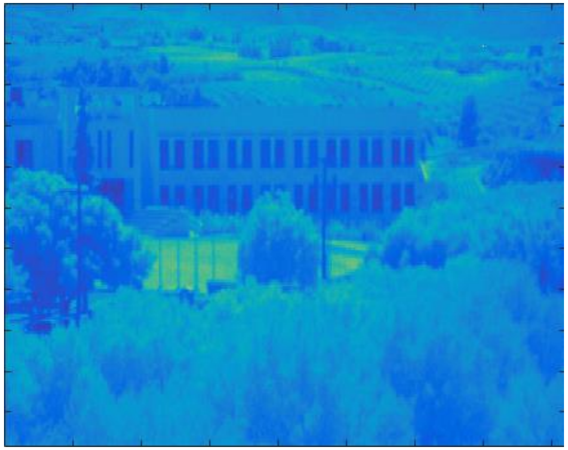


(x2) SCDL, **684.73** nm,
SSIM: 0.96

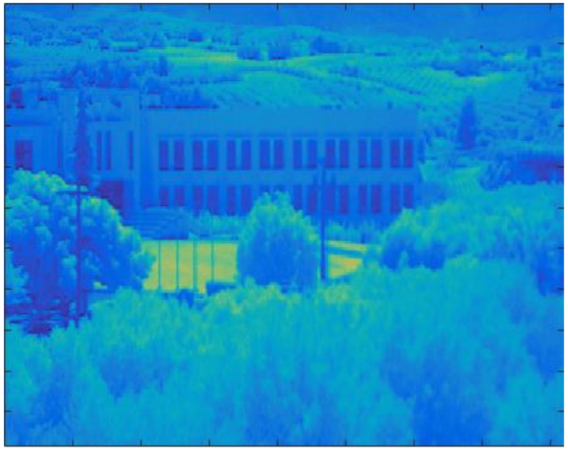


(x2) SCDL, **724.24** nm,
SSIM: 0.94

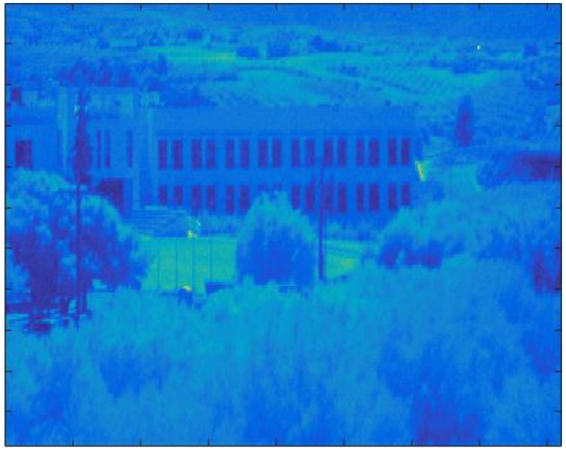
# FORTH Building: Comparison with the state-of-the-art



