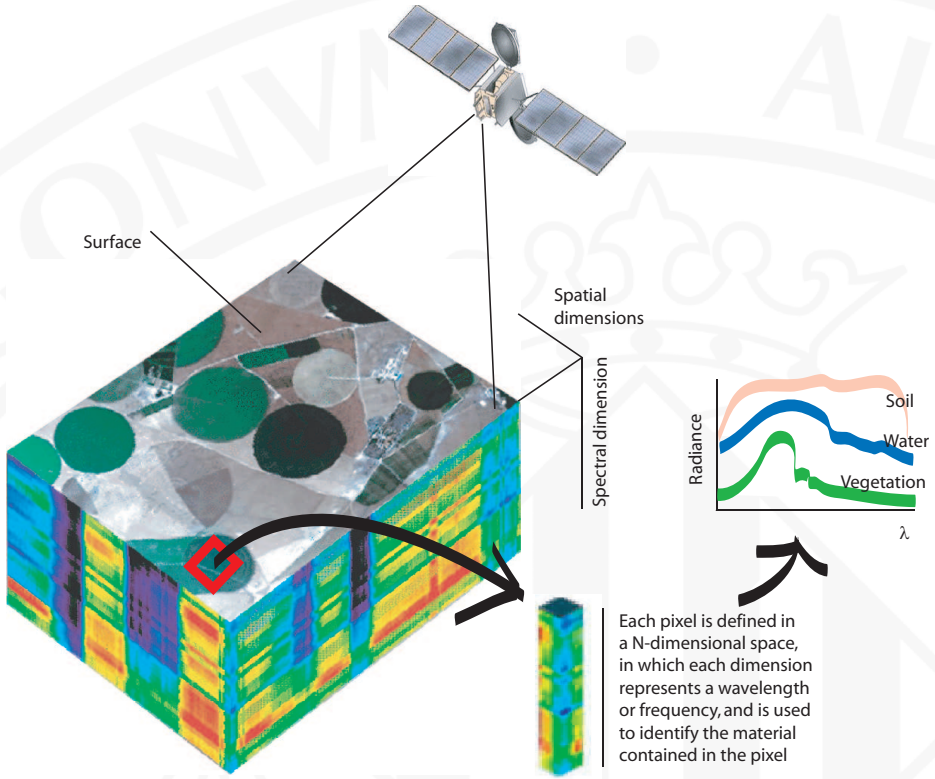
Machine Learning in Remote Sensing

Gustau Camps-Valls

Image Processing Laboratory (IPL), Universitat de València, Spain
Gustau.camps@uv.es, http://isp.uv.es

1. **Introduction to remote sensing image processing**
   - Observing Earth from space with MS/HSI sensors
   - Statistical challenges: dimensionality and overfitting
   - The image representation
2. **Feature extraction from remote sensing images**
   - Spatial and spatial-spectral features
   - Computer-vision based features
3. **Supervised remote sensing image classification**
   - Standard classifiers: LDA, k-NN, SVM and RF
   - Evaluating classification performance: OA, conf.mat, kappa
   - Fusing information: optical, radar and LiDAR features
4. **Deep convolutional neural networks**
   - Exploiting spatio-spectral image features
   - Contextual and spatio-temporal classification
5. **Target detection in remote sensing images**
   - Detecting one class of interest
   - Main approaches and methods: OSP and SAM
6. **Change detection in remote sensing images**
   - Supervised and unsupervised
   - Bitemporal and multitemporal
7. **Unmixing and abundance estimation**
   - Endmember determination and identification
   - The unmixing problem: standard methods and advances
8. **Retrieval of biophysical parameters**
   - VI-based, LUT inversion, regression-based
9. **Bibliography, source code and resources**

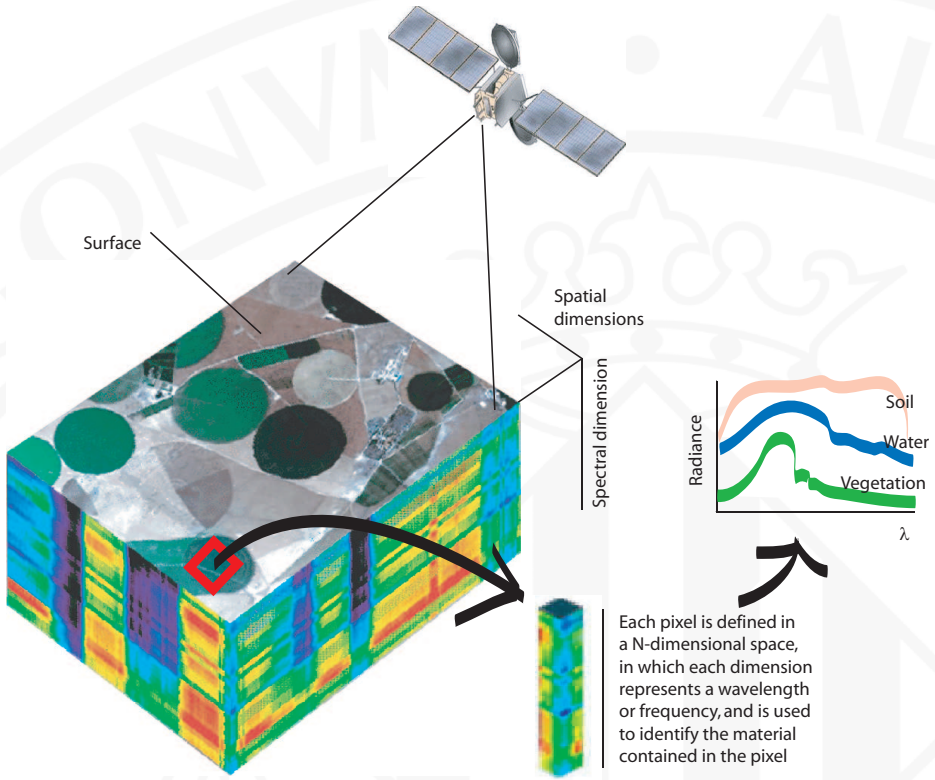
Part 1: Introduction to hyperspectral image processing



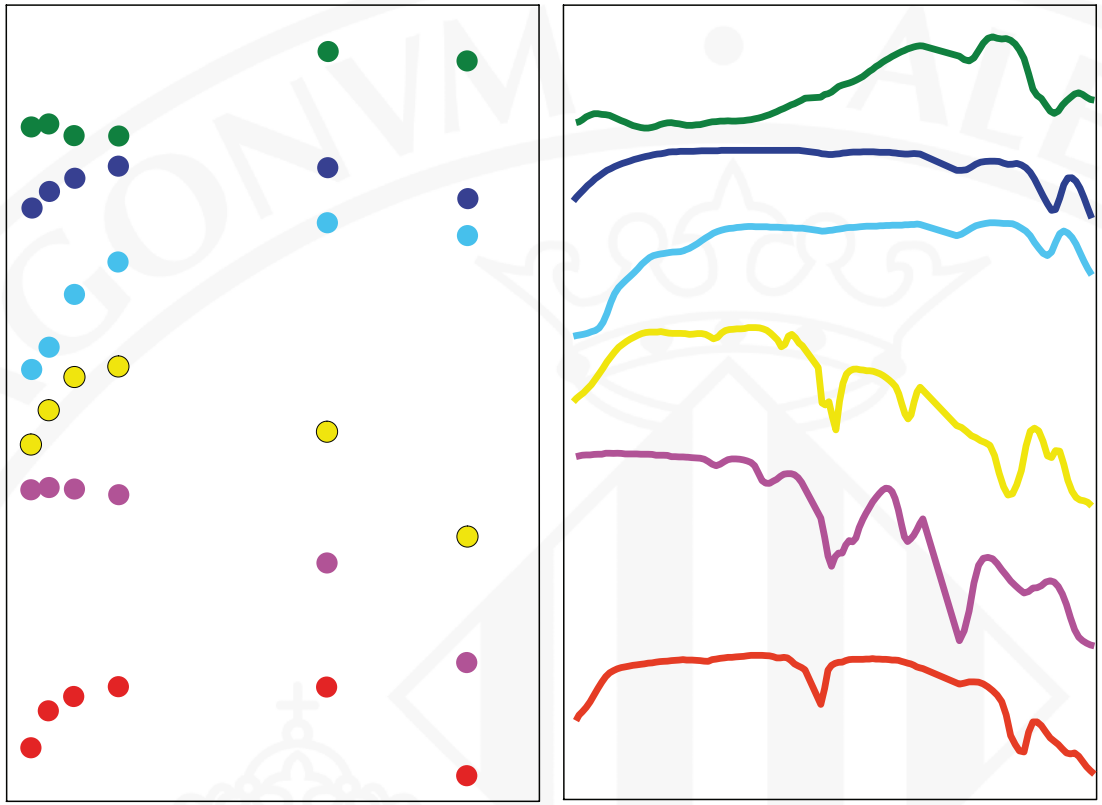
Lillesand08 *"Monitor and model the processes on the Earth surface and their interaction with the atmosphere"*

Liang04 *"Obtain quantitative measurements and estimations of geo-bio-physical variables"*

Manolakis02 *"Identify materials on the land cover analyzing the acquired spectral signal by satellite/airborne sensors"*

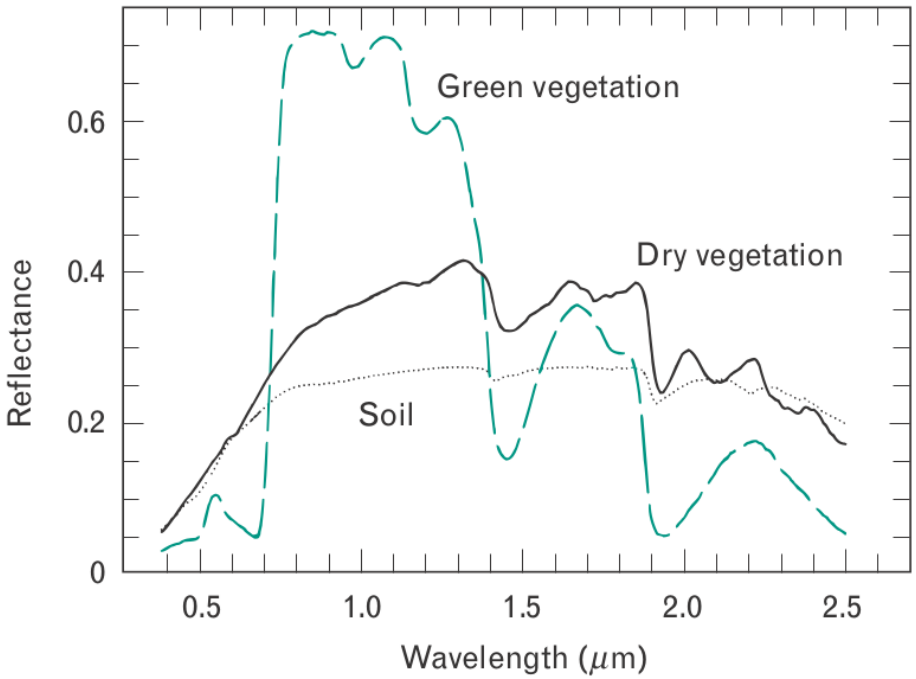


- Materials in a scene reflect, absorb, and emit electromagnetic radiation in a different way depending of their molecular composition and shape.
- Remote sensing exploits this physical fact and deals with the acquisition of information about a scene at a short, medium or long distance.
- Image spectroscopy allows to identify materials in the scene with unprecedented accuracy



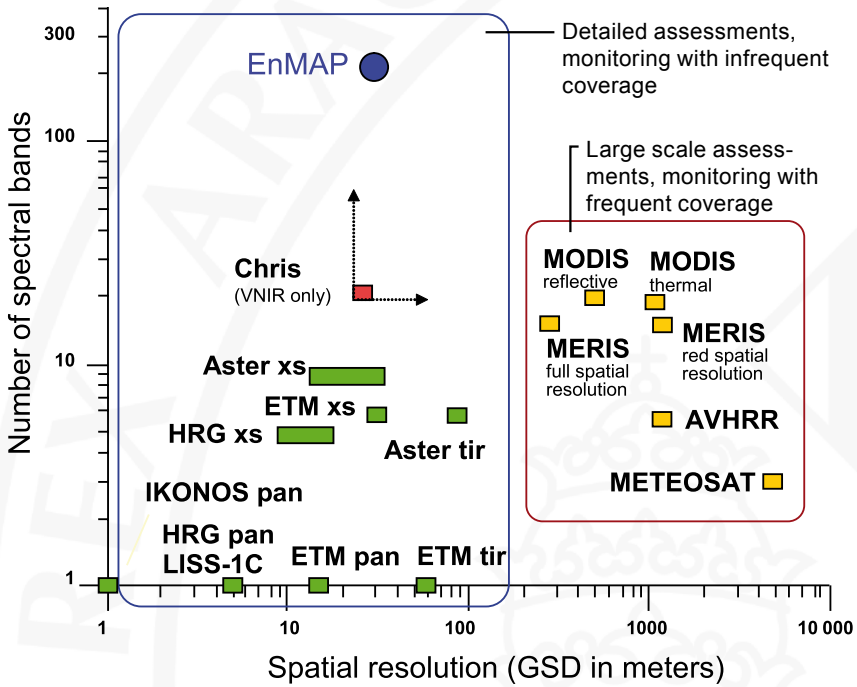
Multispectral Hyperspectral

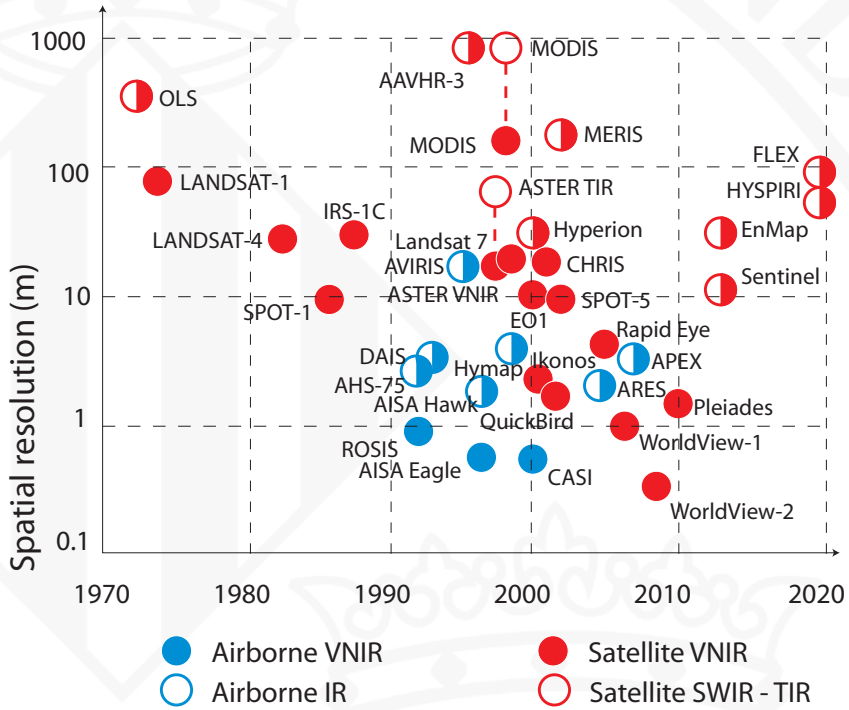
- Hyperspectral signals allow finer material characterization
- Absorption, depth, re-emissions and modulated particular spectral features
- Accurate identification of chemical components and bio-chemical processes



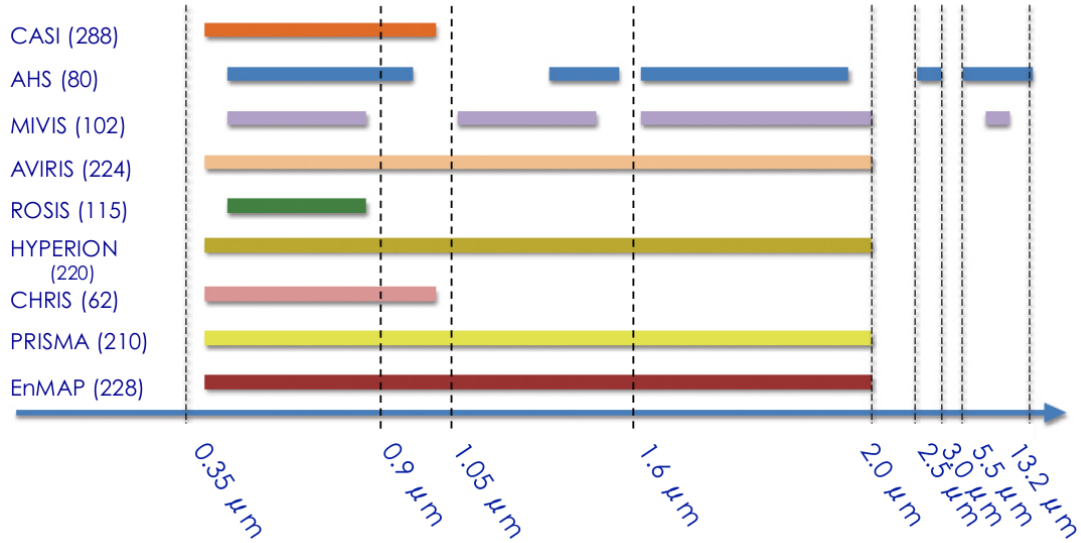
- Different materials produce different electromagnetic radiation spectra
- The spectrum shows absorptions and emissions at different wavelengths
  ≫ e.g. reflectance for soil, dry vegetation, and green vegetation
- The high spectral resolution preserves important aspects of the spectrum (e.g., shape of narrow absorption bands), and makes differentiation of different materials on the ground possible
- The spectral information contained in a hyperspectral image pixel can therefore indicate the various materials present in a scene

Left: Performance comparison of the main air- and space-borne multi- and hyperspectral systems in terms of spectral and spatial resolution.
Right: Evolution of the spatial-spectral resolution through the years.





Credits: http://www.enmap.de/



Barnsley04,Cutter04 PROBA/CHRIS

Ungar03 EO1/Hyperion

Kaufmann08,Stuffler07 EnMAP (Environmental Mapping and Analysis Program, GFZ/DLR, Germany)

Stoll03,Moreno06 FLEX (ESA proposal)

Green08 HyspIRI (NASA GSFC proposal)

Trishchenko07 MEOS

ZASat ZASat (South African proposal, University of Stellenbosch)

HIS HIS (Chinese Space Agency)

HERO HERO - Hyperspectral Environment and Resource Observer, Canadian Space Agency

Some fields of application...

Geology
- Mineral detection
- Cover homogeneity

Forestry
- Infected trees
- Status monitoring
- Forest clearing

Sea/ice/coastal
- Oil spills monitoring
- Water quality

Precision agriculture
- Crop stress location
- Crop productivity

Atmosphere
- Air quality, pollutants
- Global/local change

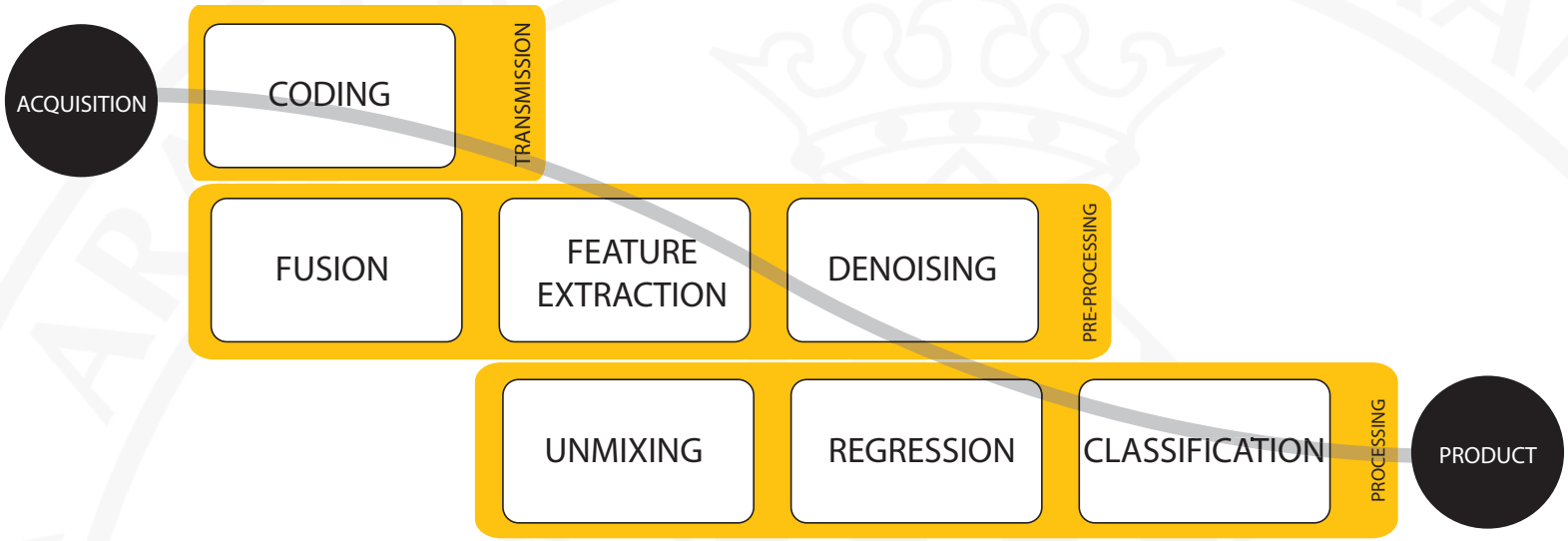
Land management
- Crop monitoring/phenology
- Land use/cover change

Defense
- Target detection
- Mine detection

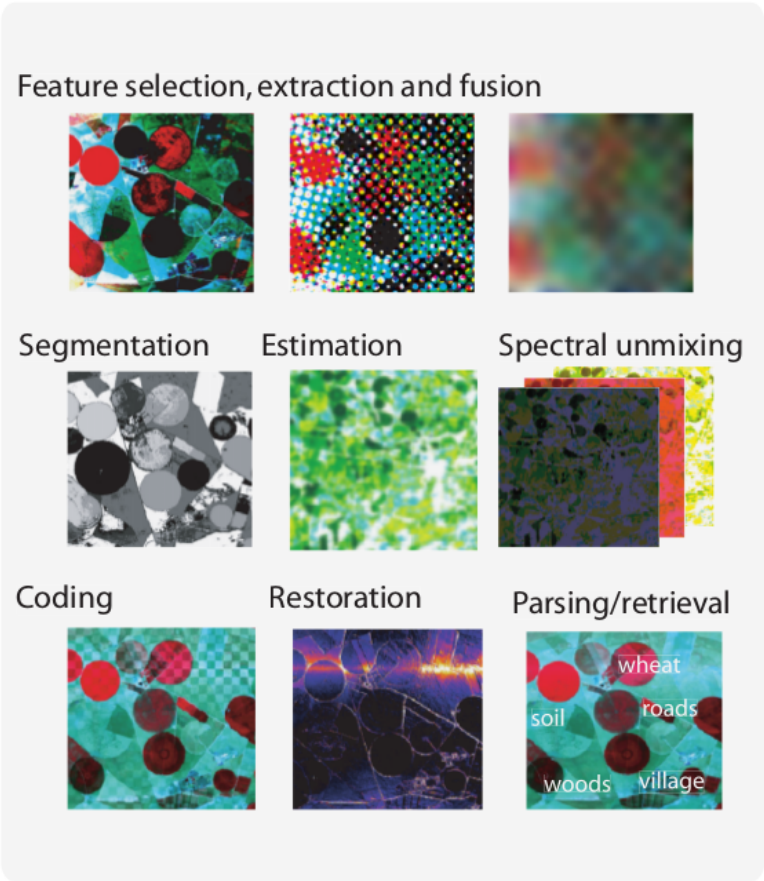
Public safety
- Logistics & operations
- Fire risk, floods

Regulation & Policy making
- Urban growth
- Settlements, population movements

A standard image processing chain:



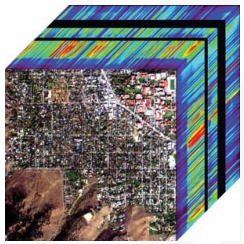
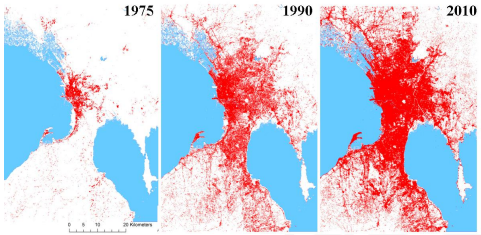
- Many steps and by-products from signal/image acquisition to the product
- Transmission $\longrightarrow$ Preprocessing $\longrightarrow$ Processing
- A wide diversity of problems and dedicated tools

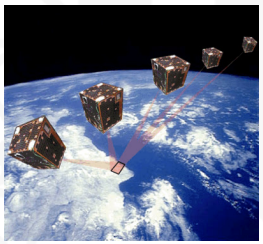
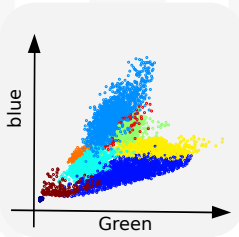


1. Select best features (channels, spatial) that describe the problem (classification, retrieval)
2. Extract (lin/nonlin) combinations of spectral channels that best describe the problem
3. Combine panchromatic and optical bands to improve products
4. Automatically find groups of pixels in the image (for screening, detection)
5. Estimate geo-bio-physical parameters and variables (temperature, LAI, etc) from spectra
6. Estimate the spectral components (pure pixels, endmembers) in a 'mixed' pixel
7. Compress images for storage and transmission, while keeping most of the information
8. Remove noise and distortions due to acquisition (sun glint) or transmission (vertical stripes)
9. Assign semantic classes to objects (pixels, patches, regions) in the scene

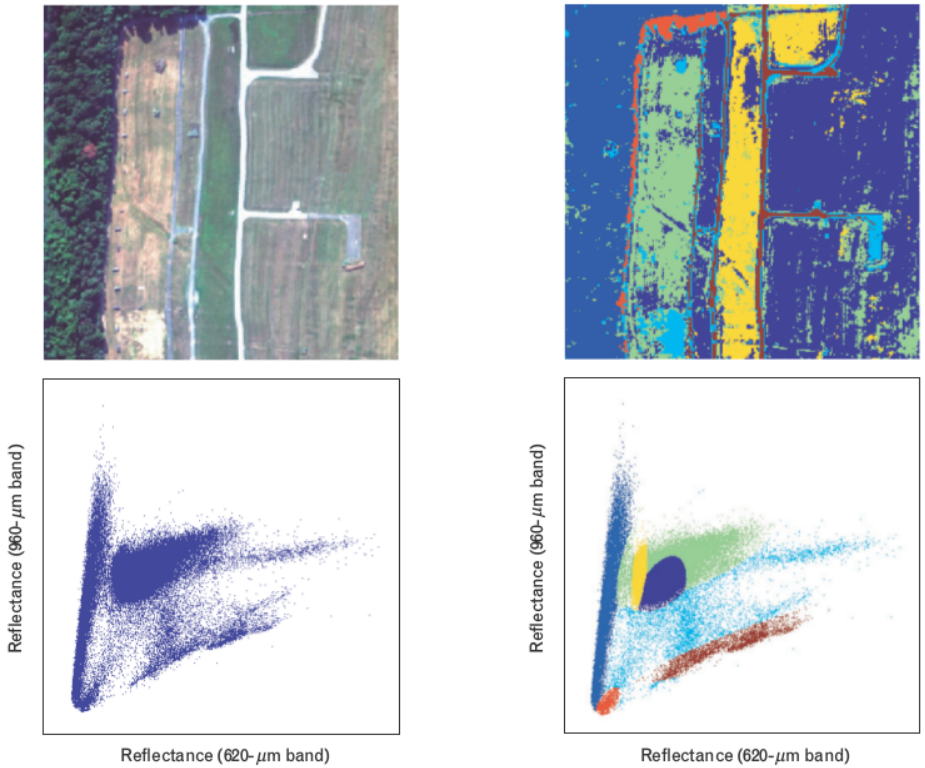
Characteristics of remote sensing data:

- High spectral resolution $\rightarrow$ moderate spatial resolutions (mixed pixels, subpixel targets)
- High dimensional data: multi-temporal, multi-angular and multi-source fusion
- Non-linear and non-Gaussian feature relations
- Few supervised (labeled) information is available (high cost)
- Tons of data to process in (near) real-time

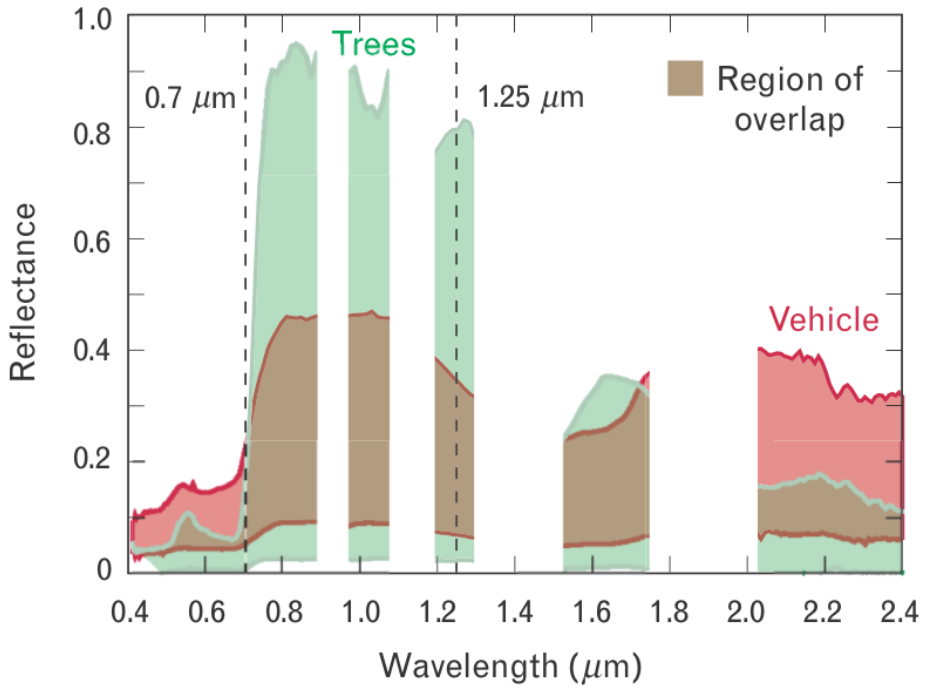

Representation of images: the feature space



- Pixels (or eventually patches) become points in a geometric feature space
- Axes have physical meaning, e.g. reflectances
- Relations between features reveal non-linear and non-Gaussian structures

Credits: Image from Manolakis02.

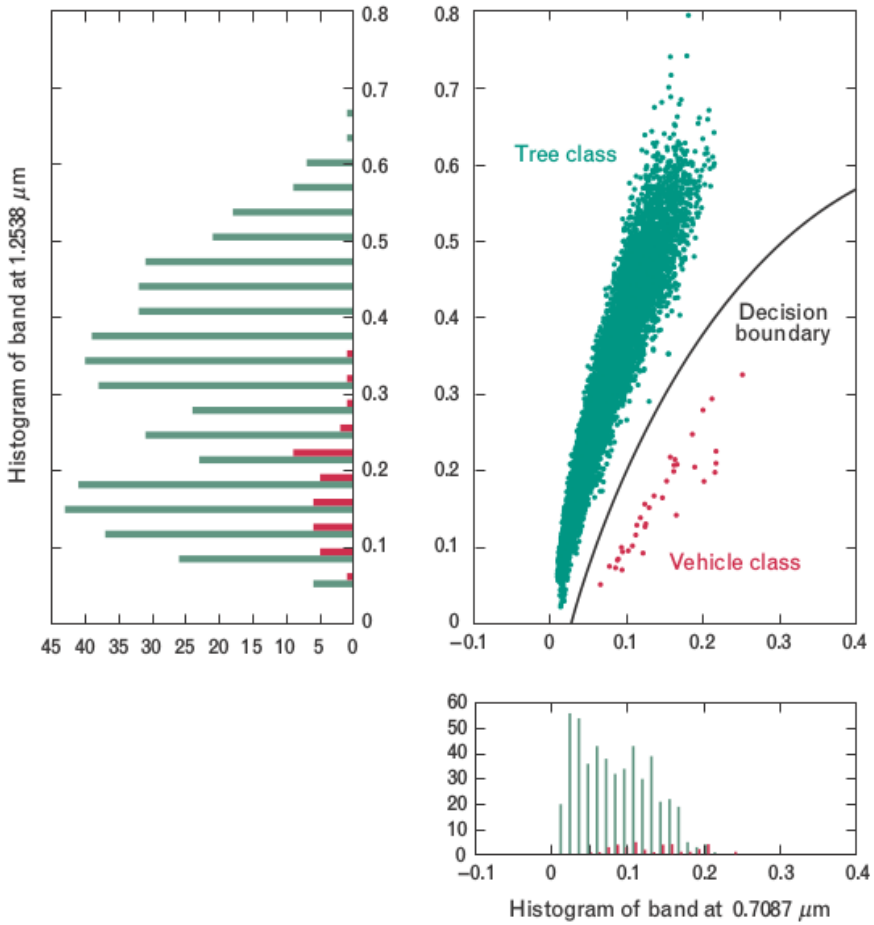
Spectral variability poses problems for discrimination:



- In overlapping spectral regions, discrimination is almost impossible with just a single band

Credits: Image from Manolakis02.

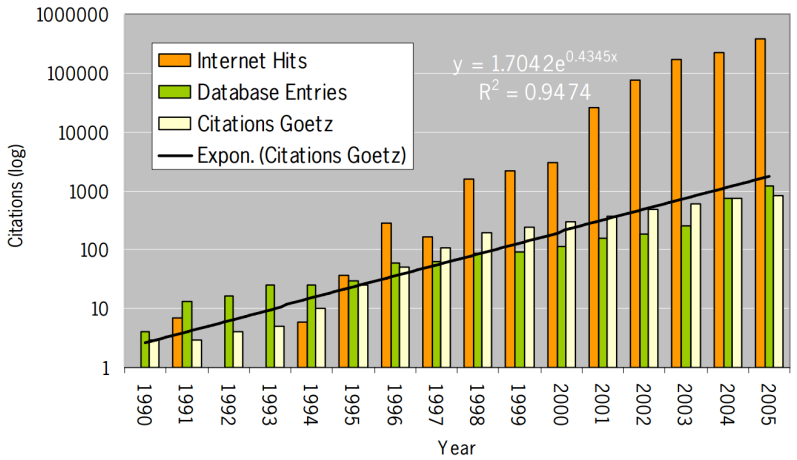
Combination of bands solves the problem:



- Simultaneous exploitation of the spectral bands at 0.7μm and 1.25μm makes discrimination possible
- More bands lead to linear separability theoretically

Credits: Image from Manolakis02.

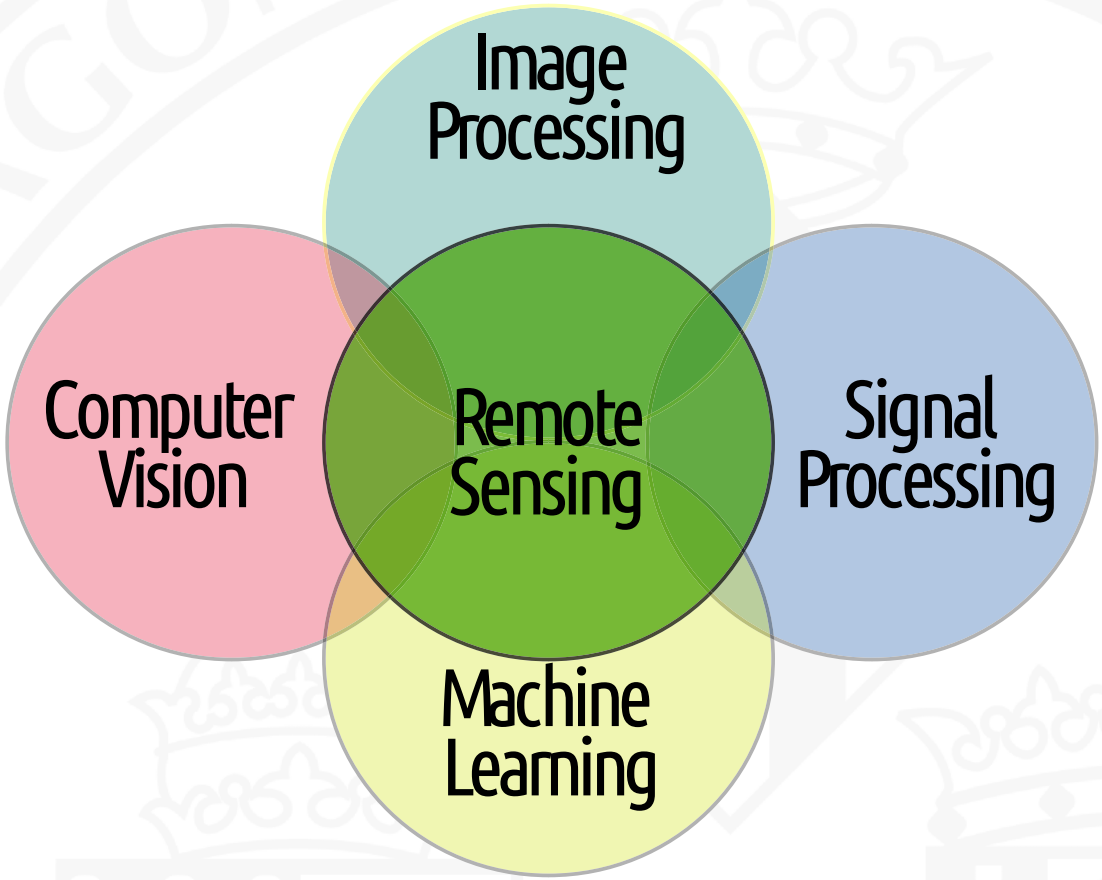
Hyperspectral imaging is an interdisciplinary, ever-growing field of Science:



- Hyperspectral images provide a unique source of information for many real-life applications:
  - Identify materials in the land cover
  - Update land cover and land use maps
  - Detect targets of interest (in both civilian and military applications)
  - Estimate the abundance and mixture of materials per pixel
  - Estimate biophysical parameters
- High dimensionality of data pose many processing problems
  - Curse of dimensionality: Few labeled samples in high dimensional spaces
  - Many high-dim unlabeled pixels: huge computational cost and redundancy issues
  - Ancillary information typically included: how? when? useful?

Credits: Image from Schaepman09 – 'Earth system science related imaging spectroscopy–An assessment'.

We will live at the intersection:



Part 2: Supervised hyperspectral image classification

Hyperspectral image classification is a challenging problem!

- **Philosophical problems:** infinite diversity of the Earth covers
  - What is a class? How many classes in the scene?
  - What is a forest? How many forest classes are there?
- **Methodological problems:**
  - High dimensionality of pixels and scarcity of labels
  - Hughes phenomenon, overfitting and generalization capabilities
- **Practical and operational problems:**
  - High cost for gathering labeled data (economic, time, resources)
  - Acquisition process and distortions in the images imply strong nonlinearities
  - Atmospheric and illumination effects may ruin the validation data
  - Heavy image preprocessing: geometric and atmospheric corrections
  - Need expert knowledge in pre- and postprocessing

Statistical classifiers have been readily applied to the problem:

Parametric

Assume a particular density distribution
LDA, GMM

Non-parametric

No assumption about the data distribution
k-NN, NNETS, TREES, SVM

Supervised

Need labeled input-output pairs
LDA, k-NN, TREES, SVM

Unsupervised

No need labels
k-means, EM-GMM, SOM

Semisupervised

Use both labeled and unlabeled data
Laplacian SVM, TSVM, graphs

One-class

Interest in detecting just one class
SAM, OSP, RX, OC-SVM

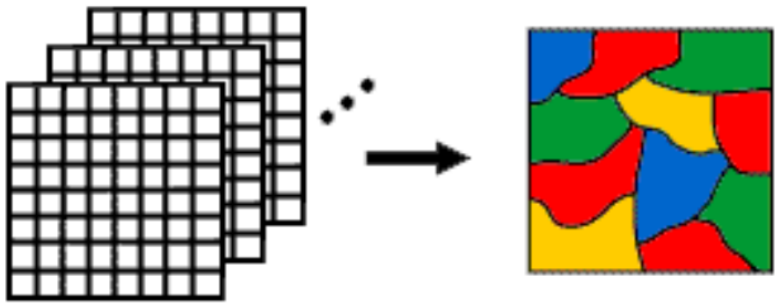
- Not too much success in parametric classifiers, as some assumptions break
- Currently, nonparametric classifiers and committees of experts excel!
- k-NN: good compromise between accuracy and computational cost
- Support vector machines (SVM) typically outperform the rest

Classifiers:

- Linear discriminant analysis (linear, quadratic, Mahalanobis)
- k-Nearest neighbors (KNN)
- Neural networks (NNETS)
- Support Vector Machines (SVM)

Analysis:

- Accuracy of classifiers (OA, Kappa, Confusion matrix)
- Robustness to dimensionality (apply before PCA?)
- Robustness to number of labeled samples
- Computational cost



Linear discriminant analysis (LDA): "Fits a Gaussian to each class data"

- Linear discriminant analysis ('linear'): Fit a multivariate Gaussian to each group/class through a joint covariance matrix

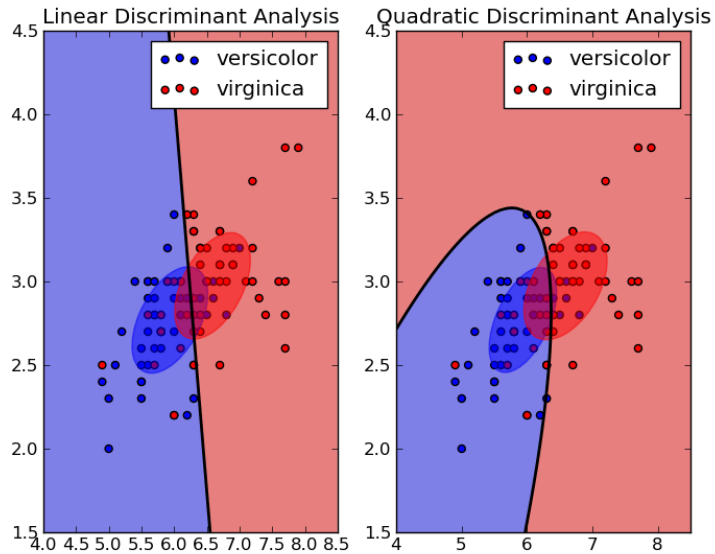
```
>> yp=classify(Xtest,Xtrain,Ytrain,'linear');
```

- Linear discriminant analysis ('quadratic'): Fit a multivariate Gaussian to each group/class through a class-dependent covariance matrix

```
>> yp=classify(Xtest,Xtrain,Ytrain,'quadratic');
```

- Linear discriminant analysis ('mahalanobis'): Fit a multivariate Gaussian to each group/class through a class-dependent Mahalanobis distance

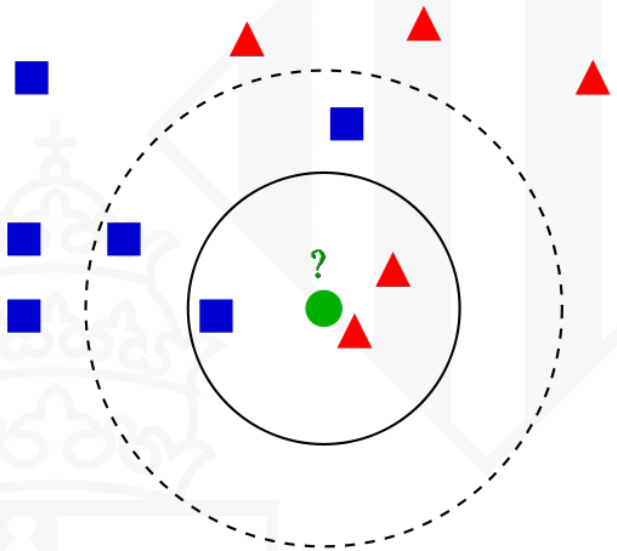
```
>> yp=classify(Xtest,Xtrain,Ytrain,'mahalanobis');
```



k nearest neightbor (k-NN): "is a non-parametric memory-based classifier that assigns the test label from the closest training point(s)"

- We can play around with the notion of distance (e.g. Euclidean, SAM, etc.)
- k-NN is a rather slow method with many samples and high k
- $k = 1$ use to work in real applications!