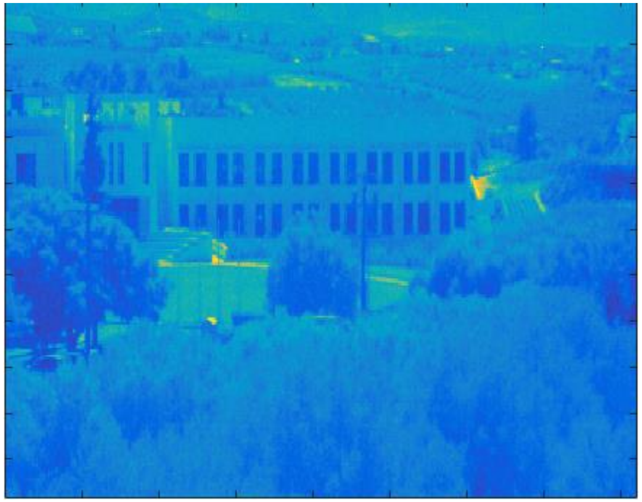
(x2) Cubic, PSNR: 26.41 dB



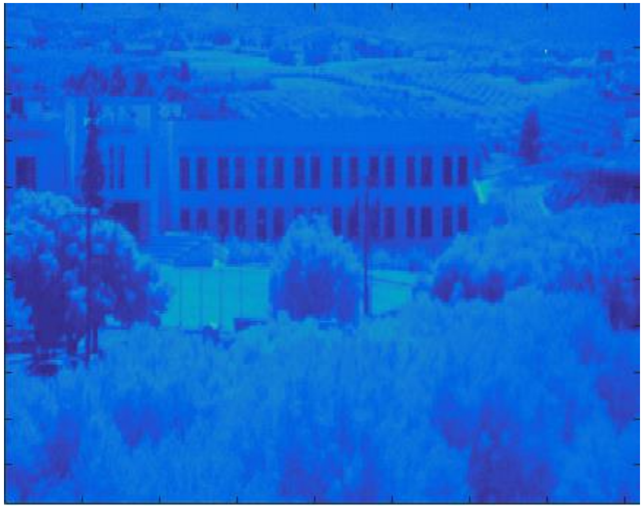
(x2) RWL1-SF, PSNR: 33.59 dB



(x2) SSR- K-SVD, PSNR: 34.47 dB

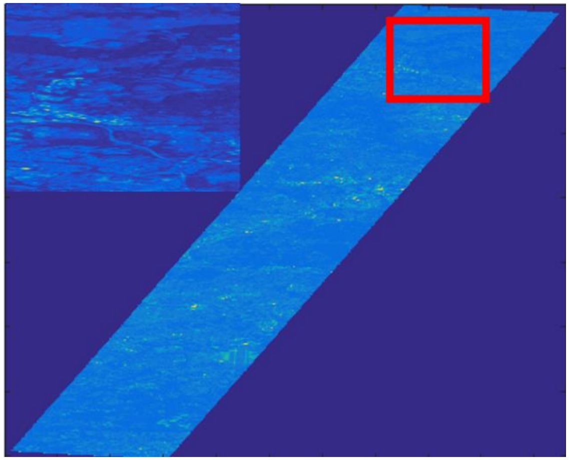


**(x2) Ground Truth, 669.81 nm**

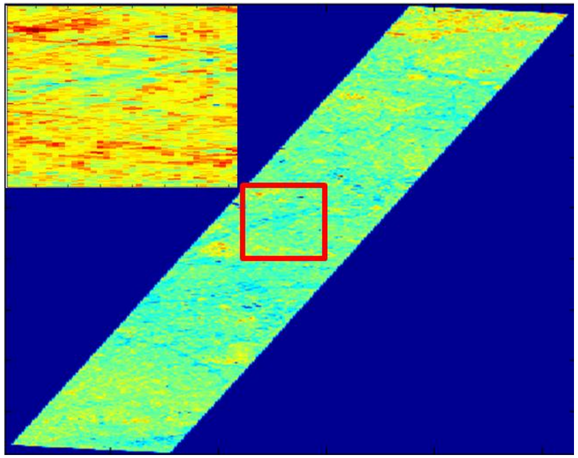


**(x2) SCDL, PSNR: 36.20 dB**

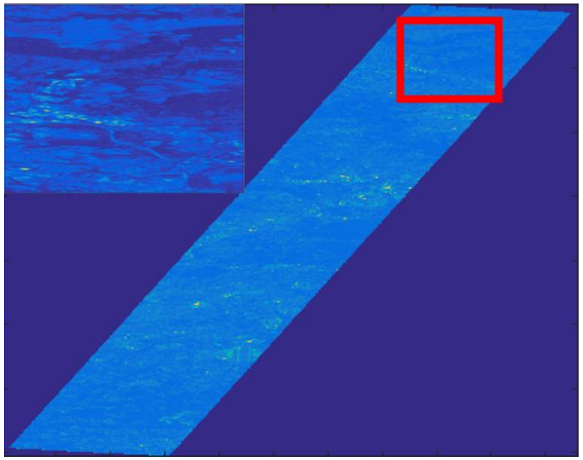
# Hyperion Data Recovery (x4 - zoom)



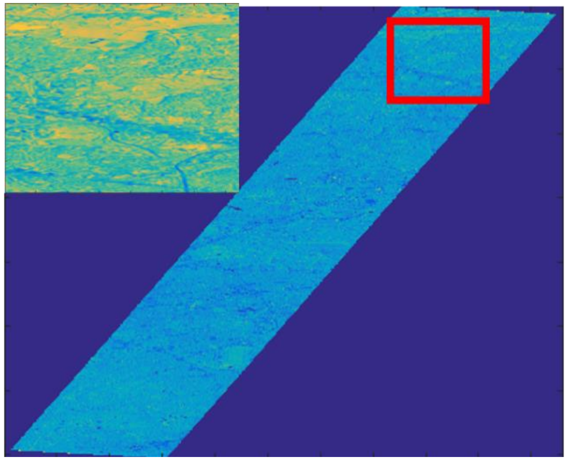
20th Band, *x4*, SSIM: 0.99



54th Band, *x4*, SSIM: 0.98



20th Band, Ground Truth



54th Band, Ground Truth

**Parameters**

- **Full Spectrum**

✓ **Input: 67 spectral bands from the VNIR region (437 – 833 nm)**

✓ **Sub-sampling factor : *x4***

✓ **High Res. Dictionary: 67 bands**

✓ **Low Res. Dictionary: 17 bands**

**PSNR Recovery of the 3D-cube:**

- **SCDL– SSR → 46.8 db**

# Agenda

- Sparse Coding for Image Processing Applications
- Coupled Dictionary Training
- **Learning Deep Features**

# Feature Learning

Raw data → Feature representation → Learning algorithm

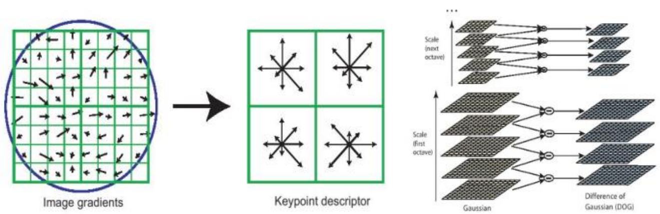
Find a better way to represent images than pixels!!

Image

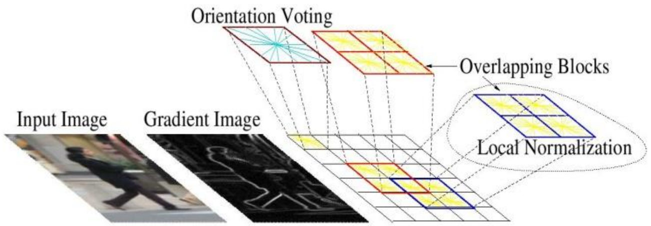
Low-level vision features

Recognition

- **Computer Vision Features**




**S**cale **I**nvariant **F**eature **T**ransform **(SIFT)**

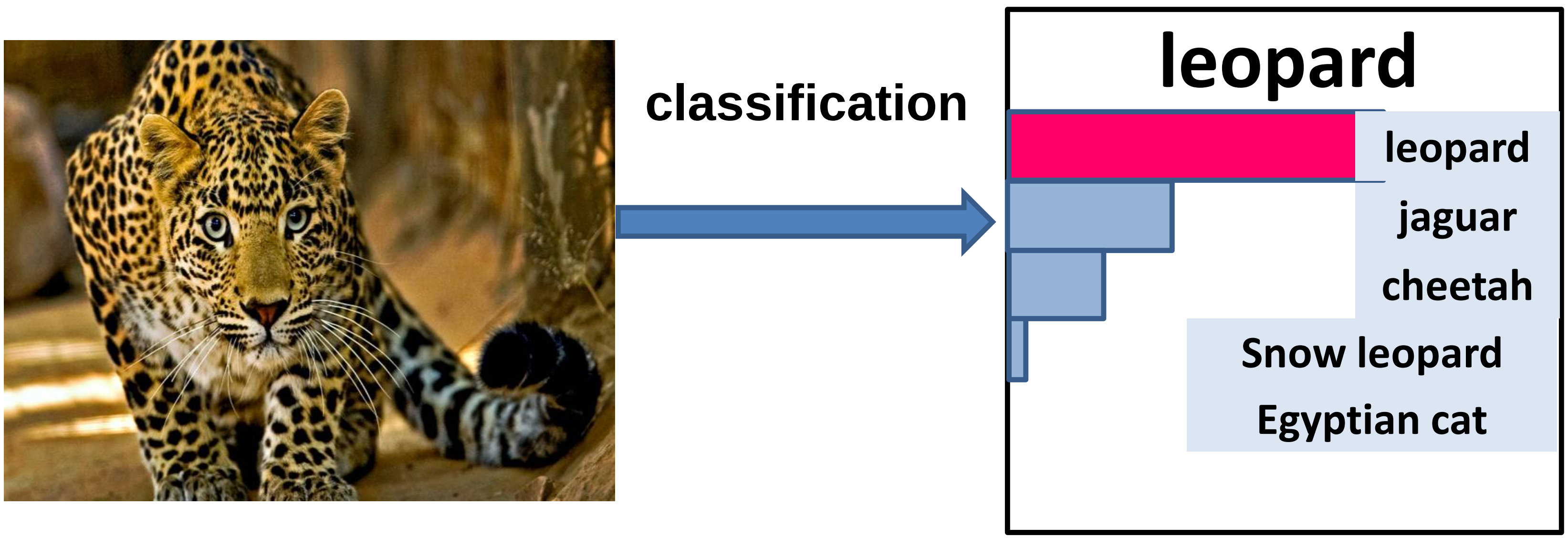



**H**istogram of **O**riented **Gradients** **(HoG)**

- **Limitation: *What is the optimal feature for each application?***

# Deep Learning: Big Picture

- **Challenge**: *How could an artificial vision system learn appropriate internal representations automatically, the way humans seem to by simply looking at the world?*



# Why using deep learning?

- Traditional **("shallow")** architectures

**Input Image/ Video Sequences** → **Hand-designed feature extraction** → **Trainable Classifier** → **Object Class**

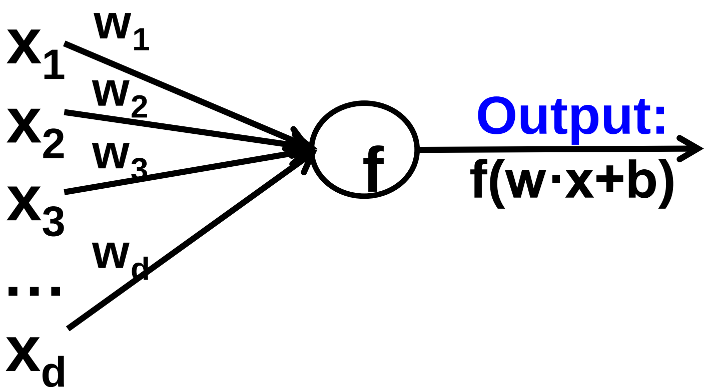
- Vs.... "**deep**" architectures

**Input Image/ Video Sequences** → **Layer 1** → **Layer 2** … → **Layer N** → **Object Class**

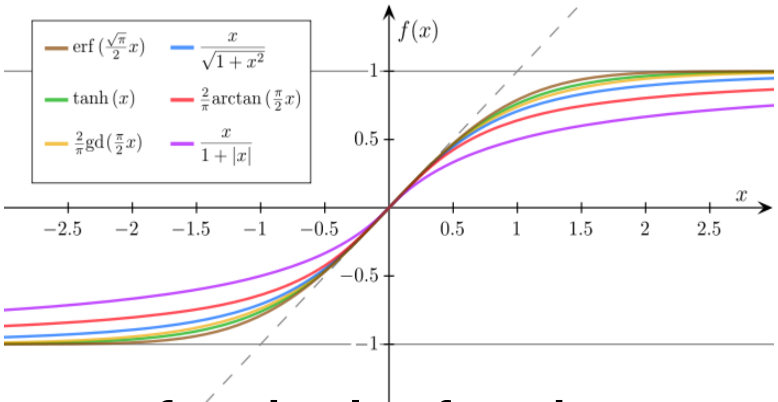
**Advantage: Learn a *feature hierarchy* all the way from pixels to classifier!**

# Background: Typical Neural Networks

- **A neuron**



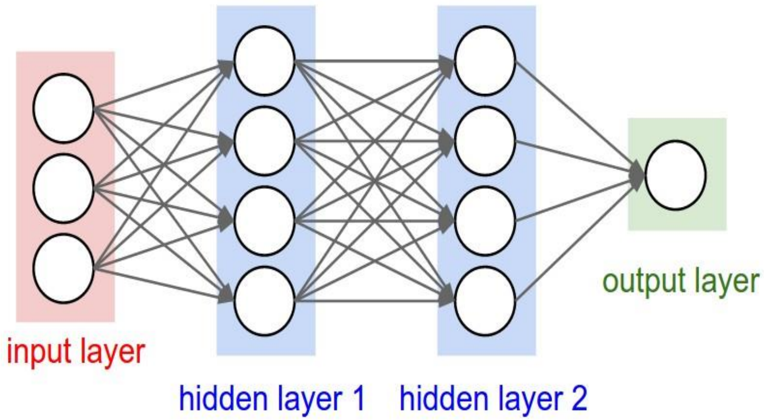

**Pixel values**




***f*: activation function**

- **Multi-layer Neural Network**
  - ✓ Nonlinear classifier
  - ✓ Learning can be done by *gradient descent*