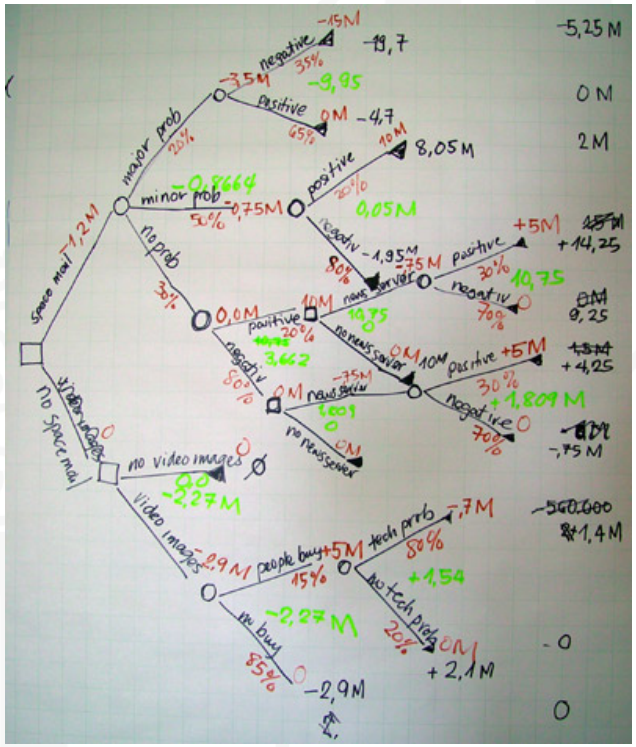
```
>> yp = knnclassify(Xtest,Xtrain,Ytrain,'euclidean');
```



Decision trees (TREES): "are non-parametric classifiers that adjust threshold values per feature in a hierarchical structure"

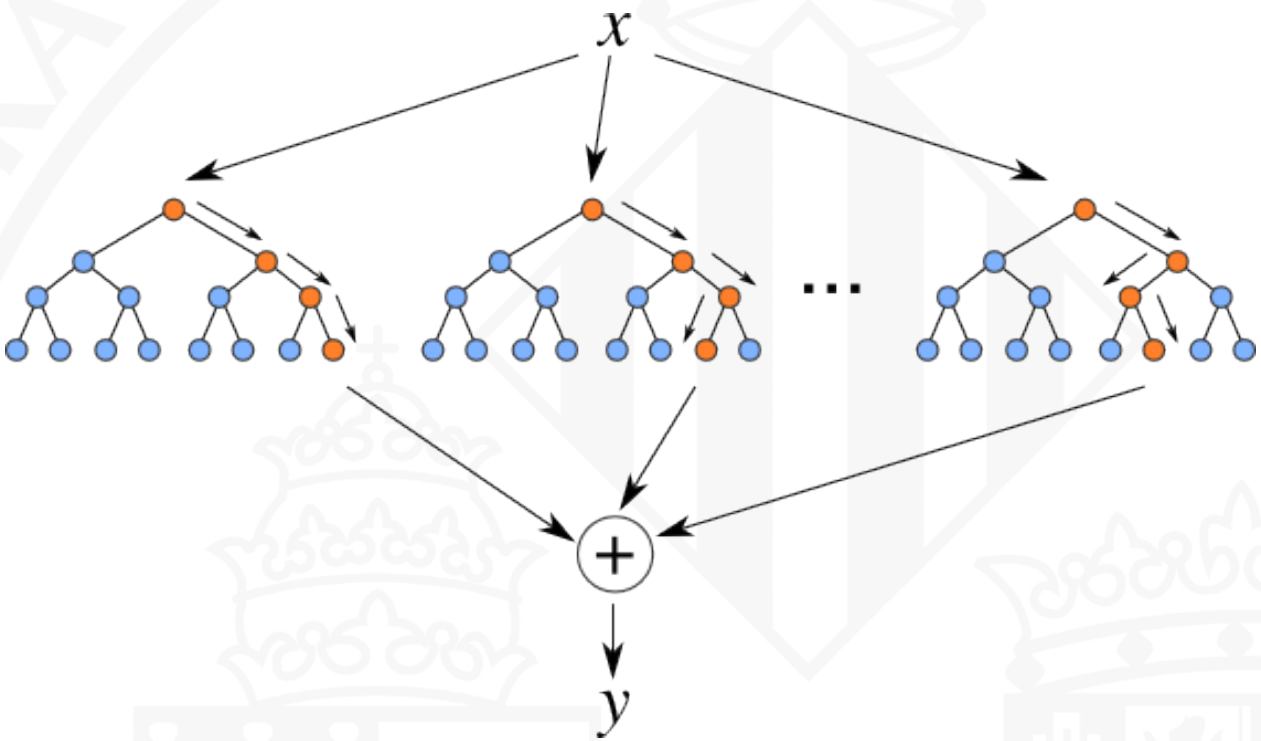
- TREES typically optimize the information transmitted from father to sons
- TREES are fast to learn/adjust and apply
- TREES allow to study the problem through trees visualization
- TREES are however limited to simple linear boundaries
- TREES provide a moderate success rate

```
>> tree = treefit(Xtrain,Ytrain,'method','classification');
>> ypred = treeval(tree,Xtest);
```



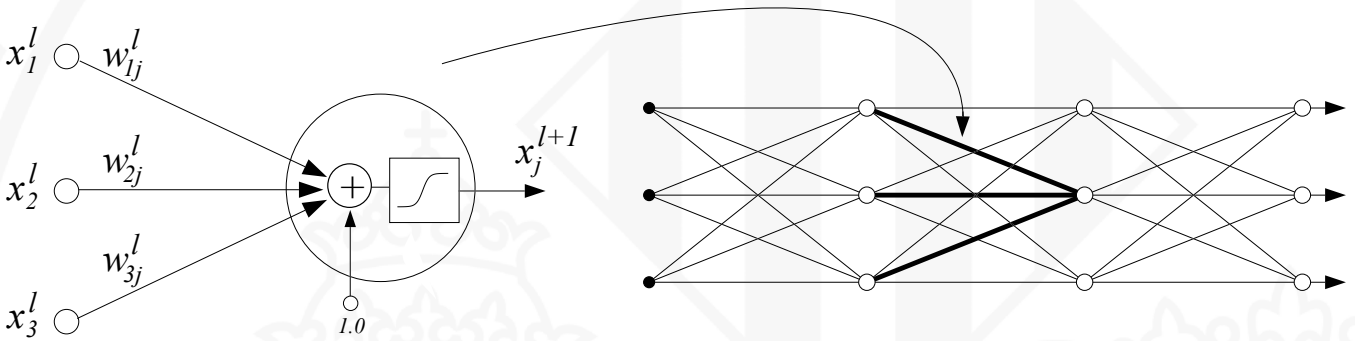
Random forests (RF): "evolution of TREES through the combination of trees learned from sets of randomly selected features"

```
>> ntrees = 200
>> forest = TreeBagger(ntrees,Xtrain,Ytrain);
>> Ypred = predict(forest,Xtest);
```



Neural networks (NNETS): "adjust a fully-connected nonlinear hierarchical structure made of simple neurons (point-wise nonlinearity) by minimizing the MSE in the output layer"

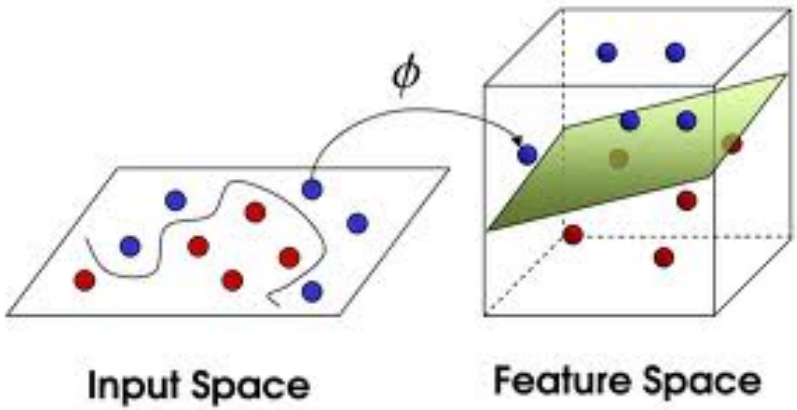
- Binary problems: binary coding of the output ($y \in \{0, 1\}$).
- Multiclassification: as many output neurons as classes



Support Vector Machines (SVM): "non-parametric kernel method that fits an optimal linear hyperplane separating the classes in a higher dimensional representation (feature) space"

- SVMs optimize two parameters: C to adjust the level of regularization (prevent overfitting) and the σ parameter of the RBF kernel (mapping space dimensionality)
- SVMs are fast to train and apply in moderate size problems
- SVMs are slow with many labeled examples
- SVMs generally outperform the rest in hyperspectral image classification

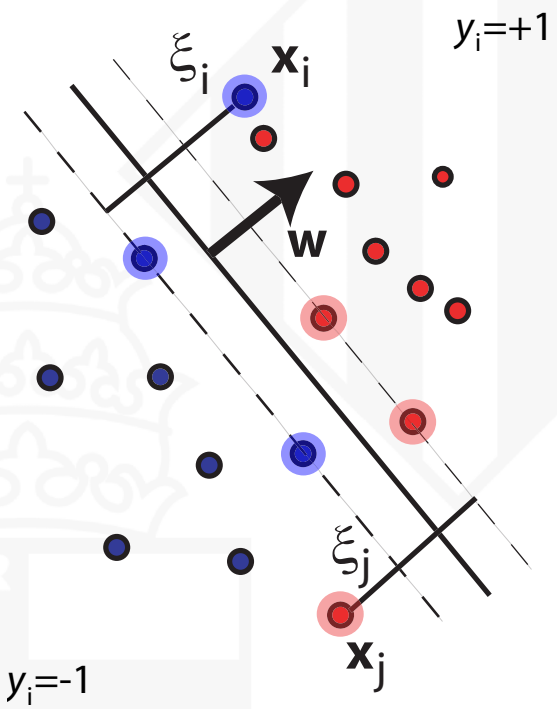
```
>> ypred = svm_classify(Xtest,X,Y);
```



- **The solution of the SVM:**

$$\hat{y}_j = f(\mathbf{x}_j) = sign(\mathbf{w}^\top \phi(\mathbf{x}_j) + b) = sign\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_j, \mathbf{x}_i) + b\right)$$

- **The solution is sparse:** only few examples $\mathbf{x}_i$ with $\alpha_i \neq 0$ are important
- **Support vectors:** those that define the margin and are misclassified examples



Valid kernels must be symmetric and positive definite similarity measures

- Linear:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$$

- Polynomial:

$$K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + 1)^d$$

- Gaussian Function (RBF):

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$$

- Hyperbolic Tangent:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \tanh(a(\mathbf{x}_i^\top \mathbf{x}_j) + b)$$

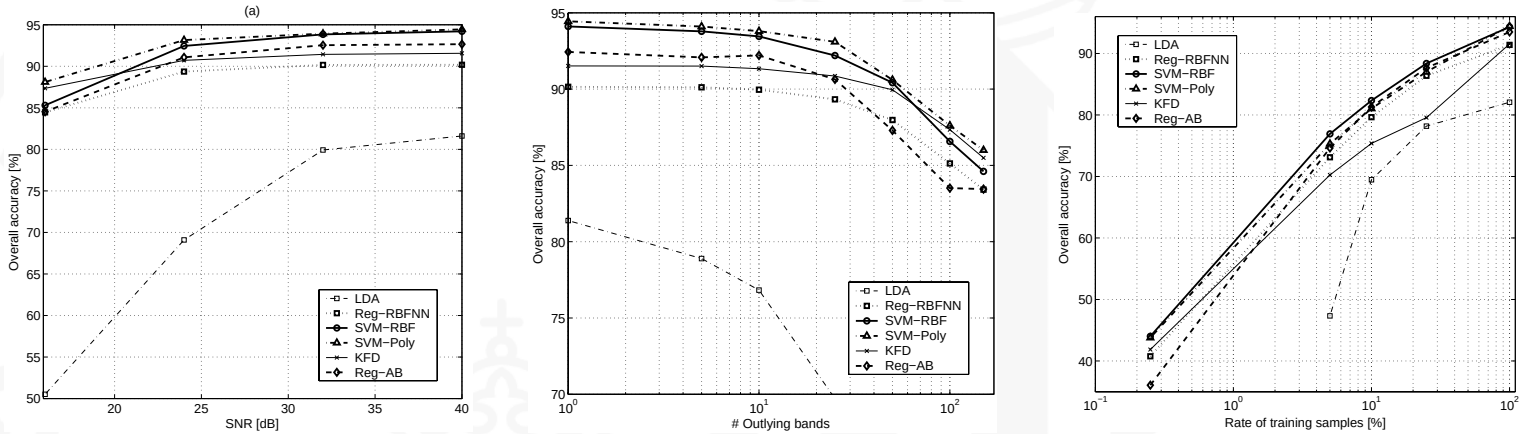
- Build new kernels...

$$K(\mathbf{x}_i, \mathbf{x}_j) = K_1(\mathbf{x}_i, \mathbf{x}_j) + K_2(\mathbf{x}_i, \mathbf{x}_j)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = K_1(\mathbf{x}_i, \mathbf{x}_j) \cdot K_2(\mathbf{x}_i, \mathbf{x}_j)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = \eta K_1(\mathbf{x}_i, \mathbf{x}_j), \quad \eta > 0$$

Example 1: Pixel-wise hyperspectral image classification

- Standard image: 9 crop classes, Indiana (USA), 1999.
- AVIRIS sensor: 220 bands, 145×145 pixels.
- *Only spectral information is considered at this point.*

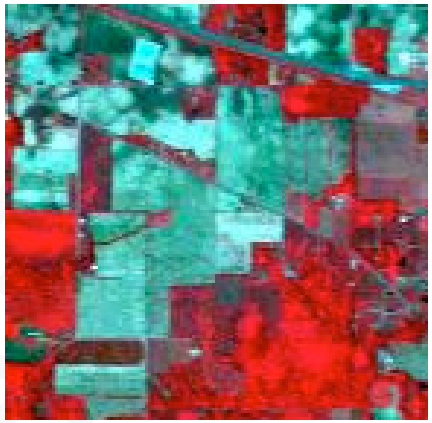
Accuracy and robustness



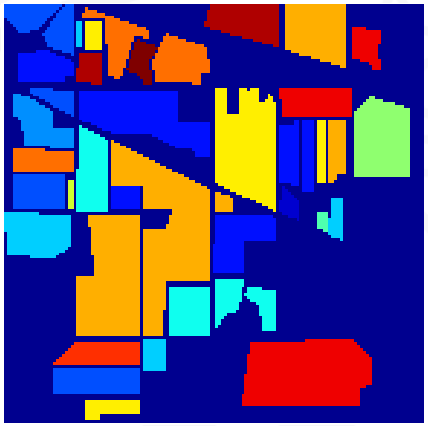
- Non-linear SVM (RBF kernel) yields the best results when compared to LDA and RBF neural nets.
- SVMs show an important gain when working with low number of samples and high dimension, high levels of input noise, and moderate computational cost

Visual inspection

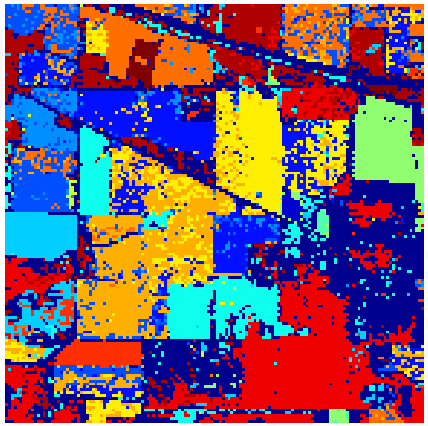
RGB



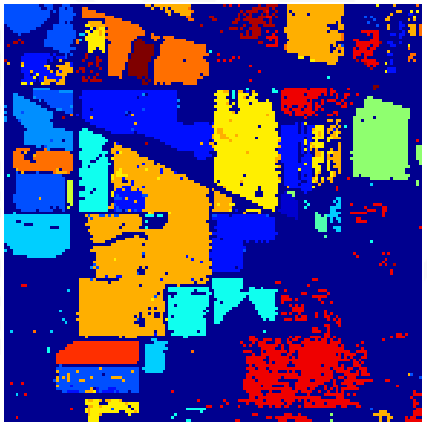
Ground truth



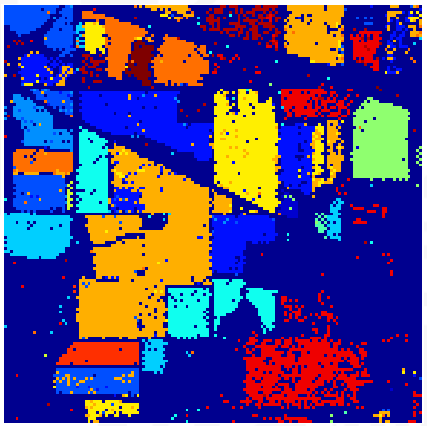
LDA (59.72%)



NNETS (83.43%)



SVM (88.27%)



Example 2: Spatial-spectral multispectral image classification

- Multispectral image: 9 crop classes, Zürich, 2002.
- Quickbird sensor: 4 bands + 22 spatial features (top/bottom hat).
- *Both spatial and spectral information is considered.*

Accuracy and robustness without contextual information:

Training pixels		OA [%]					Kappa				
		LDA	Trees	k-NN	SVM	MLP	LDA	Trees	k-NN	SVM	MLP
115	μ	60.43	68.62	68.43	**74.99**	72.94	0.53	0.61	0.61	**0.69**	0.67
	σ	(5.13)	(3.85)	(1.63)	(2.25)	(1.55)	(0.06)	(0.05)	(0.02)	(0.03)	(0.02)
255	μ	60.19	71.25	73.65	**77.31**	76.32	0.53	0.64	0.67	**0.72**	0.71
	σ	(3.25)	(1.79)	(3.79)	(1.23)	(1.20)	(0.03)	(0.02)	(0.05)	(0.02)	(0.02)
1155	μ	62.82	76.78	**80.92**	79.49	79.41	0.56	0.71	**0.76**	0.74	0.74
	σ	(2.08)	(0.90)	(0.47)	(0.73)	(0.38)	(0.02)	(0.01)	(0.01)	(0.01)	(0.01)
2568	μ	62.68	78.59	**81.38**	80.42	79.42	0.56	0.74	**0.77**	0.76	0.74
	σ	(1.94)	(0.32)	(0.24)	(0.34)	(1.09)	(0.02)	(0.01)	(0.01)	(0.01)	(0.01)

- Nonparametric methods (SVMs, MLP) excel
- Lazy learner k-NN shows good performance with enough samples
- Poor performance of linear parametric classifiers as the LDA

Example 2: Spatial-spectral multispectral image classification

- Multispectral image: 9 crop classes, Zürich, 2002.
- Quickbird sensor: 4 bands + 22 spatial features (top/bottom hat).
- *Both spatial and spectral information is considered.*

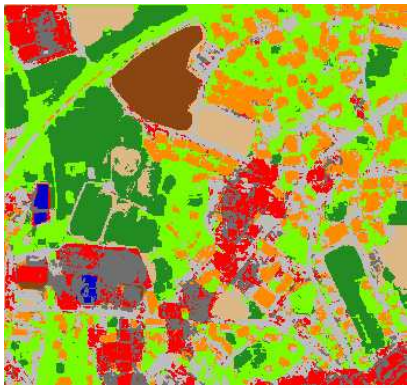
Accuracy and robustness with contextual information:

Training pixels		OA [%]					Kappa				
		LDA	Trees	k-NN	SVM	MLP	LDA	Trees	k-NN	SVM	MLP
115	μ	72.93	71.00	75.69	**83.37**	77.37	0.67	0.65	0.70	**0.80**	0.72
	σ	(2.85)	(2.97)	(1.28)	(2.40)	(2.48)	(0.03)	(0.03)	(0.02)	(0.03)	(0.03)
255	μ	77.23	73.47	80.53	**85.91**	80.61	0.72	0.68	0.76	**0.83**	0.76
	σ	(1.41)	(1.64)	(1.34)	(1.94)	(0.99)	(0.02)	(0.02)	(0.02)	(0.02)	(0.01)
1155	μ	78.35	80.45	87.32	**88.03**	84.29	0.74	0.76	0.84	**0.85**	0.81
	σ	(0.69)	(0.73)	(0.63)	(1.68)	(1.77)	(0.01)	(0.01)	(0.01)	(0.02)	(0.02)
2568	μ	78.61	81.59	**87.26**	87.17	85.10	0.74	0.77	**0.84**	**0.84**	0.82
	σ	(0.57)	(0.89)	(0.61)	(0.85)	(1.05)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)

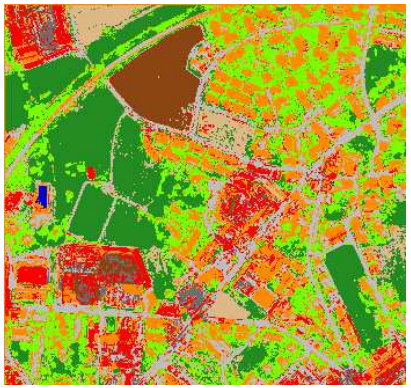
- Contextual information is beneficial for all models, +5% and 10%
- Contextual information improves SVM and NN much more
- Without spatial features, k-NN is the best option!

Ground survey

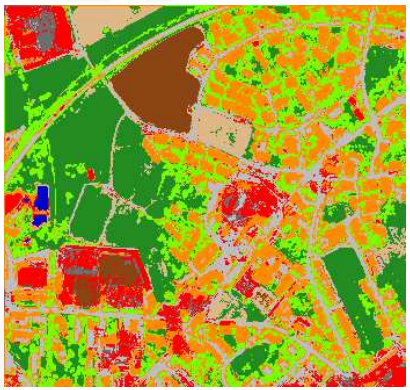
LDA (78.35, 0.74)



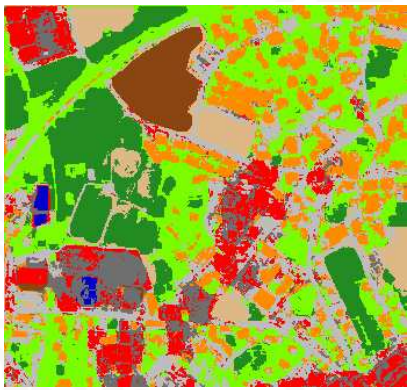
Class. tree (80.45, 0.76)



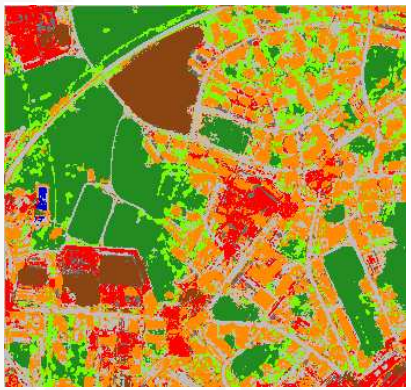
k-NN (87.32, 0.84)



SVM (88.03, 0.85)



MLP (84.29, 0.81)



- SVM and *k*-NN return detect all major structures of the image
- McNemar's test confirmed visual estimation of the quality
- SVM map is significantly better than the others, followed by the *k*-NN and NN maps

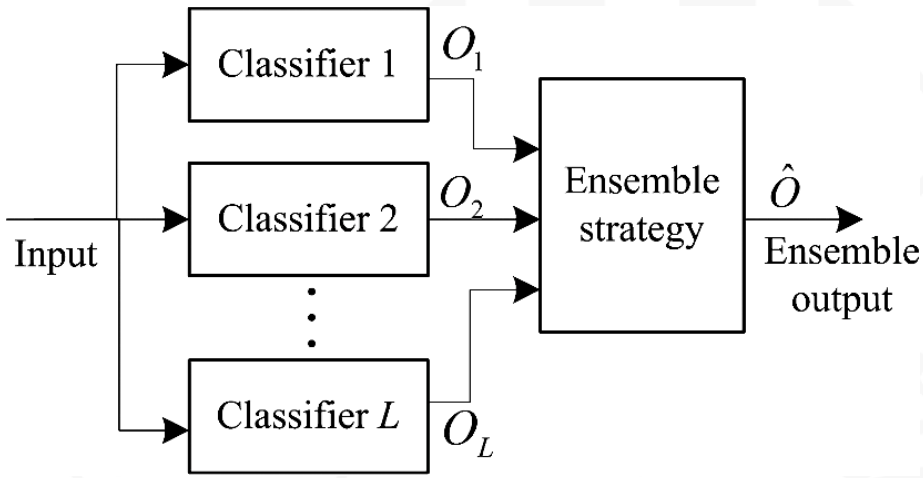
Hyperspectral image classification needs strong regularization:

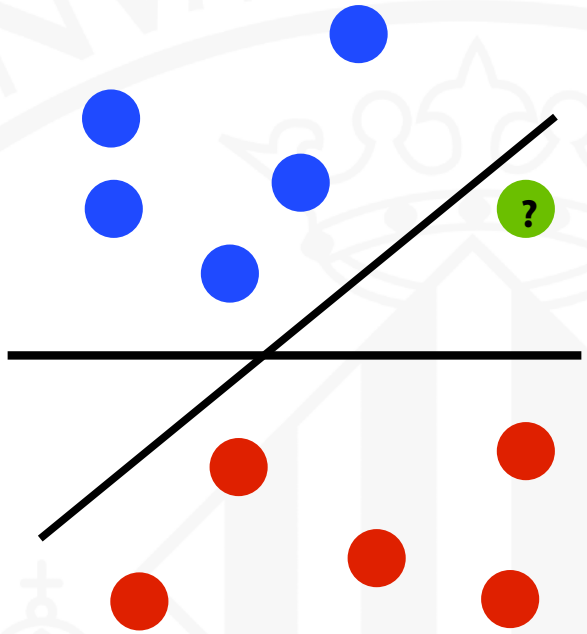
- SVM imposes regularization naturally by maximum margin
- Advanced classification focuses on other forms of regularization:
  - Reduce dimensionality via feature selection and extraction
  - Include information contained in unlabeled samples
  - Include synthetically generated data encodes invariance properties
  - Impose spatial homogeneity of images: include spatial information
  - Include multisource data: SAR, LiDAR
  - Include ancillary information from expert's knowledge (VIs, ecosystems maps, climate regions, etc)

Model combination

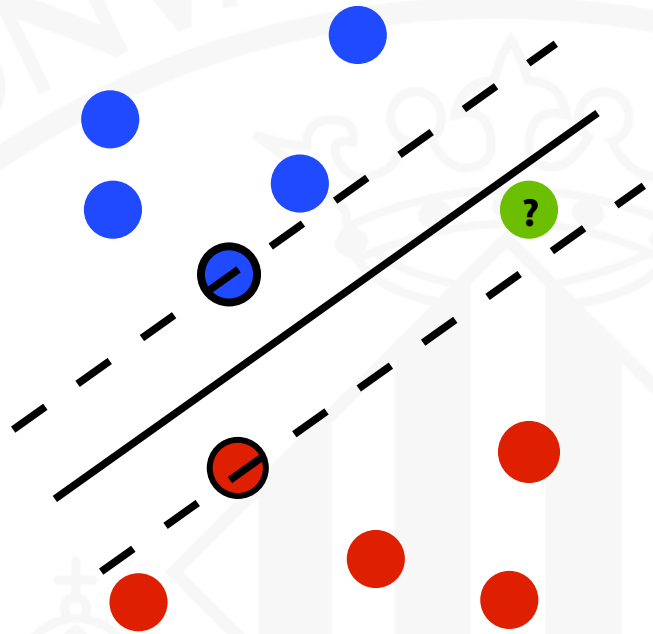
Proposal: ensemble of classifiers

- Max-vote
- Mode
- Asssign weights according to accuracy

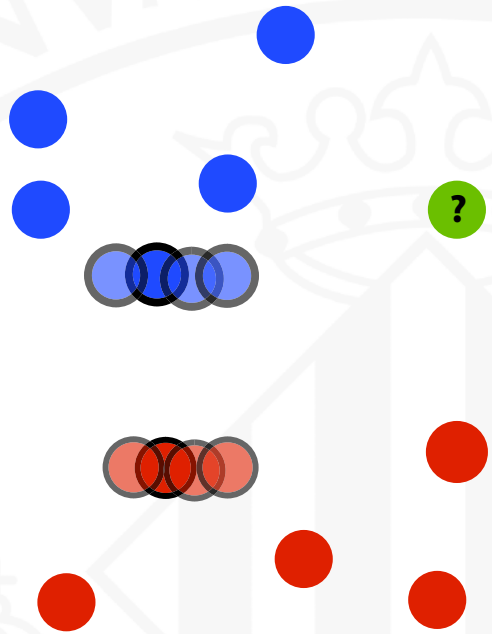




- The example assumes invariance to horizontal transformations
- Given the training data, the point ? is hard to classify
- Modify the SVM to incorporate prior knowledge

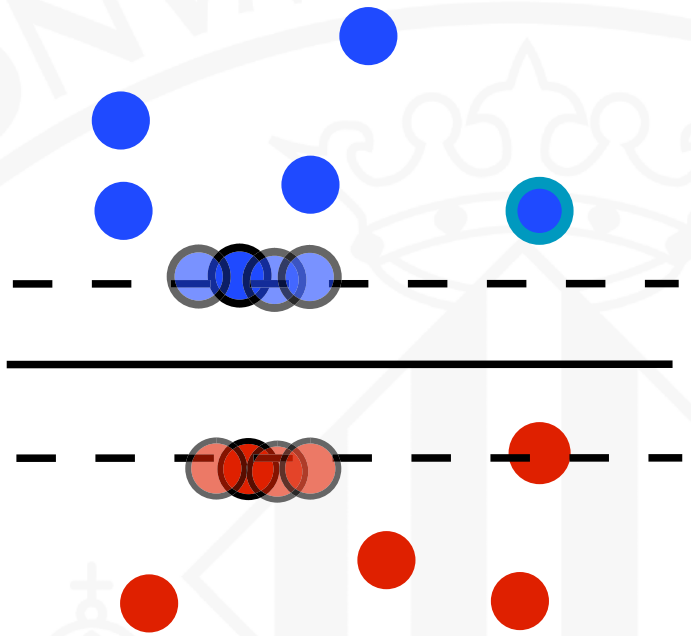


Step 1 Train a SVM and find the SVs



Step 1 Train a SVM and find the SVs

Step 2 VSVs: perturbate SVs to which the solution should be invariant



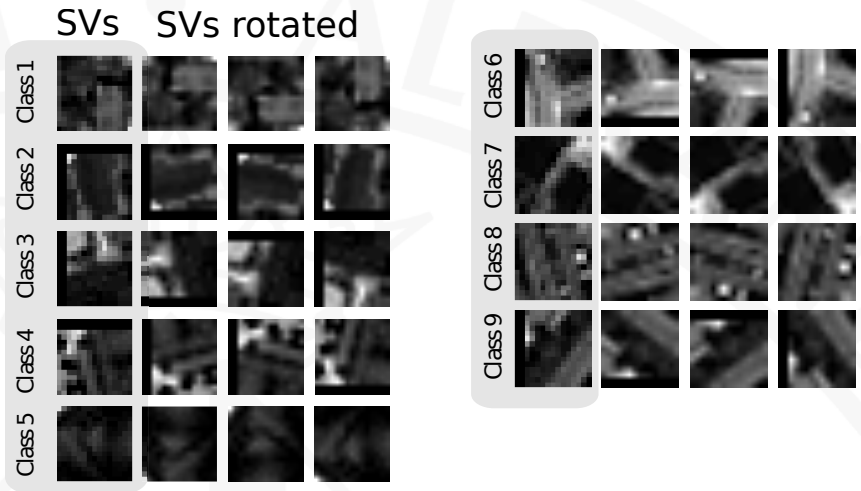
Step 1 Train a SVM and find the SVs

Step 2 VSVs: perturbate SVs to which the solution should be invariant

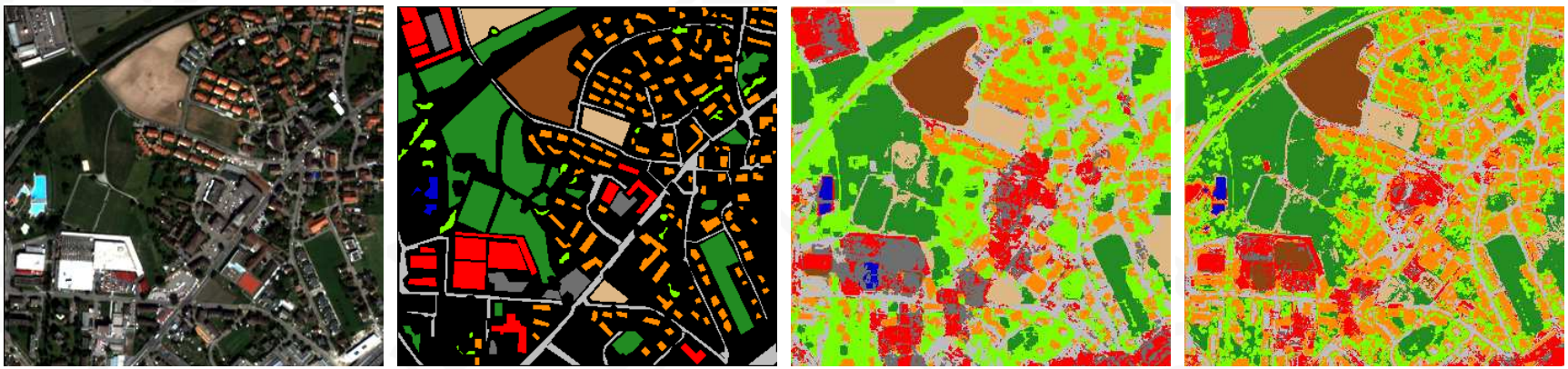
Step 3 Train a SVM with both SVs and VSVs

Example 1: encoding invariance to rotations:

- Quickbird image + 18 spatial features
- Size: 329×347 pixels
- 9 classes
- VSVM encodes invariance to rotation!



RGB | GT | SVM (76.14, 0.73) | **VSVM (83.15, 0.80)**

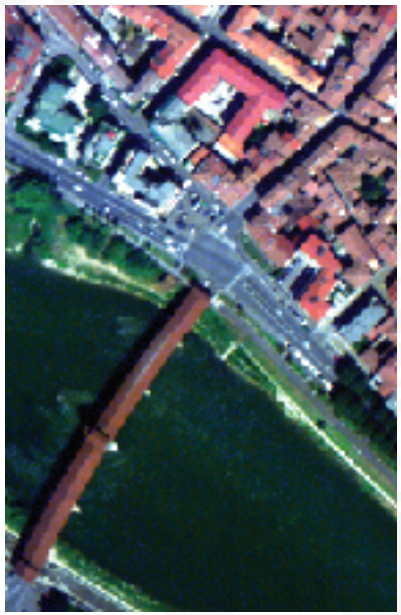


- Both classifiers show high classification scores
- VSVM improves classification score over +7%
- VSVM is however more computationally demanding

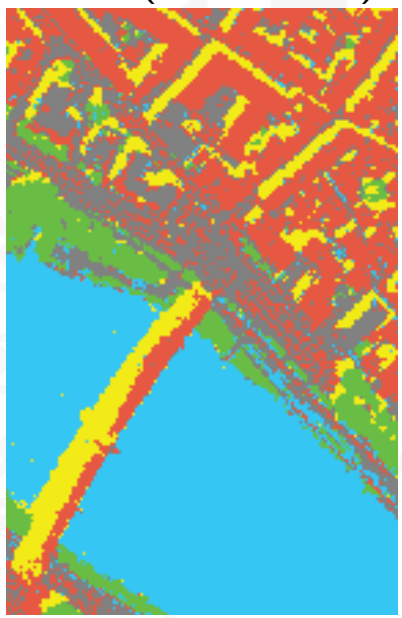
Example 2: encoding invariance to shadows and illumination changes:

- Multispectral image acquired by DAIS7915 over Pavia (Italy) [CampsValls11]
  - 9-class urban classification problem
  - Dominated by directional features and relatively high spatial resolution
  - Presence of shadows in the streets and the bridge
  - 50 training spatial-spectral samples only (4×4 patches)
- Invariance coding: exponential-decay function in [0.5, 1.76]μm [Yamazaki09]

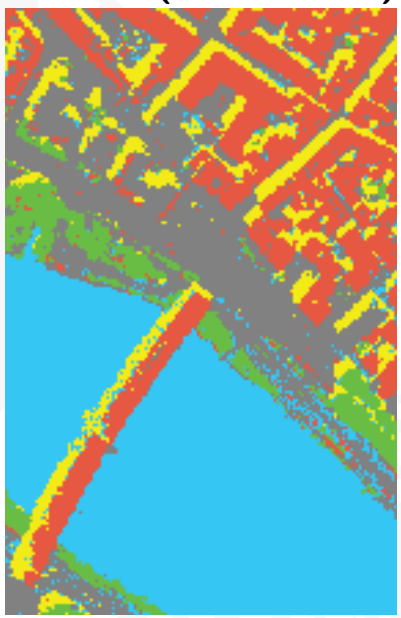
RGB



SVM (0.79±0.11)

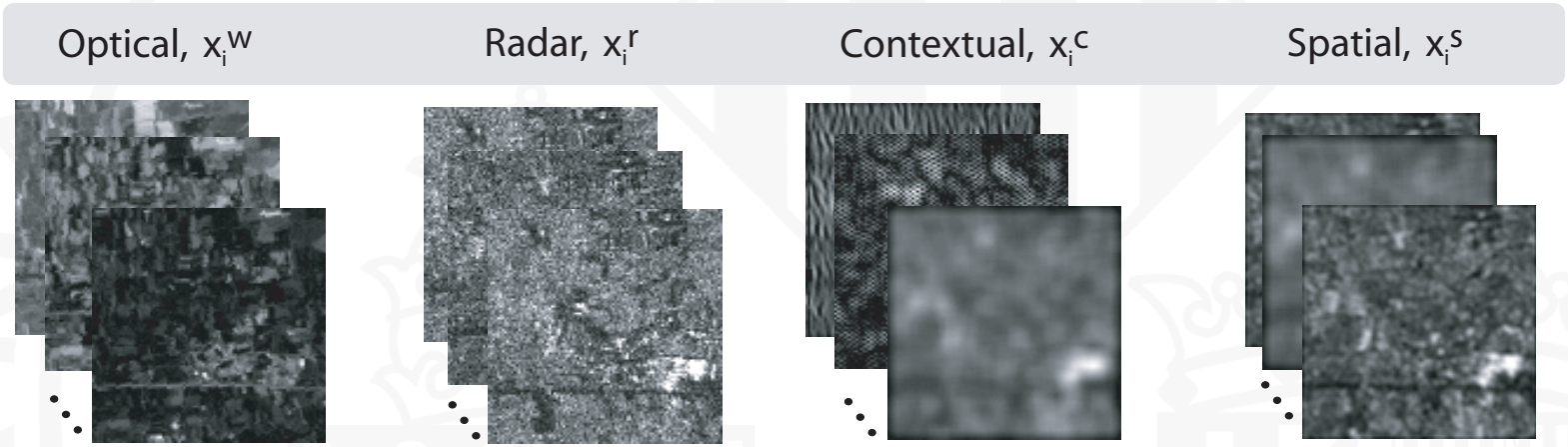


VSVM (0.84±0.09)



How to integrate multi-source information?

- Spatial features
- Textural features
- Time-varying features
- Multi-sensor features
- Multi-angular features



Taxonomy of spatial-spectral classification approaches:

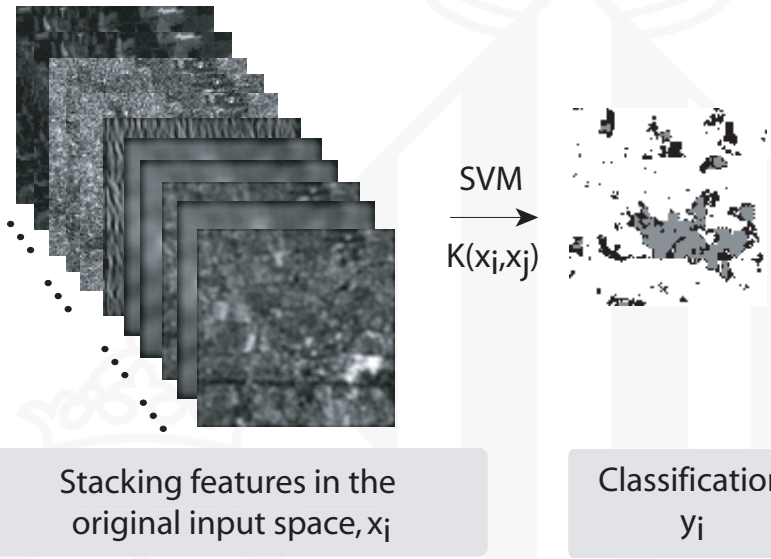
Type of Approach	Model	Idea
Spatial filters extraction	Co-occurrence	Extract texture based on statistics of pairs of pixels in a neighborhood
	EMP	Multiscale mathematical morphology (based on size)
	EMAP	Multiscale mathematical morphology (variety of attribute types)
Spatial-spectral segmentation	Segmentation and classification based on majority voting	All pixels are assigned to the most frequent class inside a segmented region
	Segmentation and classification based on markers	Most reliably classified pixels are selected as "region markers" for segmentation
	Semi-supervised hierarchical clustering tree	Returns both classification and confidence maps. Active learning used to select informative samples.
Advanced spatial-spectral classification	Composite and multiple kernels	Balances between spatial and spectral information with dedicated kernels
	Graph kernels	Takes into account higher order relations in each pixel neighborhood
	MRF	Markov Random Field Modeling (probabilistic)

Stacked approach

- Stacking features that characterize a pixel:

$$\mathbf{x}_i \leftarrow [\mathbf{x}_i^{\omega}, \mathbf{x}_i^{c}, \mathbf{x}_i^{r}, \mathbf{x}_i^{\rho}, \mathbf{x}_i^{s}, \mathbf{x}_i^{t}, ...]$$

- Compute matrix K and solve an SVM with the new samples $\mathbf{x}_i$.

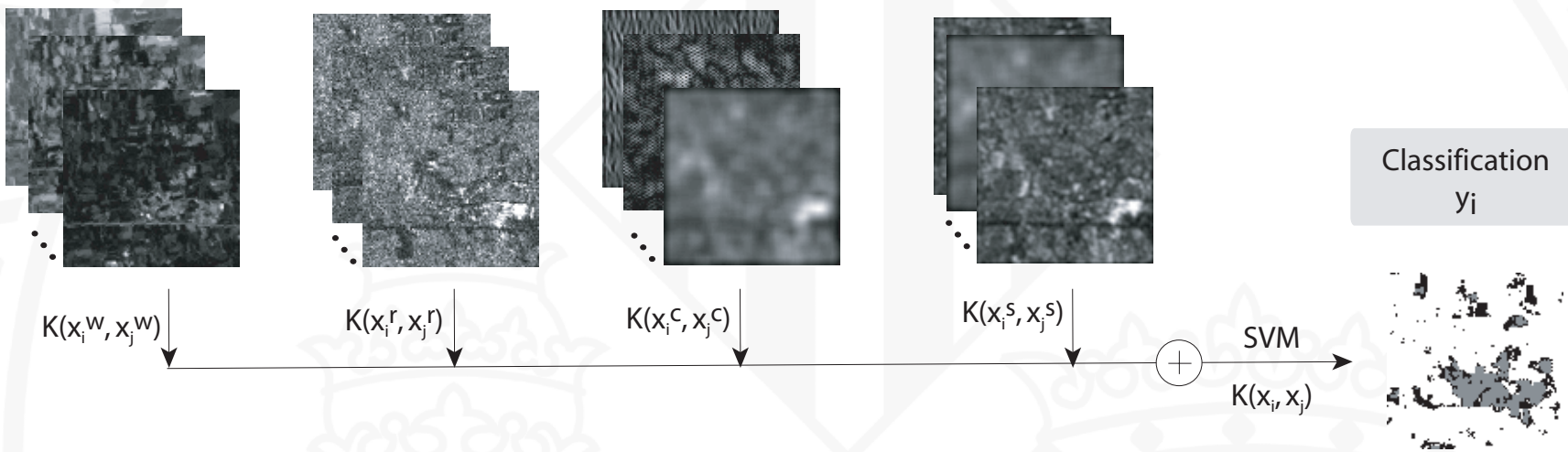


- **Problems:**
  1. Dimensionality of the samples is increased extraordinarily!
  2. Cross-relationships among features are not taken into account.
  3. This would be impractical for neural networks, for example.

Kernel-based spatial-spectral HSI classification

- Some properties of kernel methods (and SVM):

$$K(\mathbf{x}_i, \mathbf{x}_j) = K_1(\mathbf{x}_i, \mathbf{x}_j) + K_2(\mathbf{x}_i, \mathbf{x}_j)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = K_1(\mathbf{x}_i, \mathbf{x}_j) \cdot K_2(\mathbf{x}_i, \mathbf{x}_j)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = \eta K_1(\mathbf{x}_i, \mathbf{x}_j), \quad \eta > 0$$



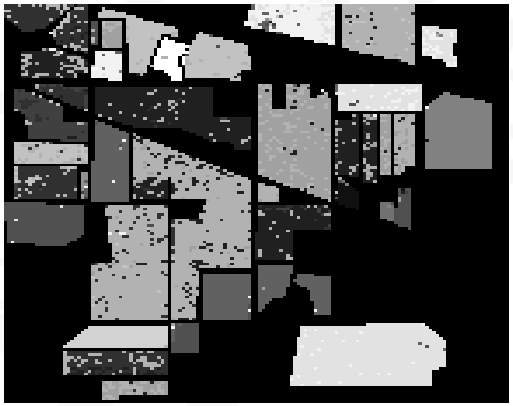
Stacking features in the kernel space implies direct sum of kernels

	Overall accuracy [%]	κ statistic
Spectral classifiers		
Euclidean [Tadjudin98]	48.23	—
bLOOC+DAFE+ECHO [Tadjudin98]	82.91	—
K_ω [CampsValls04]	88.55	0.87
Spatio-spectral classifiers [CampsValls06]		
Mean		
K_s	84.55	0.82
$K_{\{s,\omega\}}$	94.21	0.93
$K_s + K_\omega$	92.61	0.91
$\mu K_s + (1-\mu)K_\omega$	**95.97**	**0.94**
$K_s + K_\omega + K_{s\omega} + K_{\omega s}$	94.80	0.94
Mean and variance		
K_s	88.00	0.86
$K_{\{s,\omega\}}$	94.21	0.93
$K_s + K_\omega$	95.45	0.95
$\mu K_s + (1-\mu)K_\omega$	**96.53**	**0.96**

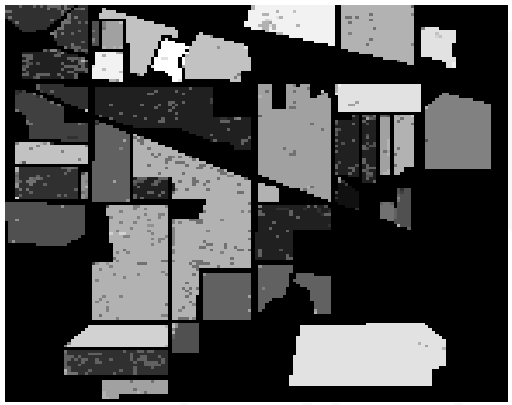
- Linear methods offer poor results
- The proposed classifiers improve results in all cases (+[5-11]%)
- Simplest kernel combinations yield very good results

Ground truth

Spatial (84.55%)



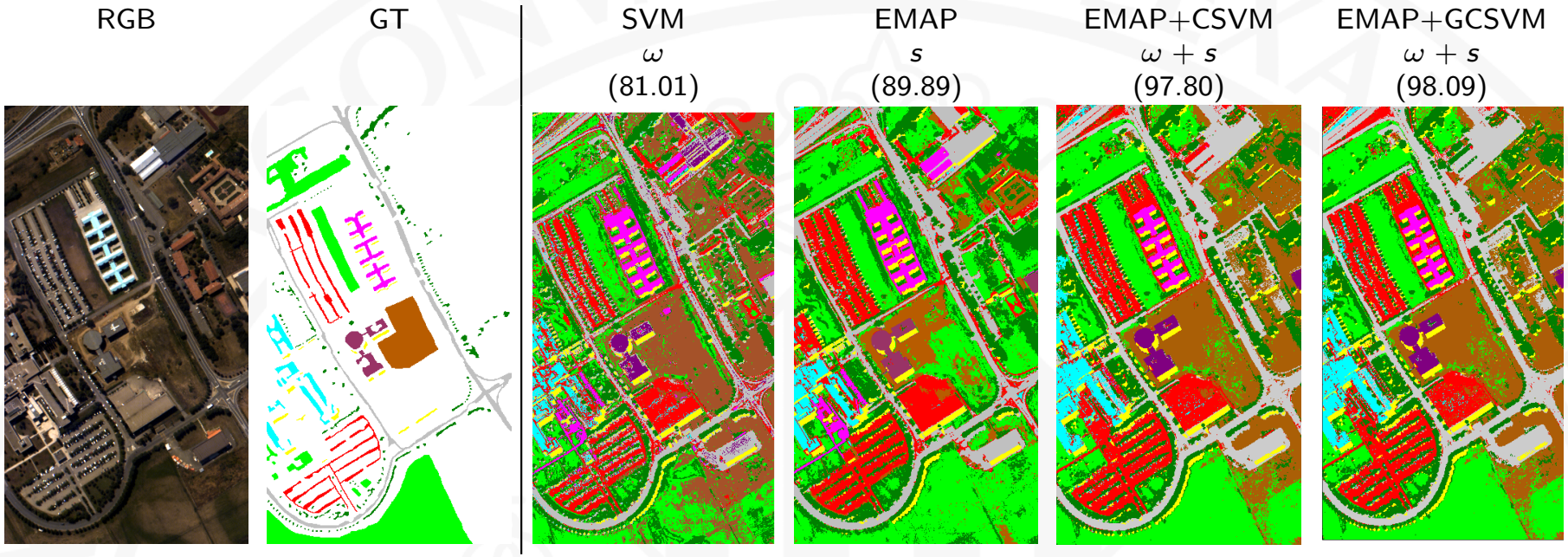
Spectral (88.55%)



Spatio-spectral (95.53%)

- More homogeneous classification maps
- State of the art results
- Easy framework for multisource data fusion

Combine advanced spatial features and composite SVM



- ROSIS-03 Pavia University area data set (103 spectral channels and spatial resolution 1.3m), 9 classes
- Spatial components:

  Benediktson11 Extended Morphological Profiles (EMP)
  CampsValls06 Cross-kernels composite SVM (CSVM)
  Li13 Generalized composite kernels (GCSVM)

Multi-sensor fusion kernels

1. Idea: Build dedicated kernels for the optical ($\mathbf{x}_i^o$), and the radar ($\mathbf{x}_i^r$) feature samples, and combine them in the kernel.
2. Three formulations:
   - *The stacked features approach:*

$$\mathbf{x}_i = [\mathbf{x}_i^o, \mathbf{x}_i^r], \qquad K_{\{o,r\}} \equiv K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

   - *The direct summation kernel:*

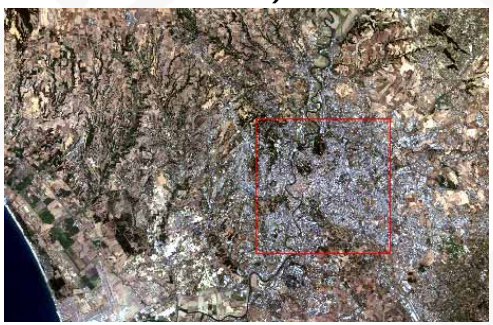
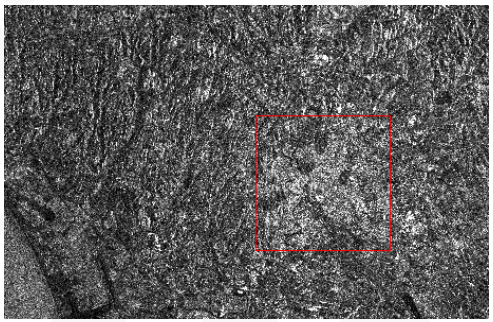
$$K(\mathbf{x}_i, \mathbf{x}_j) = K_o(\mathbf{x}_i^o, \mathbf{x}_j^o) + K_r(\mathbf{x}_i^r, \mathbf{x}_j^r)$$

   - *The cross-information kernel:*

$$K(\mathbf{x}_i, \mathbf{x}_j) = K_o(\mathbf{x}_i^o, \mathbf{x}_j^o) + K_r(\mathbf{x}_i^r, \mathbf{x}_j^r) + K_{or}(\mathbf{x}_i^o, \mathbf{x}_j^r) + K_{ro}(\mathbf{x}_i^r, \mathbf{x}_j^o)$$

Example: Detection of classes 'urban' vs. 'non-urban' [Camps-Valls08]

- 'Urban Expansion Monitoring (UrbEx) ESA-ESRIN DUP' Project
- 2 sensors (ERS2 SAR y Landsat TM)
- 2 dates (1995 and 1999) over Rome

Features and pre-processing

1. Images were co-registered with ISTAT data (at subpixel level, <15m res.)
2. SAR images were filtered for 'speckle'.
3. Original features: 7 spectral bands, 2 backscattering intensities plus coherence.
4. Additionally: (i) optical features are mean-filtered, and (ii) SAR images are Gabor-filtered, at different scales ($\theta = 1, \ldots, 4$) and orientations ({0Â°, 45Â°, 90Â°, 135Â°})

Accuracy and flexibility

- Different kernel-based methods integrating spectral, contextual, textural and temporal information.
- Different levels of complexity and versatility.
- Linear and non-linear (RBF) kernels.