→ **Back-Propagation**

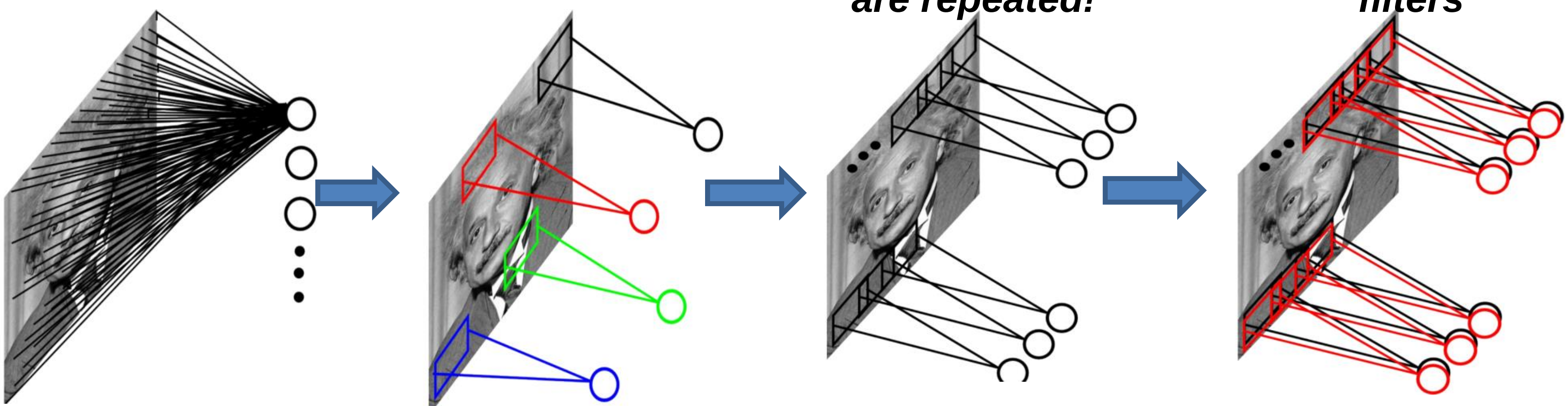


# Motivation: Convolutional Neural Networks

- **Limitations**:
  - Full connectivity of traditional neural nets → **wasteful**!
  - Tremendous number of parameters → **over-fitting**!
  - Example: 1000 x 1000 image→ **1B parameters!!!**
- **Key intuition!**

***Interesting features are repeated!***

***Learn multiple filters***



**1000 x 1000 input**
**1 M Hidden units**
**→ 1B params.**

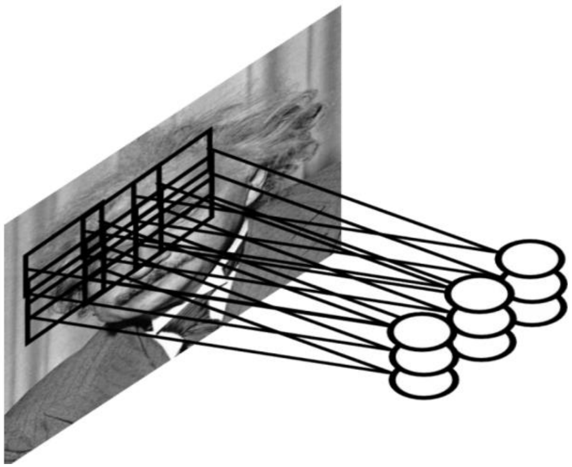
**Filter size: 10x10**
**→ 1 M params.**

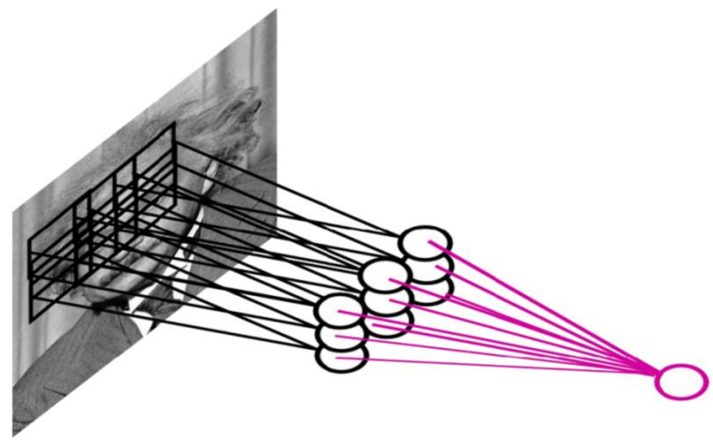
**Convolution with learned kernels!**

**100 filters filter size: 10 x 10 →**
**10 K params.**

# Convolutional Neural Networks (CNN's)

- **Advantages**
  - Translation invariance
  - Tied filter weights





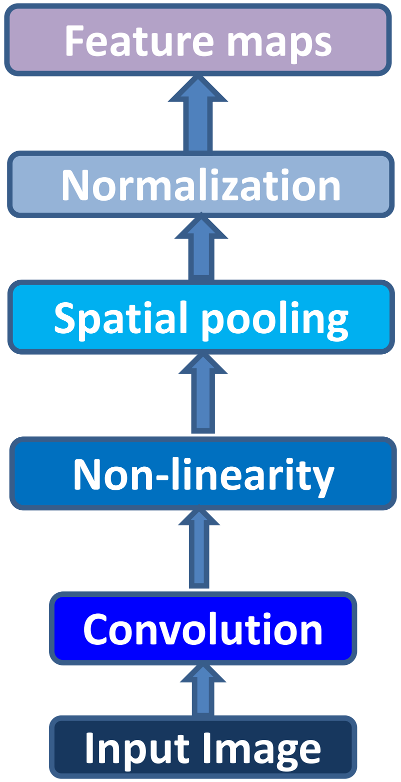
❖ **Question**: ***How to detect the accurate position of the eye?***

❖ **Answer**: **By pooling (max or average) filter responses at different locations**