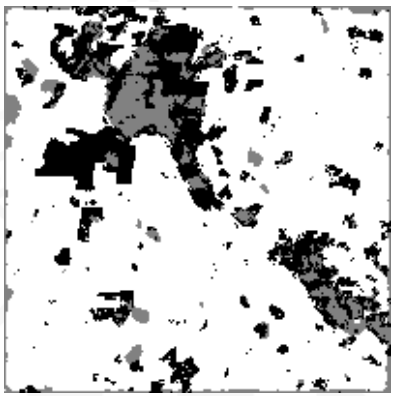
	Spatio spectral	**Multi-sensor**	**Temporal** Sum	Crossed	Weighted
SVM (LIN)	Sum	Standard	83.2 (0.45)	68.2 (0.61)	70.4 (0.64)
	Crossed	Standard	81.4 (0.49)	69.2 (0.62)	71.4 (0.63)
	Sum	Sum	**84.1 (0.51)**	70.2 (0.63)	**73.4 (0.72)**
SVM (RBF)	Sum	Standard	91.4 (0.67)	83.1 (0.70)	89.5 (0.78)
	Crossed	Standard	92.1 (0.69)	89.2 (0.71)	88.8 (0.77)
	Sum	Sum	**93.2 (0.77)**	**94.3 (0.78)**	**93.3 (0.81)**

- All the *proposed temporal kernels* improve the results of (1) considering the spectral info alone, (2) or even the spatio-spectral information.
- The *weighted summation kernel* is the best choice.
- In all cases, *non-linear RBF kernels* yield better results.

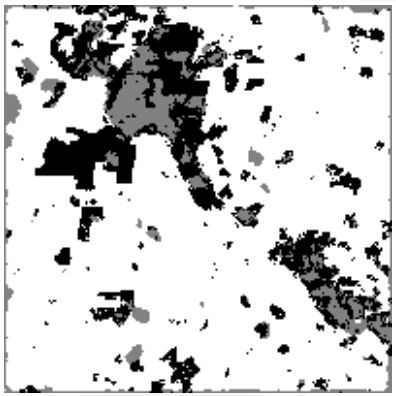
Visual inspection

Ground truth, 1999

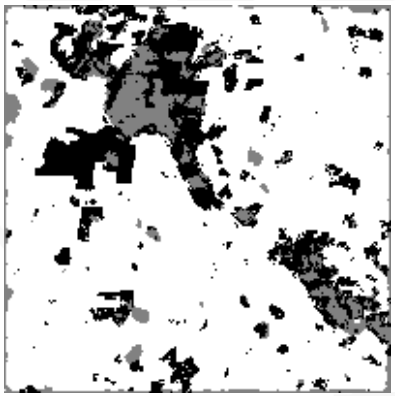
Sum (0.77)



Crossed (0.78)



Weighted (0.81)



non-urban **urban** **unknown**

What's LiDAR?

- **LI**ght **D**etection **A**nd **R**anging
- Active Sensing System
- Day or Night operation.
- Ranging of the reflecting object based on time difference between emission and reflection.

What's NOT LiDAR?

- NOT Light/Laser Assisted RADAR

  –RADAR uses electro-magnetic (EM) energy in the radio frequency range; LIDAR does not.

- NOT all-weather

  – The target MUST be visible. Some haze is manageable, but fog is not

- NOT able to 'see through' trees

  – LIDAR sees around trees, not through them. Fully closed canopies (rain forests) cannot be penetrated

- NOT a Substitute for Photography

  – For MOST users, LIDAR intensity images are NOT viable replacements for conventional or digital imagery

Credits: Jiunn-Der (Geoffrey) Duh, Portland, USA

LiDAR Characteristics

- Vertical accuracy for commercial applications at 15 cm on discrete points
- Collects millions of elevation points per hour
- Produces datasets with much greater density than traditional mapping
- Some systems capable of capturing multiple returns per pulse and/or intensity images

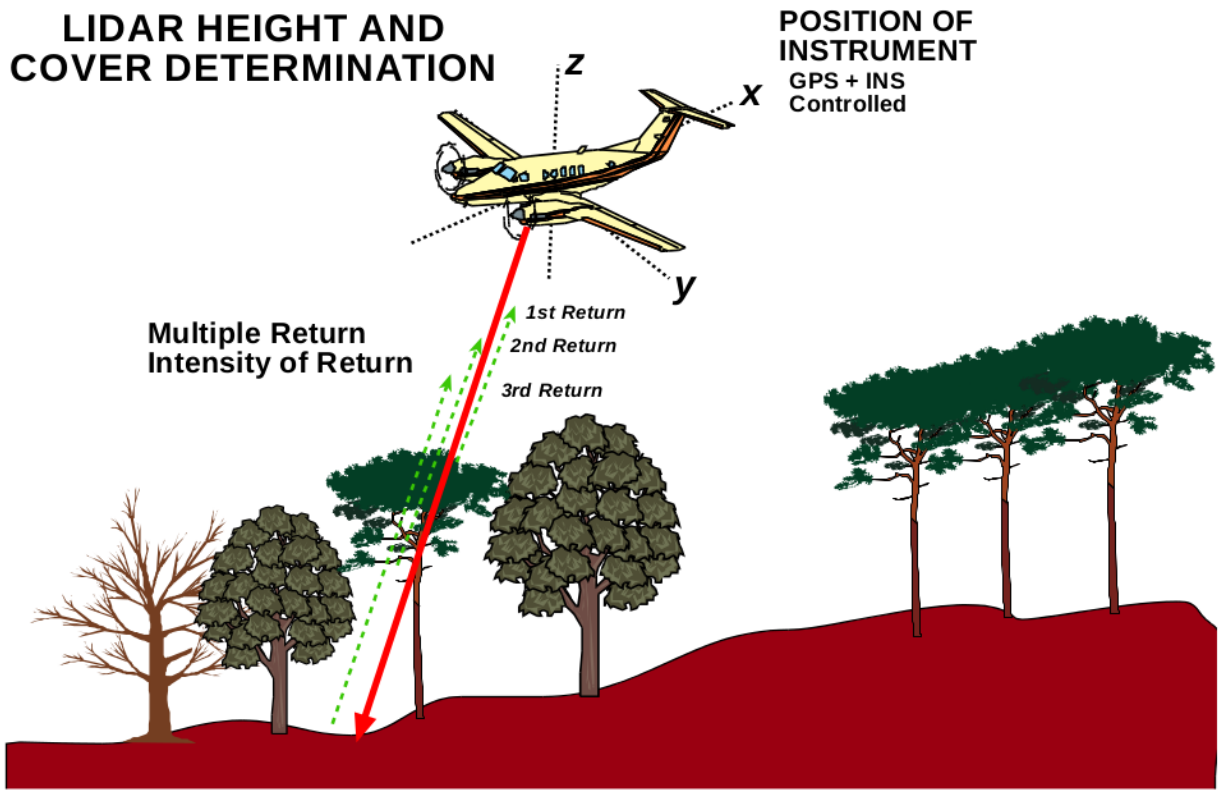
LiDAR Operational Theory

- A pulse of light is emitted and the precise time is recorded
- The reflection of that pulse is detected and the precise time is recorded
- Using the constant speed of light, the delay can be converted into a "slant range" distance.
- Knowing the position and orientation of the sensor, the XYZ coordinate of the reflective surface can be calculated

Credits: Jiunn-Der (Geoffrey) Duh, Portland, USA

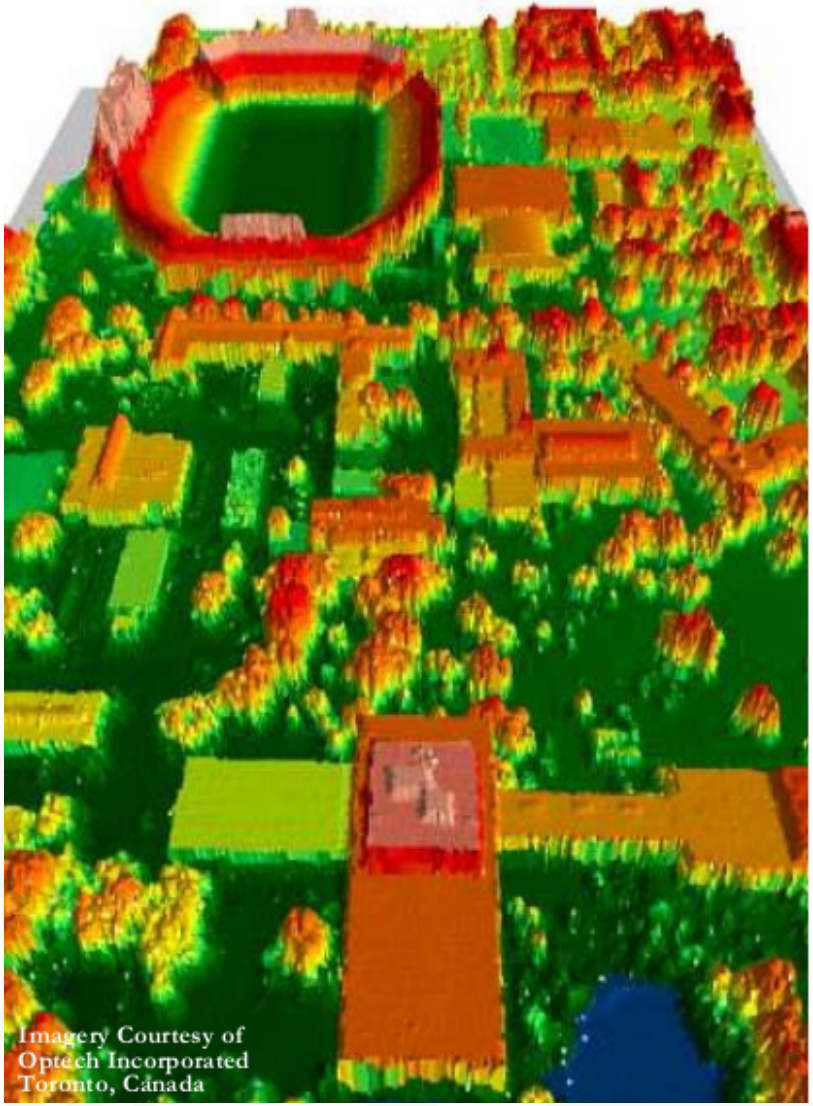
Multiple-Returns vs. Single-Return Systems

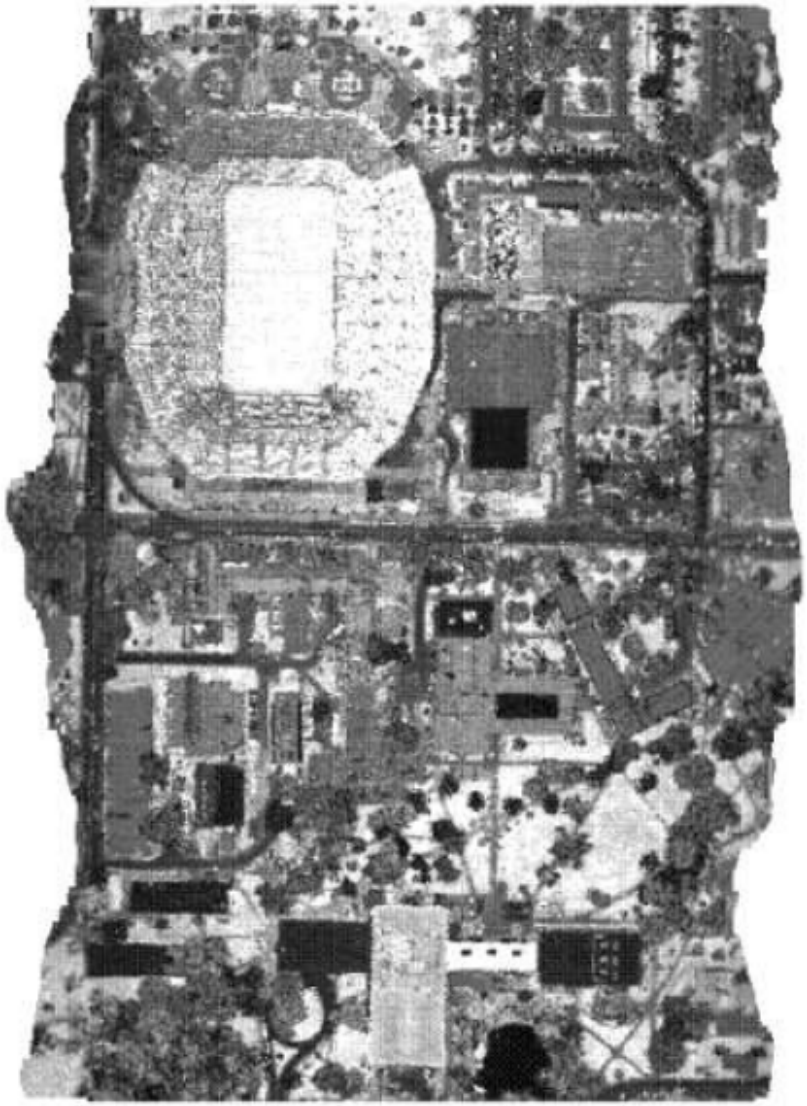
- Single-Return systems: returns come from the canopy top
- Multiple-Return systems: first returns also from the canopy top, but successive returns will come from lower surfaces, such as vegetation and the ground



Credits: Jiunn-Der (Geoffrey) Duh, Portland, USA

LiDAR return intensity

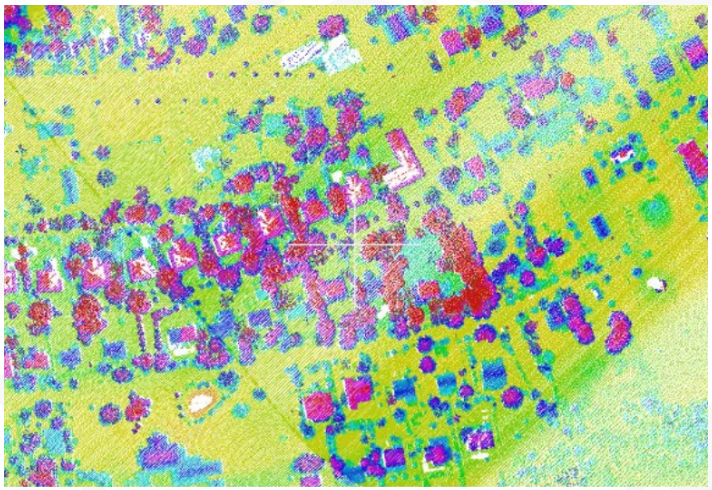




Credits: Jiunn-Der (Geoffrey) Duh, Portland, USA

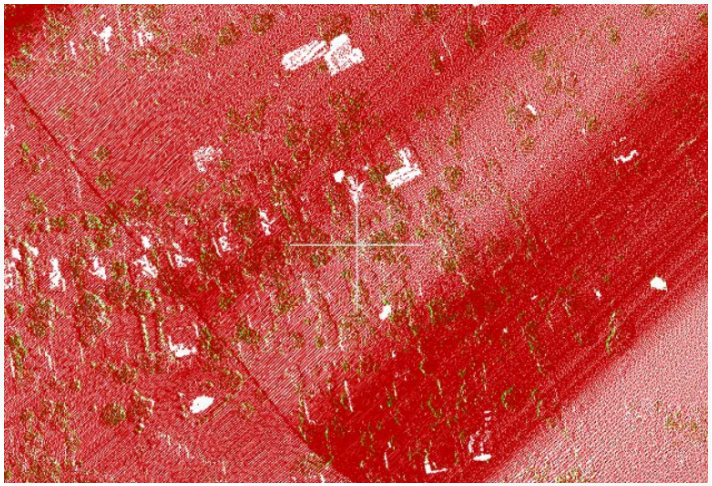
LiDAR points colored to represent different attributes of the data

Elevation

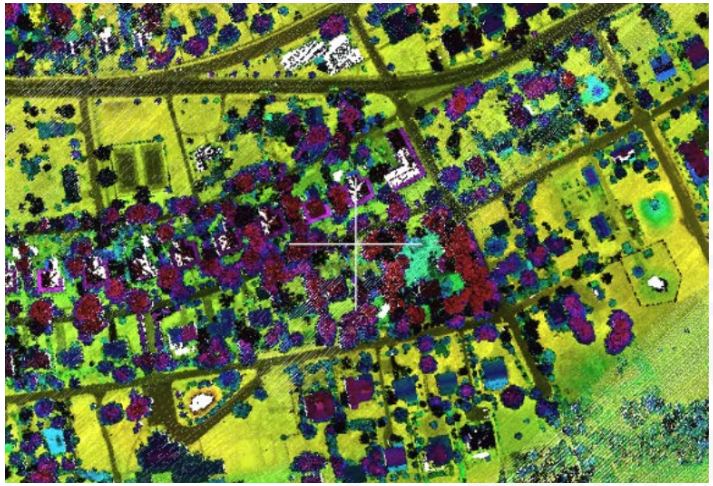


Intensity

Return number



Intensity+Elevation



Credits: Carter, et al. NOAA Coastal Services Center

Applications of LiDAR:

Water resources

- Floodplain mapping
- Storm water management
- Shoreline erosion

Geology

- Sinkhole identification
- Geologic/geomorphic mapping

Transportation

- Road and culvert design
- Cut and fill estimation
- Archaeological site id

Agriculture

- Erosion control
- Soils mapping
- Precision farming

Water quality

- Watershed modeling
- Wetland reconstruction
- Land cover/use mapping

Forestry

- Forest characterization
- Fire fuel mapping

Fish and wildlife management

- Drainage and water control
- Walk-in accessibility
- Habitat management

Emergency management

- Debris removal
- Hazard mitigation

LiDAR and Hyperspectral image fusion is a successful and active field

Elakshe08 coastal mapping by HSI (road vs water) + LiDAR (buildings)

Swatantrana11 biomass estimation by HSI (VIs) + LiDAR (vegetation structure)

Shimoni09 detect vehicles under shadows by rule-based HSI+LiDAR fusion

Zhang11 detect objects under shadows by HSI (remove direct illumination) + LiDAR (shadow-independent structures)

Lemp05 classification of urban areas using LiDAR for segmentation and HSI for region labeling

Sugumaran07 identification of tree species in a urban environment (structure matters)

Koetz07 classify fuel composition using SVM with composed HSI+LiDAR features

Naidooa12 classify savanna tree species using RF over HSI+LiDAR feature space

Pedergnana12 image classification using extended morphological attribute profiles (EAPs) from HSI and LiDAR

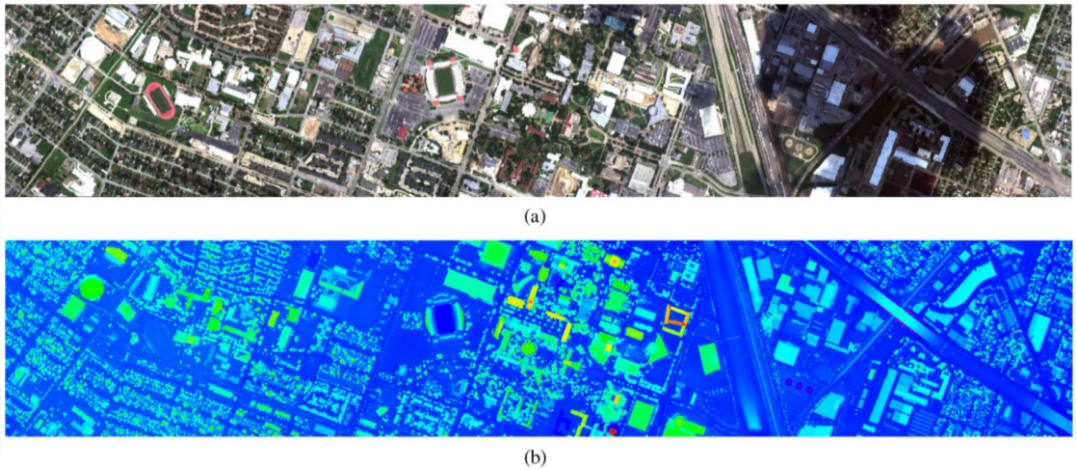
GRSS DF-TC competition 2013:

- HSI from CASI1500 sensor (144 bands, 380–1050 nm)
- LiDAR-derived digital surface model (DSM), spatial res. 2.5 m
- 15 classes, challenging problem, diversity of classes
- DSM represents elevation (in [m]) above sea level (Geoid 2012 A model)
- Note a large cloud shadow only for validation, avoid training there!

Classes

Class name	Training set	Test set	Class color
Healthy grass	198	1053	
Stressed grass	190	1064	
Synthetic grass	192	505	
Tree	188	1056	
Soil	186	1056	
Water	182	143	
Residential	196	1072	
Commercial	191	1053	
Road	193	1059	
Highway	191	1036	
Railway	181	1054	
Parking lot 1	192	1041	
Parking lot 2	184	285	
Tennis court	181	247	
Running track	187	473	

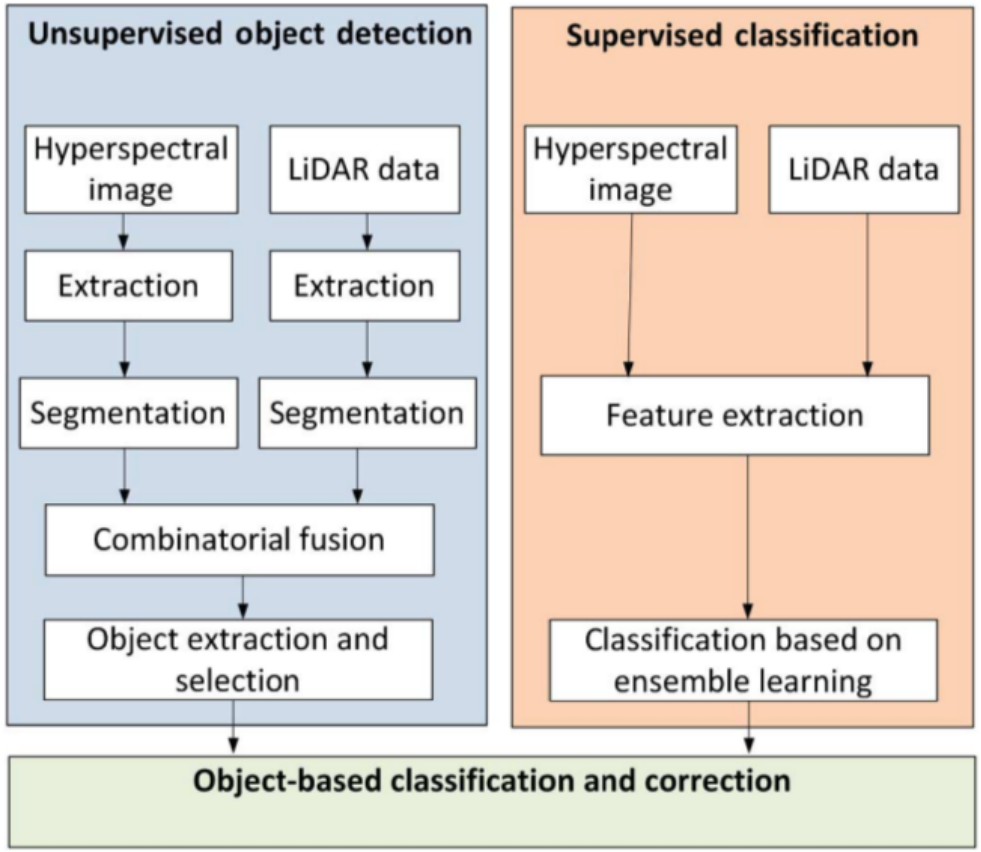
HSI + LiDAR-derived DSM



(a) (b)

Credits: Figures from Debes, et al. IEEE-JSTARS 2013. Special thanks to Dr. Saurabh Prasad @ University of Houston, USA.

Unsupervised+Supervised processing chain



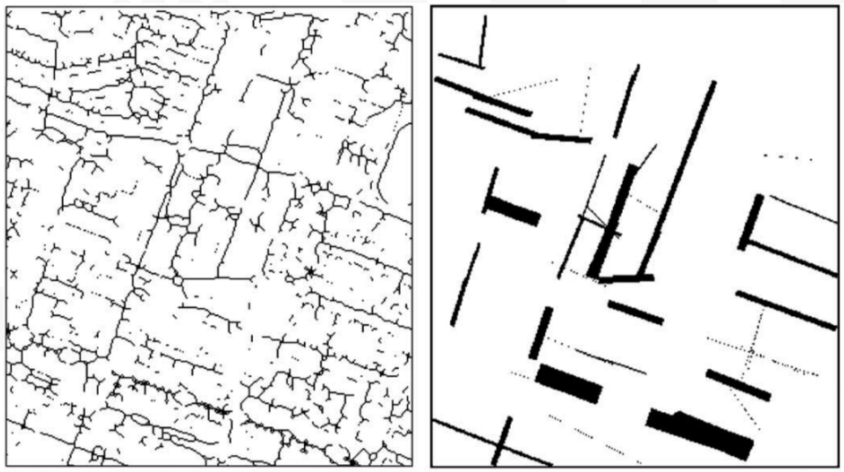
- Unsupervised Object Detection extracts objects (e.g. buildings, streets)
- Supervised Classification module: FE+classification via ensemble learning
- Object-based classification and correction combines results

1: Unsupervised module: Combinatorial fusion rules improve the detection

- Low vegetation, high elevation, large extension $\longrightarrow$ Buildings

- Low vegetation, low elevation, large extension $\longrightarrow$ Parking lots (after MP)

- Low vegetation, low elevation, small extension $\longrightarrow$ Streets (after Hough)



2: Supervised module: feature diversity plus random forests!

☐ Feature extraction

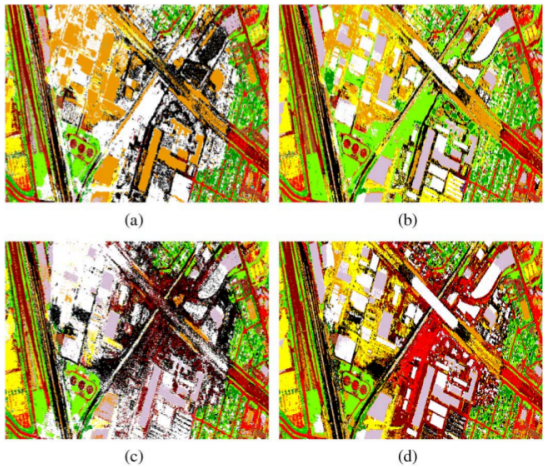
- ATGP unmixing *to* 50 endmembers $\rightarrow$ 50 abundance maps as features
- MNF features
- Vegetation index and water vapor absorption
- LiDAR-derived elevation map
- Topology features (such as gradients)

☐ Classification

- Random forest classifier

```
>> ntrees = 200
>> forest = TreeBagger(ntrees,Xtrain,Ytrain);
>> Ypred = predict(forest,Xtest);
```

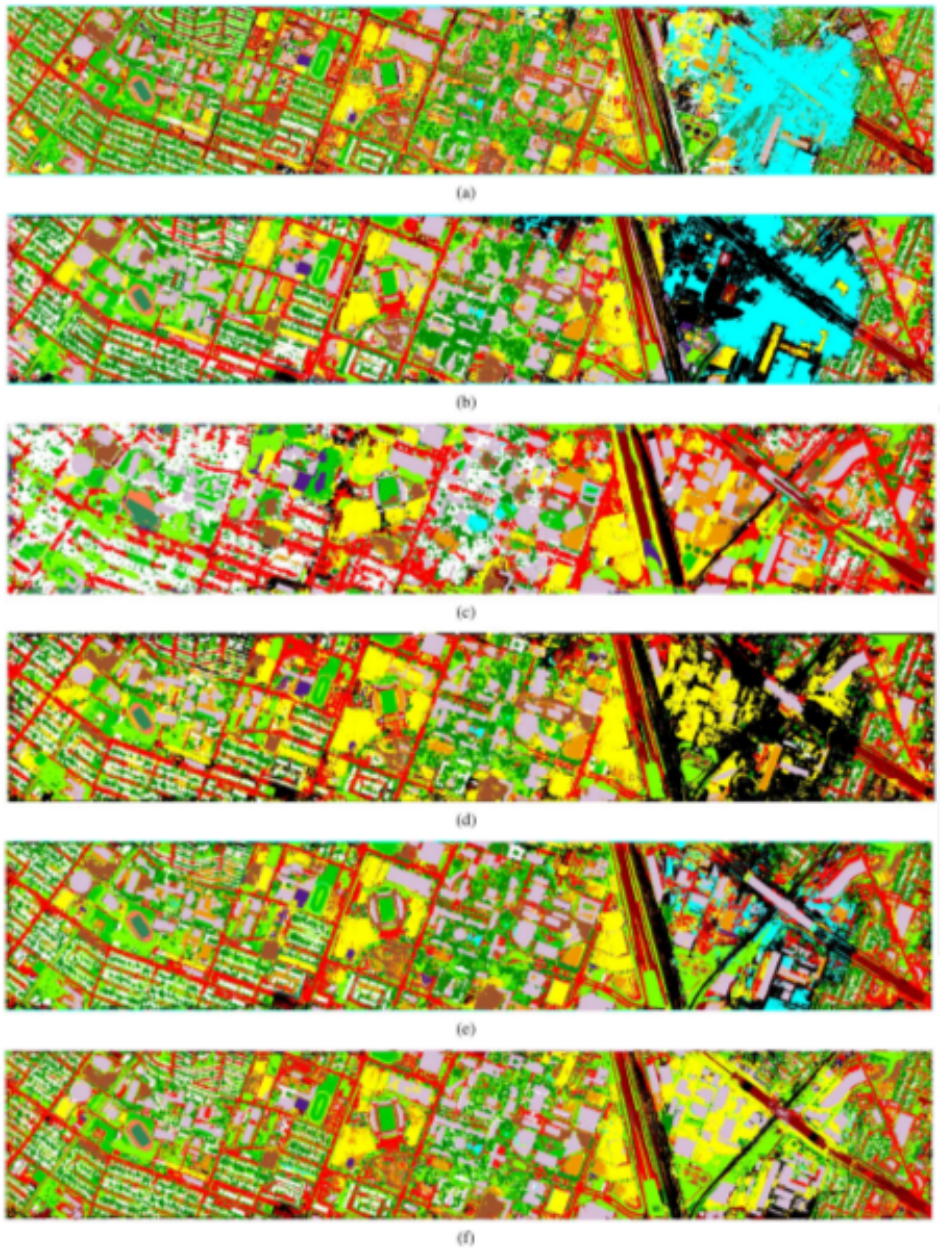
- RF1 with shadow-covered areas
- RF2 with shadow-free areas



3: Object-Based Classification and Correction

- Unsupervised branch provides information on the object level
- Supervised branch provides the required class label
- **Combination:**
  - Object correction implements a voting scheme for every object based on class uncertainties
  - Post-classification segmentation performs a label reassignment on the pixel level by modeling the classification outcome as an MRF

Class	SVM	2x RF	MMS Cor.	Post Seg.
Healthy grass	82.24	83.38	83.48	83.47
Stressed grass	82.99	97.74	97.56	97.56
Synthetic grass	99.60	99.60	99.80	100.00
Tree	89.87	98.30	97.63	97.63
Soil	98.56	99.24	98.58	100.00
Water	83.92	95.10	87.41	88.11
Residential	79.85	90.95	87.13	90.95
Commercial	43.11	94.11	95.06	97.44
Road	66.38	83.10	85.65	89.90
Highway	82.62	52.51	94.40	96.91
Railway	81.69	85.01	81.69	93.64
Parking 1	75.31	84.05	80.69	96.35
Parking 2	66.32	82.10	70.52	75.79
Tennis court	98.38	99.59	100.00	100.00
Running track	97.67	97.46	98.10	100.00
OA (%)	80.10	88.10	90.60	94.40
AA (%)	81.90	89.50	90.50	93.90
κ	0.784	0.871	0.898	0.940



a hyperspectral data,
b MPs of hyperspectral data,
c MPs of LiDAR data,
d the stacked features
e features fused with graphs
f the proposed fused features

Why not considering additional information?

- Vegetation indices, e.g. NDVI
- Clustering maps
- Max vote of all trained classifiers
- Abundance maps
- Ecosystems maps
- Climate regions
- ...

Nice idea, yet problematic: dimensionality increases again!
Solution: feature selection together with sparse classifiers!

- Hyperspectral image classification is a challenging problem
- High dimensional feature spaces scarcely populated!
- Statistical approaches:
  - Supervised algorithms
  - ~~Unsupervised algorithms~~
  - ~~Semisupervised algorithms~~
  - ~~One-class and target detection~~
- Kernel methods are the current state-of-the-art classifiers
- More info in the classifiers implies improved signal model
  - More samples (by sampling or synthesizing)
  - More meaningful features
  - More concurrent sensors (SAR, LiDAR, VHR, etc)
  - Additional ancillary information
  - Multitemporal information

Part 3: Feature extraction from remote sensing images

Extracting features from remote sensing images is essential to:

- Compress information for storage/transmission
- Reduce (spatial and spectral) redundancy
- Make image processing algorithms more robust (to noise, ♯labels, dim.)
- Visualize data characteristics
- Understand the underlying physical relations

Extracted features can be either:

1. Spectral:
   - Physically-based spectral features
   - Statistical multivariate methods: linear and nonlinear
2. Spatial/contextual
   - Standard image processing descriptors
   - Advanced computer vision descriptors
3. ~~Spatio-spectral: extract features from spectral patches or regions~~

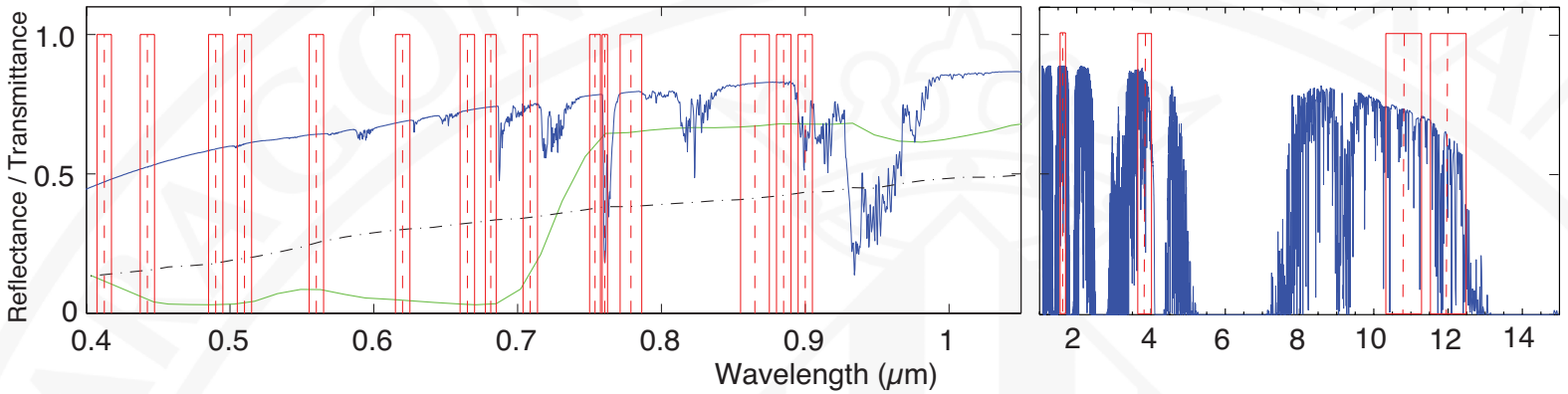
Motivation:

- Measured spectral signal at the sensor depends on the illumination, the atmosphere, and the surface
- Physically-inspired features before applying a machine learning algorithm
- Adapt standard feature extraction methods, such as PCA, to include knowledge about the physical problem

Two case studies:

1. Cloud screening with spectral feature extraction from MERIS and AATSR
2. Vegetation monitoring by vegetation indices

Example 1: Cloud screening with spectral features from MERIS+AATSR



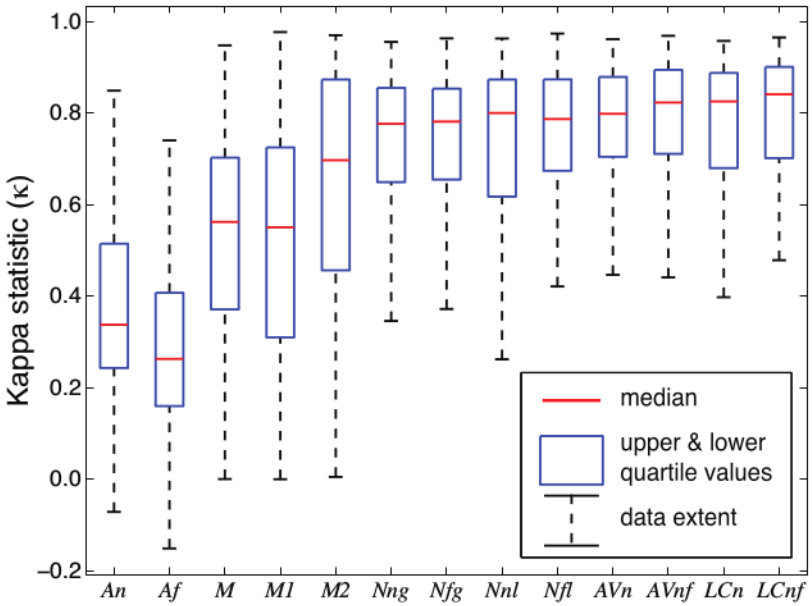
MERIS and AATSR channel locations (red boxes) superimposed to a reflectance spectra of healthy vegetation (green thin solid line), bare soil (black dash-dotted line), and the atmospheric transmittance (blue solid line)

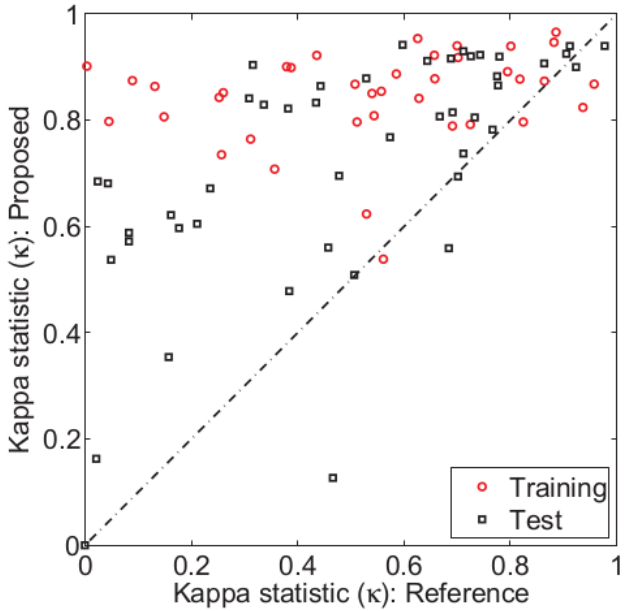
- The spectral bands free from atmospheric absorptions contain information about the surface reflectance
- Other spectral bands are mainly affected by the atmosphere
- Cloud features extracted from MERIS and AATSR products are needed to discriminate clouds from surface

Credits: Figure from Gomez-Chova07.

Sensor	Cloud Feature	Channels Involved	Reference
MERIS	Brightness & Whiteness (VIS)	VIS bands [1-8]	GomezChova07
MERIS	Brightness & Whiteness (NIR)	NIR bands [9 10 12 13 14]	GomezChova07
MERIS	Brightness & Whiteness	VIS&NIR bands (without 11 & 15)	GomezChova07
MERIS	O_2 absorption	754, 761, 778 nm	GomezChova07
MERIS	WV absorption	885, 900nm	GomezChova07
MERIS	Surface Pressure	761&754nm	Lindstrot09
MERIS	Surface Pressure	761/754nm ratio	MERIS_handbook
MERIS	Bright over Land (sand)	443/754nm ratio	MERIS_handbook
MERIS	Bright over Land (ice)	709/865nm ratio	MERIS_handbook
MERIS	Cirrus over Ocean/Land	761/754nm ratio ; 865nm	MERIShandbook
MERIS	Bright Clouds	450nm	Preusker08
MERIS	Snow Test (reflectance)	865/890 NDI	Preusker08
MERIS	Cloud 412 reflectance	412/443nm ratio	Kokhanovsky08
MERIS	Cloud 412 reflectance	412/443nm difference	Kokhanovsky08
MERIS	Cloud mask 1	412/681nm ratio	Guanter08
MERIS	Cloud mask 2	412/708nm ratio	Guanter08
MERIS	Hue-Saturation-Value transf.	665, 560, 442nm	Gonzalez07
AATSR	Gross Cloud	12μm	AATSR_handbook
AATSR	Thin Cirrus	11/12μm difference	AATSR_handbook
AATSR	11/12μm Nadir/Forward	11μm nad/fwd ; 11/12μm	AATSR_handbook
AATSR	Visible Channel Cloud Test	870,670,550nm NDI	Prata02
AATSR	Snow Test	1.6μm 550nm NDI	Prata02
AATSR	Reflectance Gross Cloud	670nm	Birks07
AATSR	Reflectance Ratio	870/670nm ratio	Birks07
AATSR	Albedo	3.7μm	Birks07
AATSR	Thermal Difference	11/12μm difference	Birks07
AATSR	Thermal Gross Cloud	11μm	Birks07
AATSR	11μm Nadir/Forward	11μm nad/fwd	Muller08
AATSR	865 Nadir/Forward	865 nad/fwd	Muller08

Credits: Table from Gomez-Chova07.





- Combination of MERIS and AATSR features improves cloud detection
- Cloud detection over 84 MERIS/AATSR images improves the 'BEAM Cloud Probability Processor'

Credits: Figures taken from Gomez-Chova10.

Example 2: Vegetation monitoring with spectral indices

- The estimation of land/vegetation parameters from remote sensing images helps to determine their status and processes therein
- Standard parameters: Leaf chlorophyll content (Chl), leaf area index (LAI), and fractional vegetation cover (FVC)
- Simple relations to predict bio-physical parameters from VIs:

$$\begin{aligned} y &= \sum_{i=1}^{n} a_i \mathrm{VI}^i \\ y &= a + b\mathrm{VI}^c \\ y &= a \ln(b - \mathrm{VI}) + c \end{aligned} \tag{1}$$

  where VI is a combination (typically ratios) of reflectance values in n specific channels
- VIs can be either computed using digital numbers, TOA radiance/reflectance, or surface radiance/reflectance

The Normalized Difference Vegetation Index (NDVI) is a widely used index:

$$\text{NDVI} = \frac{\text{NIR} - \text{R}}{\text{NIR} + \text{R}}$$

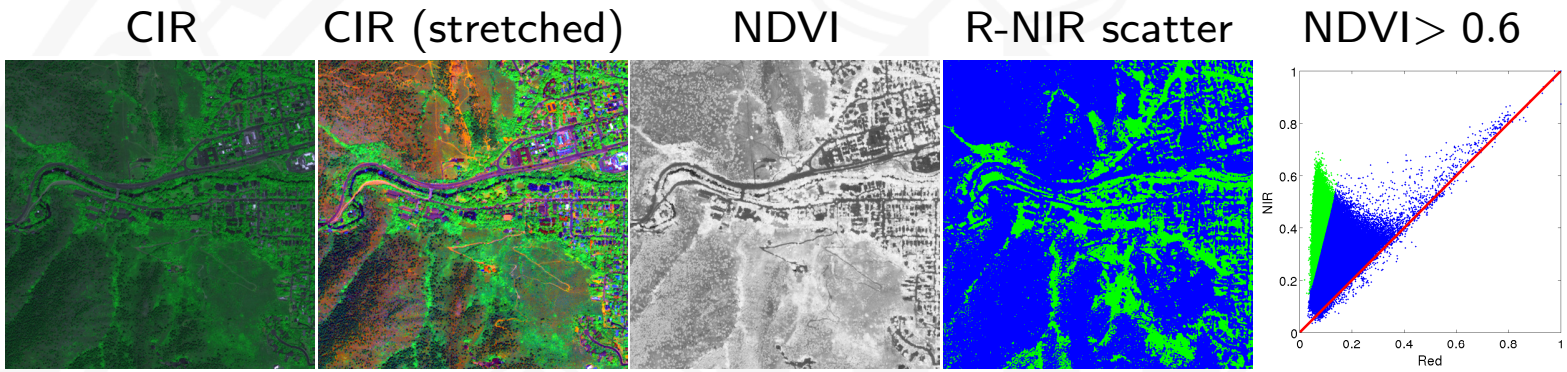


Figure : Landsat image acquired over a residential area containing different land classes (asphalt, forest, buildings, grass, water, etc.). Left to right: standard color-infrared (CIR) composite, stretched CIR and NDVI image, thresholded NDVI image, and the scatter plot of all image pixels in Red versus NIR space.

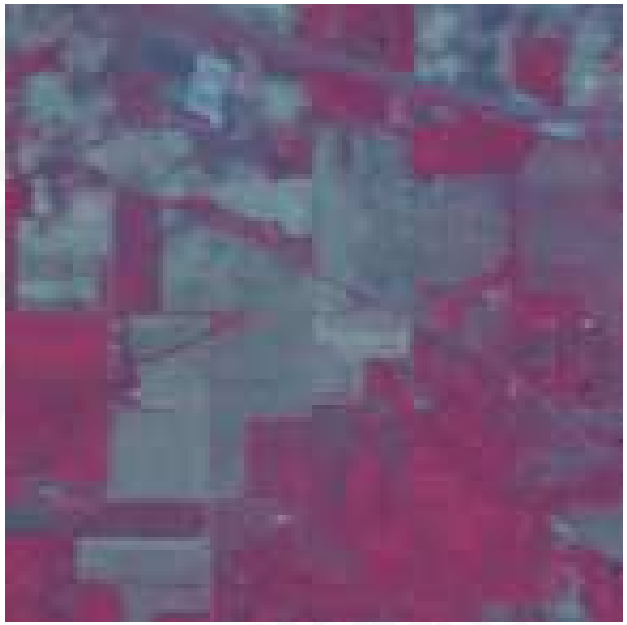
Method	Formulation	ρ
GI	R_{672}/R_{550}	0.52 (0.09)
GVI	$(R_{682}\text{-}R_{553})/(R_{682}+R_{553})$	0.66 (0.07)
Macc	$(R_{780}\text{-}R_{710})/(R_{780}+R_{680})$	0.20 (0.29)
MCARI	$[(R_{700}\text{-}R_{670})\text{-}0.2(R_{700}\text{-}R_{550})]/(R_{700}/R_{670})$	0.35 (0.14)
MCARI2	$1.2[2.5(R_{800}\text{-}R_{670})\text{-}1.3(R_{800}\text{-}R_{550})]$	0.71 (0.12)
mNDVI	$(R_{800}\text{-}R_{680})/(R_{800}+R_{680}\text{-}2R_{445})$	0.77 (0.12)
mNDVI_{705}	$(R_{750}\text{-}R_{705})/(R_{750}+R_{705}\text{-}2R_{445})$	0.80 (0.07)
mSR_{705}	$(R_{750}\text{-}R_{445})/(R_{705}+R_{445})$	0.72 (0.07)
MTCI	$(R_{754}\text{-}R_{709})/(R_{709}+R_{681})$	0.19 (0.26)
mTVI	$1.2[1.2(R_{800}\text{-}R_{550})\text{-}2.5(R_{670}\text{-}R_{550})])$	0.73 (0.07)
NDVI	$(R_{800}\text{-}R_{670})/(R_{800}+R_{670})$	0.77 (0.08)
NDVI2	$(R_{750}\text{-}R_{705})/(R_{750}+R_{705})$	0.81 (0.06)
NPCI	$(R_{680}\text{-}R_{430})/(R_{680}+R_{430})$	0.72 (0.08)
NPQI	$(R_{415}\text{-}R_{435})/(R_{415}+R_{435})$	0.61 (0.15)
OSAVI	$1.16(R_{800}\text{-}R_{670})/(R_{800}+R_{670}+0.16)$	0.79 (0.09)
PRI	$(R_{531}\text{-}R_{570})/(R_{531}+R_{570})$	0.77 (0.07)
PRI2	$(R_{570}\text{-}R_{539})/(R_{570}+R_{539})$	0.76 (0.07)
PSRI	$(R_{680}\text{-}R_{500})/R_{750}$	0.79 (0.08)
RDVI	$(R_{800}-R_{670})/\sqrt{(R_{800}+R_{670})}$	0.76 (0.08)
SIPI	$(R_{800}\text{-}R_{445})/(R_{800}\text{-}R_{680})$	0.78 (0.08)
SPVI	$0.4[3.7(R_{800}\text{-}R_{670})\text{-}1.2(R_{530}\text{-}R_{670})]$	0.70 (0.08)
SR	R_{800}/R_{680}	0.63 (0.12)
SR1	R_{750}/R_{700}	0.74 (0.07)
SR2	R_{752}/R_{690}	0.68 (0.09)
SR3	R_{750}/R_{550}	0.75 (0.07)
SR4	R_{672}/R_{550}	0.76 (0.10)
SRPI	R_{430}/R_{680}	0.76 (0.09)
TCARI	$3[R_{700}\text{-}R_{670})\text{-}0.2(R_{700}\text{-}R_{550})(R_{700}/R_{670})]$	0.53 (0.13)
TVI	$0.5[120R_{750}\text{-}R_{550})\text{-}200(R_{670}\text{-}R_{550})]$	0.70 (0.10)
VOG	$R_{740}/(R_{720}$	0.76 (0.06)
VOG2	$(R_{734}\text{-}R_{747})/(R_{715}+R_{726})$	0.72 (0.09)
NAOC	Area in $[643, 795]$	0.79 (0.09)

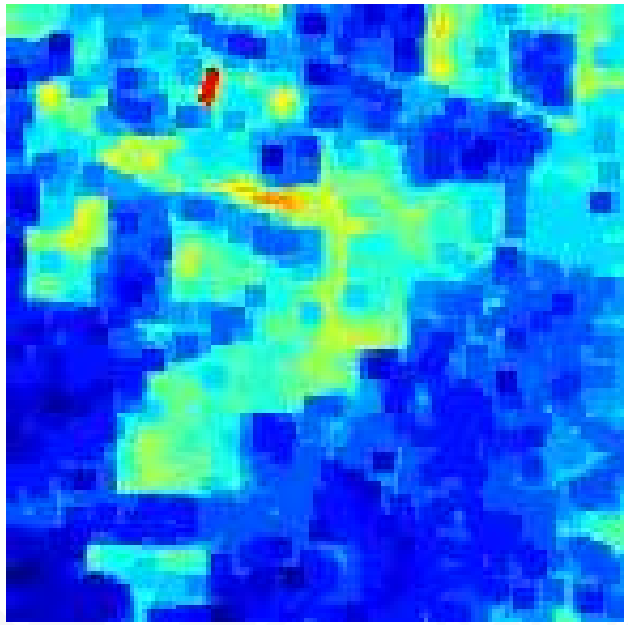
Credits: Table from Verrelst11.