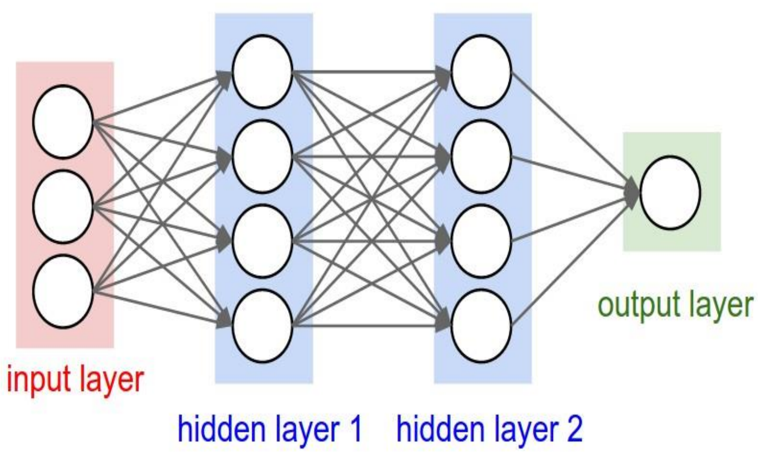
✓ ***robustness to the exact spatial location of the feature!***

# Convolutional Neural Networks (CNN's)

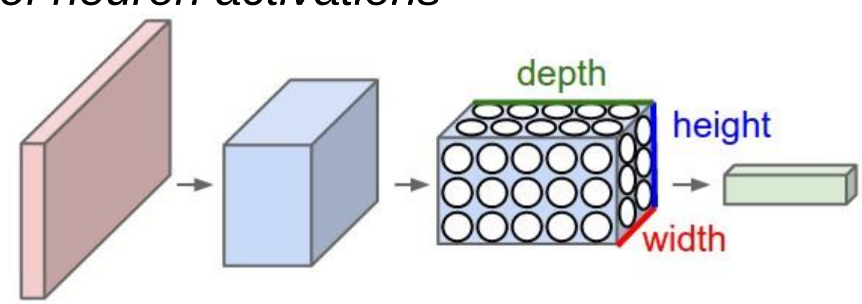
- **Typical structure of a convolutional layer**
  - **Pooling & normalization → optional**
  - **Training →** multi-layer neural nets (stacked)
  - **Final layer →** fully connected layer
  - **Output size: number of classes!**
- **Regular neural networks vs. CNN's**




**Typical 3-layer NN**

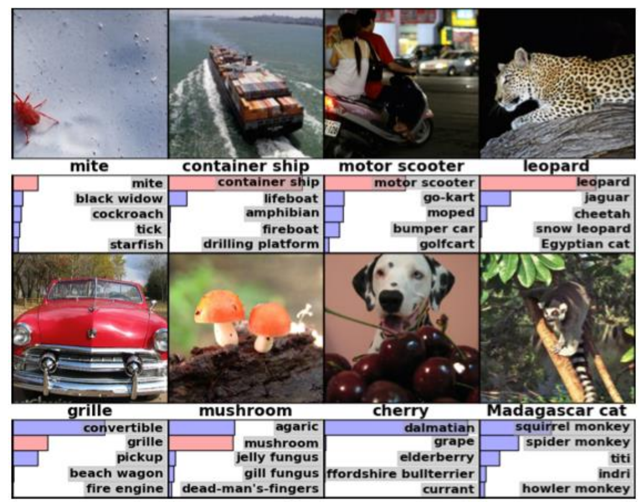



- **Convolutional NN:** *Transforms the 3D input volume to a 3D output volume of neuron activations*



# Application: ImageNet -2010 Contest [1]

- 1.2 million high-resolution images
- 1,000 different classes
- 50,000 validation images, 150,000 testing images
- Top-1 error 47.1% best in contest, 45.7% best published
- Top-5 errors 28.2% best in contest, 25.7% best published

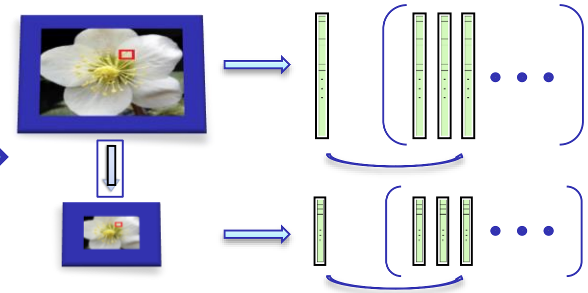



[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

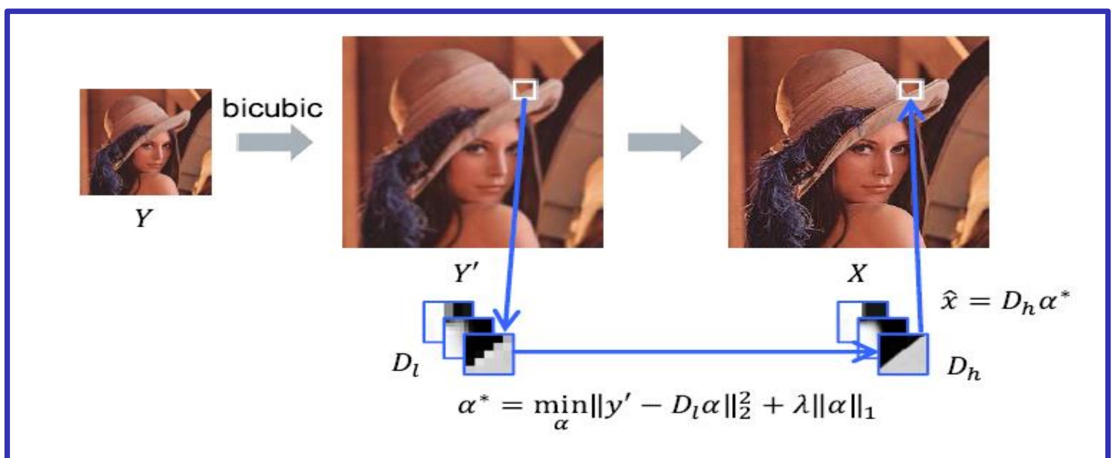
# Application: Image Super-Resolution

- **Traditional Sparse-based Super-Resolution based on coupled trained dictionaries**