Erosion: "Replace pixel with the minimum surrounding pixel over SE."

```
>> se = strel('disk',3); O = imerode(I,se);
```

Erosion, disk 3x3



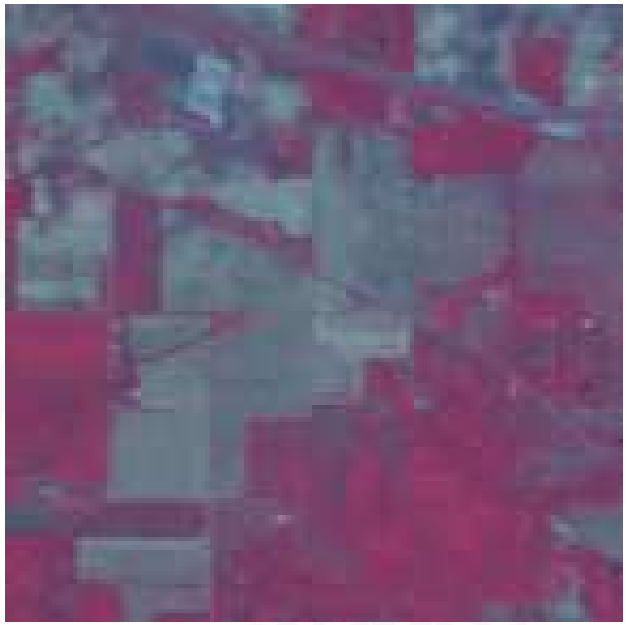


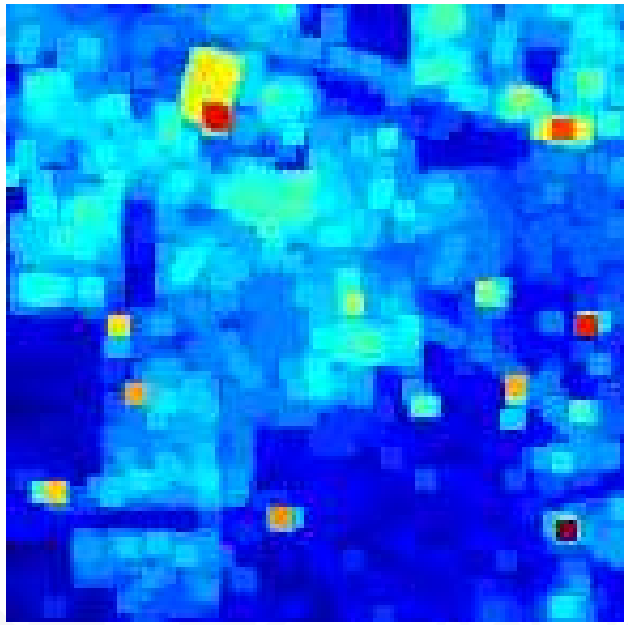
- Darker features than the surroundings are enlarged
- Brighter features than the surroundings shrink

Dilation: "Replace pixel with the maximum surrounding pixel over SE."

```
>> se = strel('disk',3); O = imdilate(I,se);
```

Dilation, disk 3x3

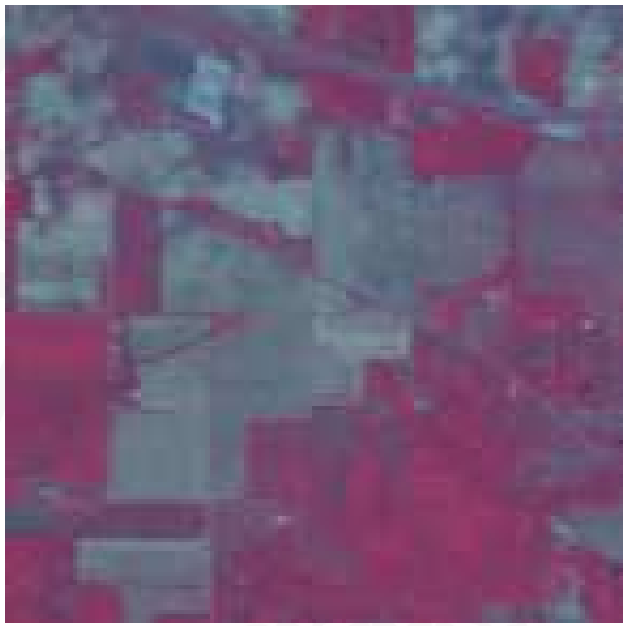




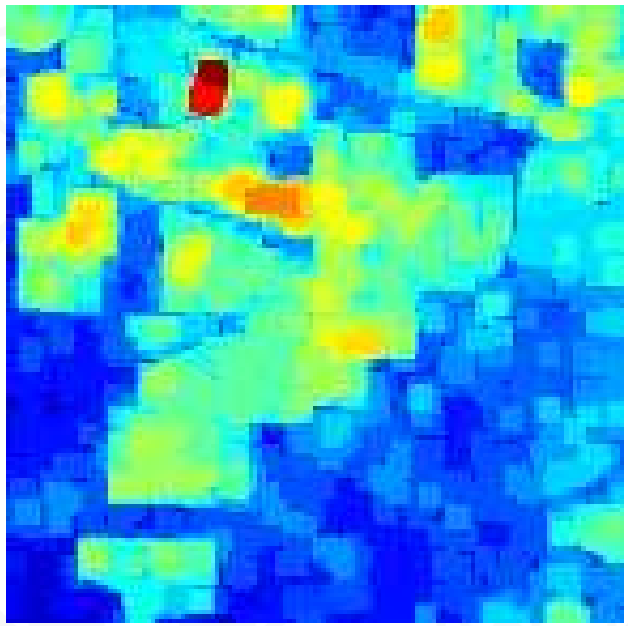
- Brighter features than the surroundings are enlarged
- Darker features than the surroundings shrink

Opening: "Erosion followed by dilation"

```
>> se = strel('disk',3); O = imopen(I,se);
```



Opening, disk 3x3

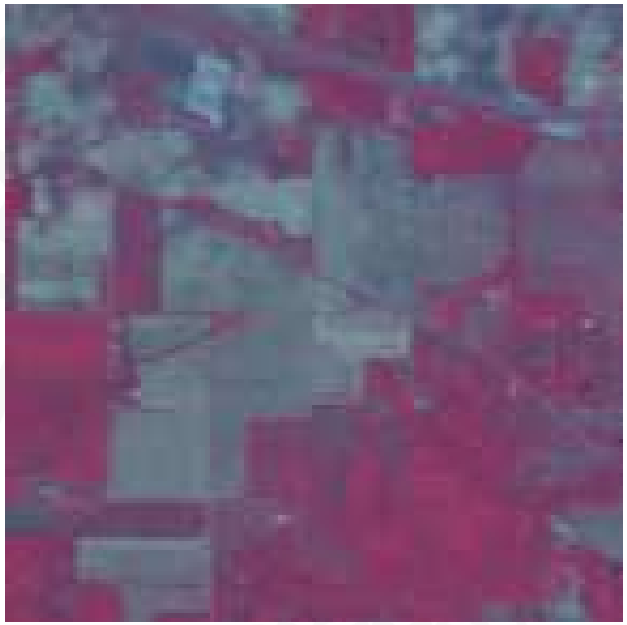


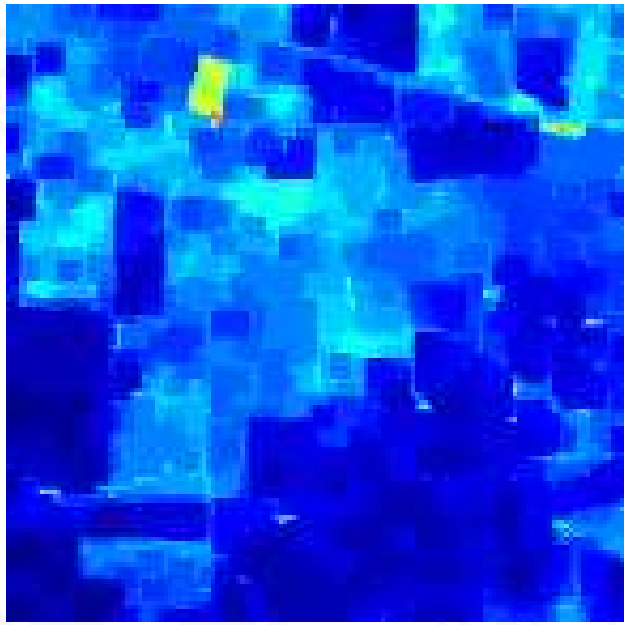
- Brighter features than the surroundings and smaller than the SE disappear
- Other features (dark, or bright and large) remain unchanged

Closing: "Dilation followed by erosion."

```
>> se = strel('disk',3); C = imclose(I,se);
```

Closing, disk 3x3



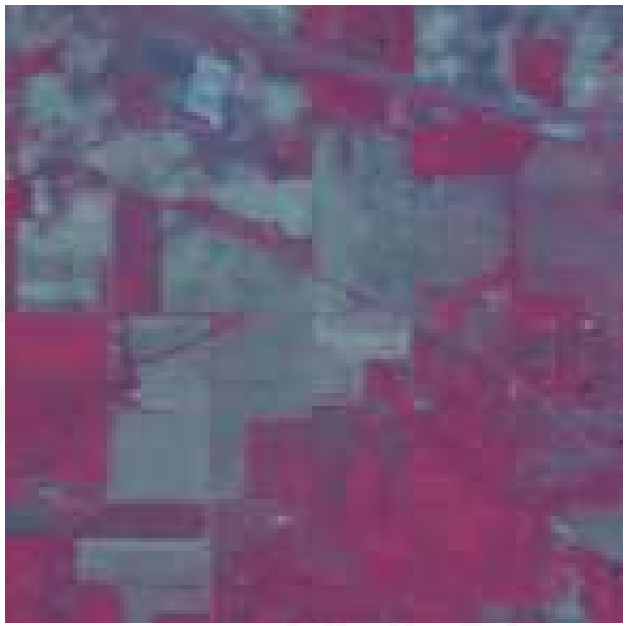


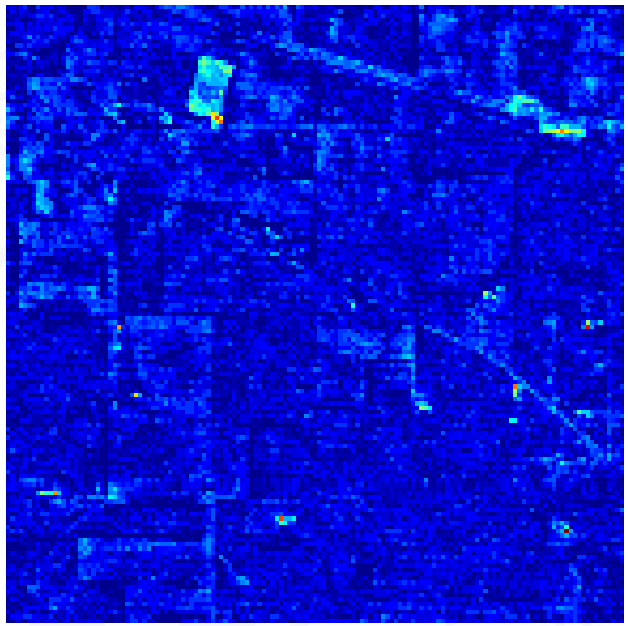
- Darker features than the surroundings and smaller than the SE disappear
- Other features (bright, or dark and large) remain unchanged

Top hat: "Open and then subtract the result from the original image"

```
>> se = strel('diamond',5); T = imtophat(I,se);
```

Top hat, diamond 3x3



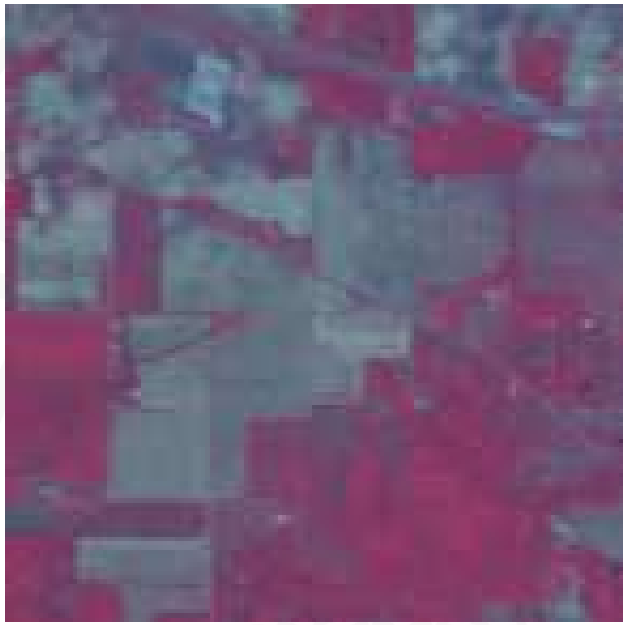


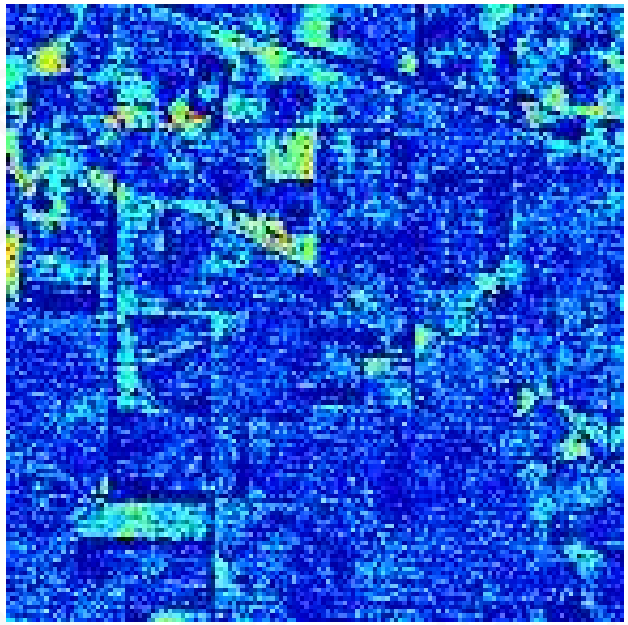
- Emphasizes distinct (sharp peaks) structures, extracts small elements and details from given images
- Useful to correct for uneven illumination (improve contrast)

Bottom hat: "Closing and then subtracts the result from the original image"

```
>> se = strel('diamond',5); B = imbothat(I,se);
```

Bottom hat, disk 3x3

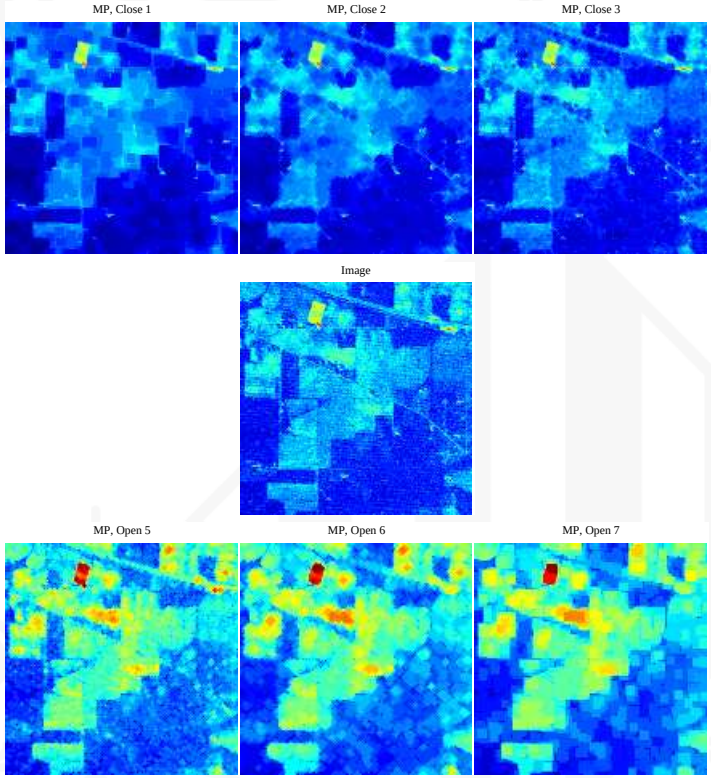




- Emphasizes distinct (sharp valleys) structures
- Useful to correct for uneven illumination (improve contrast)

Morphological profile: "Openings and closings with increasing SE"

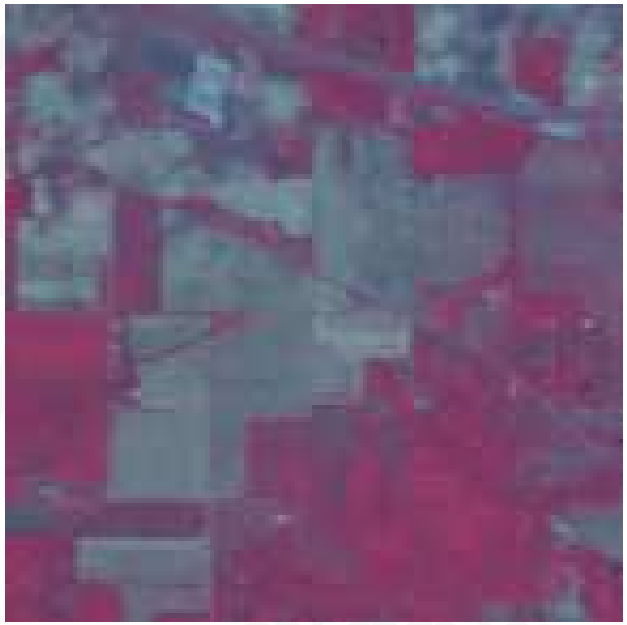
```
>> se = strel('diamond',5); repeat opening-closing operations;
```



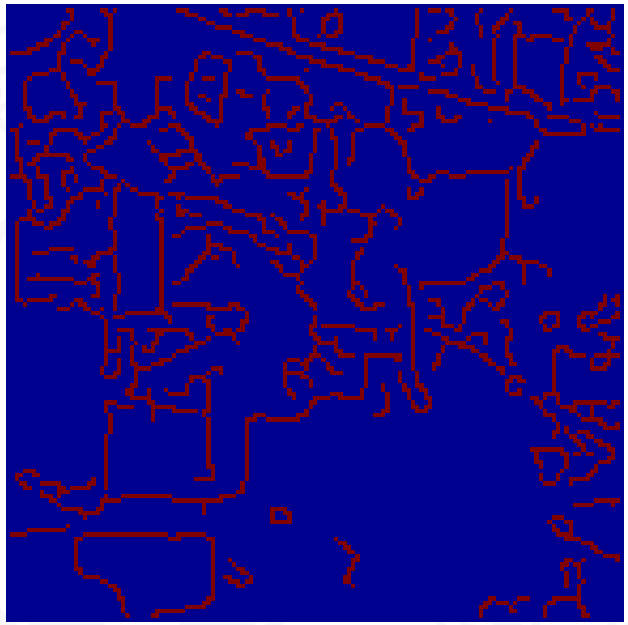
- Pixels turn into a sequential analysis of fine-to-coarse relations
- Useful as a feature vector for processing (e.g. classification)

Edges: "Detecting discontinuities in images"

```
>> EDGES1 = edge(I,'canny'); EDGES2 = edge(I,'prewitt');
```



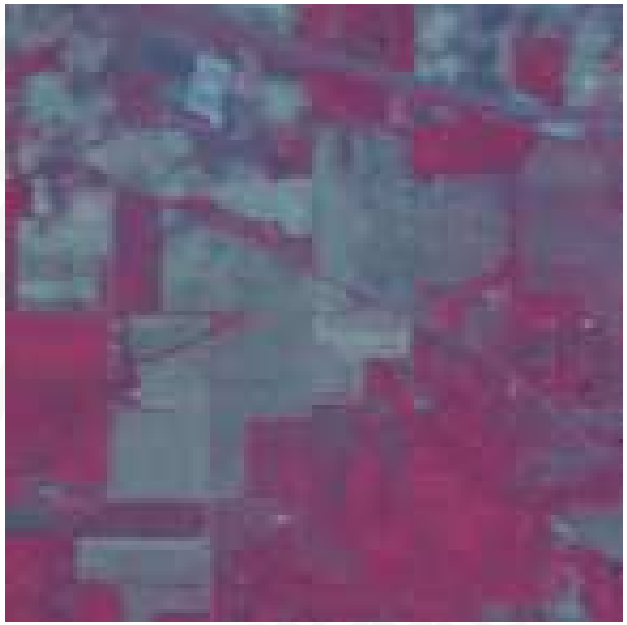
Canny edges



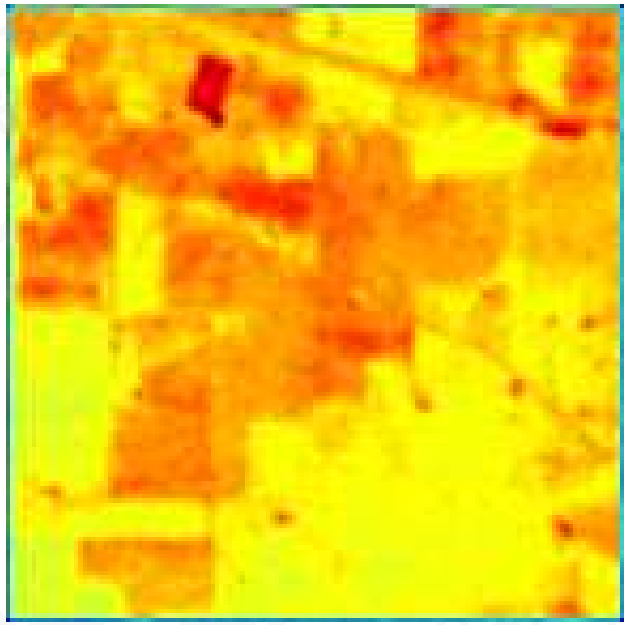
- Useful feature to detect boundaries in urban monitoring
- Useful feature for object delineation

Mean filter: "Average intensity values around every pixel"

```
>> H = ones(3); S = imfilter(I,H);
```



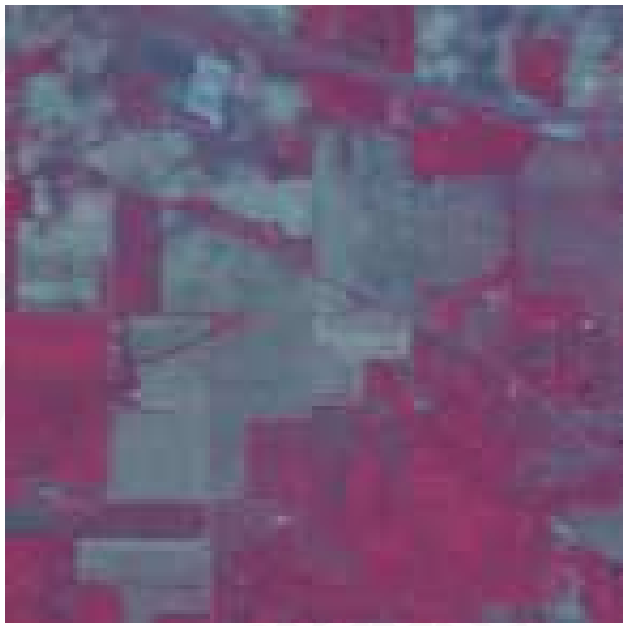
Mean filter, 5x5 window



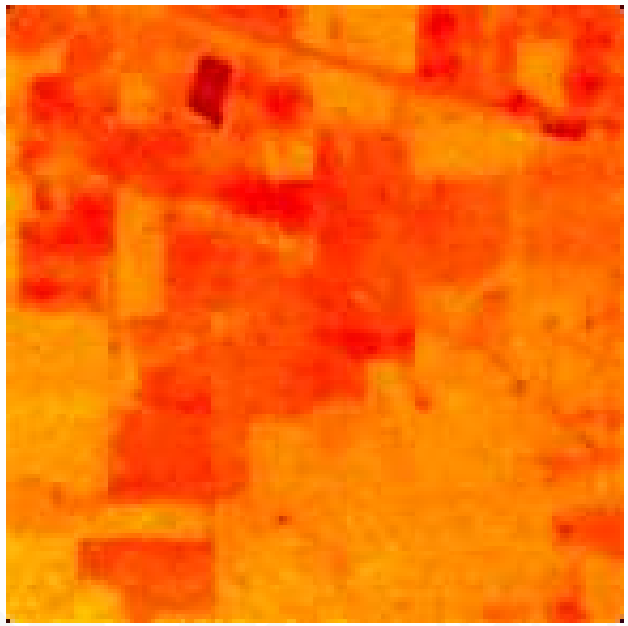
- Useful for noise removal and smoothing
- Simple yet efficient to account for spatial pixel relations

Median filter: "Replace a pixel with the median value of the neighborhood"

```
>> S = medfilt2(I);
```



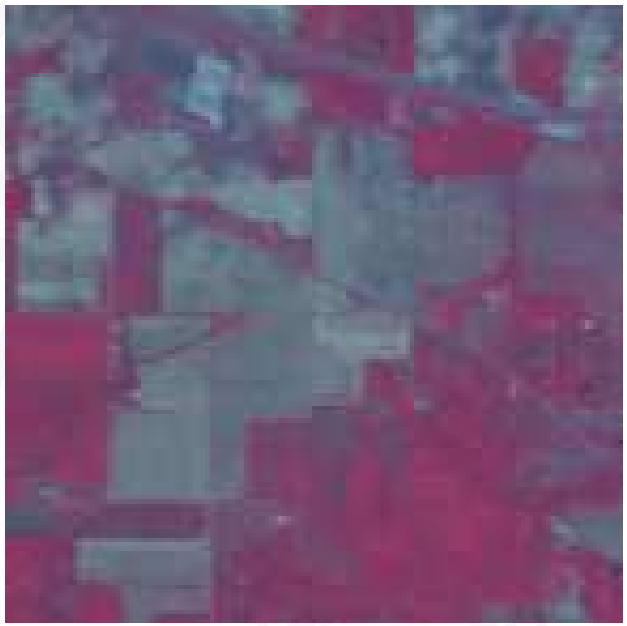
Median filter, 3x3 window



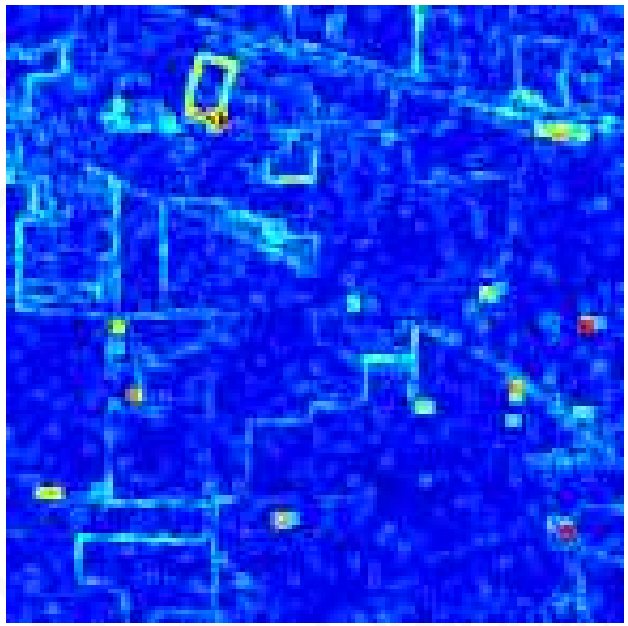
- Useful for impulsive noise removal and invariance encoding
- Simple yet efficient to account for spatial pixel relations

Standard deviation: "Replace a pixel with the local standard deviation value of the neighborhood"

```
>> S = stdfilt(I);
```



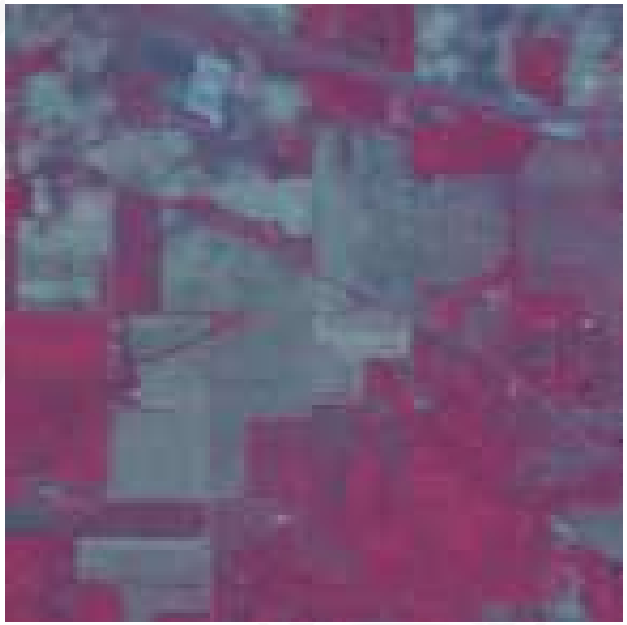
Local standard deviation, 3x3 window



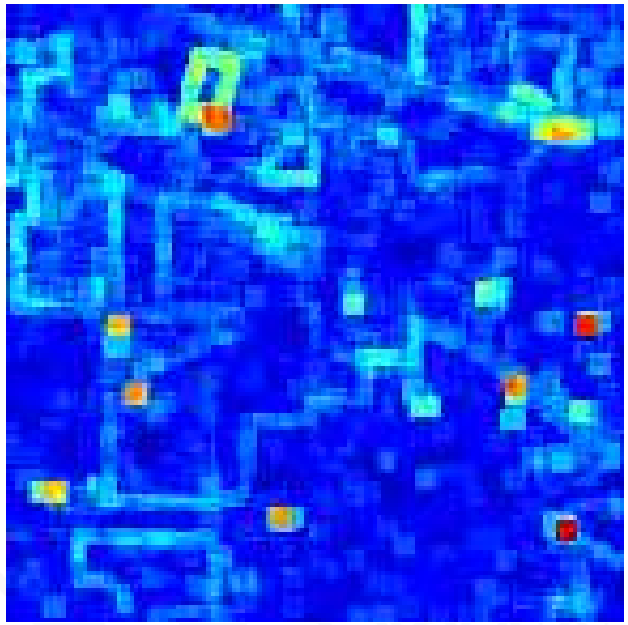
- Useful to detect borders and edges
- Captures the spatial variability of the intensity image

Range filter: "Replace a pixel with the range (max − min) value of the neighborhood"

```
>> R = rangefilt(I,ones(5));
```



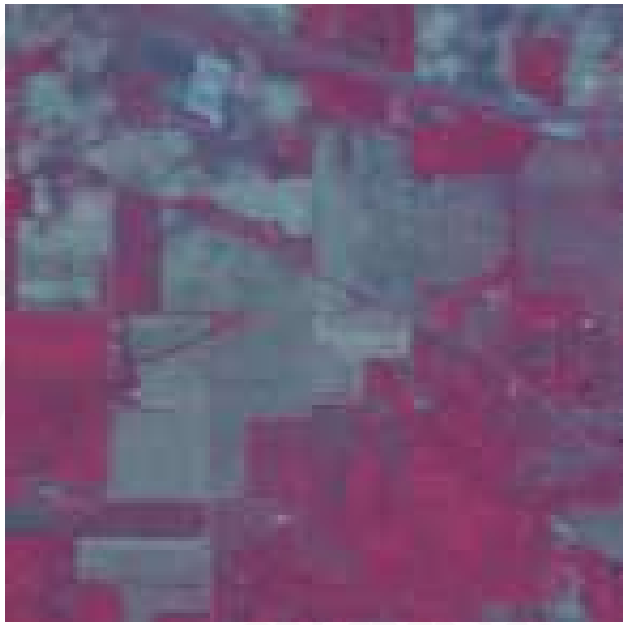
Local range filter, 5x5 window



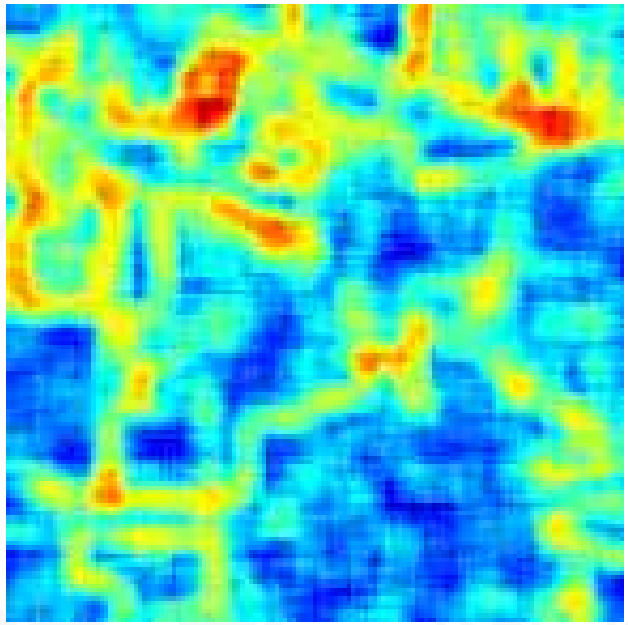
- Useful for edge detection
- Useful for range filtering

Local entropy: "Replace a pixel with the entropy value of the neighborhood"

```
>> H = entropyfilt(I/max(I(:)));
```



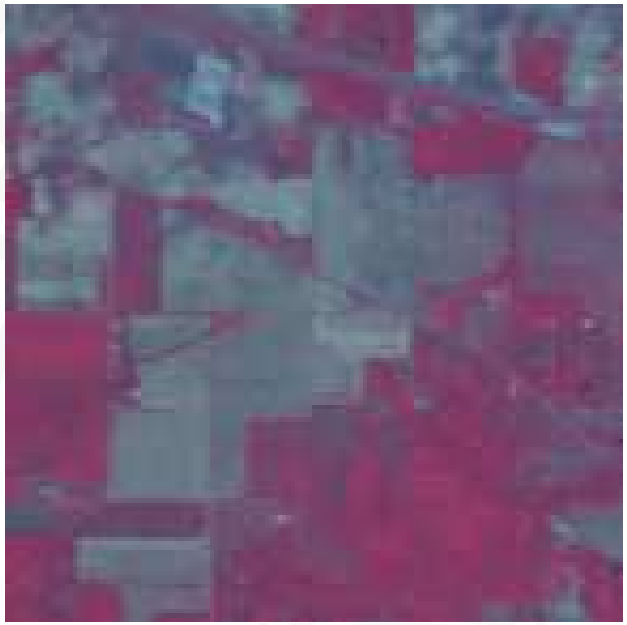
Local entropy, 9x9 window



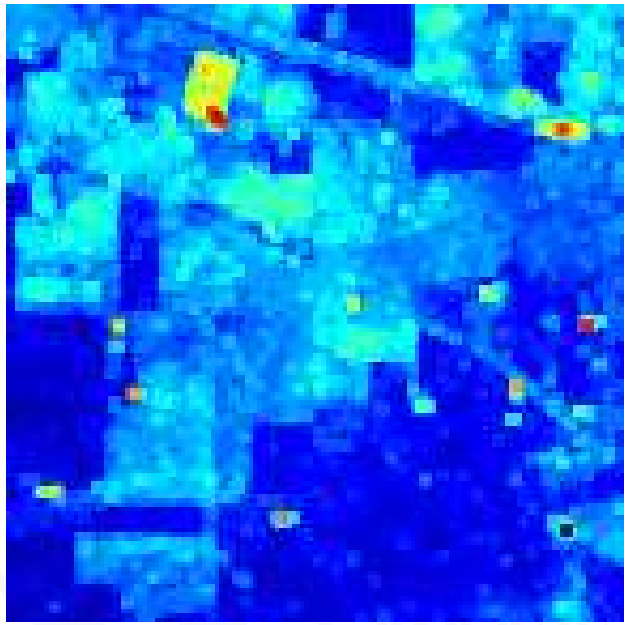
- Useful for edge detection
- Useful for saliency and detection of anomalies

Max pooling filtering: "Replace a pixel with the maximum value of the neighborhood"

```
>> maxpool = ordfilt2(I,9,true(3));
```



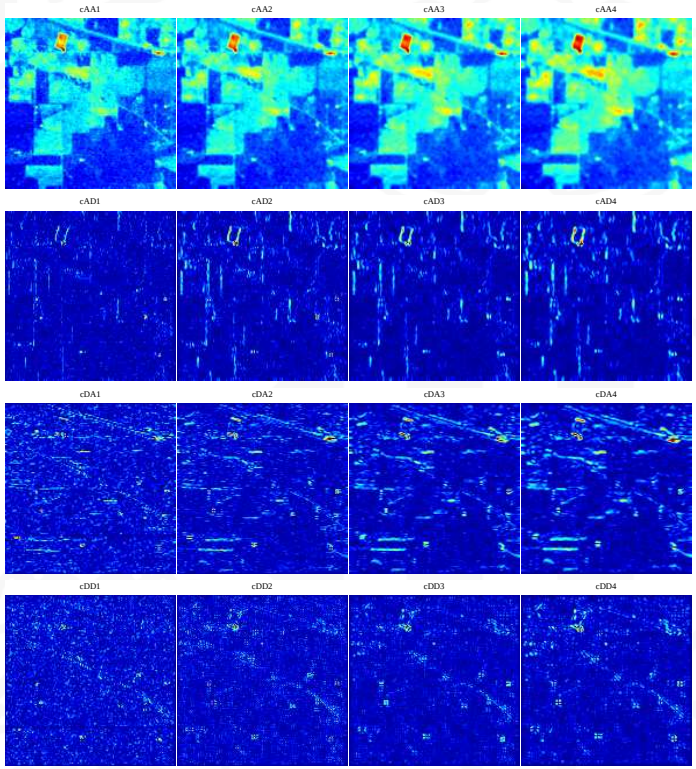
Max pooling, 3x3 window



- Efficient to encode invariance to rotation
- Useful in object detection

Haar wavelet decomposition: "performs a multilevel 2-D nondecimated wavelet decomposition with *n* scales and 3 orientations"

```
>> n = 4; w = 'db1'; >> WT = ndwt2(I,n,w);
```

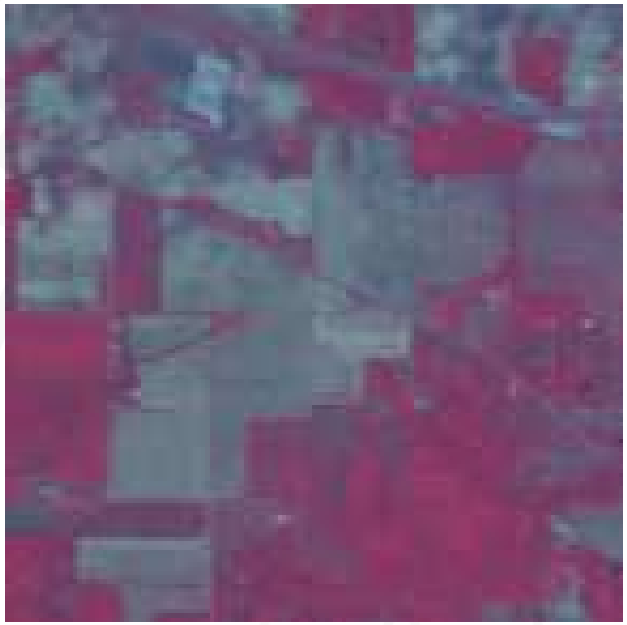


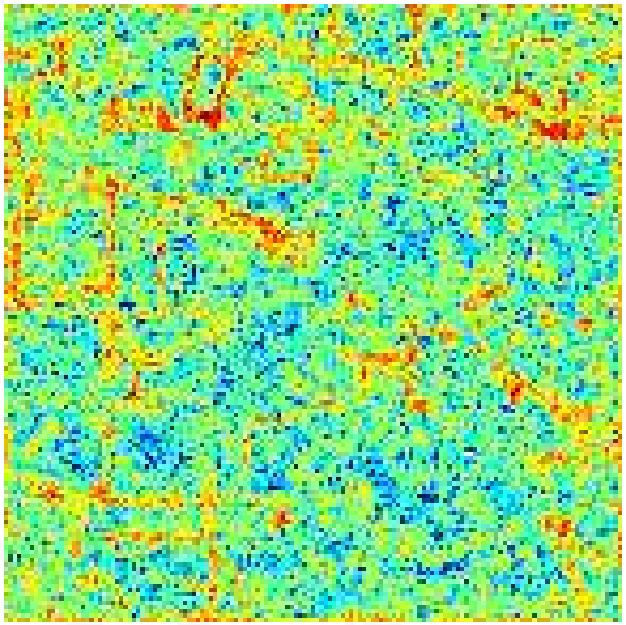
- Multiscale analysis of spatial and frequency pixel relations
- Stacking features is robust to noise and powerful for discrimination

Markov random fields: "Models a pixel with a Markov chain of the surrounding pixels, and computes a statistic on the model weights"

```
>> fun = @(x) entropy(lsfit(x));
>> M = nlfilter(I,[3 3],@fun);
```

Entropy of the Markov random field

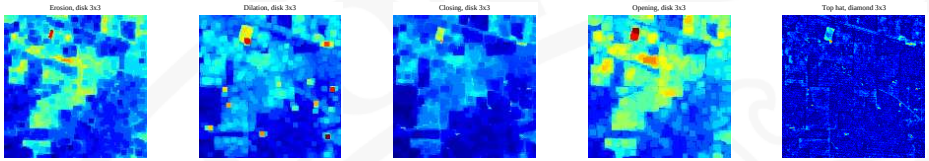




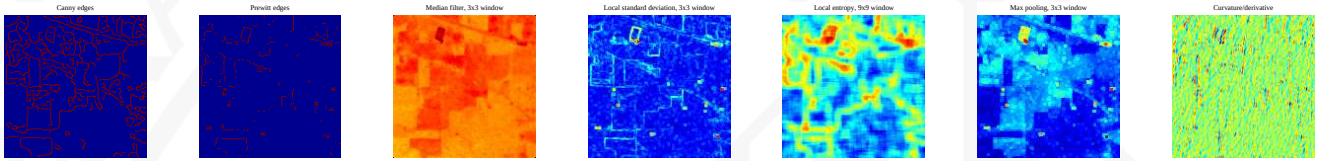
- A simple linear predictive model is useful to capture textures
- Computationally demanding and several free parameters

Standard image processing spatial features

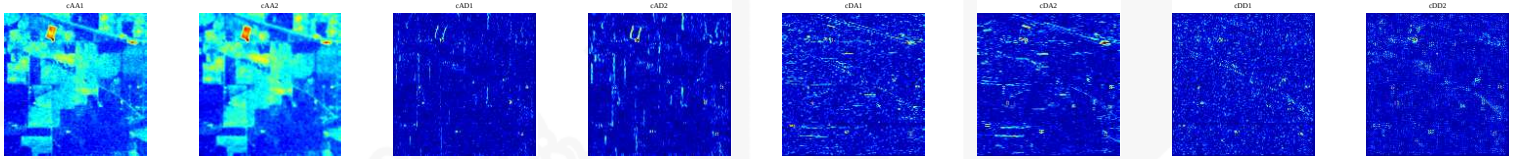
- Mathematical morphology: erosion, closing, tophats



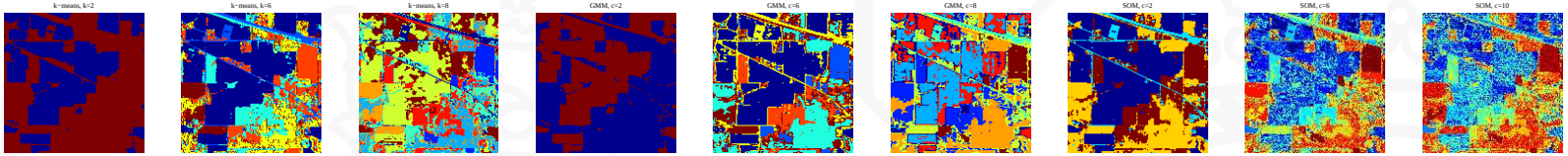
- Edges and invariants (median, max-pooling, entropy)



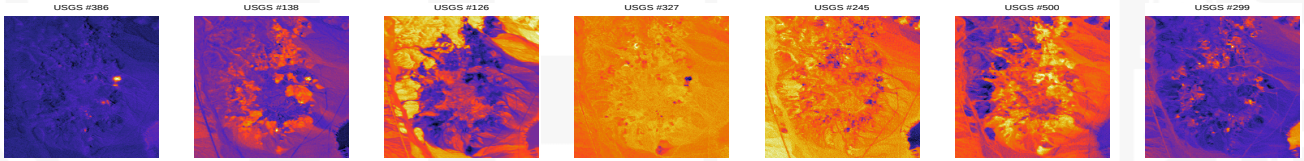
- Multiscale wavelet decompositions



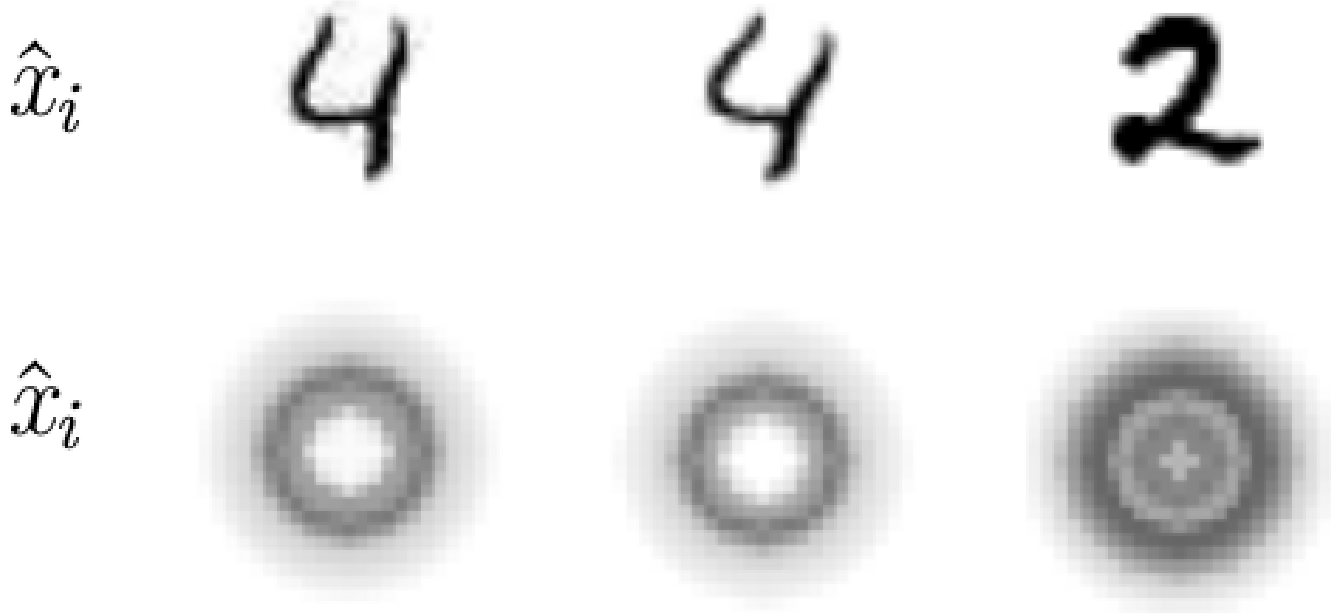
- Clustering (*k*-means, fuzzy *c*-means, GMM-EM, hierarchical, SOM)



- Abundance maps (from linear spectral unmixing)

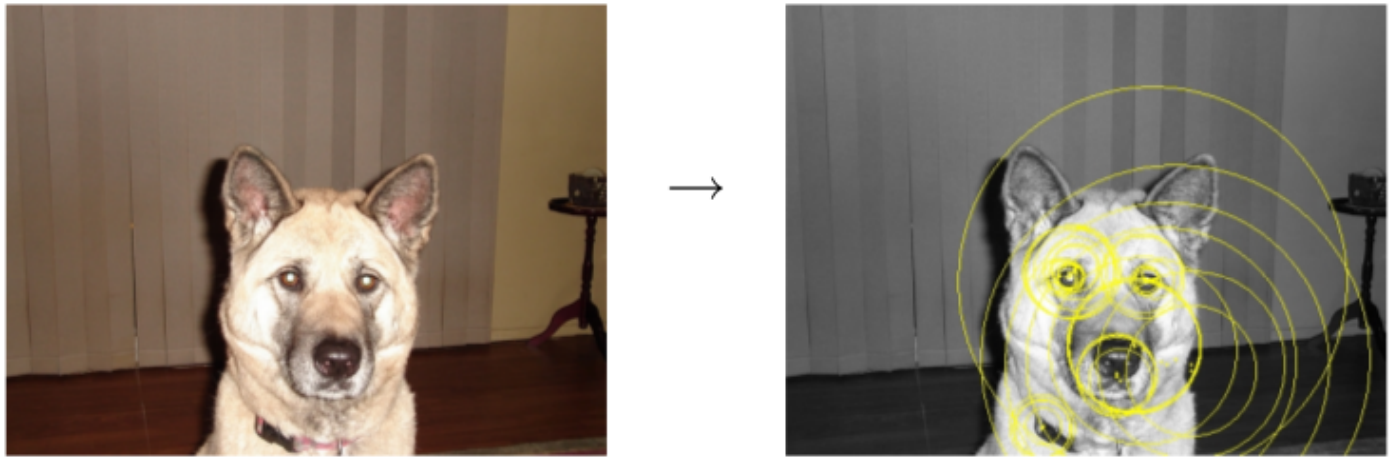


Encoding invariances via virtualization



- Work with transformed samples, not with the input examples
- Encode all kind of invariances of the data

Interest point operators

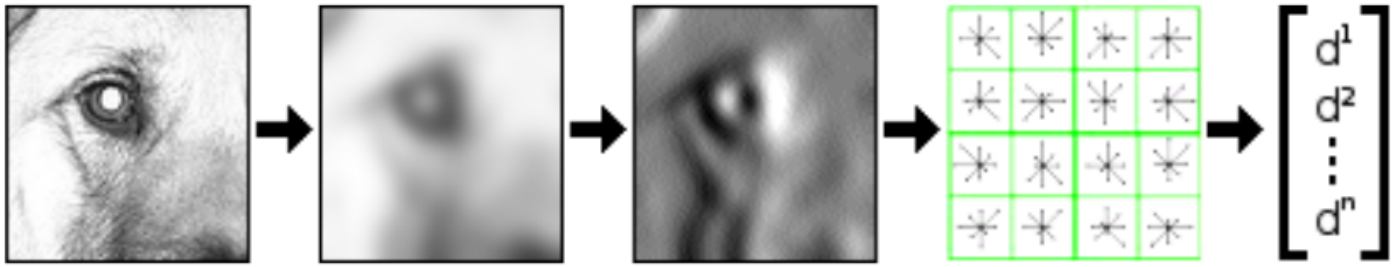


- Many methods to detect the regions of interest in images
- An image is summarized with several characteristics of interest
- Many methods: SIFT, HOG, ...

Advanced edge-based descriptors

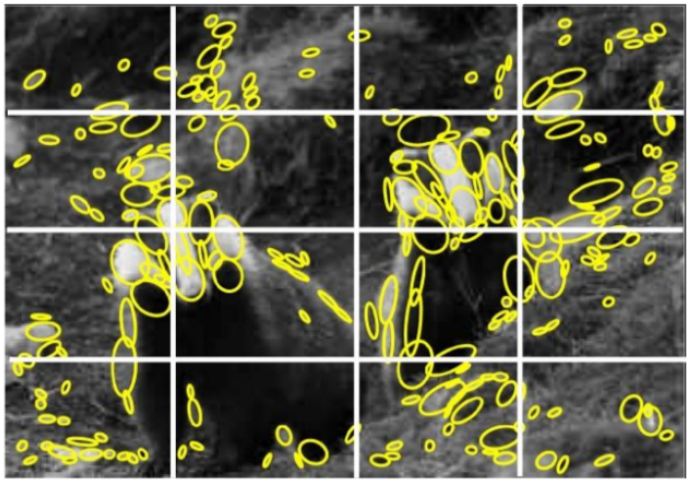
- Compute edges
- Describe directional histograms
- An image is summarized with histograms in color/shape/attention

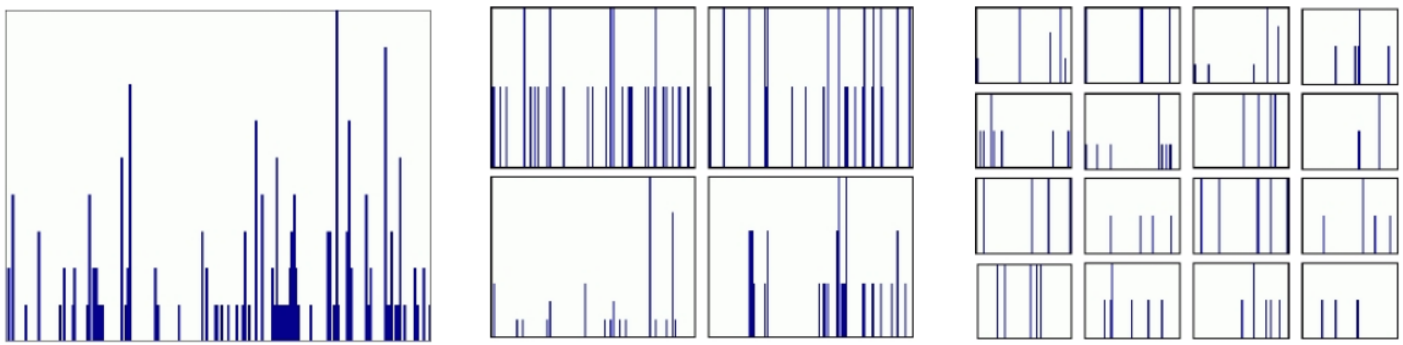
Sparse local descriptors



- Look at local features
- Compute many directional histograms
- An image is summarized with several characteristics of interest

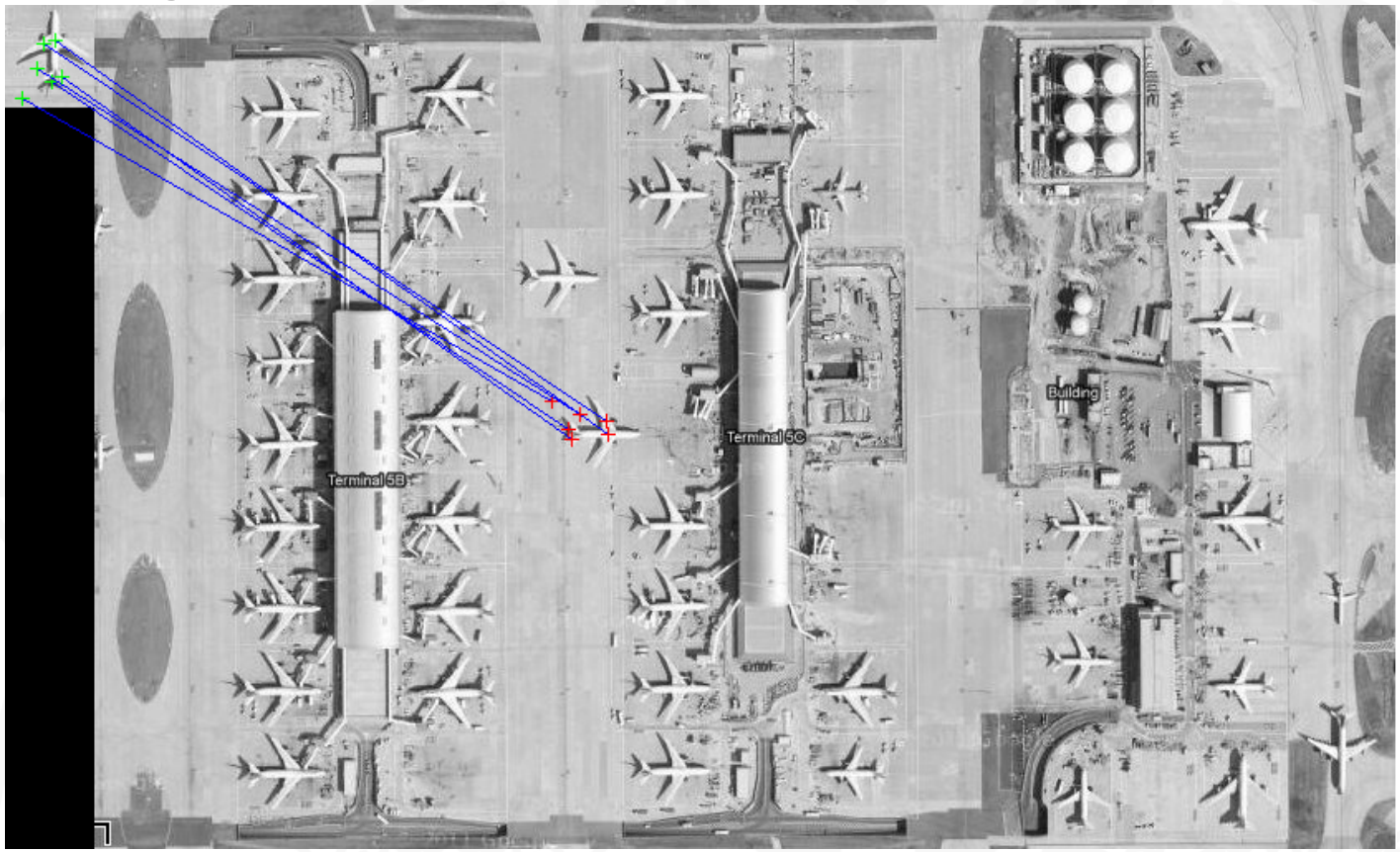
Local Bag-of-words/features Local





- Split the scene in tiles
- Identify anisotropic regions with blobs/Gaussians
- Describe the subregions with (multiscale) histograms on blobs

Corresponding pairs



- Many methods that identify corresponding pairs in images
- Related to target detection
- Useful to identify objects of interest

- Dimensionality reduction is essential before classification or regression
- High number of correlated features leads to collinearity, overfitting, and Hughes phenomenon
- Most of the spectral feature extractors are based on multivariate analysis: *"project data onto a subspace that maximize explained variance, minimize correlation, minimize error, etc."*
- Linear methods are simple and intuitive, yet often not appropriate (nonlinearity, non-Gaussianity)
- Nonlinear methods give improved expressive power

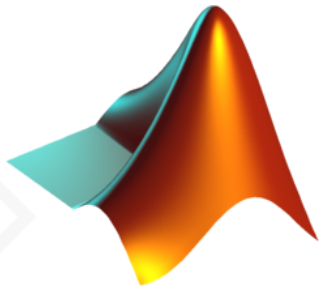
Principal component analysis (PCA)

- *"Find projections maximizing the variance of the data:"*

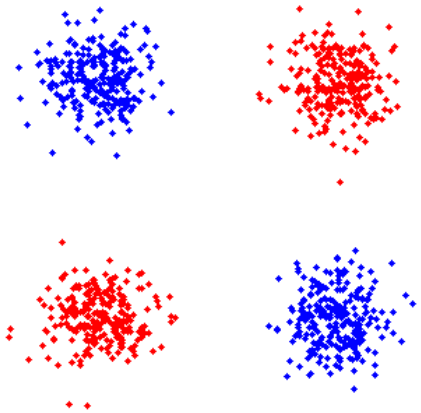
$$\begin{array}{lll} \text{PCA:} & \text{maximize:} & \text{Tr}\{(\mathbf{XU})^\top(\mathbf{XU})\} = \text{Tr}\{\mathbf{U}^\top\mathbf{C}_{xx}\mathbf{U}\} \\ & \text{subject to:} & \mathbf{U}^\top\mathbf{U} = \mathbf{I} \end{array}$$

- **The Matlab PCA code:**

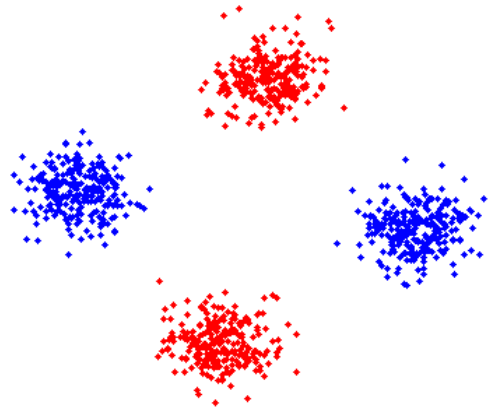
```
>> C = cov(X);
>> [U L] = eigs(C,d);
>> Xtest_projected = Xtest*U;
>> Xtest_projected = Xtest*U(:,1:np);
```



- **Pros & cons:**
  - ✓ Simplicity
  - ✓ Easy to understand
  - ✓ Leads to convex optimization problems
  - × Unsuitable for non-linear problems
  - × More dimensions than points?



Original data



PCA

Orthonormalized PLS (OPLS)

- *"OPLS chooses the projection* $\mathbf{U}$ *that minimizes the MSE error using a linear regression:"*

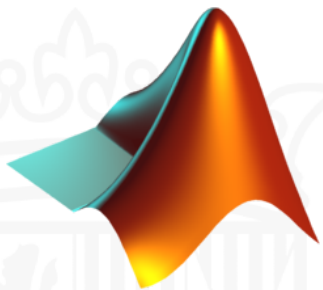
$$\begin{array}{lll} \text{OPLS:} & \text{find:} & \mathbf{U} = \arg\min\{\|\mathbf{Y} - (\mathbf{XU})\mathbf{W}\|_F^2\} \\ & \text{where:} & \mathbf{W} = (\mathbf{XU})^{\dagger}\mathbf{Y} = ((\mathbf{XU})^{\top}\mathbf{XU})^{-1}\mathbf{XUY} \end{array}$$

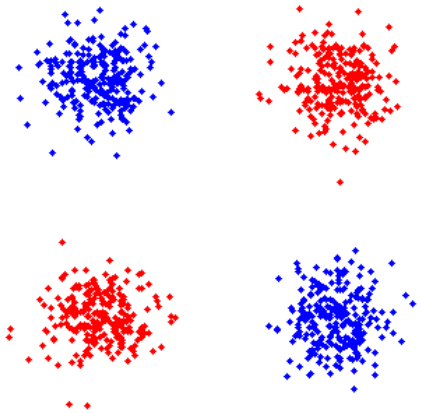
- *"... which can be rewritten as"* [Worsley98]

$$\begin{array}{lll} \text{OPLS:} & \text{maximize:} & \text{Tr}\{\mathbf{U}^{\top}\mathbf{X}^{\top}\mathbf{YY}^{\top}\mathbf{XU}\} \\ & \text{subject to:} & (\mathbf{XU})^{\top}(\mathbf{XU}) = \mathbf{I} \end{array}$$

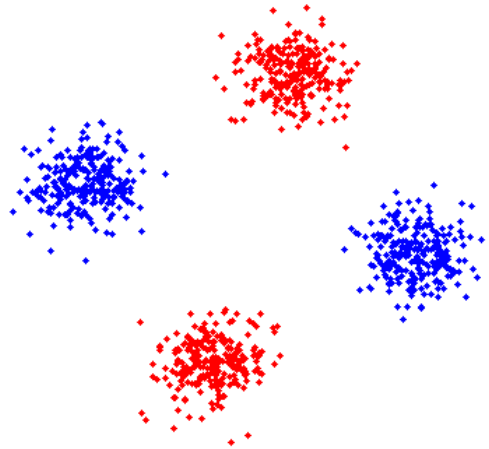
- **The Matlab OPLS code**

```
>> [U,D] = eig((X'*Y)*(Y'*X),X'*X);
>> [U,D] = eig(inv(X'*X)*(X'*Y)*(Y'*X));
>> [U,D] = eigs((X'*Y)*(Y'*X),X'*X,d);
>> Xtest_projected = Xtest*U;
>> Xtest_projected = Xtest*U(:,1:np);
```

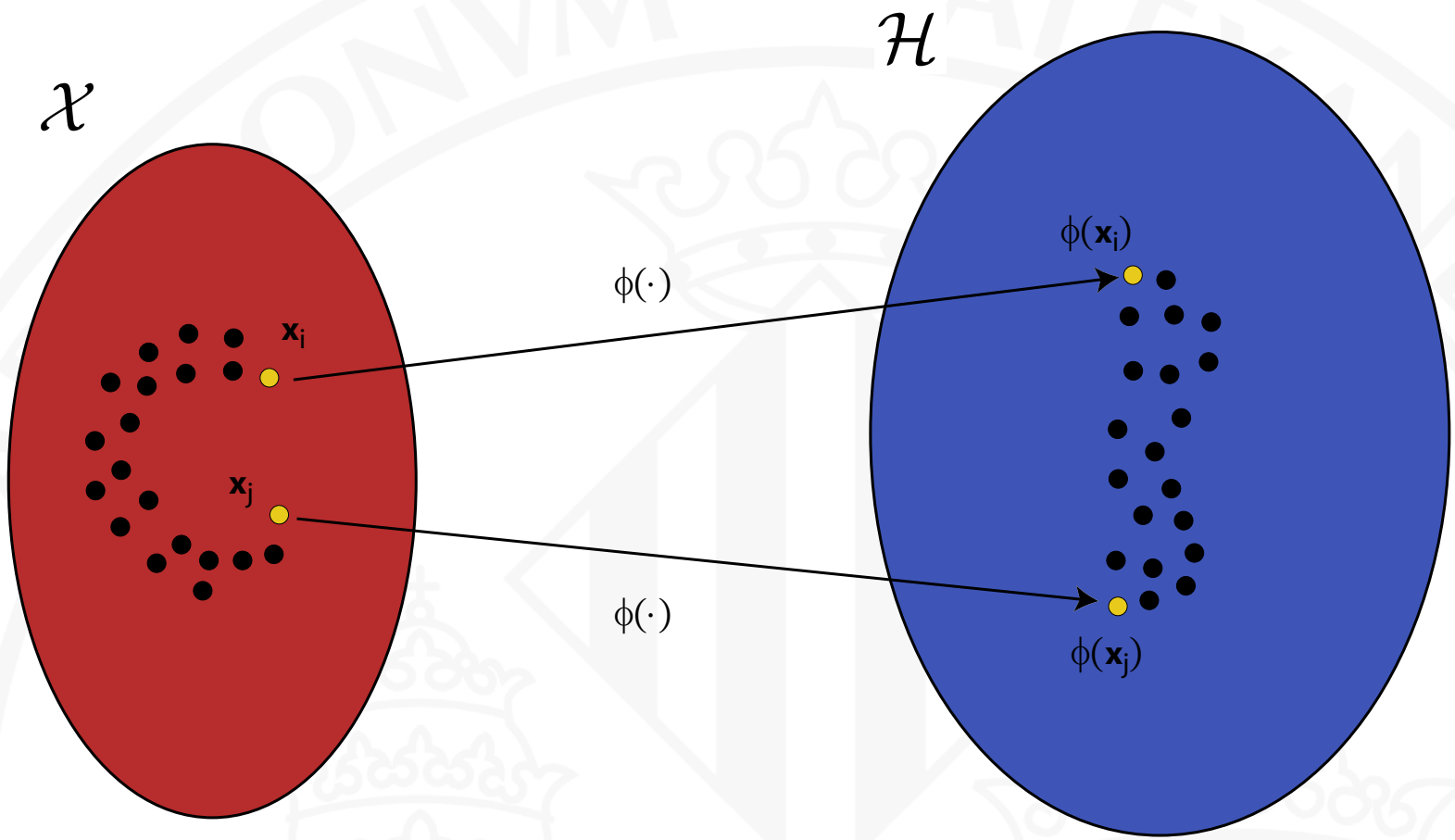




Original data



OPLS



Similarity measure

$\mathbf{x}_i^\top \mathbf{x}_j$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

1. Map the points in $\mathcal{X}$ to a higher dimensional space $\mathcal{H}$:

$$\mathbf{X} \rightarrow \mathbf{\Phi}$$

2. Express model parameters in $\mathcal{H}$ as a linear combination of mapped data

$$\mathbf{w} = \mathbf{\Phi}^\top \boldsymbol{\alpha}$$

3. Replace the dot (scalar) products by a kernel function:

$$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top$$

4. Out-of-sample predictions:

$$\mathcal{P}(\mathbf{X}_{test}) = \mathbf{\Phi}_{test}\mathbf{w} = \mathbf{\Phi}_{test}\mathbf{\Phi}^\top\boldsymbol{\alpha} = \mathbf{K}(\mathbf{X}_{test}, \mathbf{X})\boldsymbol{\alpha}$$

Valid kernels must be symmetric and positive definite similarity measures

- Linear:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$$

- Polynomial:

$$K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + 1)^d$$

- Gaussian Function (RBF):

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$$

- Hyperbolic Tangent:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \tanh(a(\mathbf{x}_i^\top \mathbf{x}_j) + b)$$

- Build new kernels...

$$K(\mathbf{x}_i, \mathbf{x}_j) = K_1(\mathbf{x}_i, \mathbf{x}_j) + K_2(\mathbf{x}_i, \mathbf{x}_j)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = K_1(\mathbf{x}_i, \mathbf{x}_j) \cdot K_2(\mathbf{x}_i, \mathbf{x}_j)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = \eta K_1(\mathbf{x}_i, \mathbf{x}_j), \quad \eta > 0$$

Kernel principal component analysis (KPCA)

- *"Find projections maximizing the variance of the mapped data"*

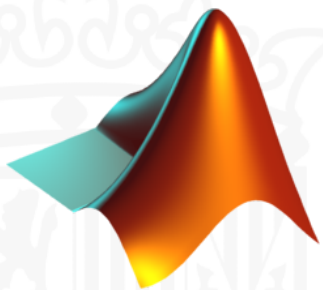
$$\begin{array}{lll} \text{KPCA:} & \text{maximize:} & \text{Tr}\{(\mathbf{\Phi U})^\top(\mathbf{\Phi U})\} = \text{Tr}\{\mathbf{U}^\top\mathbf{\Phi}^\top\mathbf{\Phi U}\} \\ & \text{subject to:} & \mathbf{U}^\top\mathbf{U} = \mathbf{I} \end{array}$$

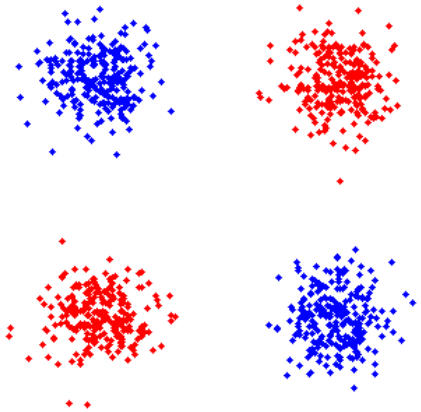
- Representer's theorem: $\mathbf{U} = \mathbf{\Phi}^\top\mathbf{A}$, $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]^\top$:

$$\begin{array}{lll} \text{KPCA (2):} & \text{maximize:} & \text{Tr}\{\mathbf{A}^\top\mathbf{K}\mathbf{K}\mathbf{A}\} \\ & \text{subject to:} & \mathbf{A}^\top\mathbf{K}\mathbf{A} = \mathbf{I} \end{array}$$

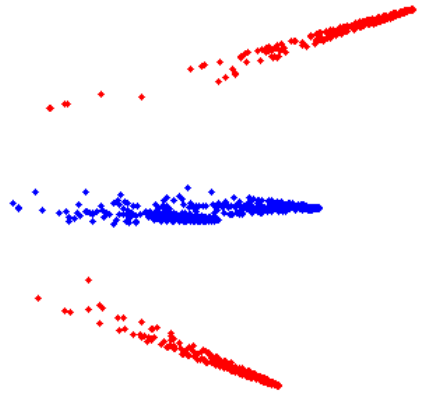
- Including Lagrange multipliers $\mathbf{\Lambda}$: $\mathbf{KA} = \mathbf{\Lambda A}$
- Project new data: $\mathcal{P}(\mathbf{X}_*) = \mathbf{\Phi}_*\mathbf{U} = \mathbf{\Phi}_*\mathbf{\Phi}^\top\mathbf{A} = \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{A}$
- **The Matlab KPCA code**

```
>> K = kernelmatrix('rbf',X,X,sigma);
>> K = kernelcentering(K);
>> [A L] = eigs(K,n);
>> Ktest = kernelmatrix('rbf',Xtest,X,sigma);
>> Xtest_projected = Ktest*A;
>> Xtest_projected = Ktest*A(:,1:np);
```





Original data



KPCA

Kernel Orthonormalized Partial Least Squares (KOPLS)

- *"Choose the projection that minimizes the MSE:"* [Worsley98]

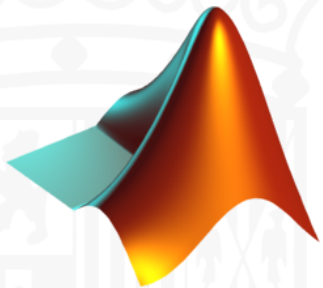
$$\begin{aligned} \text{KOPLS:} \quad & \text{maximize:} \quad \text{Tr}\{(\mathbf{\Phi U})^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{\Phi U}\} \\ & \text{subject to:} \quad (\mathbf{\Phi U})^\top \mathbf{\Phi U} = \mathbf{I} \end{aligned}$$

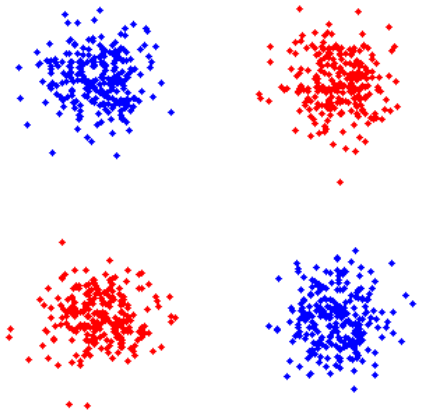
- Representer's theorem: $\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$, $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]^\top$:
- Including Lagrange multipliers $\mathbf{\Lambda}$, this problem is equivalent to

$$\begin{aligned} \text{KOPLS:} \quad & \text{maximize:} \quad \text{Tr}\{\mathbf{A}^\top \mathbf{K}_x \mathbf{K}_y \mathbf{K}_x \mathbf{A}\} \\ & \text{subject to:} \quad \mathbf{A}^\top \mathbf{K}_x \mathbf{K}_x \mathbf{A} = \mathbf{I} \end{aligned}$$

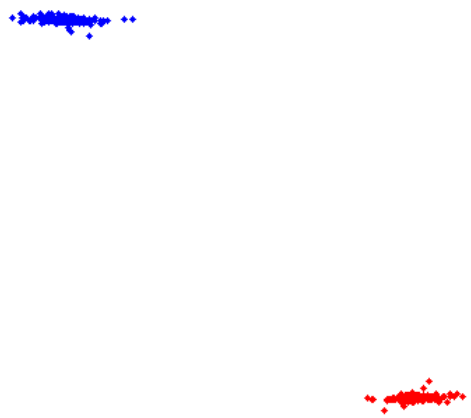
- This is a generalized eigenproblem: $\mathbf{K}_x \mathbf{K}_y \mathbf{K}_x \mathbf{A} = \mathbf{\Lambda} \mathbf{K}_x \mathbf{K}_x \mathbf{A}$
- Project new data: $\mathcal{P}(\mathbf{X}_*) = \mathbf{\Phi}_* \mathbf{U} = \mathbf{\Phi}_* \mathbf{\Phi}^\top \mathbf{A} = \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{A}$
- **The Matlab KOPLS code**

```
>> Kx = kernelmatrix('rbf',X,X,sigma);
>> Kx = kernelcentering(K);
>> Ky = Y*Y';
>> Ky = kernelcentering(Ky);
>> [A, L] = eigs(Kx*Ky*Kx,Kx*Kx,n);
>> Xtest_projected = K(Xtest,X)*A;
>> Xtest_projected = K(Xtest,X)*A(:,1:np);
```



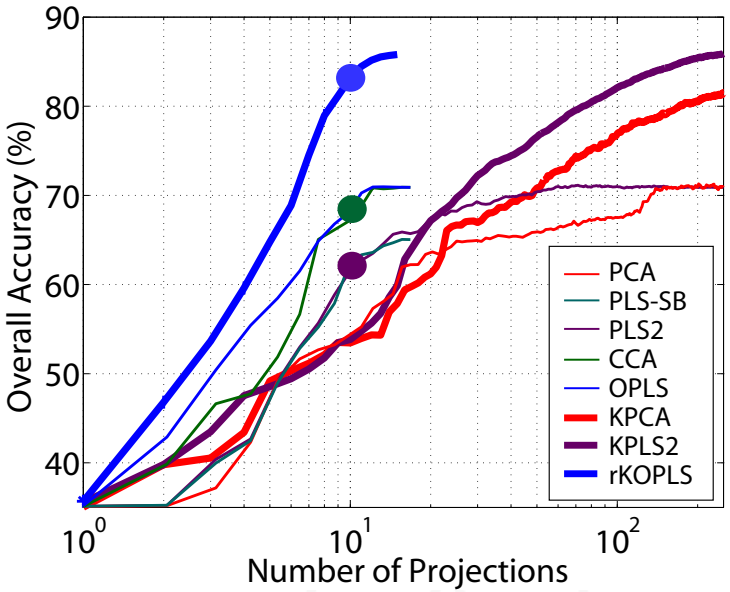


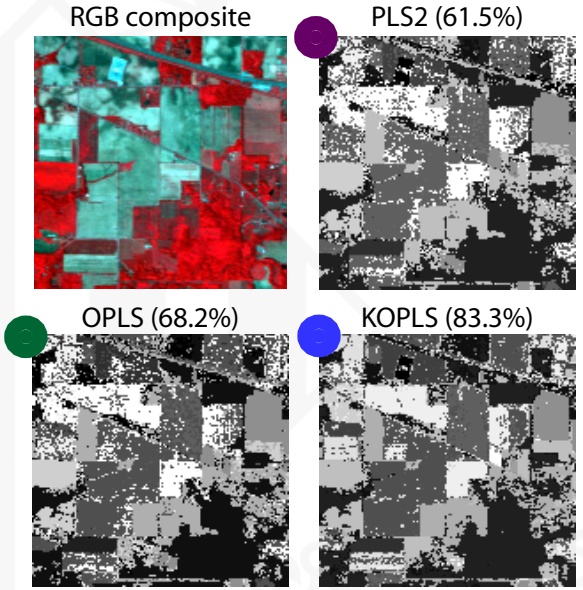
Original data



KOPLS

- **Data:**
  - AVIRIS image taken over NW Indiana's Indian Pine test site in June 1992
  - 145 $\times$ 145 image size, 220 features (bands), 16 land cover classes
  - 80% for training and 20% for testing
  - Classifier: linear classifier on top of different number of features
- **Results:**





  - Supervised feature extraction often better than unsupervised
  - Higher accuracies lead to smoother maps
  - kOPLS excels in performance, needs few components
  - kOPLS reduce false alarm rates in large homogeneous vegetation areas

- Extracting features from remote sensing images is essential to:
  - Compress information for storage/transmission
  - Reduce (spatial and spectral) redundancy
  - Visualize data characteristics
- Spectral features rely either on physical prior knowledge or statistical techniques that optimize a sensible criterion
- Spatial features rely on image processing operations builing on the classical smoothness assumption in the image space
- Linearity and Gaussianity are strong assumptions in general
- Nonlinear methods using kernels can be convenient due to high robustness to low-sized datasets and high input space dimensionality

Part 4: Neural networks in remote sensing

History (I). conventional techniques

- Poynting (1884) invents a predictive moving average (MA) model
- Hooker (1901), Spencer (1904), Anderson (1914), Nochmals (1915) generalize the results to higher order polynomials
- Use in census and exchange stock market applications until 1920

History (II). classic techniques

- Yule (1927) develops the auto-regressive (AR) model
- Smoothers (exponential, polynomic, ...)
- Decomposition methods.
- Wahba (1970): Spline smoothing.
- Box & Jenkins (1972): methodology for ARMA identification
- "Threshold Autoregressive Model" (TAR).
- Hastie (1979): MARS.

History (III). Modern techniques.

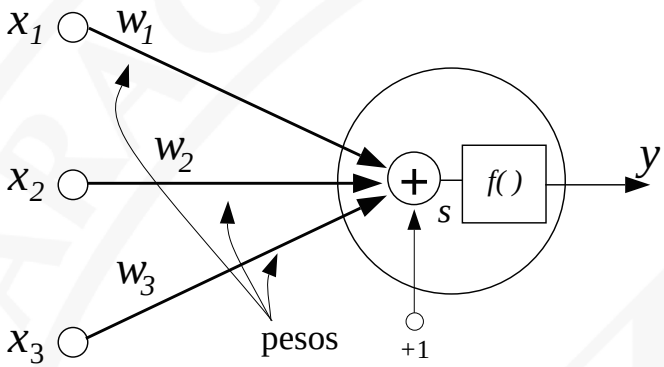
- Rumelhart & McClelland (1984): formalize MLP.
- Elman (1988): recurrent networks
- Lapedes & Farber (1990): chaotic time series prediction
- Waibel et al. (1989): TDNN.
- Competiciï¿½n de Santa Fe (1991): FIR networks.
- Principe (1992): IIR/gamma nets
- Vapnik, Schölkopf, Smola, Cristianini (1991-2000): SVM, SVR, kernels
- Neal (1996): Relation between neural nets and Gaussian Processes.
- Tipping (2001): RVM

History (IV). Latest developments.

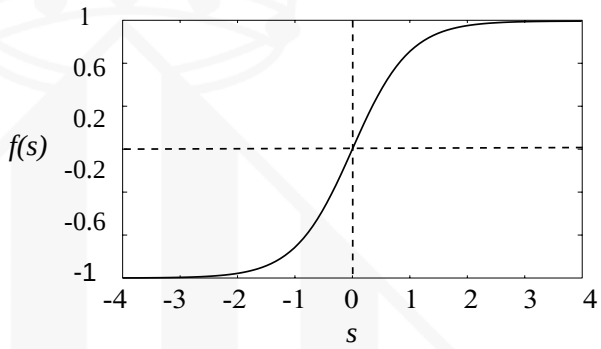
- LeCun, Hinton, Bengio (2002-2012): Deep neural nets
- Reichstein and Camps-Valls (2019): Physics-aware neural networks

A model for a static single neuron

- McCulloch and Pitts (1943) introduce the first artificial neuron model



(a)



(b)

- Output is a weighted sum of inputs

$$y = sgn(f(s)) = sgn\left(\sum_{i=1}^{N} w_i x_i + w_0\right)$$

An early nonlinear function

- In 1958 Rosenblatt presents a new approximation: the nonlinear extension (*sigmoid*):

$$f(s) = 1/(1 + e^{-s}) \tag{2}$$

- Neurons are also known as *adalines* or *perceptrons*
- Weights are adjusted by minimizing the error $e(n)$, summarized as a squared loss function:

$$J(k) = \sum_{n=1}^{N} e(n)^2, \tag{3}$$

where

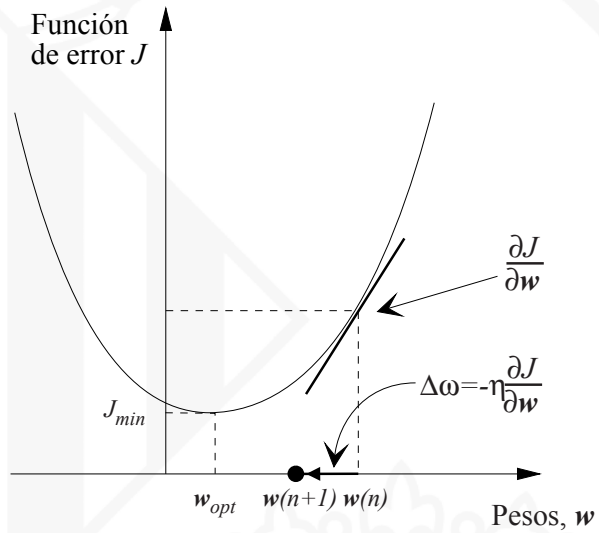
$$e(n) = d(n) - y(n). \tag{4}$$

... and the method to fit the weights!

1. *Iteration* $k = 0$. *Init weights.*
2. *Iteration* $k = k + 1$.
3. *Compute output with* (2).
4. *Compute instantaneous error with* (4).
5. *Descend along the error surface:*

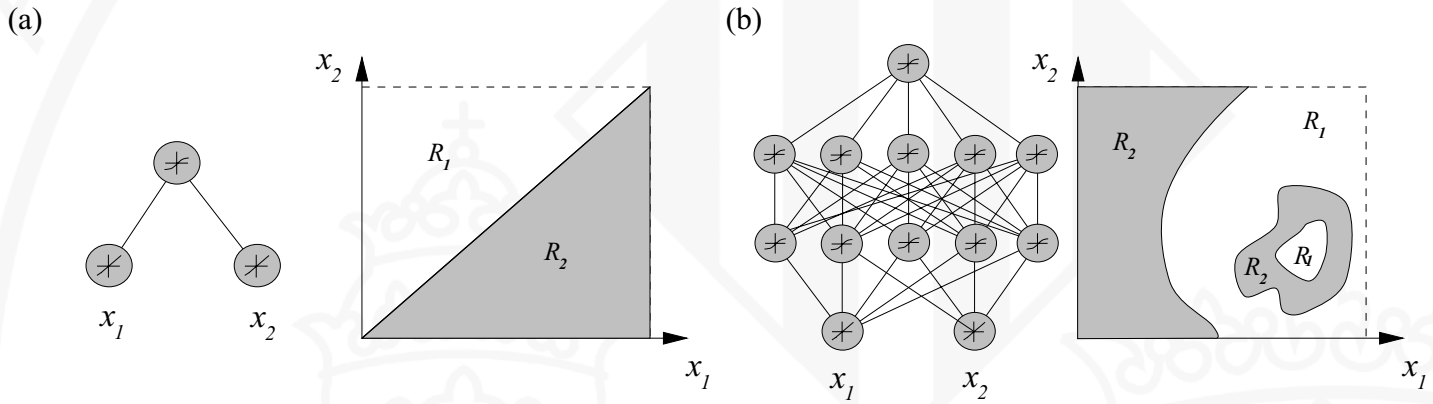
$$\Delta \mathbf{w} = -\eta \nabla J = -\eta \frac{\partial J}{\partial \mathbf{w}}$$

6. $\mathbf{w}^{k+1} = \mathbf{w}^k + \eta e(k) x(k)$
7. *Back to step 2 until convergence*



A bit more of history...

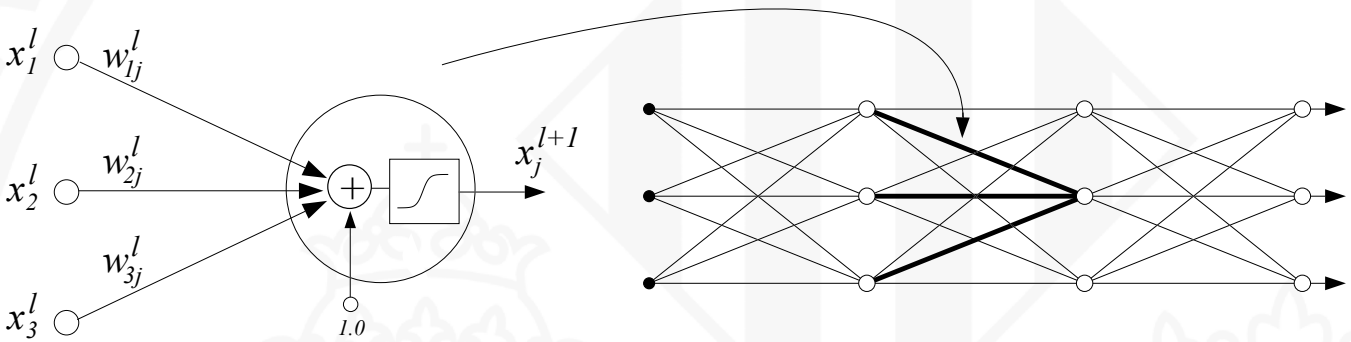
- Minsky and Papert (1969): *"perceptrons are not capable to resolve nonolinear separable problems!"*
- Rumelhart, Hinton and Williams (1986):
  1. Backpropagation algorithm
  2. Multilayer perceptron approximates arbitrary functions
  3. Werbos demonstrated this 12 years before!



- Many developments, network architectures and applications

Multilayer network and backpropagation

- **Multilayer Perceptron, MLP:** total or partially connected structure, organized in layers
- Signals are propagated from input to output



Multilayer network and backpropagation

- Input: $\mathbf{x} = [x_0, x_1, \ldots, x_N]$
- Output: $\mathbf{y} = [y_0, y_1, \ldots, y_M]$
- Each neuron's output: $x_j^{l+1} = f\left(\sum_i w_{i,j}^l x_i^l\right)$
- *Bias* in a neuron $x_0^l = \pm 1$
- Take x_i^0 as external inputs, and x_i^L as network outputs
- A neural net does a complex mapping $\mathcal{M}(\mathbf{x},\mathbf{w})$ where model weights $\mathbf{w}$ are learned from data
- Training is done with *"backpropagation"* (BP)

w_{ij}^l	weight connecting neuron i in layer $l-1$ to neuron j in layer l
w_{bj}^l	bias in neuron j in layer l
$s_j^l = \sum_i w_{ij}^l a_i^{l-1} + w_{bj}^l$	sum unit j in layer l
$a_j^l = tanh(s_j^l)$	output of neuron j in layer l
$x_i = a_i^0$	i-th external input in the net
$y_i = a_i^L$	i-th output in the net

Multilayer network and backpropagation

- Supervised learning by error correction using pairs of input-output data: $\{(\mathbf{x}_1, d_1), ..., (\mathbf{x}_P, d_P)\}$.
- The error vector is:

$$\mathbf{e} = \mathbf{d} - \mathbf{y}.$$

- We sum over all examples and outputs:

$$J = \frac{1}{P} \sum_{p=1}^{P} e_p^2,$$

- Gradient descend to fit the weights

$$\Delta \mathbf{w} = -\eta \nabla_J = -\eta \frac{\partial J}{\partial \mathbf{w}} = -\eta \frac{\partial \mathbf{e}^T \mathbf{e}}{\partial \mathbf{w}} \tag{5}$$

where η is a *learning rate* controling how fast we learn

Multilayer network and backpropagation

1. *For $k = 0$, we randomly initialize weights*
2. *Propagate examples through net $x_p = a_p^0$:*

$$s_j^l = \sum_i w_{ij}^l a_i^{l-1} + w_{bj}^l, \quad a_j^l = tanh(s_j^l) \tag{6}$$

3. *Compute for all output neurons:*

$$\delta_j^l \equiv \frac{\partial J_p}{\partial s_j^l} = f'(s_i^l) \sum_j \delta_j^{l+1} w_{ij}^{l+1}, \tag{7}$$

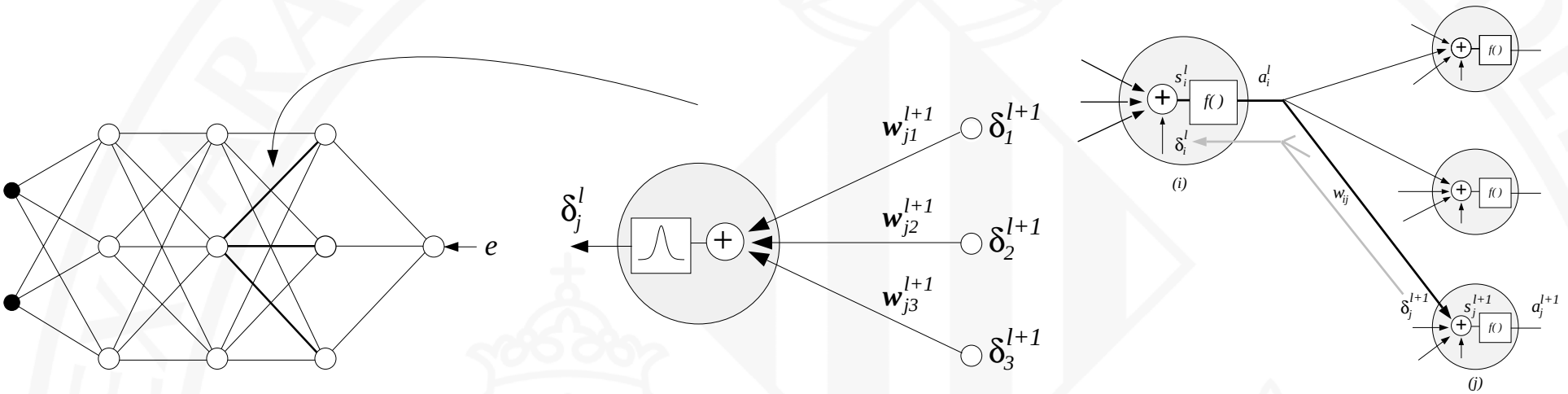
4. *Backpropagate the instantaneous errors for each neuron*

$$\delta_i^l = \begin{cases} -2e_i f'(s_i^L), & l = L. \\ f'(s_i^l) \cdot \sum_j \delta_j^{l+1} w_{ij}^{l+1}, & 1 \leq l \leq L-1, \end{cases} \tag{8}$$

5. *Evaluate updating weigthes: $\Delta w_{ij}^l = -\eta \delta_j^l a_i^{l-1}$*
6. *Back to Step 1 until convergence*

Multilayer network and backpropagation

- Subsequent delta values are computed using the chain rule



- Oher methods exist: *conjugate gradient*, Hessian approximations, Levenberg–Marquardt, etc.

- **Clasification.**
  1. Binary coding of the output ($y \in \{0, 1\}$).
  2. Multiclassification: as many output neurons as classes
- **Regression.**
  1. Real output, $y \in \mathbb{R}$.
  2. Useful when there's no time relations between the input-outputs
- **Prediction.**
  1. When we have input-output time relation, just delay inputs to define $\mathbf{x} := [x(t-1), x(t-2), ..., x(t-p)]$, where p defines a temporal window (time embedding).
  2. Do one-step ahead prediction so $\mathbf{y} = x(t)$.
  3. Compute the error as always:

$$\mathbf{e} = \mathbf{d} - \mathbf{y} \rightarrow e(t) = x(t) - \hat{x}(t), \tag{9}$$

where $\hat{x}(t) = \mathcal{N}([x(t-1), x(t-2), ..., x(t-p)], \mathbf{w})$.

Training standard neural networks

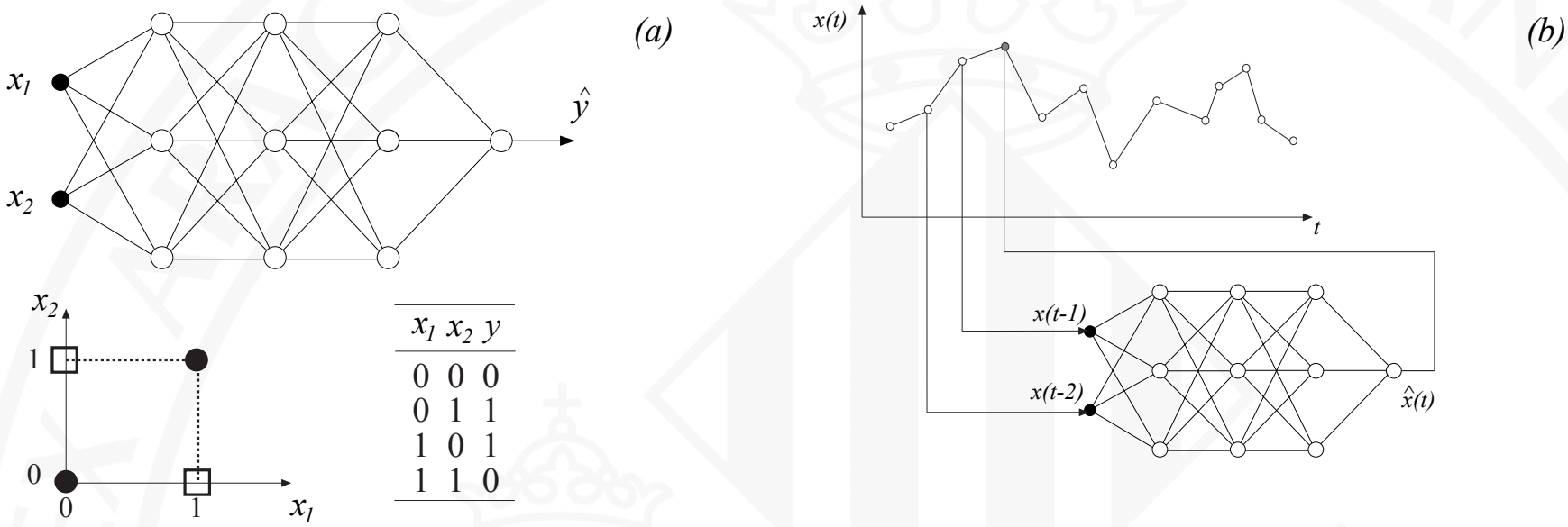
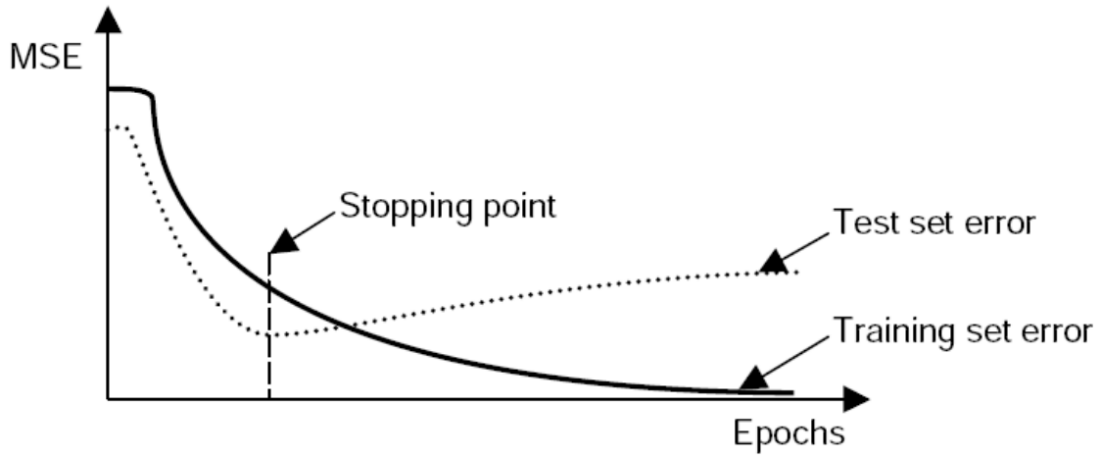


Figure : MLP configurations.