**Training phase**

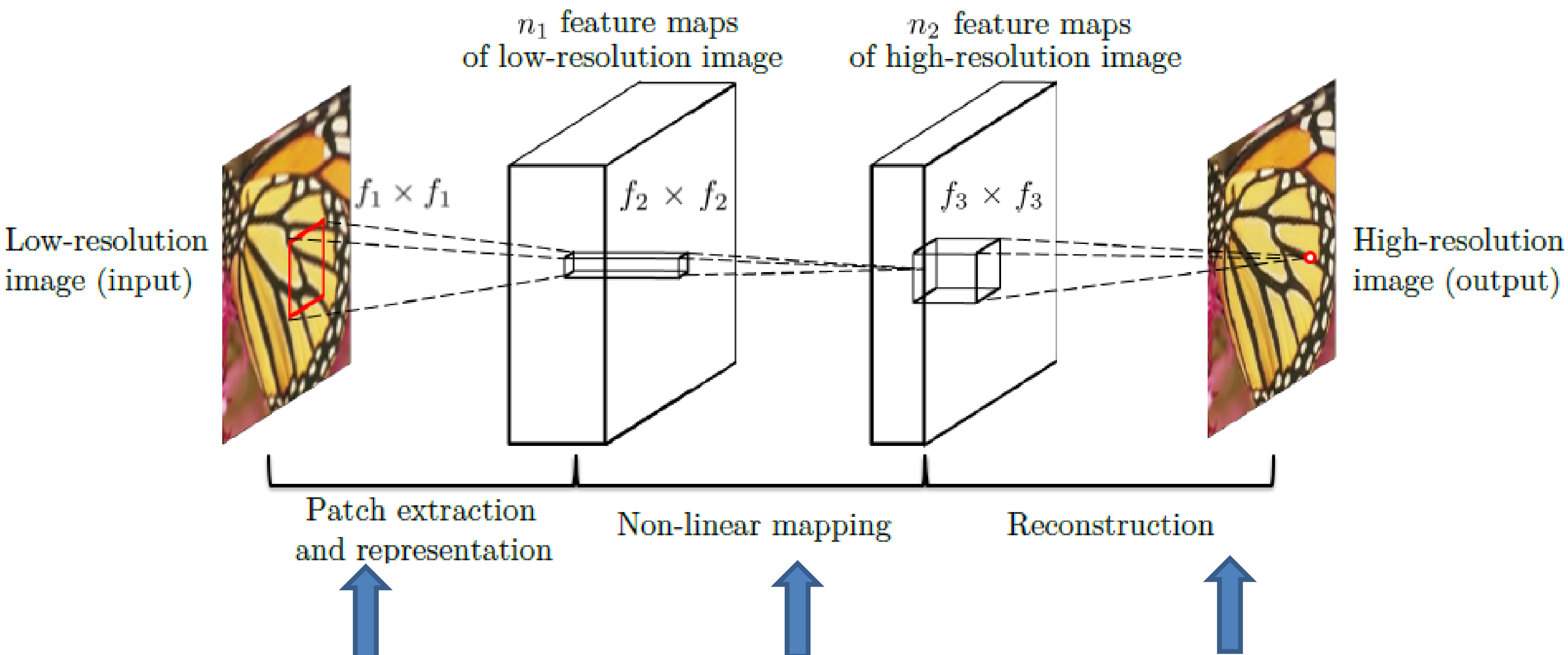


**Testing phase**




J. Yang et al. “Image Super-Resolution via Sparse Representations”. In IEEE transactions on image processing, 2010

# Application: Image Super-Resolution[1]




**1st layer**

$$F_1(\mathbf{Y}) = \max(0, W_1 * \mathbf{Y} + B_1)$$

where: $W_1 \in R^{c\, x\, f_1\, x\, f_1}$

**2nd layer**

$$F_2(\mathbf{Y}) = \max(0, W_2 * F_1(\mathbf{Y}) + B_2)$$

where: $W_2 \in R^{n_1\, x\, f_1\, x\, f_1}$

**3rd layer**

$$F(\mathbf{Y}) = W_3 * F_2(\mathbf{Y}) + B_3$$

where: $W_3 \in R^{n_2\, x\, f_2\, x\, f_2}$

[1] C. Dong, et al. “Image Super-Resolution Using Deep Convolutional Networks”, ECCV 2014

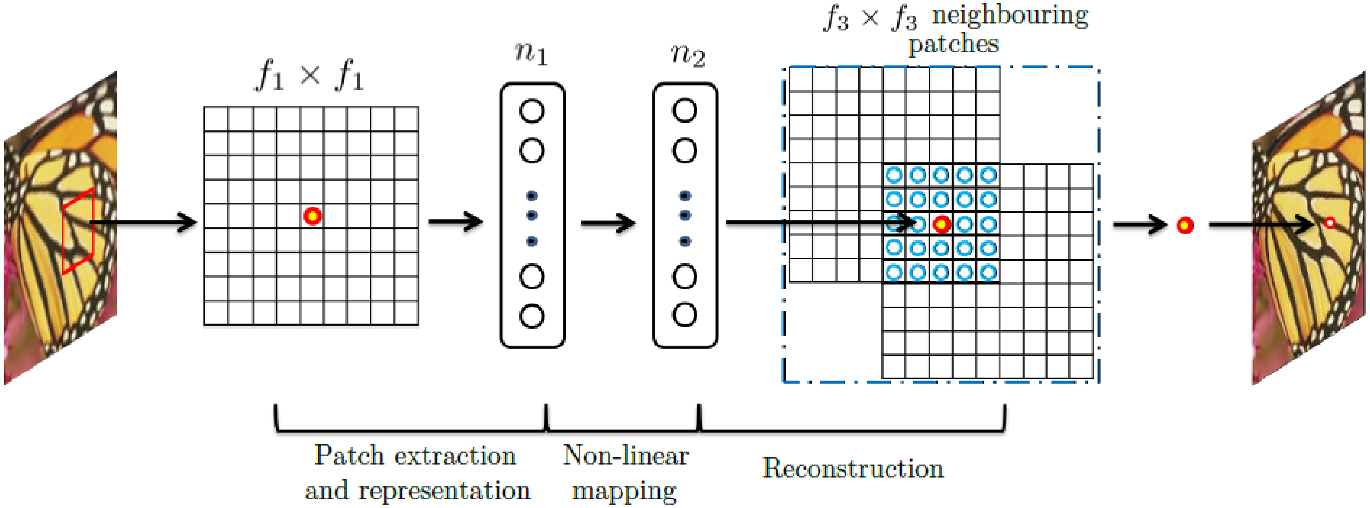
# Relationship to the sparse coding based methods