Training standard neural networks

- Split data in two sets:
  - Training: 2/3
  - Test: 1/3
- Fit/learn weights to the minimum test error
- Parameters to adjust:
  - How to initialize weights $\mathbf{w}_o$
  - Learning rate η
  - Number of epochs or iterations N_e
  - Network structure: layers ℓ and neurons/layer N_ℓ



MLPs

- Provide a mapping from $\mathcal{X} \rightarrow \mathcal{Y}$, i.e. from a features space (usually $\mathcal{X} \equiv \mathbb{R}^n$) to a label space $\mathcal{Y}$
- Are based on concatenation of "simple" functions that depend on parameters (i.e. weights)
- Are optimized by gradient descent (and its modern extensions)

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

MLPs

- Provide a mapping from $\mathcal{X} \to \mathcal{Y}$, i.e. from a features space (usually $\mathcal{X} \equiv \mathbb{R}^n$) to a label space $\mathcal{Y}$
- Are based on concatenation of "simple" functions that depend on parameters (i.e. weights)
- Are optimized by gradient descent (and its modern extensions)
- Work great, BUT:

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

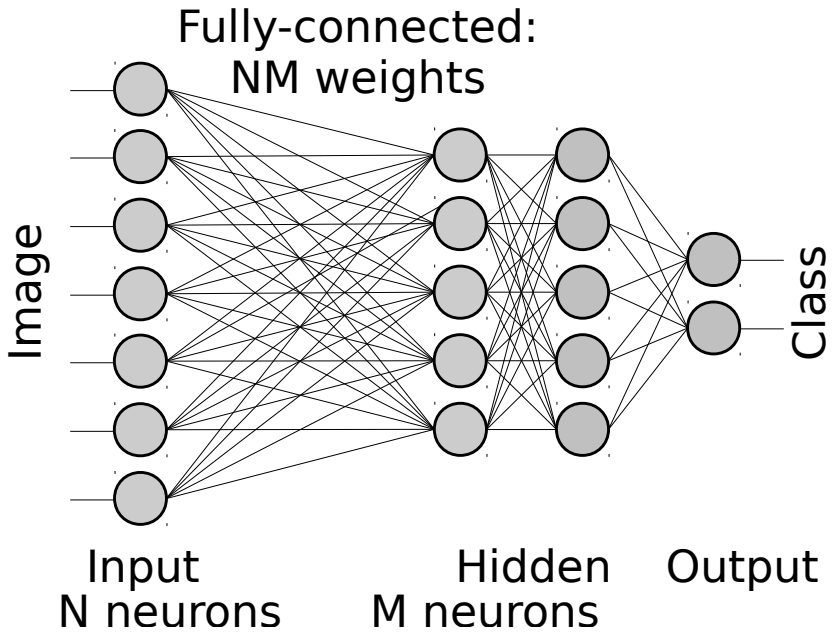
- Used by
  - Facebook: Automatic tagging
  - Google: Photo search
  - Amazon: Recommendations
  - Pinterest: Home feed personalization
  - Instagram: Search
- Buzzwords
  - Deep Learning, Deep Networks
  - Convolutional Neural Networks (CNNs), Convolutional Networks (ConvNets)
  - Note: There are more Deep Networks / Deep Learning approaches than ConvNets

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

"CNNs are inspired by biological principles in the visual cortex."

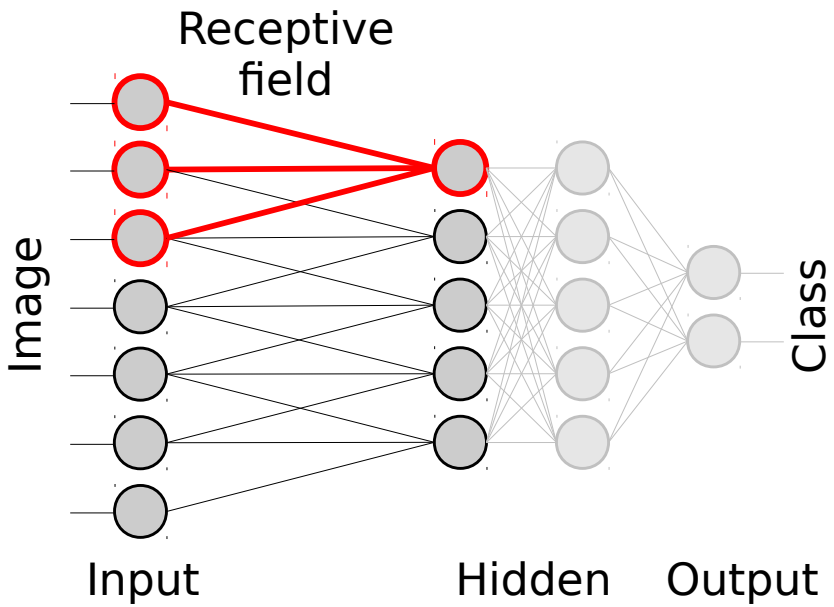
- Small regions of cells sensitive to specific regions within the visual field.
- 1962, Hubel and Wiesel
  - Neuronal cells fire only in the presence of certain structures e.g. edges of a specific orientation
  - Organized in columns
- Good selling point, BUT:
  - Extracting image features is neither new, nor the main point of ConvNets
  - Training works very differently

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka



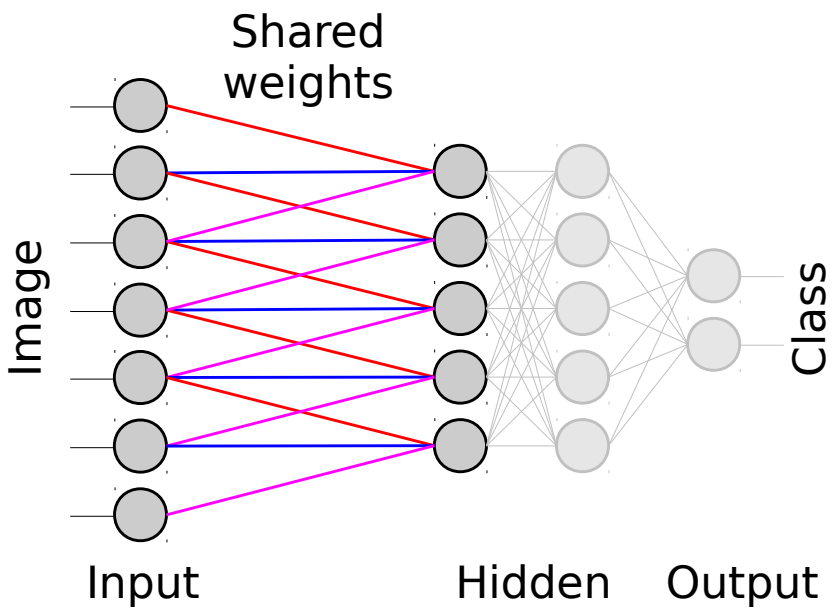
- Multiple layers of units
- All-to-all connection between two adjacent layers
- No lateral connections
- A tremendous amount of parameters in case of images $\rightarrow$ Untrainable

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka



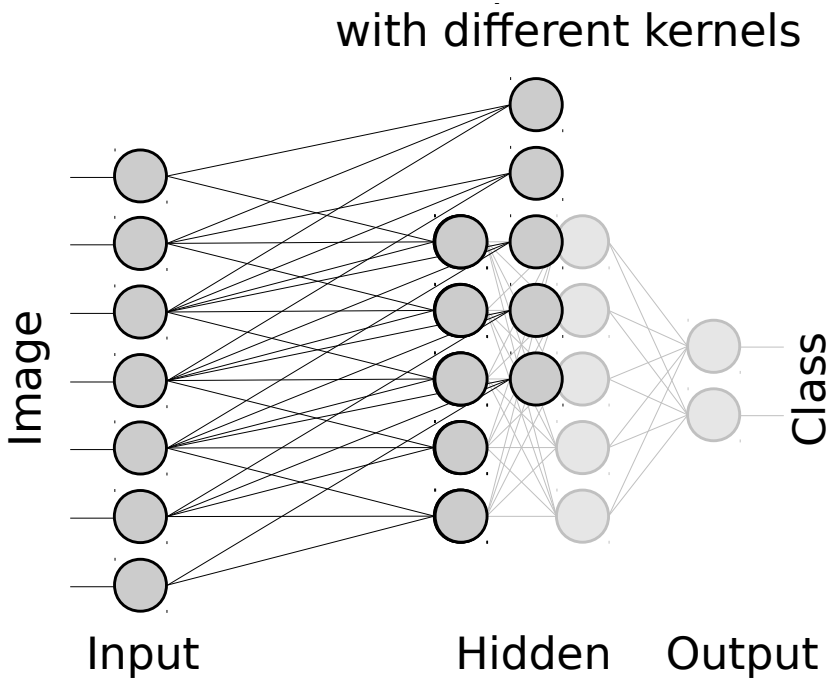
- Set most weights to zero and thus delete most connections and decrease parameters.

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka



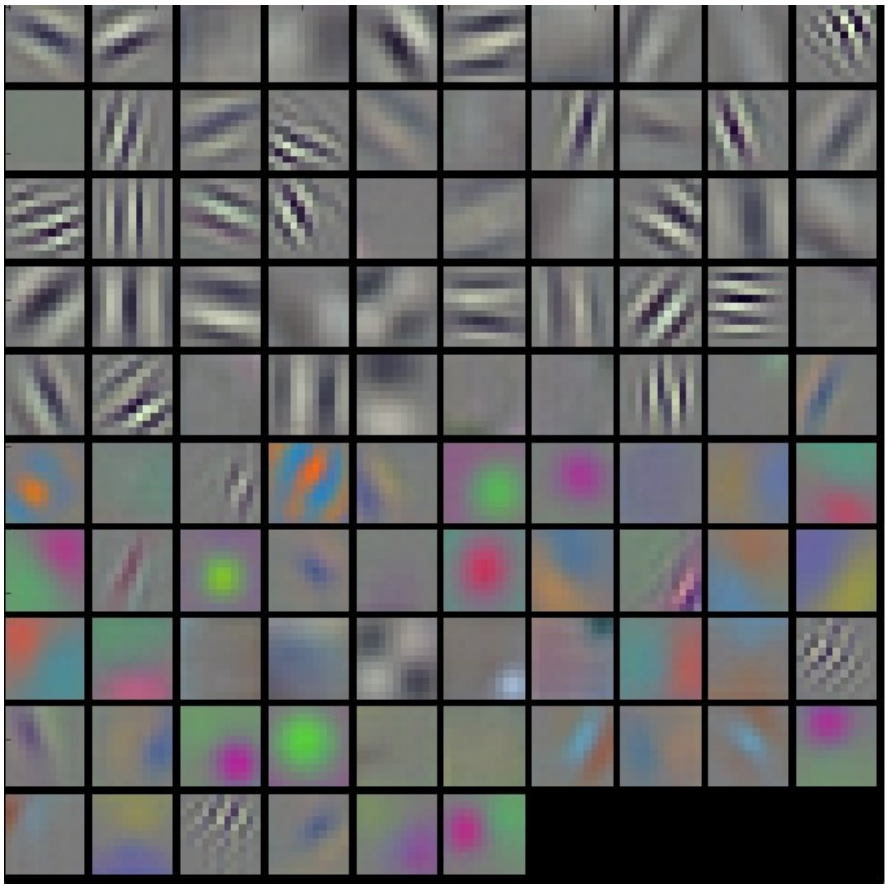
- Set most weights to zero and thus delete most connections and decrease parameters.
- Use same values for weights of different neurons within a layer.
- The multiplication of the input with identical weights for different neurons corresponds to a convolution.
- The kernel of this convolution is automatically learned.

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka



- Set most weights to zero and thus delete most connections and decrease parameters.
- Use same values for weights of different neurons within a layer.
- The multiplication of the input with identical weights for different neurons corresponds to a convolution.
- The kernel of this convolution is automatically learned.
- Use multiple convolutional layers to enable different kernels to be learned.

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka



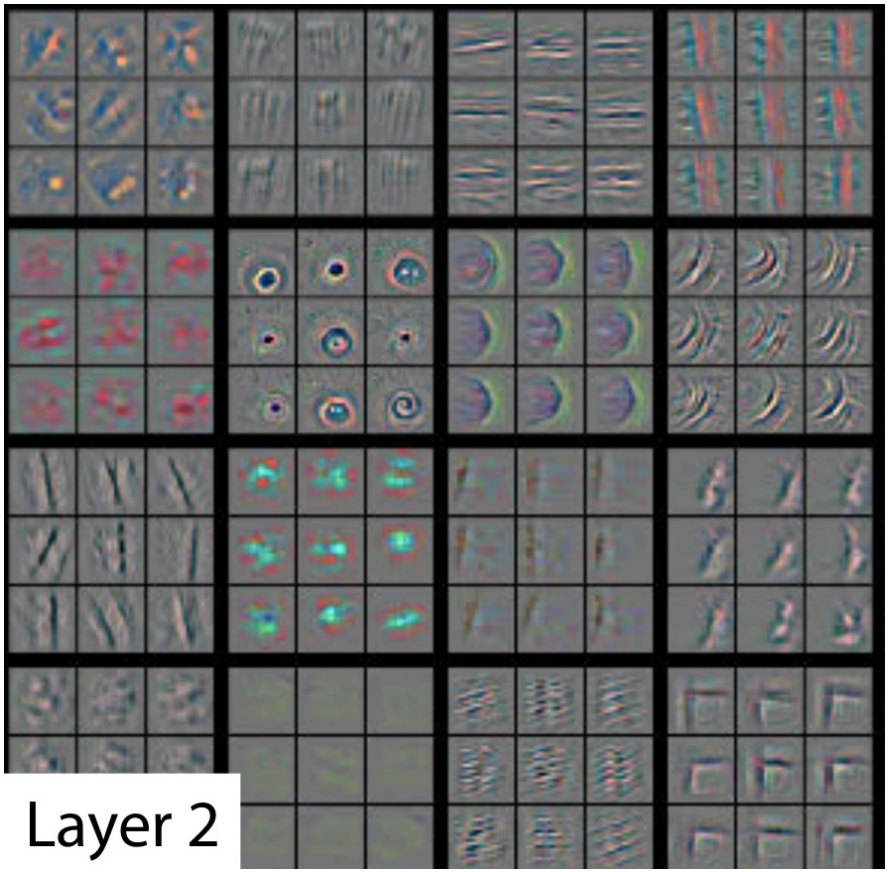
- Learned kernels of first convolutional layer of a ConvNet (AlexNet).
- Correspond mostly to edges and corners of different orientations.
- Note: Grouping is caused by network architecture (two independent streams were used to handle the large amount of data).

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Layer 1

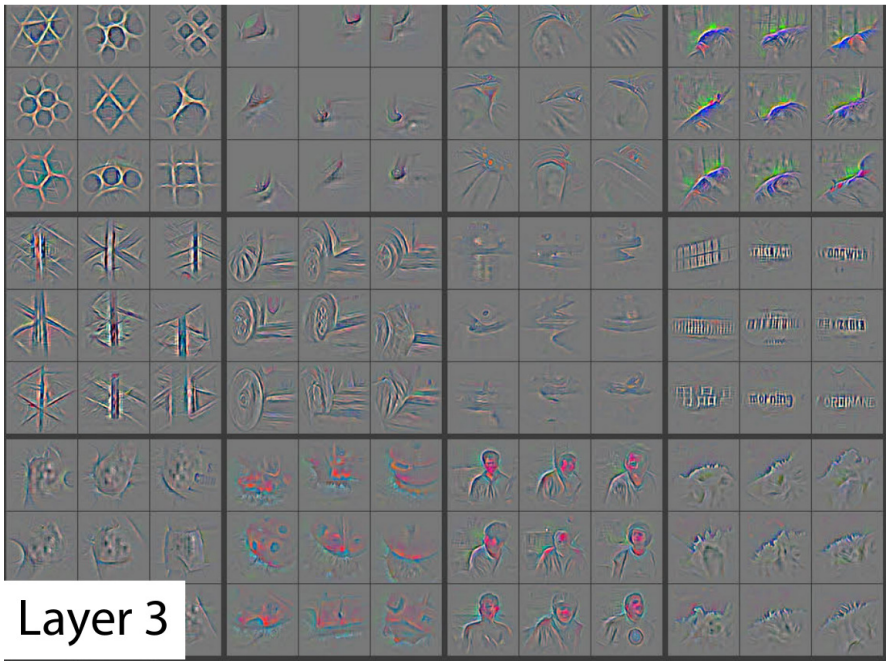
- Top nine activations in feature maps
- Projected to pixel space using a deconvolutional network
- Reconstructed patterns that cause high activations
- Note: Images taken from [Zeiler and Fergus, 2013].

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka



- Top nine activations in feature maps
- Projected to pixel space using a deconvolutional network
- Reconstructed patterns that cause high activations
- Note: Images taken from [Zeiler and Fergus, 2013].

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka



- Top nine activations in feature maps
- Projected to pixel space using a deconvolutional network
- Reconstructed patterns that cause high activations
- Note: Images taken from [Zeiler and Fergus, 2013].

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

- Top nine activations in feature maps
- Projected to pixel space using a deconvolutional network
- Reconstructed patterns that cause high activations
- Note: Images taken from [Zeiler and Fergus, 2013].

LeNet (1998)

- One of the first successful applications of ConvNets
- Digital digit / character recognition

AlexNet (2012)

- Started the hype
- Similar to LeNet, but deeper and bigger
- Stacked conv-layers
- Image classification (ImageNet Large-Scale Visual Recognition Challenge)
- Trained on 15 million annotated images from over 22,000 categories
- Trained on two GTX 580 GPUs for five to six days

ZF Net (2013)

- Similar to AlexNet
- Trained on 1.3 million annotated images
- Trained on a GTX 580 GPU for twelve days

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

VGG Net (2014)

- Simple and deep: Only 3x3 filters and 2x2 pooling
- Stacked conv-layers to increase effective receptive field size
- Used Caffe toolbox
- Trained on 4 Nvidia Titan Black GPUs for two to three weeks

GoogLeNet (2015)

- 22 layers
- Proposed inception module: Running multiple filter operations in parallel
- 12x fewer parameters than AlexNet
- Trained on multiple high-end GPUs for a week

Microsoft ResNet (2015)

- 152 layers
- Trained on an 8 GPUs for two to three weeks
- 3.6% error on ImageNet LSVRC (AlexNet: 15.4%)

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

- Caffe:
  - http://caffe.berkeleyvision.org/
  - https://github.com/BVLC/caffe
- Tensorflow:
  - https://www.tensorflow.org/
  - https://github.com/tensorflow
- Torch:
  - http://torch.ch/
  - https://github.com/torch/torch7
- Matlab:
  - Official toolbox: https://es.mathworks.com/solutions/deep-learning.html
  - External: `MatconvNet`: http://www.vlfeat.org/matconvnet/

- Increasing amount & openness of data, e.g.:
  - Pléiades: entire earth every day ($<$ 1 m resolution)
  - USGS public domain aerial images

  $\Rightarrow$ Scalability: temporal/space complexity
- Intra-class variability:

Chicago

Vienna

Austin

- Interest in semantic classes (e.g., *building*, *road*, *lane*)

  $\Rightarrow$ Need for high-level contextual reasoning (shape, patterns,...)

  $\Rightarrow$ Generalization to different locations

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Pixelwise

- Neural networks (Goel et al., 2003; Ratle et al., 2010)
- Random forests (Ham et al., 2005)
- SVMs (Camps-Valls et al., 2006)

Graph-based

- Partition trees (Valero, 2010)
- Minimum spanning forest (Bernard et al., 2012)
- Graph cut (Tarabalka et al., 2014)

Feature engineering

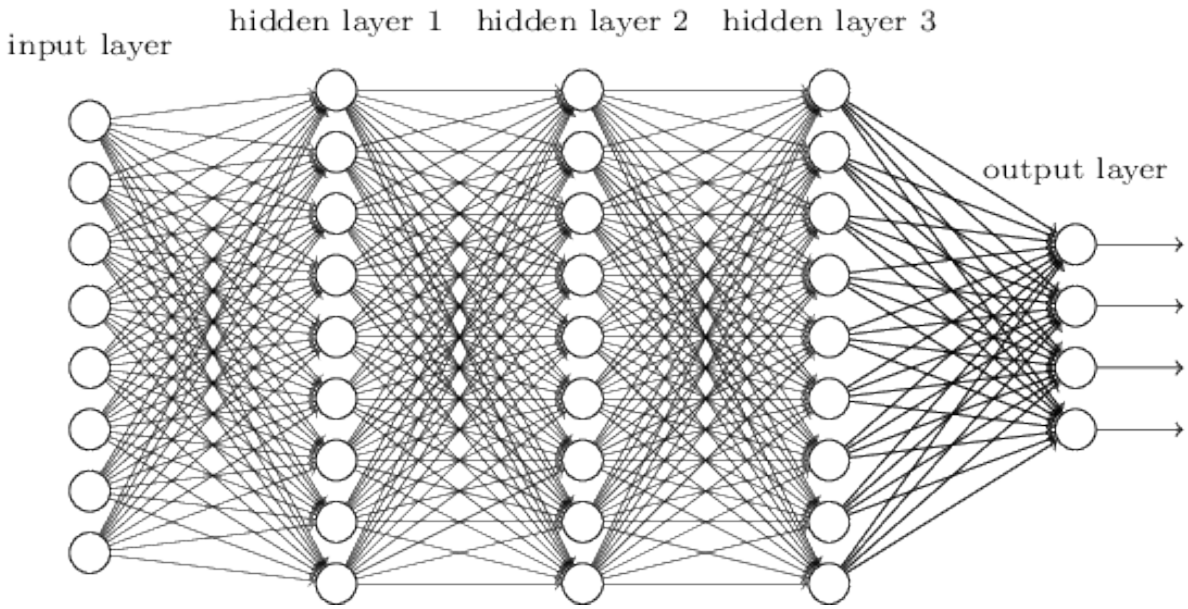
- Morphological profiles + SVMs (Fauvel et al., 2008)
- Texture + SVMs (Huang et al., 2008)
- Multiple features (Pal & Foody, 2010; Zhang et al, 2012)

Deep learned features

- Convolutional neural networks (Mnih & Hinton, 2012)
- Deep features + SVMs (Chen et al., 2014)
- Fully convolutional networks (Marmanis et al., 2016; Volpi & Tuia, 2017)

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Multilayer perceptron (MLP)

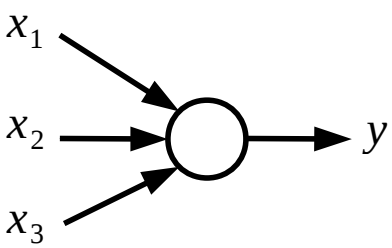


Features →

Classif. scores (e.g., "dog: 0.9")

Fully connected neuron layers

Neuron



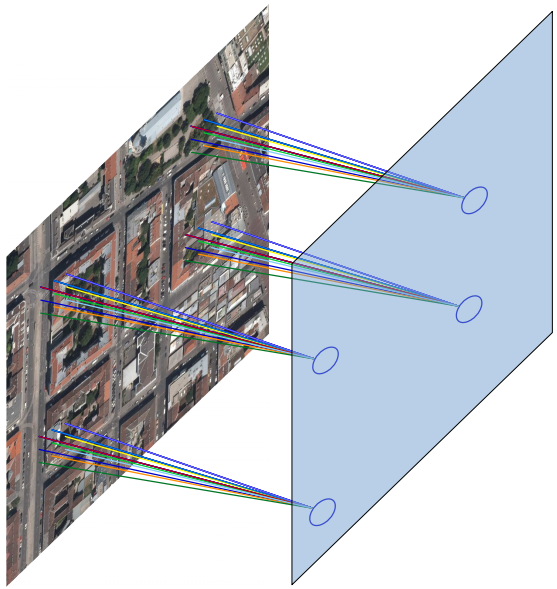
- $y = \sigma(\sum a_i x_i + b)$, σ nonlinear
- Parameters (a_i, b of all neurons) define the function
- Trained from samples by stoch. gradient descent

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

- Input: the image itself
- {*Convolutional* layers + *pooling* layers}* + MLP

Convolutional layer

Learned convolution filters $\rightarrow$ feature maps



Special case of fully connected layer:

- Only local spatial connections
- Location invariance

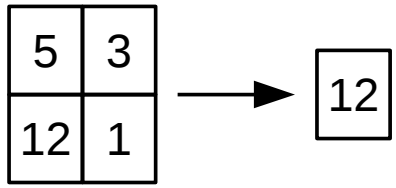
$\Rightarrow$ Makes sense in image domain (or text, time series,...)

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Pooling layers

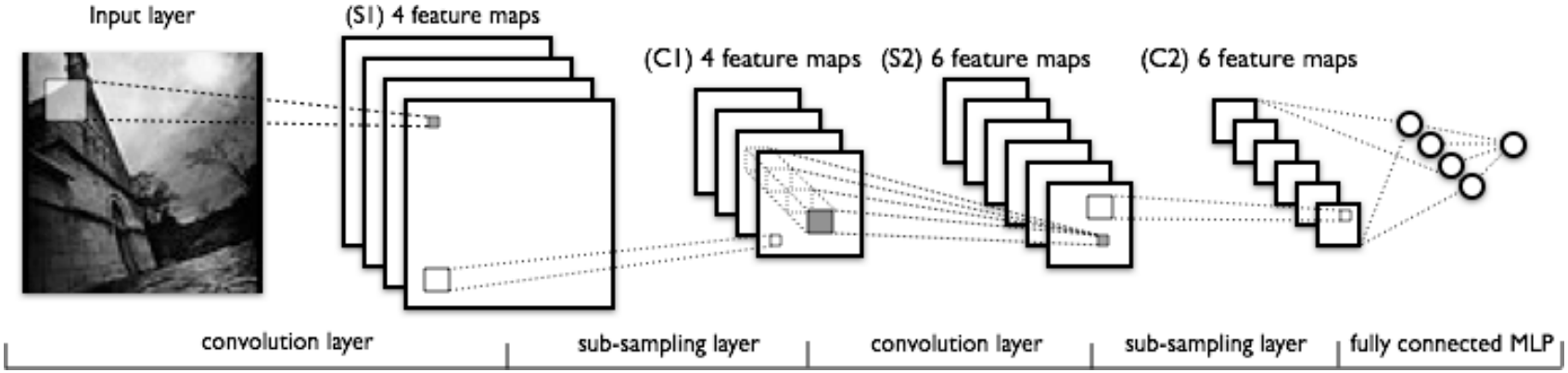
Subsample feature maps

- Increase *receptive field* ☺
- Downgrade resolution
  - Robustness to spatial variation ☺
  - Not good for *pixelwise* labeling ☹



Max pooling

Overall categorization CNN

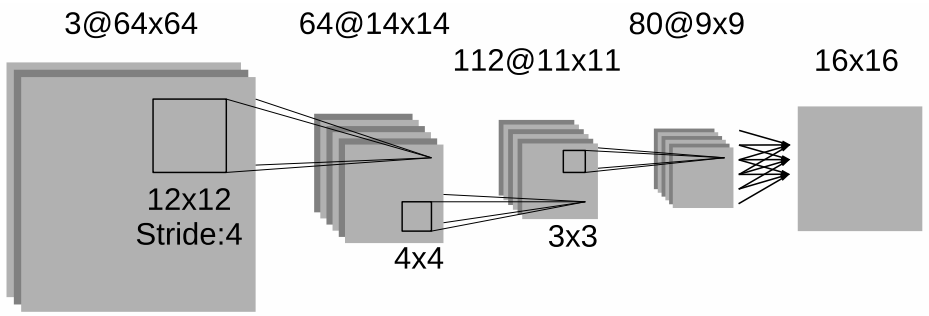


Source: deeplearning.net

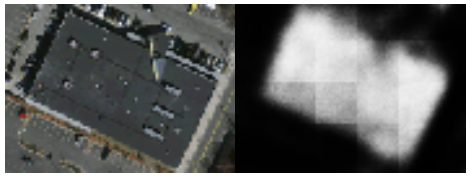
Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Pioneering works:

1. Predict and entire patch centered in input patch (Mnih, 2013)



   - Allows to learn "in-patch location" priors
     $\rightarrow$ Patch border artifacts



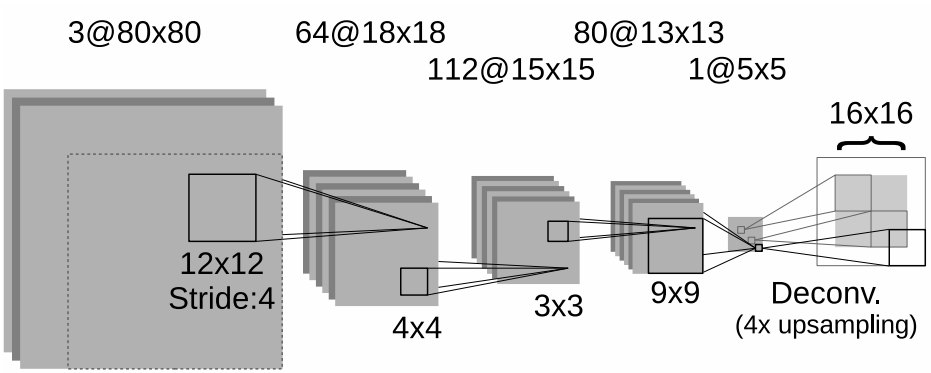
2. Predict the central pixel in the patch and shift one by one
   (e.g., Paisitkriangkrai et al., CVPR Earthvision 2015)
   - Too many redundant computations

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

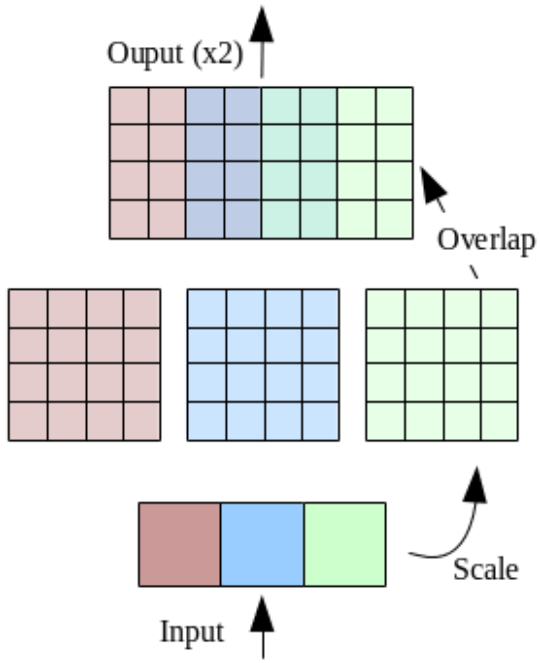
Fully convolutional networks (FCNs)

[Long et al., CVPR 2015]

- Convolutions & subsampling
- "Deconvolutional" layer to upsample



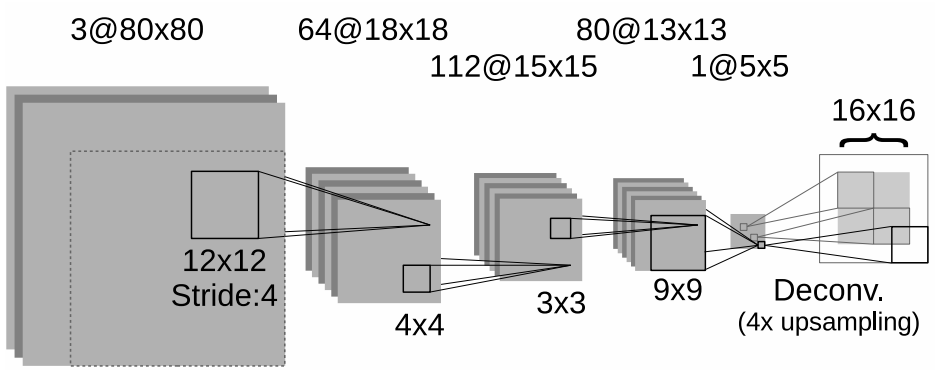
Proposed FCN for remote sensing



Deconv. layer

E. Maggiori, Y. Tarabalka, G. Charpiat, P. Alliez. "Fully convolutional neural networks for remote sensing image classification", IGARSS 2016

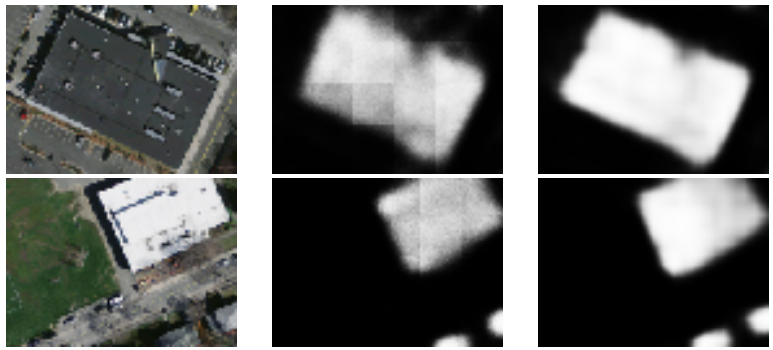
Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka



- Output size varies with input size (with fixed number of parameters)
- Location invariant (same logic used to compute every output)
- Avoid redundant computations
- *Especially* relevant in remote sensing (arbitrary tiling, azimuth)

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

- Patch artifacts removed by construction
- More accurate
- 10x faster



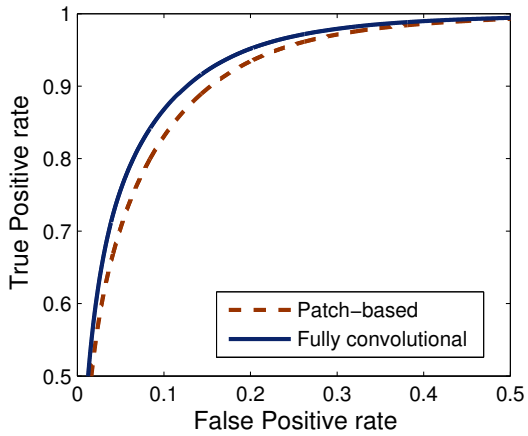
Input Patch-based FCN

Massachusetts dataset (Mnih, 2015)

Once again...

Imposing sensible restrictions

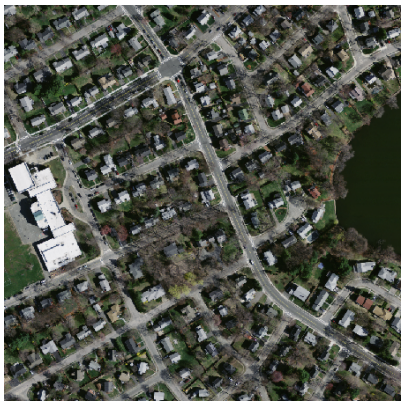
- improves the learning process,
- reduces execution times.



Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

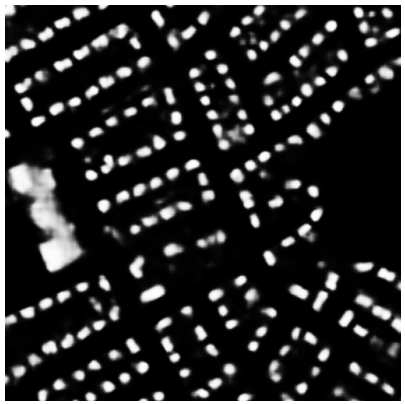
Massachusetts dataset

[Dataset: Mnih, 2013]



Color input

Reference



FCN

SVM

- Classification of 22.5 km^2 (1 m resolution): 8.5 seconds

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Frequent misregistration/omission in large-scale data sources:

Pléiades image + OpenStreetMap (OSM) over Loire department

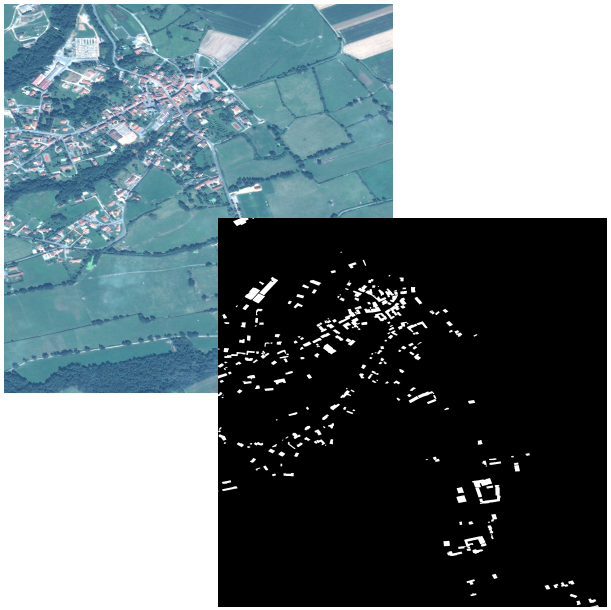
Possible strategy

Two-step training process:

1. Pretrain on large amounts of imperfect data
   $\rightarrow$ Learn dataset generalities
2. Fine-tune on a small piece of manually labeled reference

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

1. Pretrain on 22.5 km^2 Pléiades + OpenStreetMap data
2. Fine-tune on a manually labeled tile (2.5km^2, 3000×3000 px.)



Fine-tuning tile

Close-up

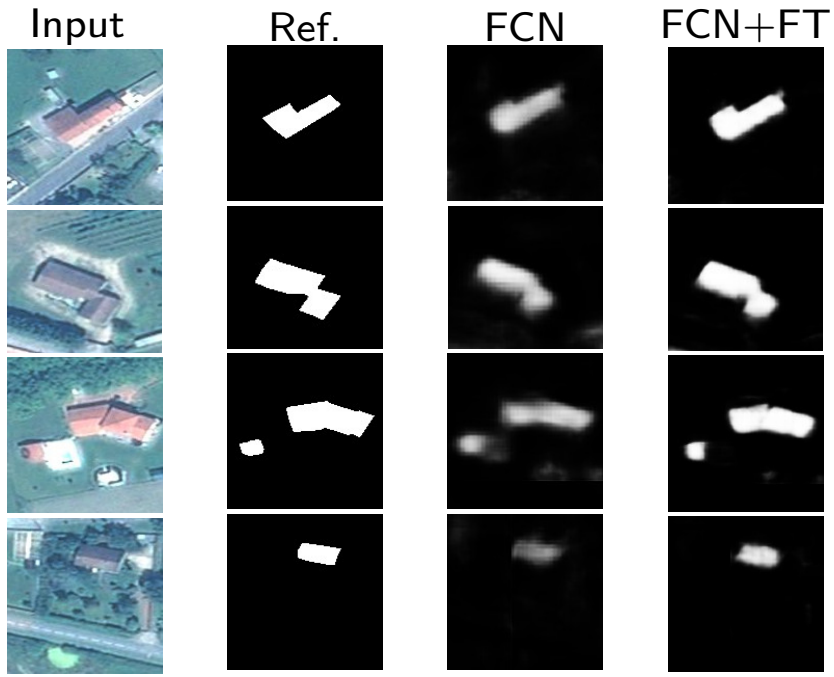
E. Maggiori, Y. Tarabalka, G. Charpiat, P. Alliez. "Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification", TGRS 2017.

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Test on a different manually labeled tile

Results

Test tile



Method	Accuracy	AUC*	IoU
FCN	99.13%	0.98154	47%
FCN + FT	99.57%	0.99836	72%

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

- Fully convolutional networks for remote sensing classification
  - FCNs have now become the standard dense labeling architecture
  - Other FCN comparisons (Kampffmeyer et al., 2016; Sherrah, 2016)
- Combining OSM + manual data sources to improve predictions
  - Growing interest in crowd-sourced data
    - Correcting OSM roads (Mattyus et al., 2016)
    - Combining diverse data sources (Kaiser, 2016)
    - OSM as an additional input (Audebert et al., 2017)

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

Recognition/localization trade-off

Subsampling:

- increases the receptive field (improving recognition)
- reduces resolution (hampering localization)

$\Rightarrow$ "Blobby" objects

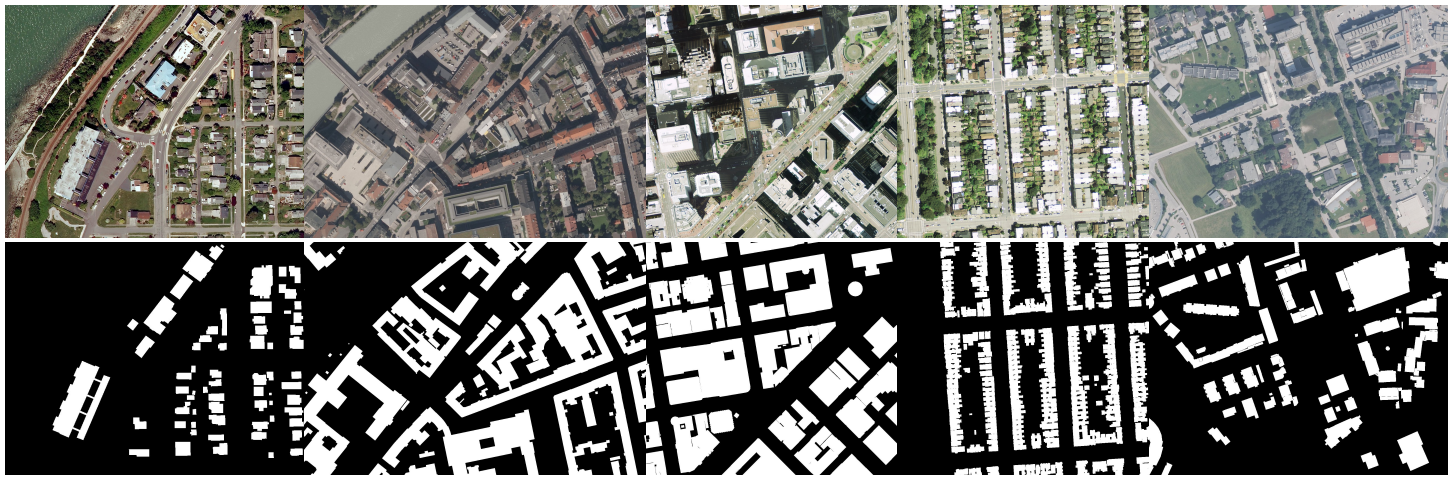
Input Ref. CNN

Solutions

1. Post-process the CNN's output (e.g., CRF)
2. Use innovative (e.g., multiscale) architectures

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

⇒ https://project.inria.fr/aerialimagelabeling/:

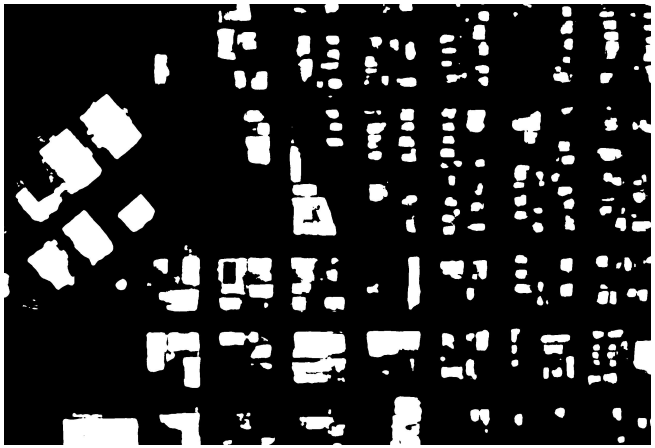


Leaderboard

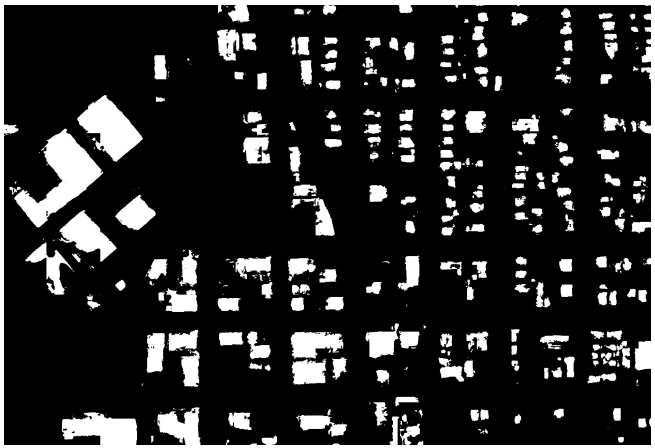
Method	Date	Bellingham		Bloomington		Innsbruck		San Francisco		East Tyrol		Overall	
		IoU	Acc.	IoU	Acc.	IoU	Acc.	IoU	Acc.	IoU	Acc.	IoU	Acc.
Inria1	3-Jan-17	52.91	95.14	46.08	94.95	58.12	95.16	57.84	86.05	59.03	96.40	55.82	93.54
Inria2	3-Jan-17	56.11	95.37	50.40	95.27	61.03	95.37	61.38	87.00	62.51	96.61	59.31	93.93
TeraDeep	5-May-17	58.08	95.88	53.38	95.61	59.47	95.26	64.34	88.71	62.00	96.57	60.95	94.41
RMIT	16-July-17	57.30	95.97	51.78	95.60	60.70	95.69	66.71	89.23	59.73	96.59	61.73	94.62

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

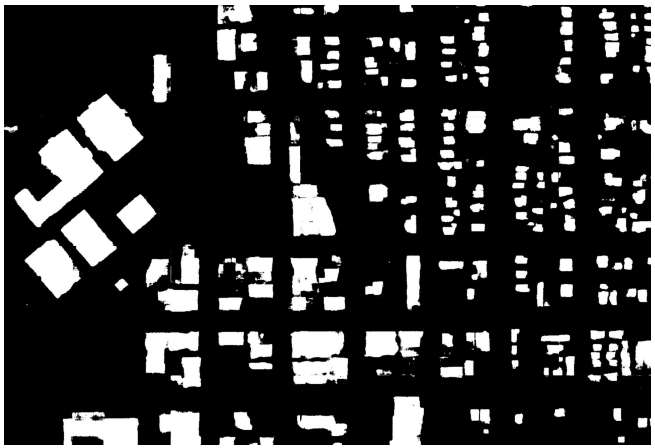
Input



Inria



TeraDeep



RMIT

Credits: Ronny Hänsch, Andreas Ley, Emmanuel Maggiori and Yuliya Tarabalka

- Image segmentation (a lot, see previous slides!)
- Object detection (not that many!)
- Image classification (a lot too!)
- Regression too! Estimate bio-geo-phys parameters! (later...)

2/ Object detection: road classification

Data:

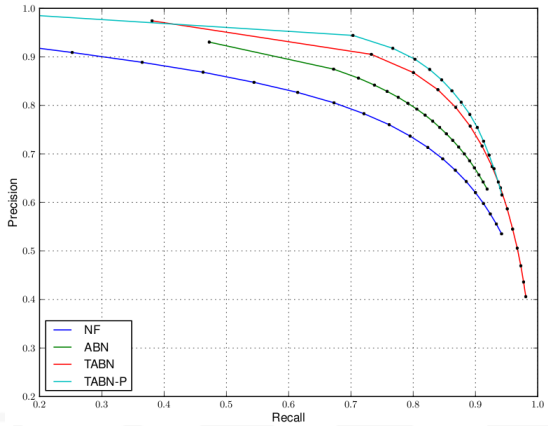
- RGB aerial images in 500 km^2, 1.2m resolution
- Two sets: good and bad co-registration

The net:

- Multilayer ($l = 3$) and RELU
- Not all neurons are connected
- Spatial filtering as input features
- Contrast normalization

2/ Object detection: road classification

- 'Precision and recall' (FNR/TPR)

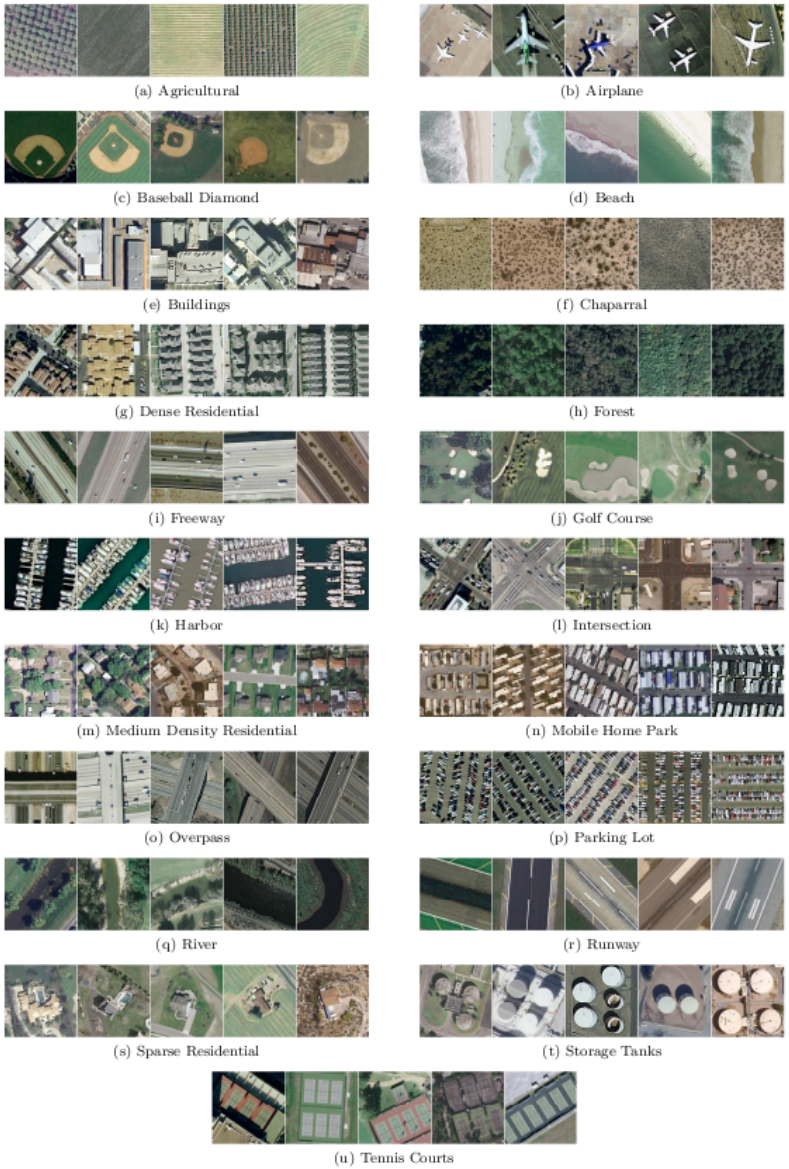


- Standard nets overfit!
- The deep nets minimize omission errors

3/ Image classification: land use detection

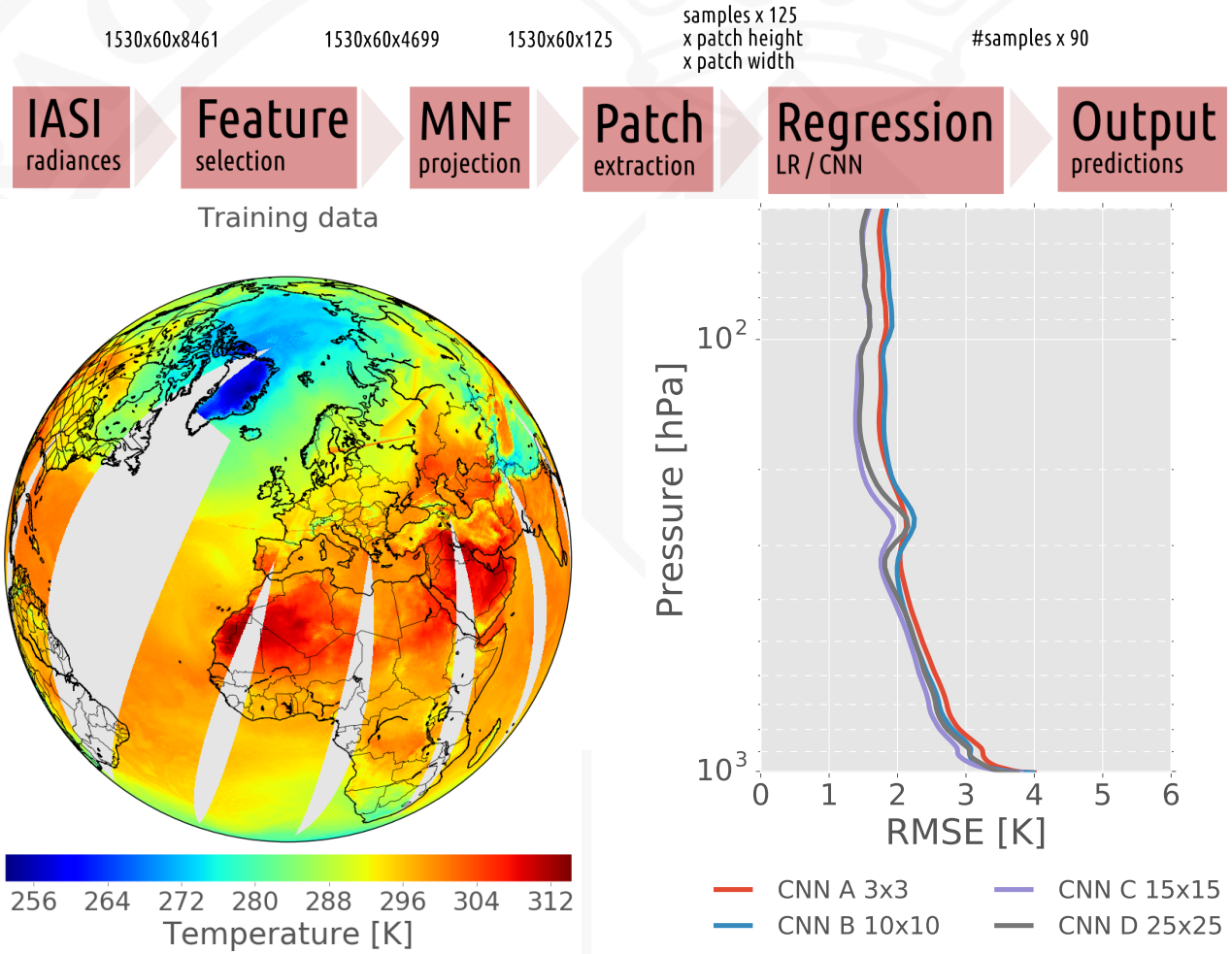
- UC Merced database, `http://vision.ucmerced.edu/datasets/`
- 21 classes, 100 RGB images each
- 80% for training, 20% for testing
- State of the art:

Feat.	**OA [%]**
SVM, inters. K [Yang10]	
BOVW	76.81
SPMK	75.29
SCK	72.52
BOVW+SCK	77.71
Color-RGB	76.71
Color-HLS	**81.19**
CNN [CampsValls13]	
1 lay. 7×7, $N_1 = 20$, $2\downarrow$	77.45
2 lay. 5×5, $N_1 = 20$, 7×7, $N_2 = 10$, $2\downarrow$	**81.63**



4/ Regression and function approximation

- Learn to estimate atmospheric parameters from infrared sounders
- Spatial-vertical relations + transfer learning



+ Many more crazy ideas of DL and AI!

Resources

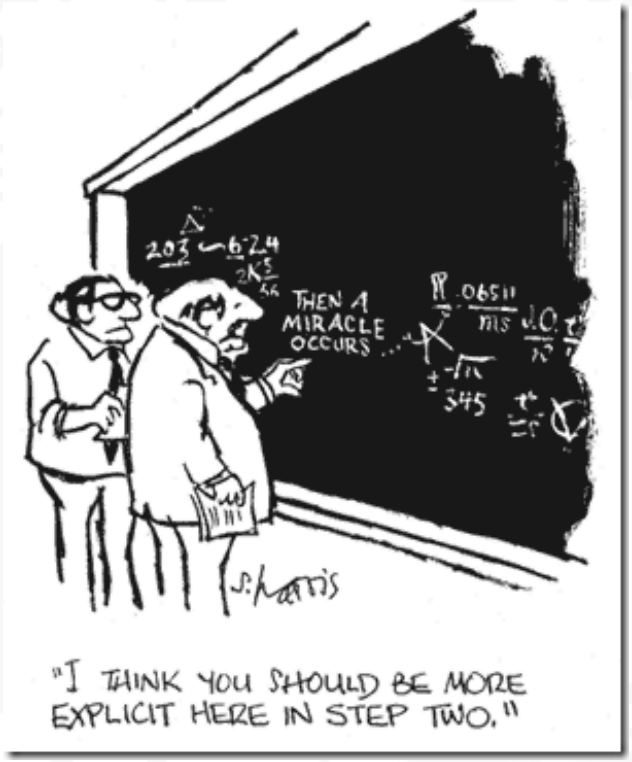
- **Google Timelapse** https://earthengine.google.com/timelapse/
- **NASA Worldview** https://worldview.earthdata.nasa.gov
- **Diseases** https://www.healthmap.org/en/
- **Water risk atlas** https://www.wri.org/applications/maps/aqueduct-atlas
- **Flood analyzer** http://floods.wri.org/
- **The anthroposphere!** https://www.gdeltproject.org/

ML/DL/AI applications

- **Wealth** http://penny.digitalglobe.com/
- **A global data refinery** http://www.descarteslabs.com/
- **One soil map** https://map.onesoil.ai
- **Land use** http://weegee.vision.ucmerced.edu/datasets/landuse.html
- **Create art** https://towardsdatascience.com/gangogh-creating-art-with-gans-8d087d8f74a1
- **Recognize faces** http://www.face-rec.org/databases/
- **Generate synthetic images** https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900

ANN features

- √ Lots of applications, mainly on detection/classification
- √ Accurate and fast at test/production time
- √ Including extra info (features, classes) is easy
- × Quite slow to train with lots of points and dimensions, GPU & HPC
- × As any other ML method, DL just interpolates!

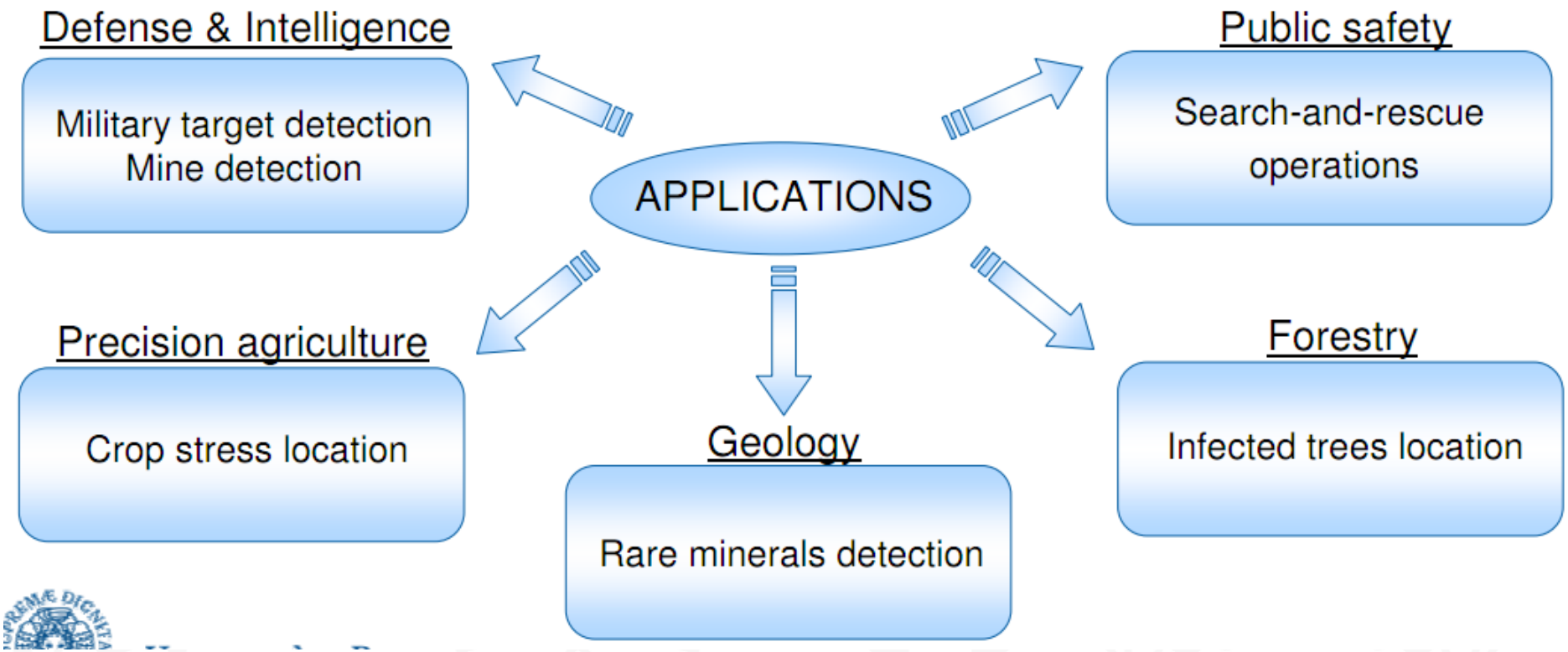


- *Deep Learning* by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT press, 2016, https://www.deeplearningbook.org/
- *Deep learning and process understanding for data-driven Earth System Science*, Reichstein, M. and Camps-Valls, G. and Stevens, B. and Denzler, J. and Carvalhais, N. and Jung, M. and Prabhat, Nature, 2019 **[and references therein!]**
- *Unsupervised Deep Feature Extraction for Remote Sensing Image Classification*, Romero, A. and Gatta, C. and Camps-Valls, G. Geoscience and Remote Sensing, IEEE Transactions on 54 (3) :1349-1362, 2016
- *Introduction Neural networks in remote sensing*, P. M. Atkinson and A. R. L. Tatnall, 699-709, 2010. https://doi.org/10.1080/014311697218700
- *Deep Learningin Remote Sensing – A comprehensive review and list of resources*, X Xiang Zhu et al. IEEE GRSM, 2017
- Web: `isp.uv.es`
- Web: `www.uv.es/gcamps`

Part 5: Target detection in remote sensing images

Outline

- Some applications relying on remote sensing data do not need the definition of many classes of interest.
- Very often, they only need to discriminate a single class from the rest, i.e. the background [Ahl04]
- Sometimes, there is no knowledge about the signature of the class of interest, but it is known that this signature is different from the rest
- **Anomaly detectors:** A model is built by looking for signatures that deviate from a model of the background: Large deviations from the mean are called *anomalies*
- **Target detectors:** If we have *spectral libraries*, the problem reduces to defining the background class and then detecting spectral signatures that are closer to the known signatures than to the background.

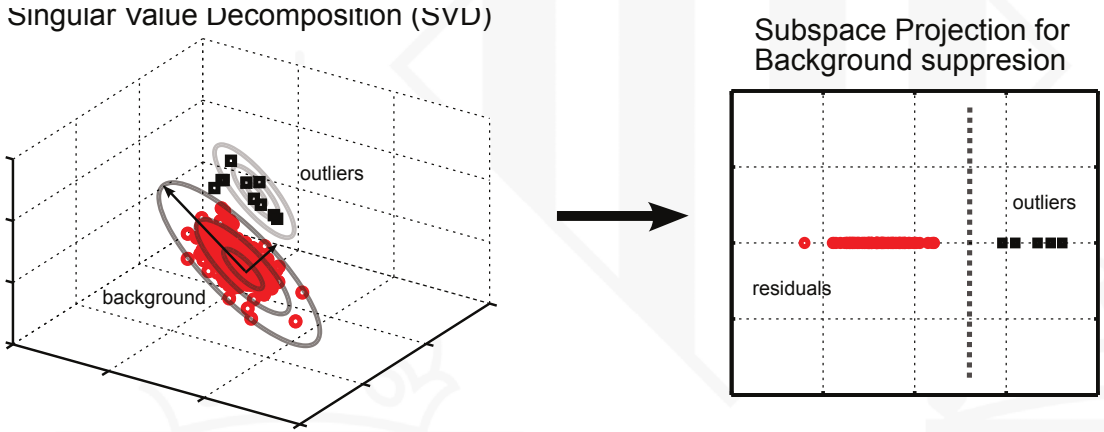

Defense & Intelligence
Military target detection
Mine detection
Public safety
Search-and-rescue operations
APPLICATIONS
Precision agriculture
Crop stress location
Geology
Rare minerals detection
Forestry
Infected trees location

- When the signature of the class to be detected is unknown, anomaly detection algorithms are used. Anomaly detection reduces to assuming (and/or defining) a model for the background class, and then look for the test samples lying far from the mean of the background distribution.
- When the target is partially or entirely known, this additional knowledge can be used to design more accurate target detectors: rather than assessing the distance from a background distribution, the pixel is compared against target and background and then assigned to the closest class.
- Taxonomy:

Table 4.3: Taxonomy of anomaly and target detectors (adapted from Ahlberg and Renhorn [2004] and Kwon and Nasrabadi [2007a]).

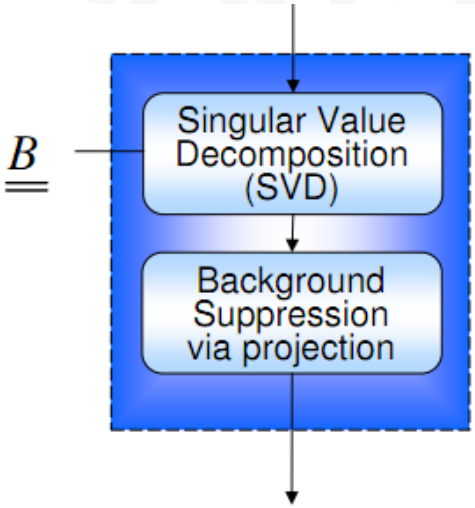
	Background model		
	Gaussian	Subspace	Nonparametric
Anomaly detectors	RX, GMM	DFFS	One Class SVM, SVDD
Target detectors	SAM, AMF	ASD, OSP, SMF, MSD	One Class SVM, SVDD, MIL, KASD, KOSP, KSMF, KASD

- OSP is the most widely used target detector in remote sensing [Ree90,Lu97]
- OSP uses a linear subspace (SVD or PCA) for the background model
- The subspace of the background basis functions is removed from the analyzed pixel, thus leaving only the part related to the target signature
- This reduced spectral signature is then matched with the target signature: if the match exceeds a given threshold, the pixel is considered as target



$\underline{\underline{B}} = [\underline{b}_1, \underline{b}_2, \ldots, \underline{b}_k]$

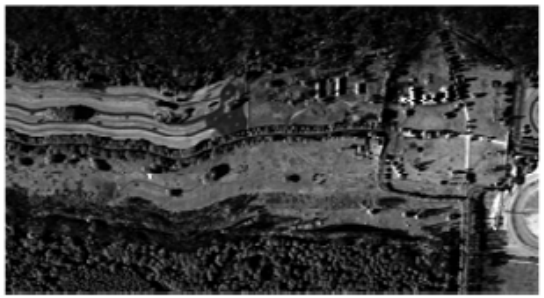
Background components matrix (eigenvectors associated to the highest eigenvalues of the SVD decomposition)



Projection Matrix:

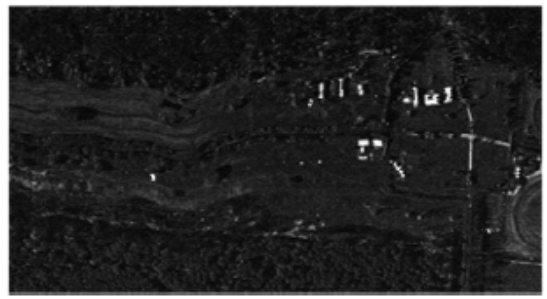
$$\underline{\underline{P}}^{\perp} = \underline{\underline{I}} - \underline{\underline{B}} \cdot \underline{\underline{B}}^{\#} = \underline{\underline{I}} - \underline{\underline{B}} \cdot \left(\underline{\underline{B}}^{T} \cdot \underline{\underline{B}} \right)^{-1} \cdot \underline{\underline{B}}^{T}$$

Energy of the original image



Background suppression

Energy of the background-suppressed image



Distance-based target detectors

- When the background and target are assumed to be Gaussian-distributed, a likelihood function can be used to decide whether a pixel belongs to one or the other
- Using a Mahalanobis distance, this target detector is called the adaptive matched filter (AMD).

$$d(x_i, x_j) = x_i^\top \Sigma^{-1} x_j$$

Spectral Angle Mapper (SAM)

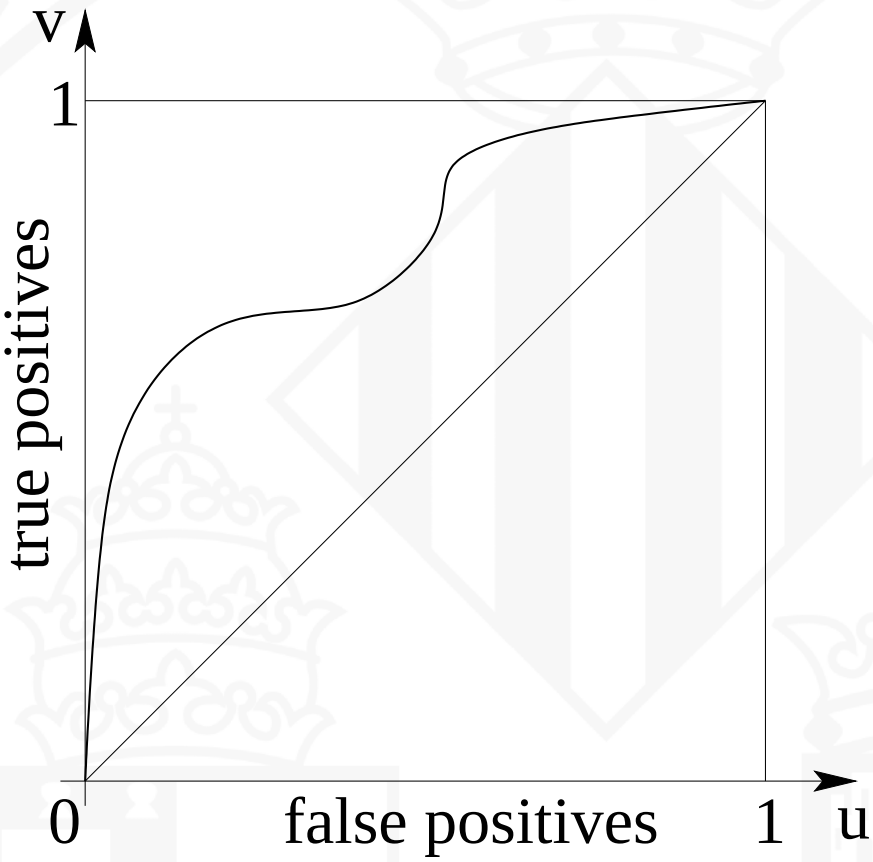
- Another possibility is to use the spectral angle mapper (SAM) as a measure of distance.
- Such measure assesses the angle between the target and the pixel to be evaluated with a dot product.

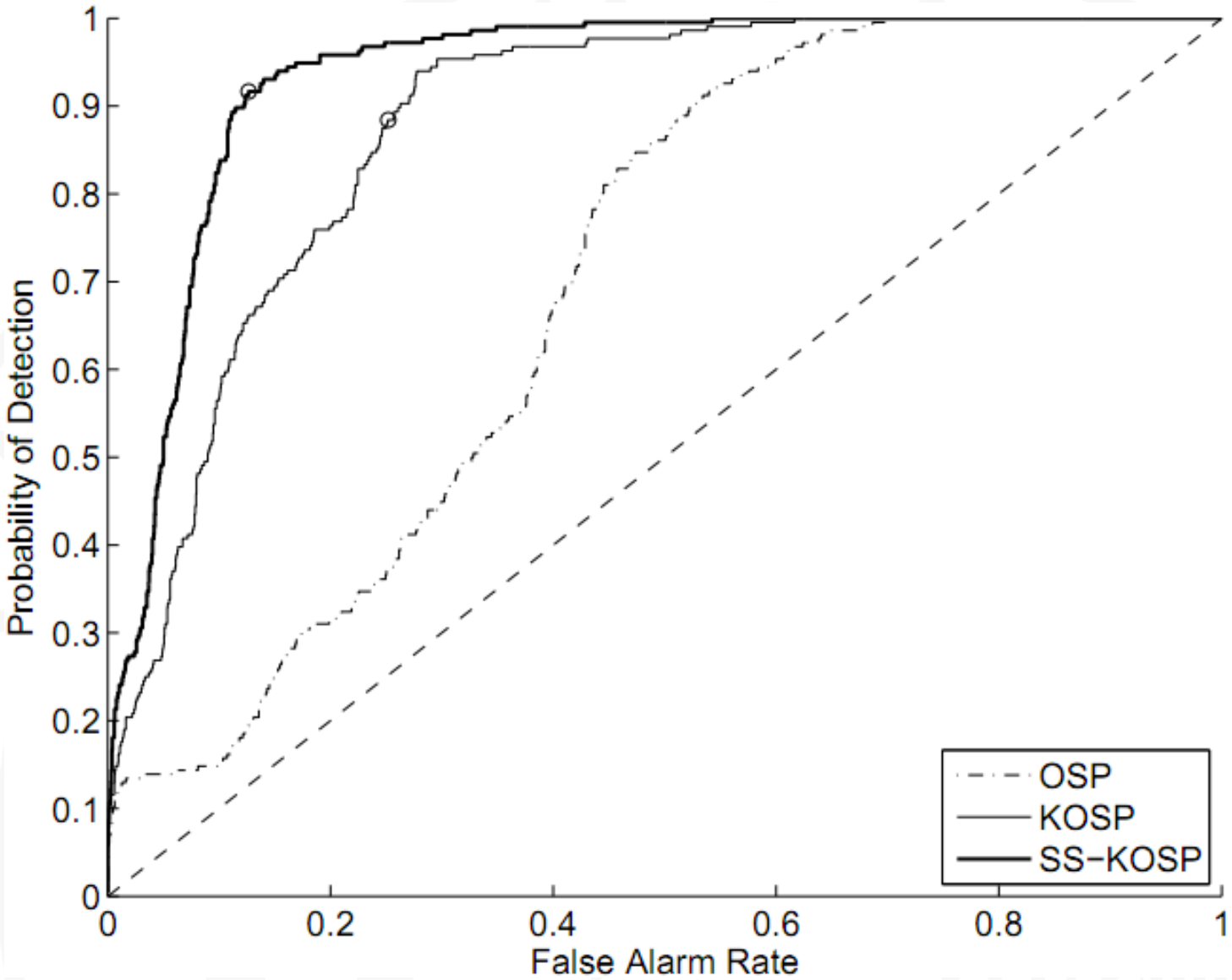
$$d(x_i, x_j) = acos\left(\frac{x_i^\top x_j}{\|x_i\|\|x_j\|}\right)$$

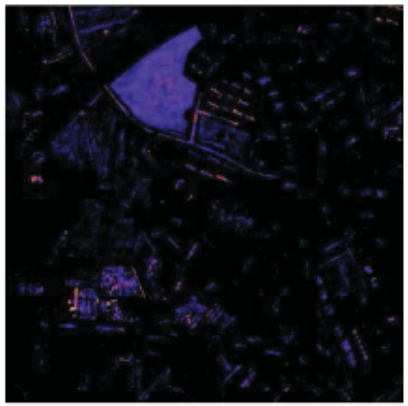
- Note that any other distance measure can be included here...

How should we tune the threshold?

- Analyze the receiver operating curve (ROC): $f(u, v|\theta)$
  - u = proportion of false positives = $P(f(x) = 1|y = -1)$
  - v = proportion of true positives = $P(f(x) = 1|y = 1)$



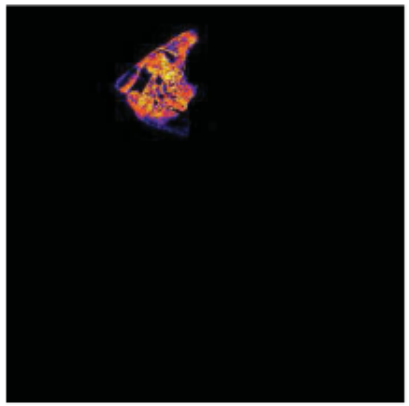

Probability of Detection
False Alarm Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1
OSP
KOSP
SS−KOSP



94.14%

0.59

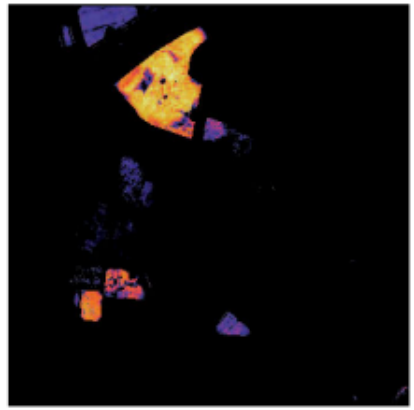
(a) OSP



99.74%

0.94

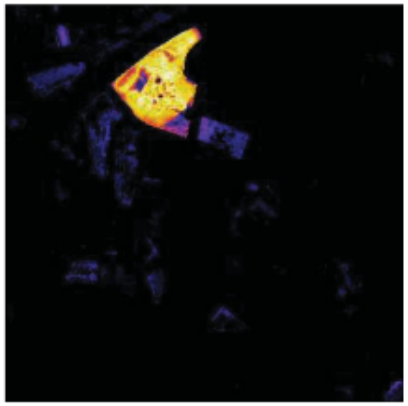
(b) KOSP, $\sigma = 0.04$



98.99% AUC

0.84 Kappa

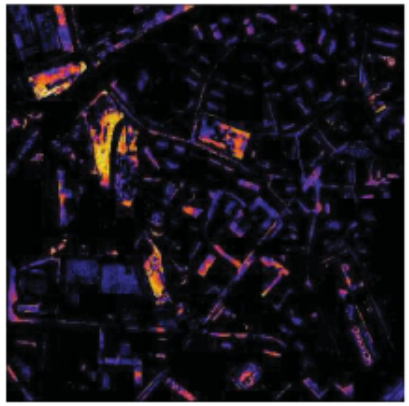
(e) SAM



99.78%

0.95

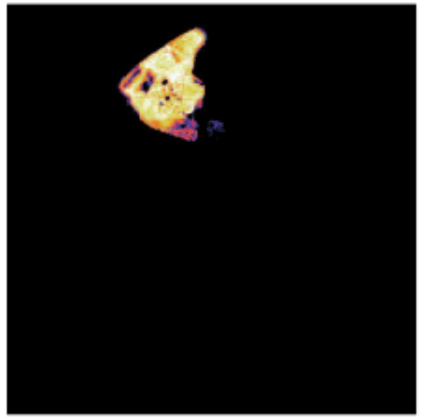
(c) KOSP, $\sigma = 0.1$



21%

-0.14

(d) KOSP, $\sigma = 0.2$



99.87% AUC

0.97 Kappa

(f) One class SVM

References

- I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," IEEE Transactions on Signal Processing, vol. 38, no. 10, pp. 1760–1770, Oct 1990.
- J. C. Harsanyi and C. I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," IEEE Transactions on Geoscience and Remote Sensing, vol. 32, pp. 779–785, 1994
- D. W. J. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," Signal Processing Magazine, IEEE, vol. 19, no. 1, pp. 58–69, Jan 2002
- K. I. Ranney and M. Soumekh, "Hyperspectral Anomaly Detection Within the Signal Subspace," IEEE Geoscience and Remote Sensing Letters, vol. 3, issue 3, pp. 312-316, vol. 3, pp. 312–316, July 2006.
- Luca Capobianco and Gustavo Camps-Valls, "Target detection with Semisupervised Kernel Orthogonal Subspace Projection. IEEE Transactions on Geoscience and Remote Sensing, 47(11), 3822-3833, 2009

http://www.uv.es/gcamps/sskosp/

Part 6: Change detection in remote sensing images

Outline

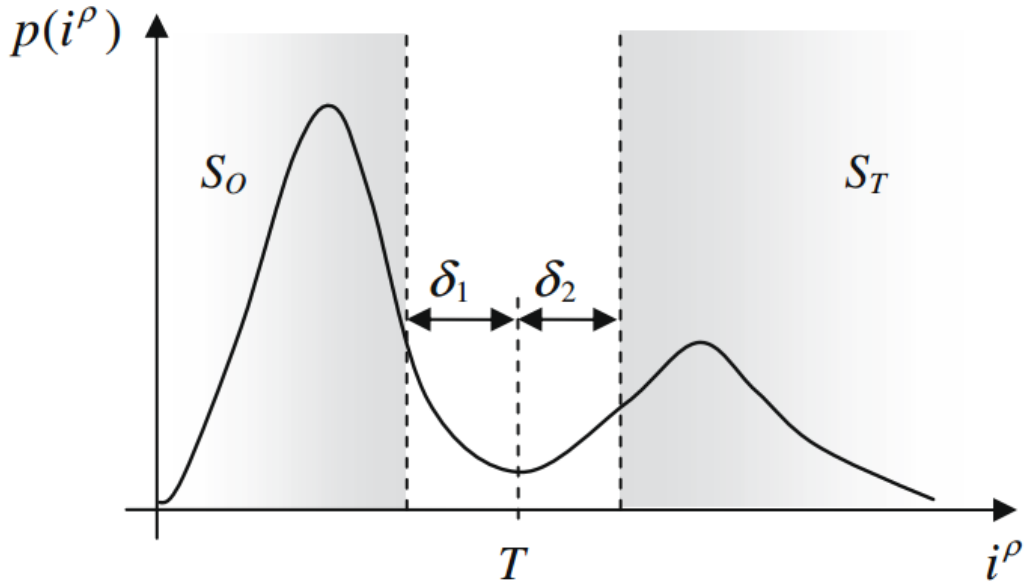
1. Definition of change detection
2. Change vector analysis (CVA)
3. Clustering approaches
4. Supervised approaches
5. Conclusions

- Change detection is attracting an increasing interest from the application domains, since it automatizes traditionally manual tasks in disaster management or developments plans for urban monitoring
- multitemporal classification and change detection are very active fields nowadays because of the increasing availability of complete time series of images and the interest in monitoring Earth's changes at local and global scales
- Three types of products: binary maps, detection of types of changes, and full multiclass change maps, thus including classes of changes and unchanged land-cover classes.
- Notationally, the problem reduces to there is a significant difference between two objects (pixels, patches, regions) taken at different times $\mathbf{x}(t_1)$ and $\mathbf{x}(t_2)$.

Many data transformations have been used to enhance the detection of changes:

- the difference image $\mathbf{x}_{\text{diff}} = |\mathbf{x}_{(1)} - \mathbf{x}_{(2)}|$, where unchanged areas show values close to zero,
- image ratioing $\mathbf{x}_{\text{ratio}} = \frac{\mathbf{x}_{(1)}}{\mathbf{x}_{(2)}}$, where unchanged areas show values close to one,
- data transformations as principal components, where changes are grouped in the components related to highest variance,
- physically based indices as NDVI, useful to detect changes in vegetation.
- the stacking of feature vectors $\mathbf{x}_{\text{stack}} = [\mathbf{x}_{(1)}, \mathbf{x}_{(2)}]$, mainly used in supervised methods (need labeled information)

- Compute the difference image $\mathbf{x}_{\text{diff}} = |\mathbf{x}_{(1)} - \mathbf{x}_{(2)}|$, where unchanged areas show values close to zero,
- Select a threshold that detects the changed pixels
- The task is not easy and often very heuristic



Unsupervised Classification - Clustering

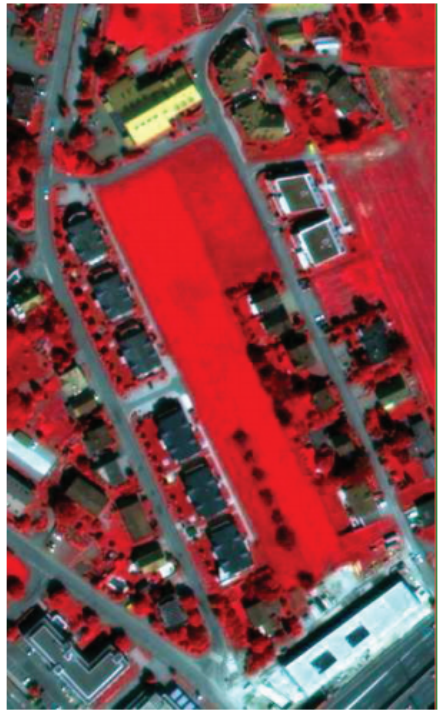
- No need of labeled examples
- Few parameters have to be set (usually the number of clusters)
- The user only minimally influence the process
- The clustering step is usually done evaluating some distance / similarity between samples

Clustering and change detection

- The algorithms automatically find two clusters: change / no change
- The images cover the same region
- We look for clusters in the difference image space, automatically assign difference vectors to a cluster
- Unsupervised methods maximize the 'automatic' component

Clustering and change detection

- The algorithms automatically find two clusters: change / no change
- The images cover the same region
- We look for clusters in the difference image space, automatically assign difference vectors to a cluster
- Unsupervised methods maximize the 'automatic' component

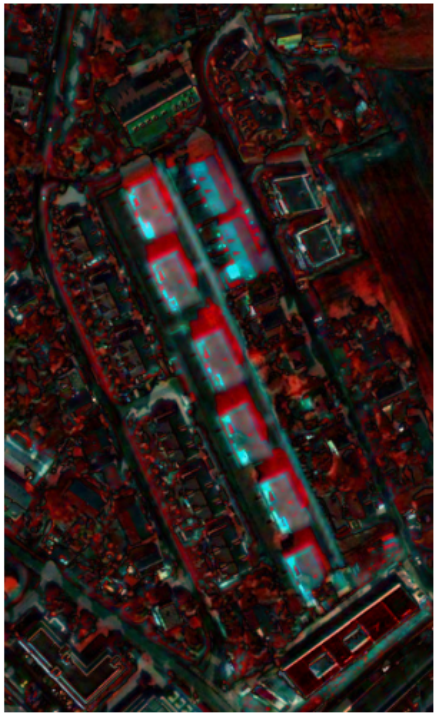


Clustering and change detection

- The algorithms automatically find two clusters: change / no change
- The images cover the same region
- We look for clusters in the difference image space, automatically assign difference vectors to a cluster
- Unsupervised methods maximize the 'automatic' component

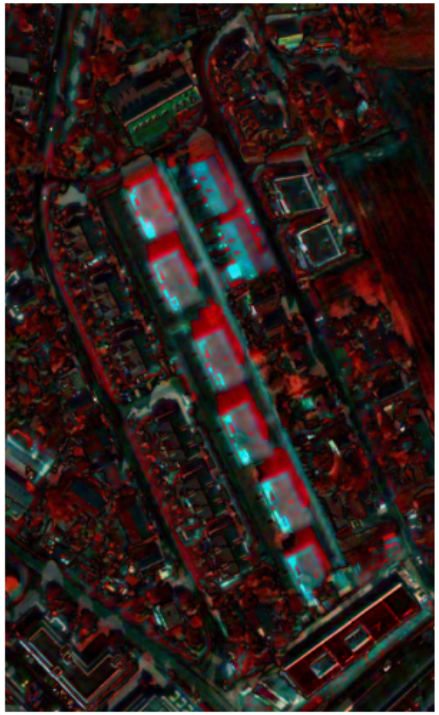
Clustering and change detection

- The algorithms automatically find two clusters: change / no change
- The images cover the same region
- We look for clusters in the difference image space, automatically assign difference vectors to a cluster
- Unsupervised methods maximize the 'automatic' component



Clustering and change detection

- The algorithms automatically find two clusters: change / no change
- The images cover the same region
- We look for clusters in the difference image space, automatically assign difference vectors to a cluster
- Unsupervised methods maximize the 'automatic' component



Partitional technique that aims at separate the observations into k groups

The k-means

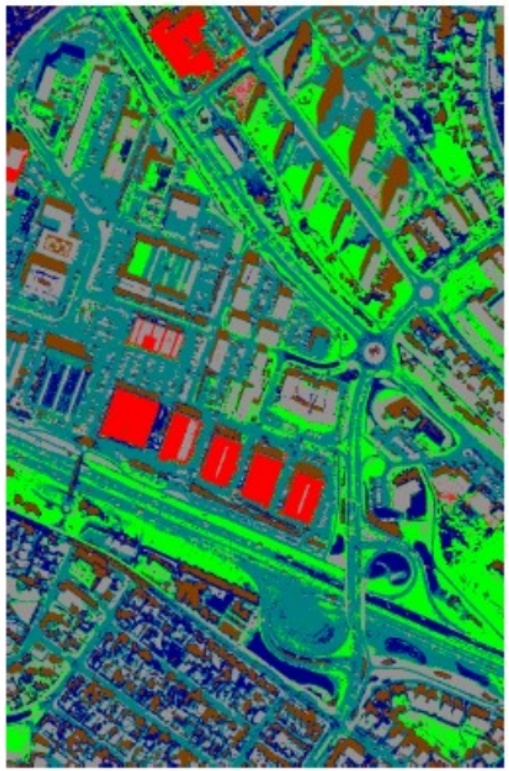
The sample $\mathbf{x}_i$ is assigned to the cluster k whose center of gravity (mean) $\mathbf{m}_k$ minimizes:

$$d^2(\mathbf{x}_i, \mathbf{m}_k) = \|\mathbf{x}_i - \mathbf{m}_k\|^2 \tag{1}$$

$$\text{where } \mathbf{m}_k = \frac{1}{|\pi_k|} \sum_{j \in \pi_k} \mathbf{x}_j \tag{2}$$

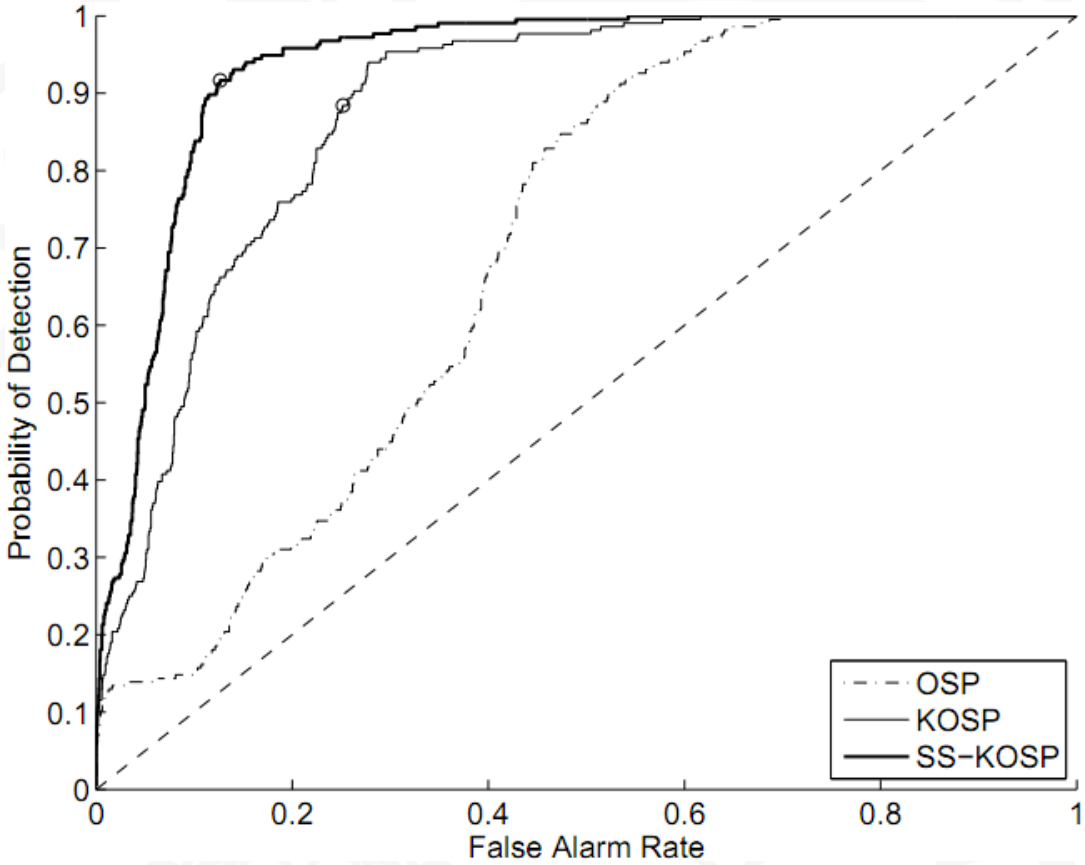
and $|\pi_k|$ are the elements of the cluster. The convergence is optimal only when the clusters are spherical at similar scales.