## Sparse Coding

- Extract LR patch & project on a LR dictionary, of size $n_1$
- Sparse coding solver, transform to HR sparse code, of size $n_2$
- Project to HR dictionary, average HR patches

## SRCNN

- Apply $n_1$ linear filters on the input image
- Non linear mapping
- Linear convolution on the $n_2$ feature maps

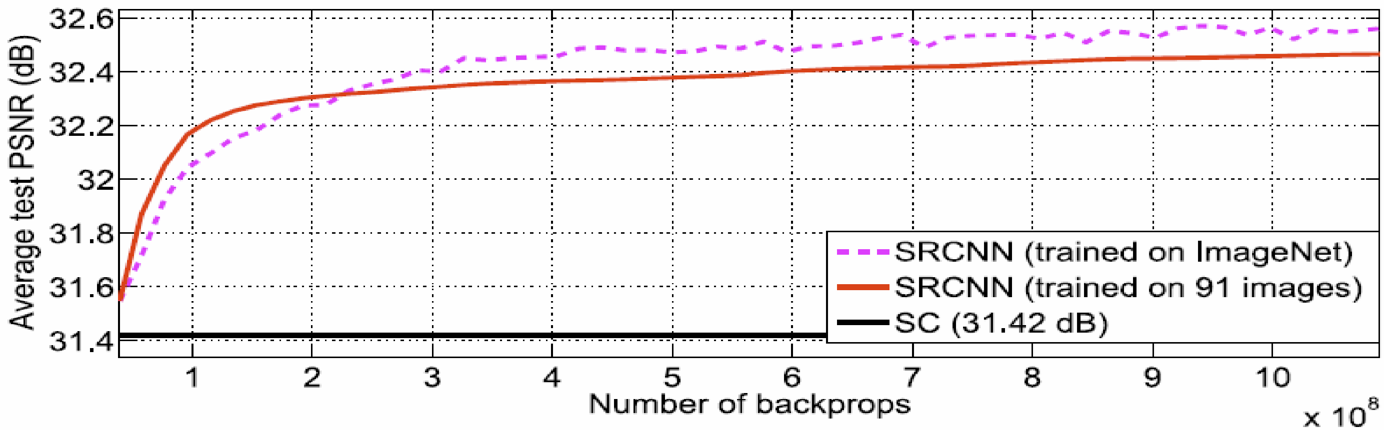


# Training Parameters (1/2)

- **Training:**
  - Small dataset: composed of 91 images
  - Large Dataset: ~ 395.909 images (ImageNet! )

- **Testing:**
  - Set 5-5 images
  - Set 14-14 images
  - ImageNet

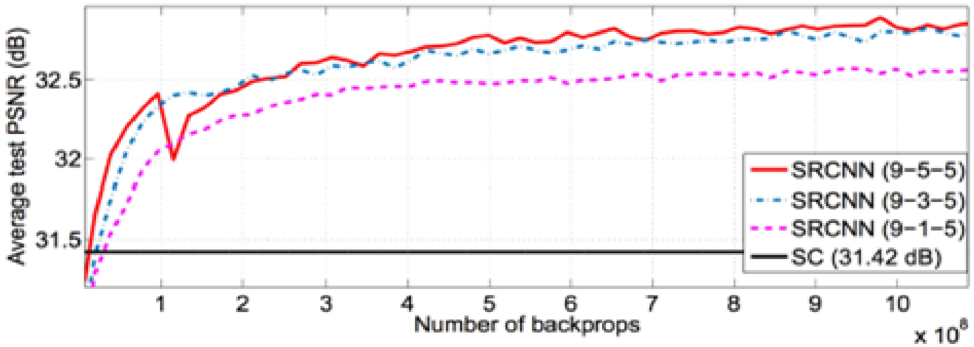
- **Scaling factors:** x3
- **The more training data the better!**



# Training Parameters (2/2)

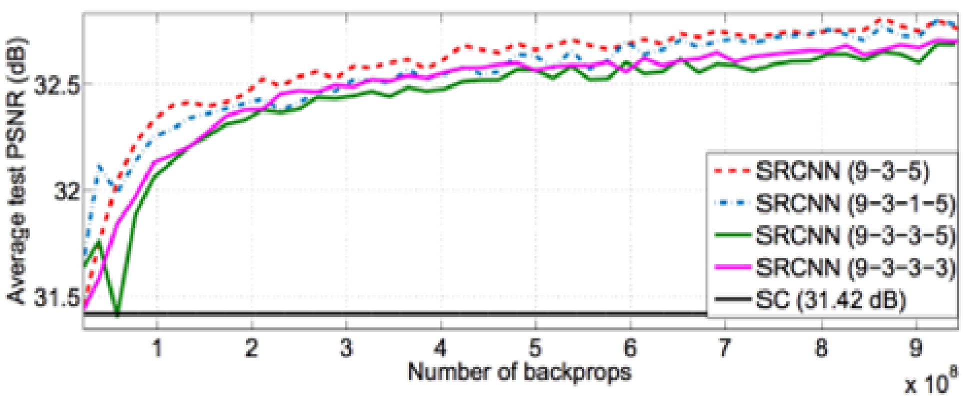
## ❑ Sensitivity effects

- **Filter size**



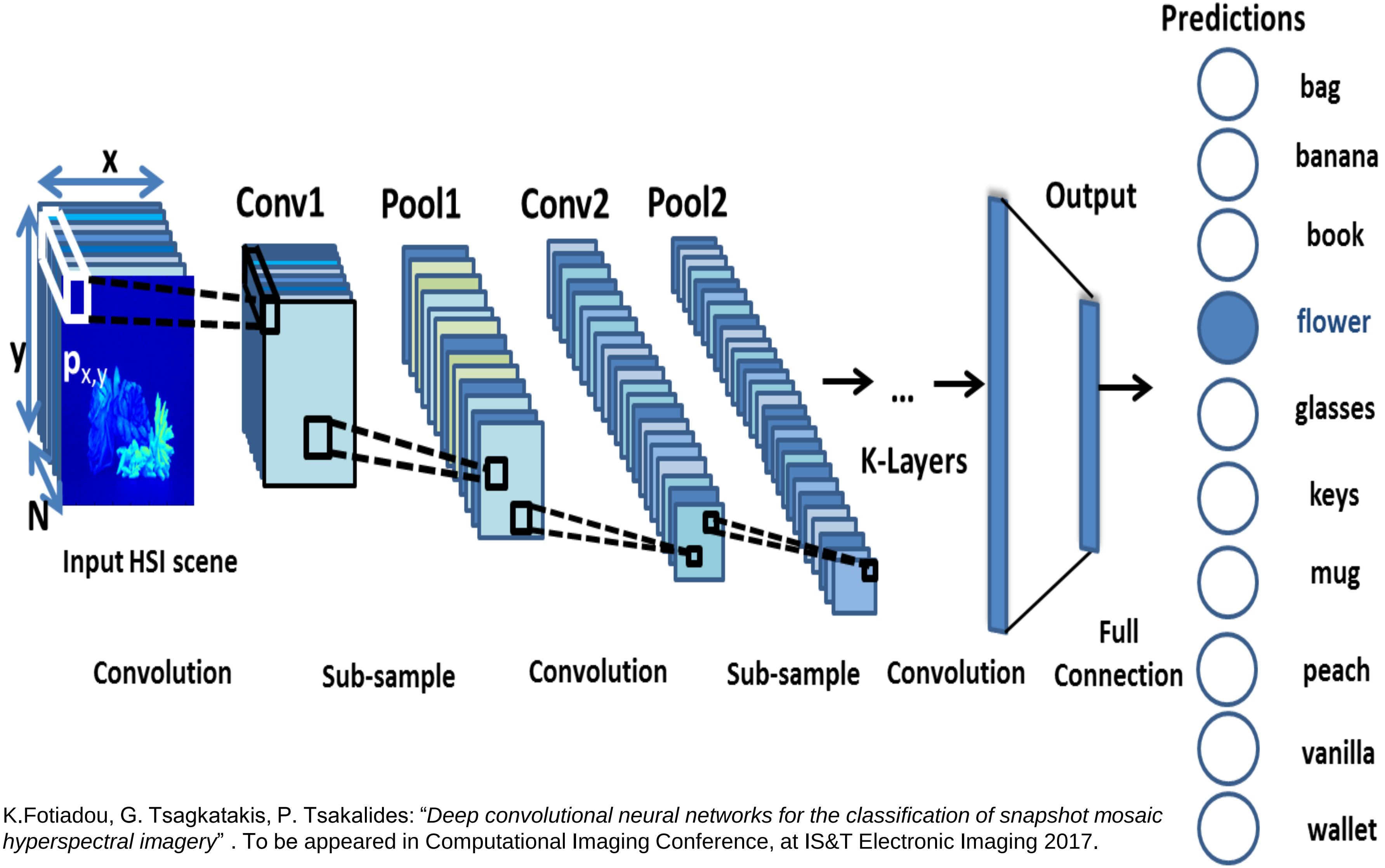

- Larger filter size → better results
- Trade-off between performance & speed

- **Deeper structure**




- The deeper the better?
- Sensitive to initialization params & learning rate

# Application: Classification of HSI images




K.Fotiadou, G. Tsagkatakis, P. Tsakalides: "*Deep convolutional neural networks for the classification of snapshot mosaic hyperspectral imagery*" . To be appeared in Computational Imaging Conference, at IS&T Electronic Imaging 2017.

# Preliminary Results

- **Experimental Setup**
  - 10 categories of indoors hyperspectral scenes
  - **<u>Training Phase:</u>**
    - Pre-trained CNN model → AlexNet [1], [2]
    - CNN architecture: 23 layers

- **<u>Testing Phase:</u>**
  - Split the sets into training & validation
  - Pick randomly 30% for training & 70% for testing
  - Extract training features using CNN
  - Train a Multiclass SVM Classifier using CNN features
  - Evaluation of the Classifier

**Proposed Mean Accuracy: 89%**


**Classifier**

**Bag → Correct !**

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[2] Vedaldi, Andrea, and Karel Lenc. "MatConvNet-convolutional neural networks for MATLAB." arXiv preprint arXiv:1412.4564 (2014).

# Conclusions

- **Linear vs. non – linear sparse representations**
  - Widely used in multiple applications in signal / image processing
- **Key component of sparse coding → the design of the proper dictionary**
  - Single vs. Coupled Dictionary Learning
  - ADMM decomposition for coupled dictionary learning
- **Feature Learning**
  - Handcrafted vs. learnt feature operators
  - Deep Learning
- **What is the current trend in feature learning ?**
  - **Convolutional Neural Networks**
    - ✓ **Applications** : Classification, Detection, Image restoration