- Supervised classification, we need some labeled samples (e.g. pixels)
- We typically stack the (spatial-)spectral vectors $\mathbf{x}_{\text{stack}} = [\mathbf{x}_{(1)}, \mathbf{x}_{(2)}]$ and train a classifier to predict $\mathbf{y} = \{0, +1\}$, i.e. no-change vs change.
- Any classifier can be used
- In practice we will try it with linear classification
- Homework: replace it with the k-nn classifier, neural nets, SVM

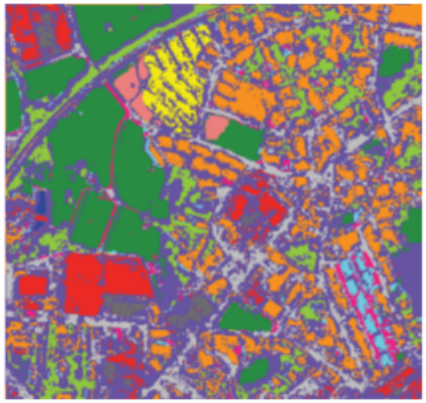


Classification

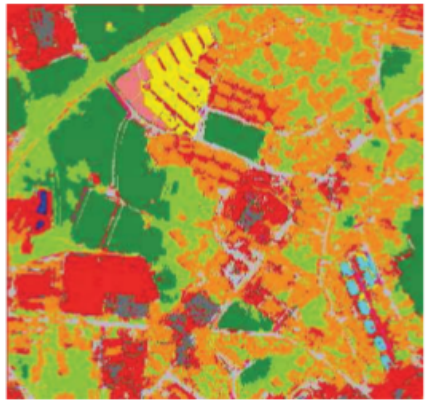
Red: big buildings
Grey: small buildings
Blue: small streets
Green: open spaces
Blue~green: avenues
Brown: shadows


Probability of Detection
False Alarm Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.4
0.6
0.8
OSP
KOSP
SS−KOSP

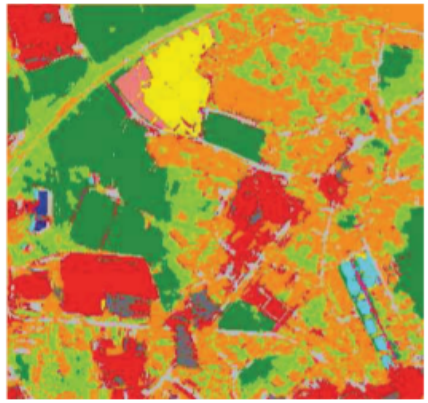
(a) PCC
(87.37, 0.85)



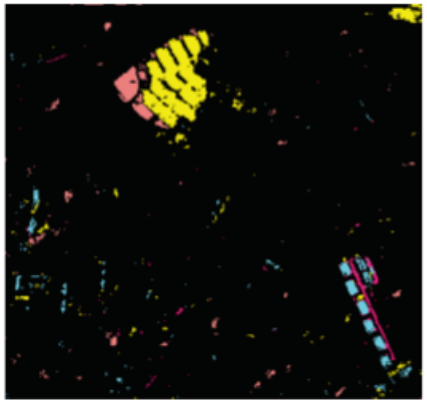
(b) Multiclass DMC
(95.11, 0.94)



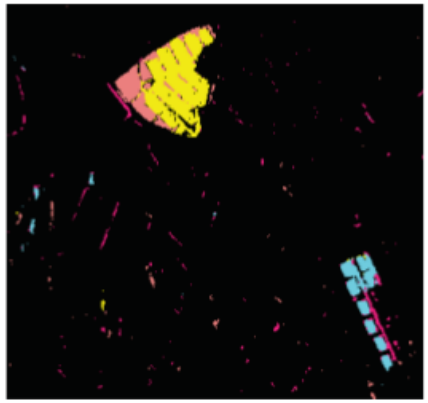
(c) SVM, summation kernel
(91.44, 0.89)



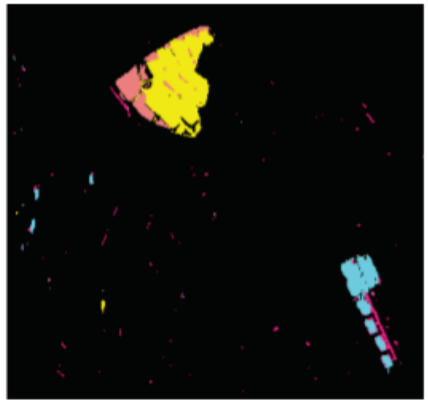
(d) DIA
(98.52, 0.93)



(e) Changes-only DMC
(99.53, 0.98)



(f) SVM, summation kernel
(99.39,0.97)



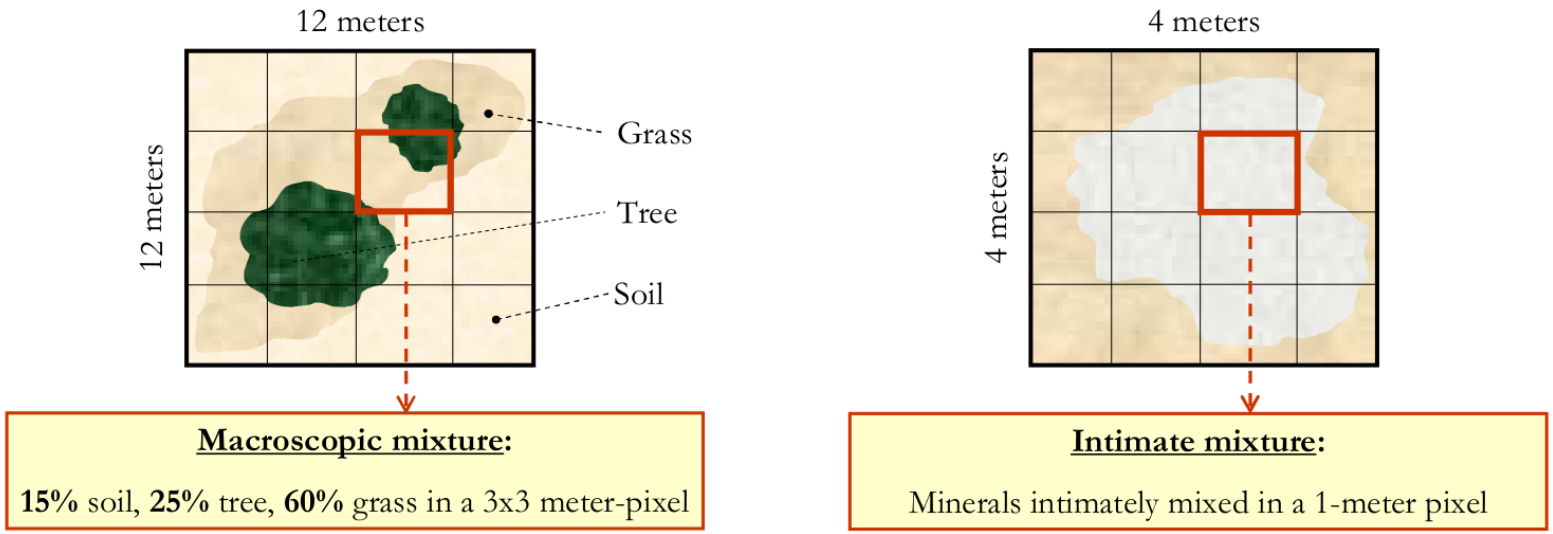
References

- L. Bruzzone and D. F. Prieto, "Automatic analysis of the difference image for unsupervised change detection," IEEE Trans. Geosci. Remote Sens., vol. 38(3), pp. 1171–1182, 2000.
- Volpi, M.; Tuia, D.; Camps-Valls, G.; Kanevski, M.; Unsupervised Change Detection with Kernels, IEEE Geosciences and Remote Sensing Letters, vol. 9, no. 6, pp. 1026-1030, 2012
- G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, "Kernel-based framework for multi-temporal and multi-source remote sensing data classification and change detection," IEEE Trans. Geosci. Remote Sens., vol. 46, no. 6, pp. 1822–1835, 2008.

Part 7: Spectral unmixing and abundance estimation

Unmixing hyperspectral pixels ...

- With a limited spatial resolution, spectral vectors are no longer pure but mixtures of the spectral signatures of the materials present in the scene
- A small fraction of the available pixels can be considered as *pure*, i.e. composed by a single material
- The field of spectral mixture analysis (or *spectral unmixing*) is devoted both to identify the most probable set of pure pixels (called *endmembers*) and to estimate their proportions (called *abudances*) in each pixel
- When the endmembers have been identified, every single pixel in the image can be synthesized as a linear (or nonlinear) combination of them



The main applications of spectral unmixing:

1 *Standard mapping applications.*

Keshava02	Excellent introduction to crop and mineral mapping
Sohn97	Abundance estimation of vegetation in deserts
Adams95	Abundance maps for image classification to detect landcover changes in the Amazonia
Roberts98	multiple endmember spectral mixture models to map chaparral
Elmore00	quantify vegetation change in semiarid environments
Goodwin05	assessed plantation canopy condition from airborne imagery using spectral mixture analysis via fractional abundance estimation
Pacheco10	crop residue mapping in multispectral images
Zhang04	deconvolution of lichen and rock mixtures
Wu2004	to monitor urban composition using ETM+ images
Dop11	extract features and then performing supervised urban image classification

The main applications of spectral unmixing:

2 *Multitemporal studies.*

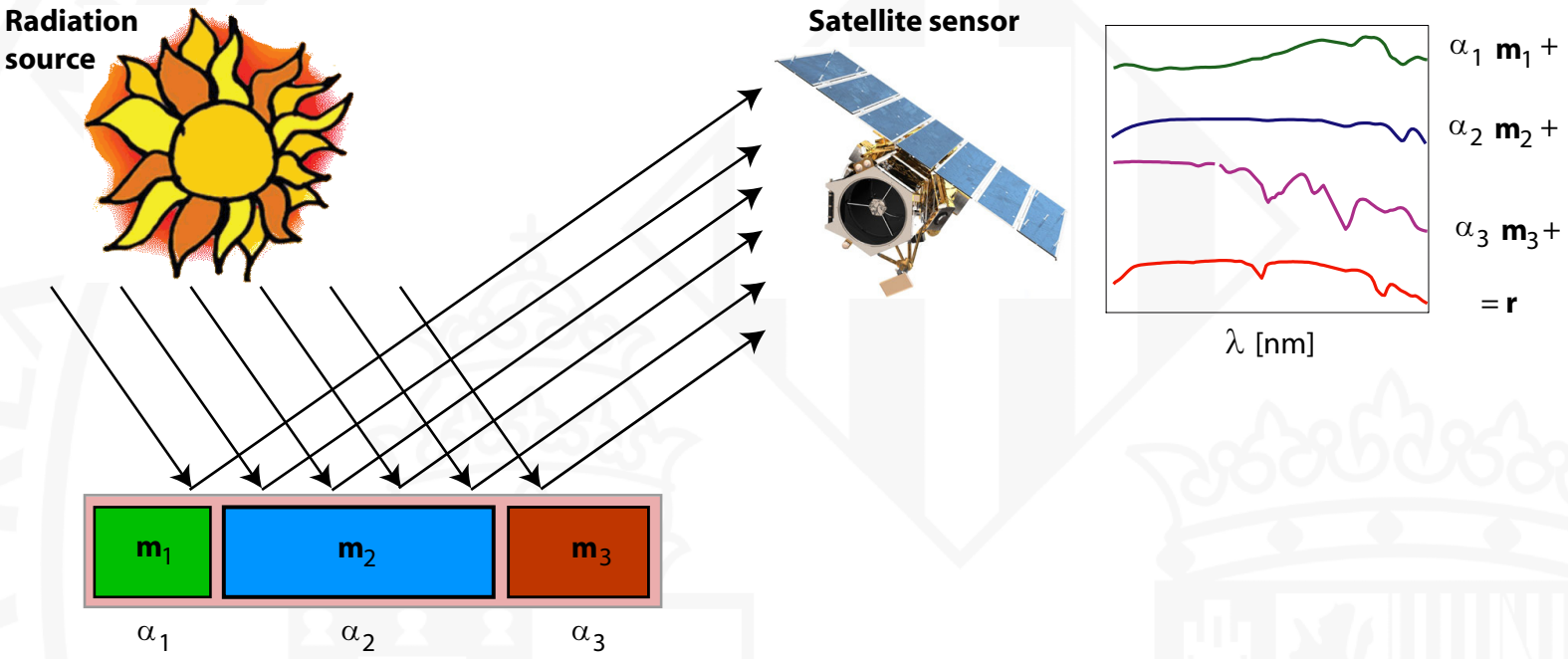
- Shoshany02 a multi-date adaptive unmixing was applied to analyze ecosystem transitions along a climatic gradient
- Lobell2004 inferred cropland distributions from temporal unmixing of MODIS data
- Gomez11 multitemporal unmixing of medium spatial resolution images was conducted for landcover mapping

3 *Multisource models.*

- Puyou94 multiple linear regression as a tool for unmixing coarse spatial resolution images acquired by AVHRR
- GarciaHaro96 alternative approach which appends the high spatial resolution image to the hyperspectral data and computes a mixture model based on the joint data set.
- Zhukov99 spatial and spectral data fusion
- Amoros11 spatial unmixing technique to obtain a composite image with the spectral and temporal characteristics of the medium spatial resolution image and the spatial detail of the high spatial resolution image

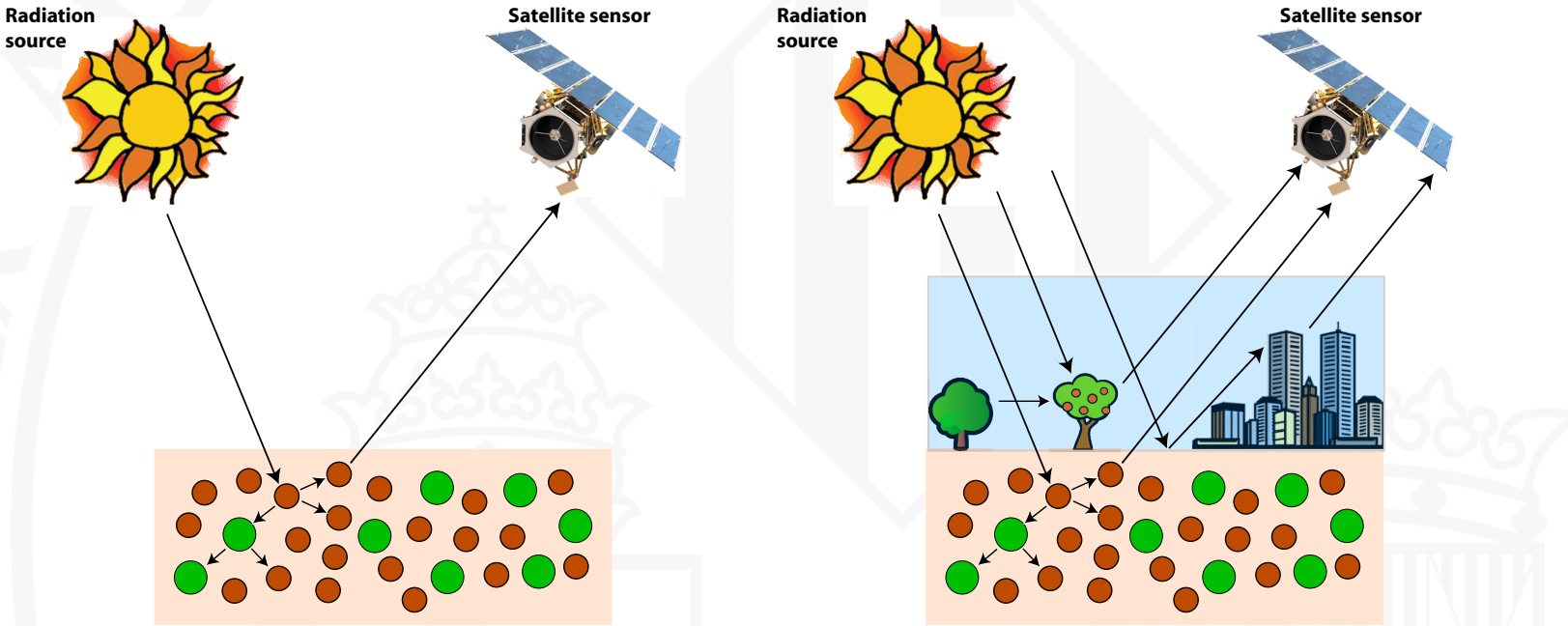
Illustration of the spectral linear mixing process

A given material is assumed to be constituted at a subpixel level by patches of distinct materials $\mathbf{m}_i$ contributing linearly through a set of weights (or abundances) α_i to the acquired reflectance $\mathbf{r}$



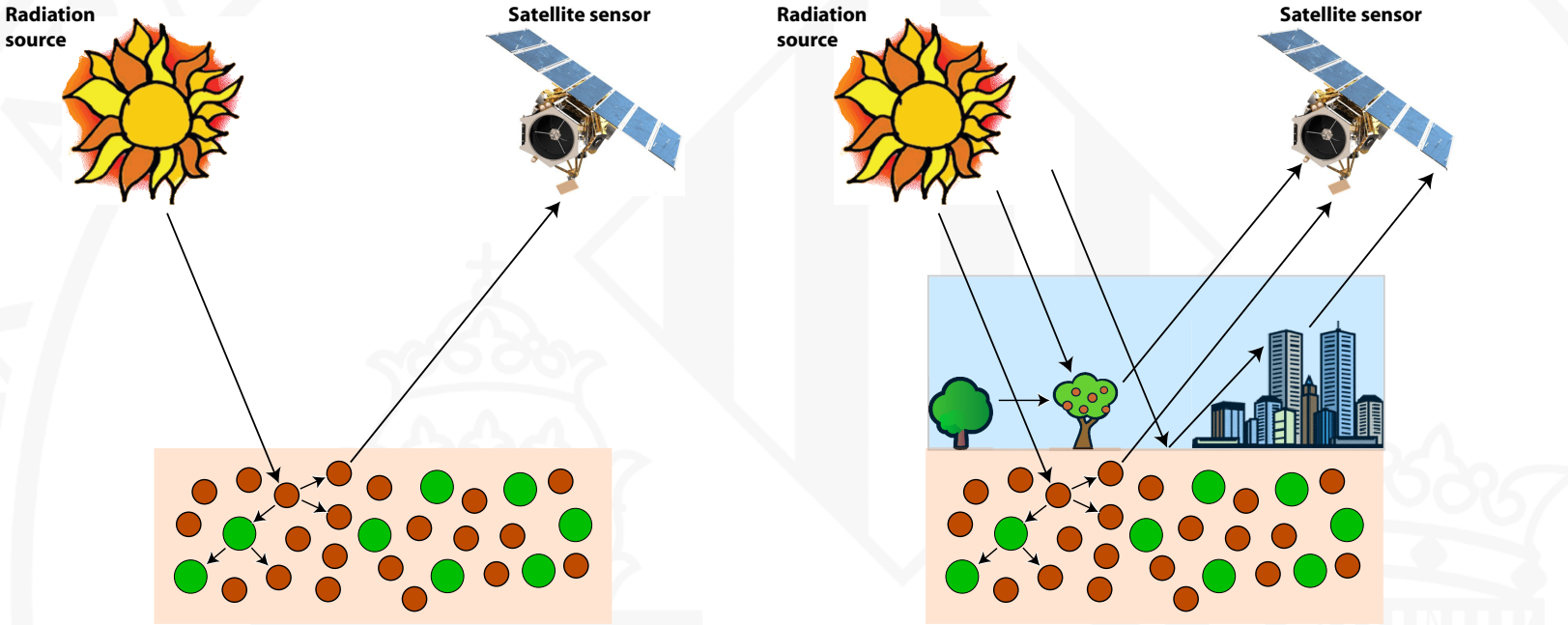
Linear versus nonlinear mixing model:

- Linear mixture model assumes that endmember substances are sitting side-by-side within the FOV
- Nonlinear mixture model:
  - Endmember components are randomly distributed throughout the FOV
  - Multiple scattering effects



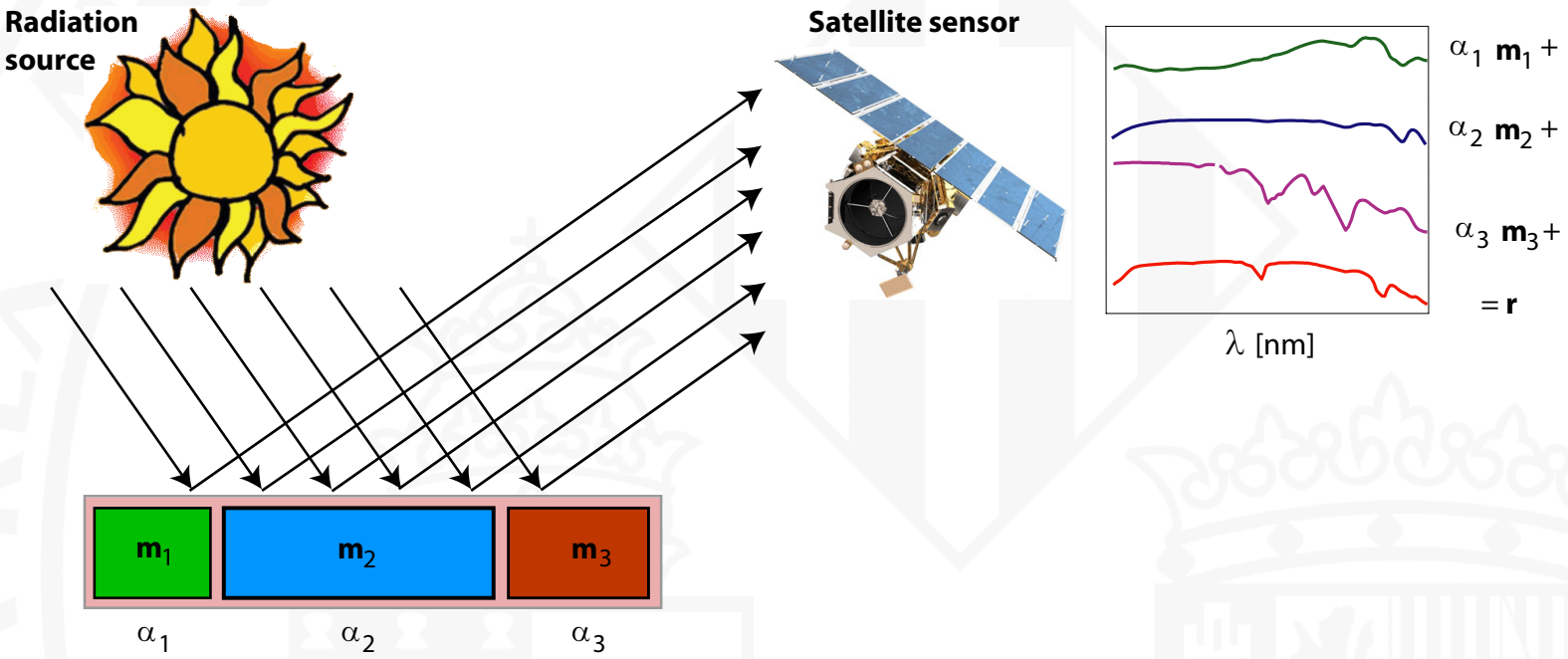
Two nonlinear mixing scenarios:

- The intimate mixture model (left): the different materials are close
- The multilayered mixture model (right): interactions with canopies and atmosphere happen sequentially or simultaneously



Let's go on with a linear unmixing model:

- Simple, tractable, mathematically convenient
- Effective in many real settings
- Acceptable approximation of the light scattering mechanisms
- Computationally feasible



The linear mixing model:

$$\boxed{\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n}}$$

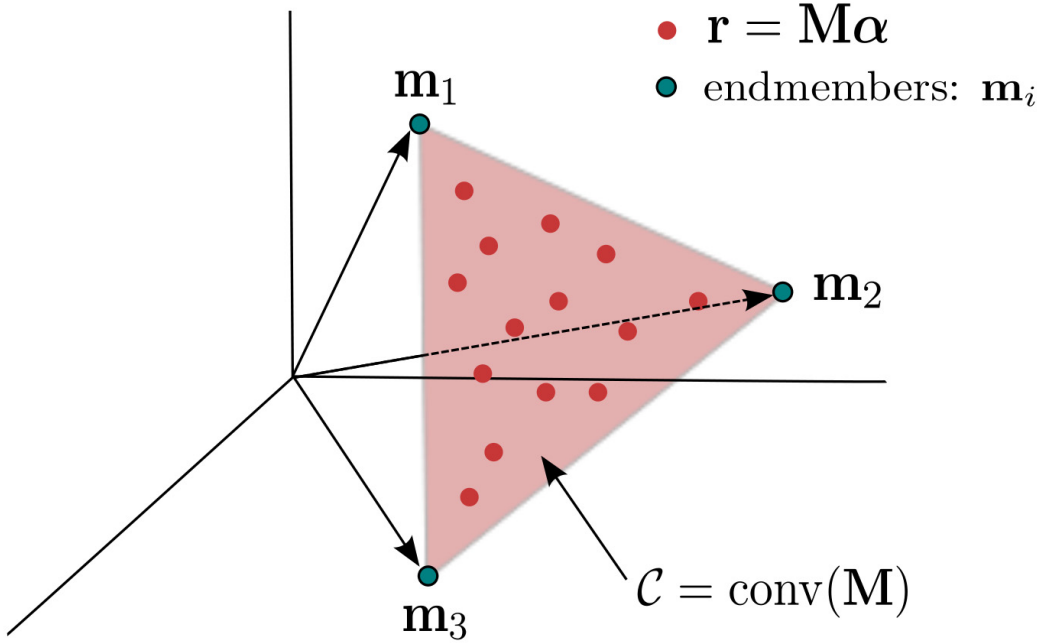
- $\mathbf{r}$ be a $B \times 1$ reflectance vector
- B is the total number of bands
- $\mathbf{m}_i$ is the signature of the ith endmember, $i = 1, \ldots, p$
- $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p]$ is the *mixing matrix* and contains the signatures of the endmembers present in the observed area,
- $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_p]^\top$ is the fractional abundance vector
- $\mathbf{n} = [n_1, \ldots, n_B]^\top$ models additive noise in each spectral channel.

The linear unmixing problem:

$$\boxed{\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n} \quad s.t. \quad \boldsymbol{\alpha} \geq 0, \quad \mathbf{1}_p^\top \boldsymbol{\alpha} = \mathbf{1}_N}$$

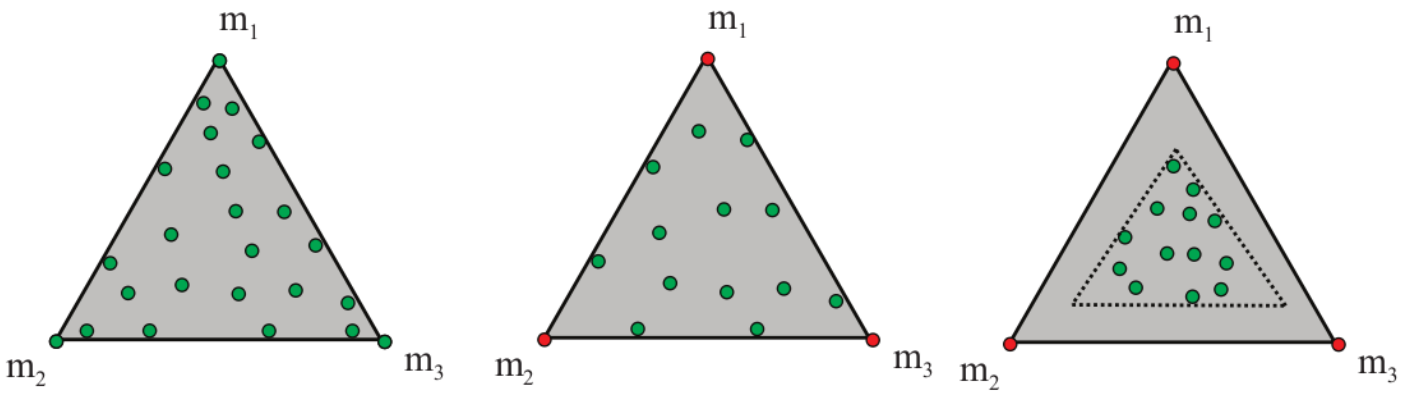
- Given a set of reflectances $\mathbf{r}_i$, $i = 1, \ldots, N$, estimate appropriate values for both $\mathbf{M}$ and $\boldsymbol{\alpha}$
- Two physically reasonable constraints:
  1. all abundances must be positive, $\alpha_i \geq 0$,
  2. they have to sum one, $\sum_{i=1}^{p} \alpha_i = 1$

The simplex representation: Illustration of the simplex set $\mathcal{C}$ for $p = 3$. Points in red denote the available spectral vectors $\mathbf{r}$ that can be expressed as a linear combination of the *endmembers* $\mathbf{m}_i$, $i = 1, \ldots, 3$, (vertices circled in green). The subspace formed defined by these endmembers is the convex hull $\mathcal{C}$ of the columns of $\mathbf{M}$



Credits: Figure from Bioucas-Dias11.

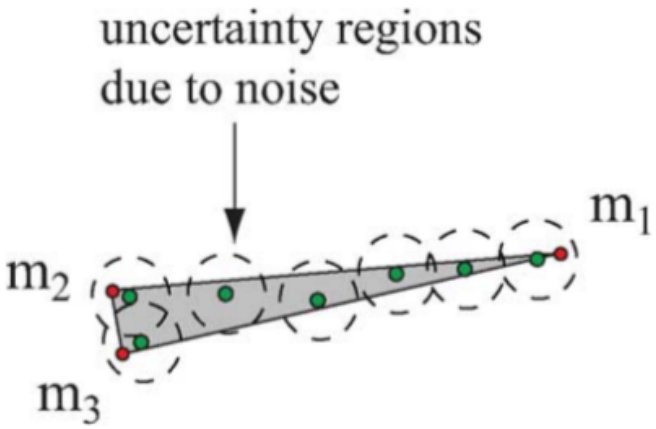
The minimum volume simplex approximation not always works:

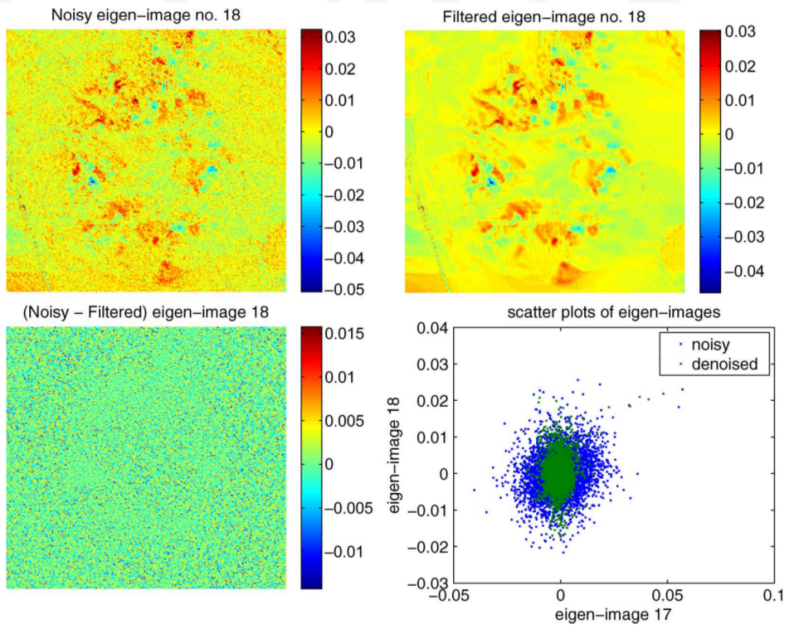


- Left and middle: identifiable!
- Right: not identifiable because of a highly mixed scenario!
- Alternative: statistical models may better capture the data distribution

Credits: Figure from Bioucas-Dias11.

The minimum volume simplex approximation may be affected by noise:





- Endmembers $\mathbf{m}_2$ and $\mathbf{m}_3$ are too close, thus $\mathbf{M}$ is badly conditioned.
- The effect of noise is evident, as represented in the uncertainty regions
- A preliminary step is to check the SNR (and eventually apply MNF):

$$\text{SNR} = \frac{\mathbb{E}[\|\mathbf{r}\|^2]}{\mathbb{E}[\|\mathbf{n}\|^2]} = \frac{\text{trace}(\mathbf{C}_r)}{\text{trace}(\mathbf{C}_n)}$$

```
>> snr=trace(cov(X))/trace(cov(N))
```

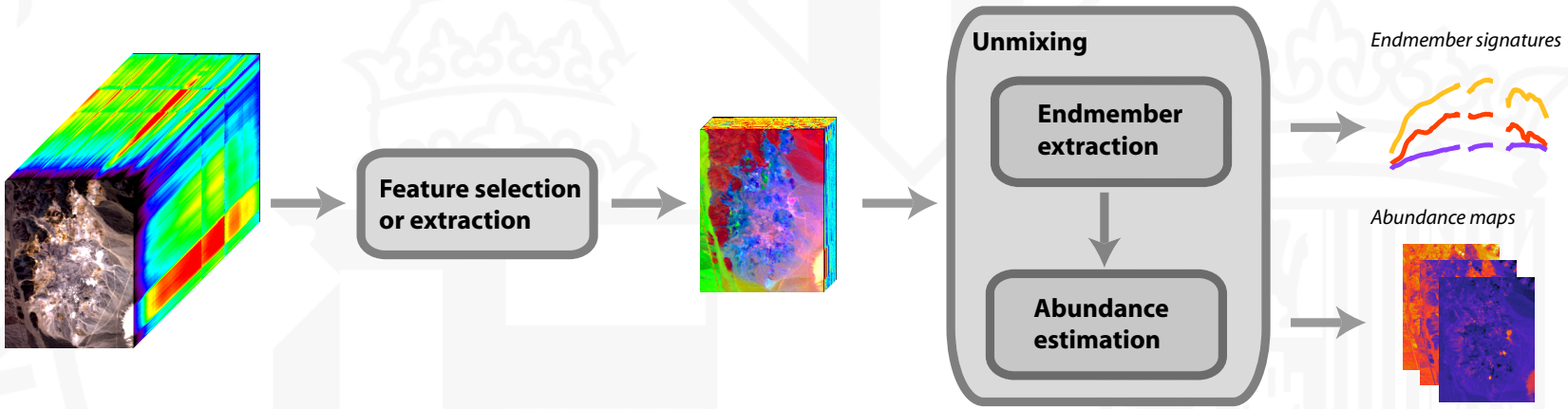
Credits: Figure from Bioucas-Dias11.

Spectral unmixing steps:

a. *Dimensionality reduction*. Spectral unmixing intrinsically assumes that the dimensionality of hyperspectral data is lower and can be expressed in terms of the endmembers. Some methods require a previous dimensionality reduction, either feature selection or extraction, e.g. PCA, MNF, ...

b. *Endmember extraction*. Search of a proper vector basis to describe all the materials in the image:
   - Find the most extreme spectra, which are the purest and those better describing the vertices of the simplex
   - Find the most statistically different pixels

c. *Abundance estimation*. Exploits linear or nonlinear regression techniques for estimating the mixture of materials, called abundance, in each image pixel, e.g. linear regression, neural networks and support vector regression



The first step in the spectral unmixing analysis tries to estimate the number of endmembers present in the scene

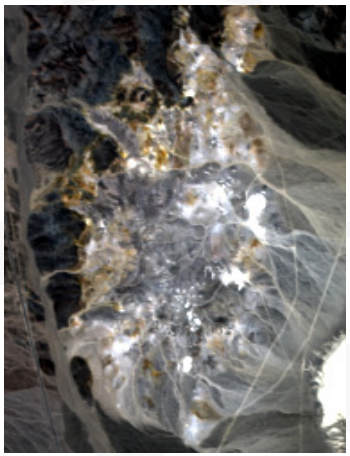
- The number of endmembers is assumed to be lower than the number of bands B
- Statistical and geometrical interpretation: spectral vectors lie in a low-dimensional linear subspace
- Endmember determination reveals the intrinsic dimensionality of the data and reduces the computational complexity of the unmixing algorithms

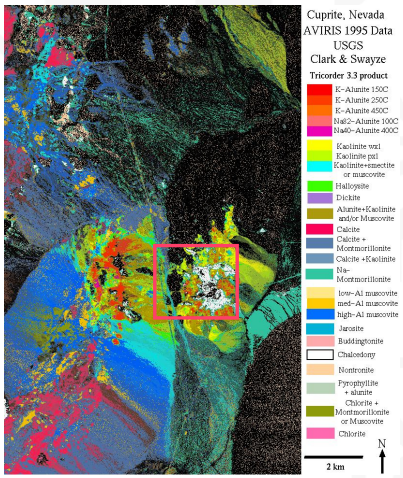
How to estimate the intrinsic dimensionality of the subspace

1. Most of the methods involve solving eigenproblems
   - Jollife86 PCA looks for the explained variance of the projected data (scores)
   - Lee90 MNF looks for the explained variance of the projected data and discounts the noisy components
   - Bioucas-Dias05 HySime looks for the explained SNR by minimizing the MSE error term
2. Information-theoretic approaches
   - Wang06 ICA and projection pursuit looks for a 'right' number of statistically independent components
   - Ifarraguerri00 Minimum description length (MDL)
   - Harsanyi93 Neyman-Pearson detection method (called HFC)
   - Chang04 Virtual dimensionality (VD) finds the highest number for which the correlation matrix have smaller eigenvalues than the covariance matrix
   - Chang04 Noise-whitened HFC (NWHFC) removes the second-order noise statistical correlation
3. Nonlinear (higher-order) methods and manifold learning
   - Bachmann06 ISOMAP
   - Yangchi05 locally linear embedding (LLE)

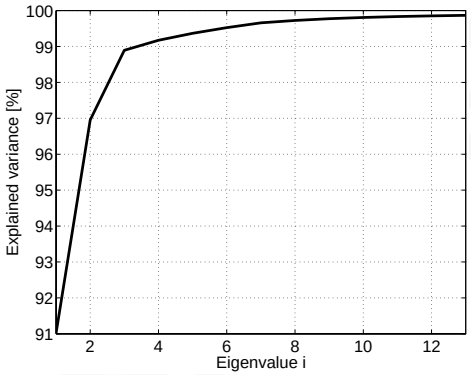
Standard benchmark hyperspectral image dataset:

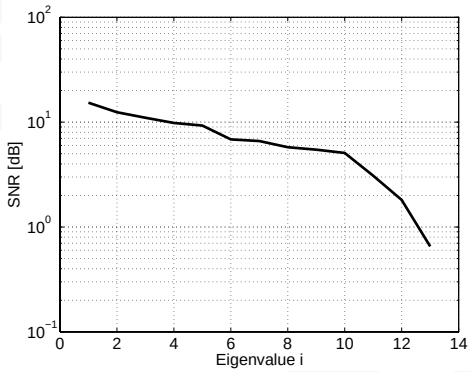
- AVIRIS Cuprite reflectance data set, `http://aviris.jpl.nasa.gov/html/aviris.freedata.html`
- AVIRIS spectrometer over the Cuprite Mining area in Nevada (USA) in 1997
- Widely used to validate the performance of many spectral unmixing and abundance estimation algorithms
- U.S. Geological Survey (USGS) in the form of various mineral spectral libraries, `http://speclab.cr.usgs.gov/spectral-lib.html`
- Many reported materials: buddingtonite, calcite alunite, kaolinite, and montmorillonite, chalcedony, dickite, halloysite, andradite, dumortierite, and sphene.
- Most mixed pixels in the scene consist of alunite, kaolinite, and muscovite

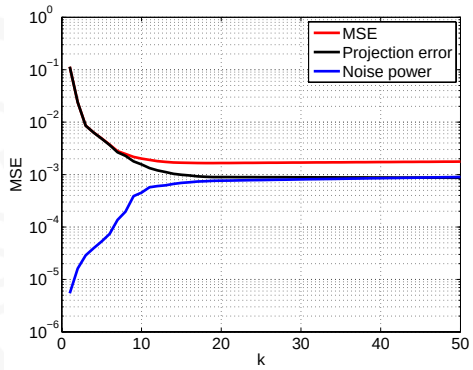




PCA, MNF, and HySime







- PCA: **8 components** retain more than 99.95% of the explained variance
- MNF yields a higher number of distinct pure pixels, **p=13**
- HySime estimates **p=18** (minimum MSE)

HFC and NWHFC are estimated with the false-alarm probability set to different values $P_f = \{10^{-2}, \ldots, 10^{-6}\}$, and give rise to p around 14

	P_F				
Method	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
HFC	23	20	17	16	**14**
NWHFC	21	18	16	**14**	12

Two main families of methods: geometrical vs statistical

Family	Method	Brief description	Auto-matic	CPU cost
Geometrical (pure-pixel)	**IEA** [Neville et al., 1999]	Iteratively selects endmembers that minimize error in the unmixed image	√	High
	VCA[Nascimento and Bioucas-Dias, 2005a]	Iteratively projects data onto a direction orthogonal to the subspace spanned by the previous endmembers	√	Low
	PPI [Boardman, 1993]	Projects spectra onto many random vectors, stores the most extreme distances to select endmembers	√	Medium
	N-FINDR [Winter, 1999]	Finds the pixels defining the simplex with maximum volume trough inflation	√	Medium
	SGA [Chang et al., 2006]	Iteratively grows a simplex by finding the vertices that yield maximum volume	√	Medium
	SMACC[Gruninger et al., 2004]	Iteratively incorporates new endmembers by growing a convex cone representing the data	×	High
Geometrical (min-volume)	**SISAL** [Bioucas-Dias, 2000]	Robust version of min-volume by allowing violation of the positivity constraint	√	Low
	CCA [Ifarraguerri and Chang, 1999]	Iteratively selects endmembers maximizing the correlation matrix eigenspectrum, forces positive endmembers	√	Low
	MVES [Chan et al., 2009]	Implements a cyclic minimization of a series of linear programming problems to find the min-vol simplex	√	Low
	MVT-NMF[Miao and Qi, 2007]	Minimizes a regularized problem: a term minimizing the approximation error of NMF and another constraining the volume of the simplex	×	Low
	ICE [Berman et al., 2004]	Similar to MVT-NMF but replaces the volume by sum of squared distances between all simplex vertices	×	Medium
	SPICE (also sparse)[Zare and Gader, 2007]	Extension of ICE with sparsity-promoting priors	√	Medium
	ORASIS [Bowles et al., 1997]	Iterative procedure with several modules involving endmember selection, unmixing, spectral libraries and spatial postprocessing	×	High
Statistical (info. theory)	**ICA** [Bayliss et al., 1997]	Standard formulation to retrieve the mixing matrix M. Assumes independence of the sources	√	Low
	DECA[Nascimento and Bioucas-Dias, 2007]	Forces a mixture of Dirichlet densities as prior for the abundance fractions	√	Low
Statistical (machine learning)	**SVDD (also geometrical)** [Broadwater et al., 2009]	Hypersphere in kernel space, rejection ratio set to zero	√	Low
	EIHA [Graña et al., 2009]	Lattice auto-associative memories + image segmentation	√	High
Statistical (sparse models)	**BP/OMP** [Pati et al., 2003]	Greedy algorithm based on orthogonal basis pursuit	√	Low
	BPDN [Chen et al., 2001]	Basis pursuit algorithm with a relaxation to solve the BP/OMP problem	√	Low
	ISMA [Rogge et al., 2006]	Iteratively finds an optimal endmembers set by examining the change in RMSE after reconstructing the original scene using the estimated fractional abundance	√	Low

Let's see the main differences

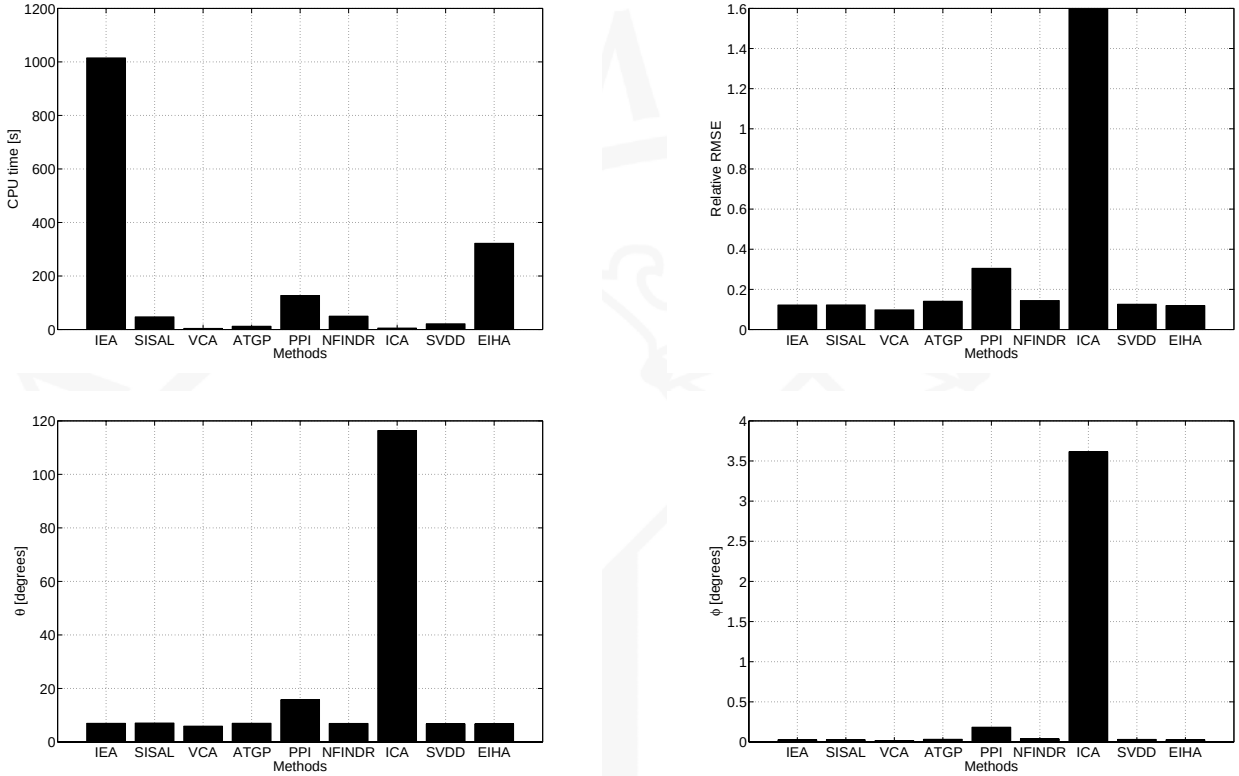
- Several representative methods: 1) pure-pixel geometrical approaches (IEA, VCA, PPI and NFINDR); 2) SISAL for geometrical minimum volume approaches; 3) ICA for the information-theoretic-based methods; and 4) SVDD target detection and EIHA for the machine learning based approaches
- Scores between estimated $\hat{\mathbf{m}}$ and closest $\mathbf{m}$ endmember in the USGS db:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{m}}_i - \mathbf{m}_i)^2}$$

$$\text{SAM} = \text{acos}\left(\frac{\hat{\mathbf{m}}^\top \mathbf{m}}{\|\hat{\mathbf{m}}\|\|\mathbf{m}\|}\right)$$

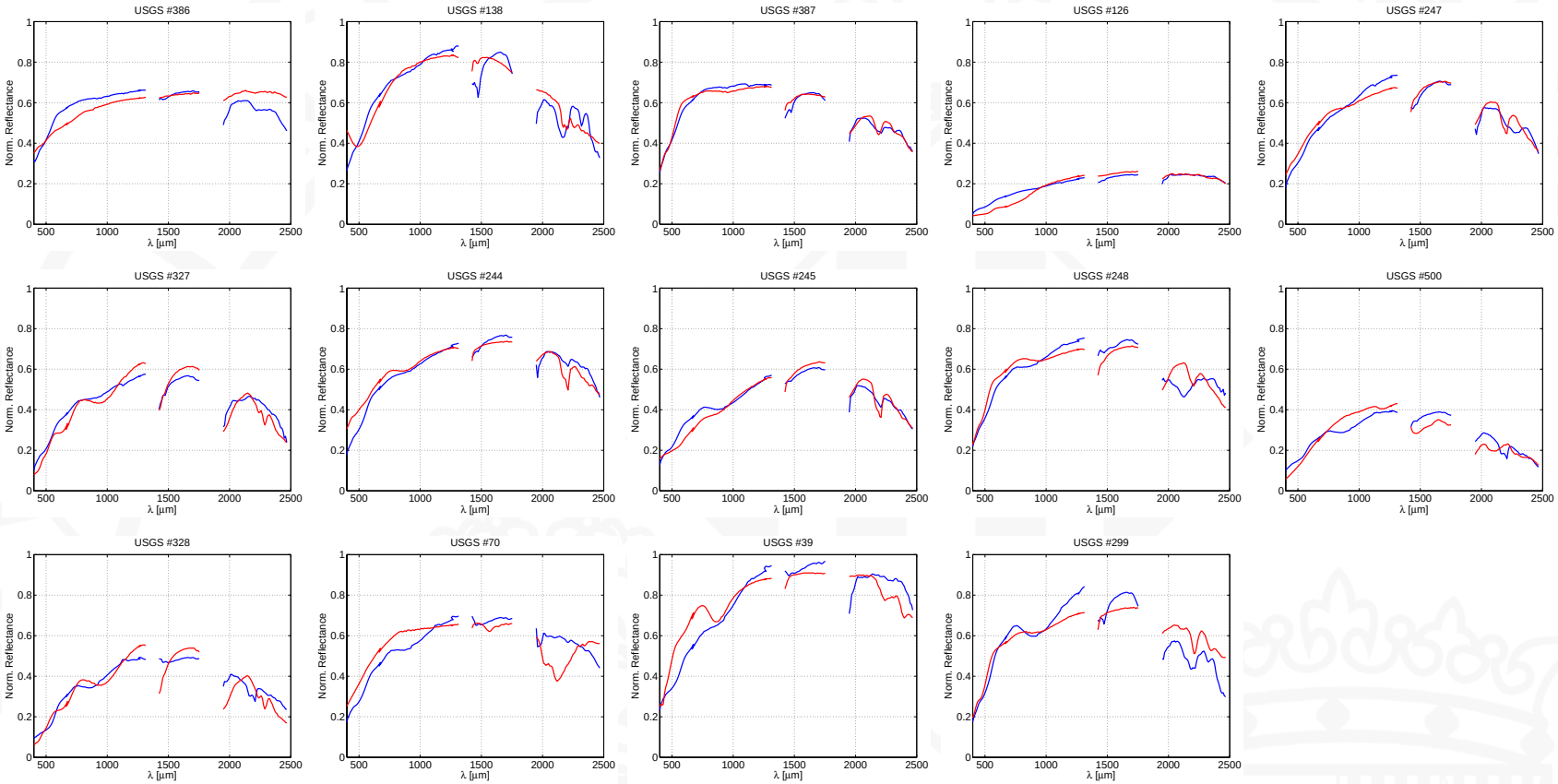
$$\text{SID}(\hat{\mathbf{m}}, \mathbf{m}) = \sum_i p_i \log\left(\frac{p_i}{\hat{p}_i}\right) + \sum_i \hat{p}_i \log\left(\frac{\hat{p}_i}{p_i}\right), \quad \mathbf{p} = \mathbf{m}/\sum_i m_i$$

- We will seek for $p = 14$ endmembers using all methods



- All methods achieve RMSE$<$0.2, except for PPI and ICA
- Similar trends are observed for SAM and SID
- VCA outperformed the rest of the methods in accuracy (in all measures)
- VCA showed very good computational efficiency, closely followed by SVDD, SISAL and IEA
- ICA does not work all (does not meet problem assumptions)
- Very good performance of SVDD

Estimated signatures are in general close to the laboratory spectra



Linear standard models for estimation

- The unconstrained least-squares problem is simply solved by

$$\hat{\alpha} = \mathbf{M}^{\dagger}\mathbf{r} = (\mathbf{M}^{\top}\mathbf{M})^{-1}\mathbf{M}^{\top}\mathbf{r}$$

- The **sum-to-one constraint** means that the LS problem is constrained by $\sum \alpha_i = 1$, which can be solved via Lagrange multipliers
- The **non-negativity constraint** is not as easy to address in closed-form

Linear advanced models for estimation

Harsanyi94 Nonnegative constrained least squares and fully constrained least squares

Keener06 Minimum variance unbiased estimator (MVUE): under the assumption of additive noise, $\mathbf{n}$, with covariance, $\mathbf{C}_n$, the minimum variance estimate of the abundances reduces to

$$\hat{\alpha} = (\mathbf{M}^{\top}\mathbf{C}_n^{-1}\mathbf{M})^{-1}\mathbf{M}^{\top}\mathbf{C}_n^{-1}\mathbf{r}$$

Li04 use wavelet features to improve the linear estimation

Debba06 used derivative spectra in simulated annealing procedures

Chang06 uses a weighted abundance-constrained linear spectral mixture analysis

Bioucas10 reviews the field, including sparse LASSO regression

Nonlinear models for estimation

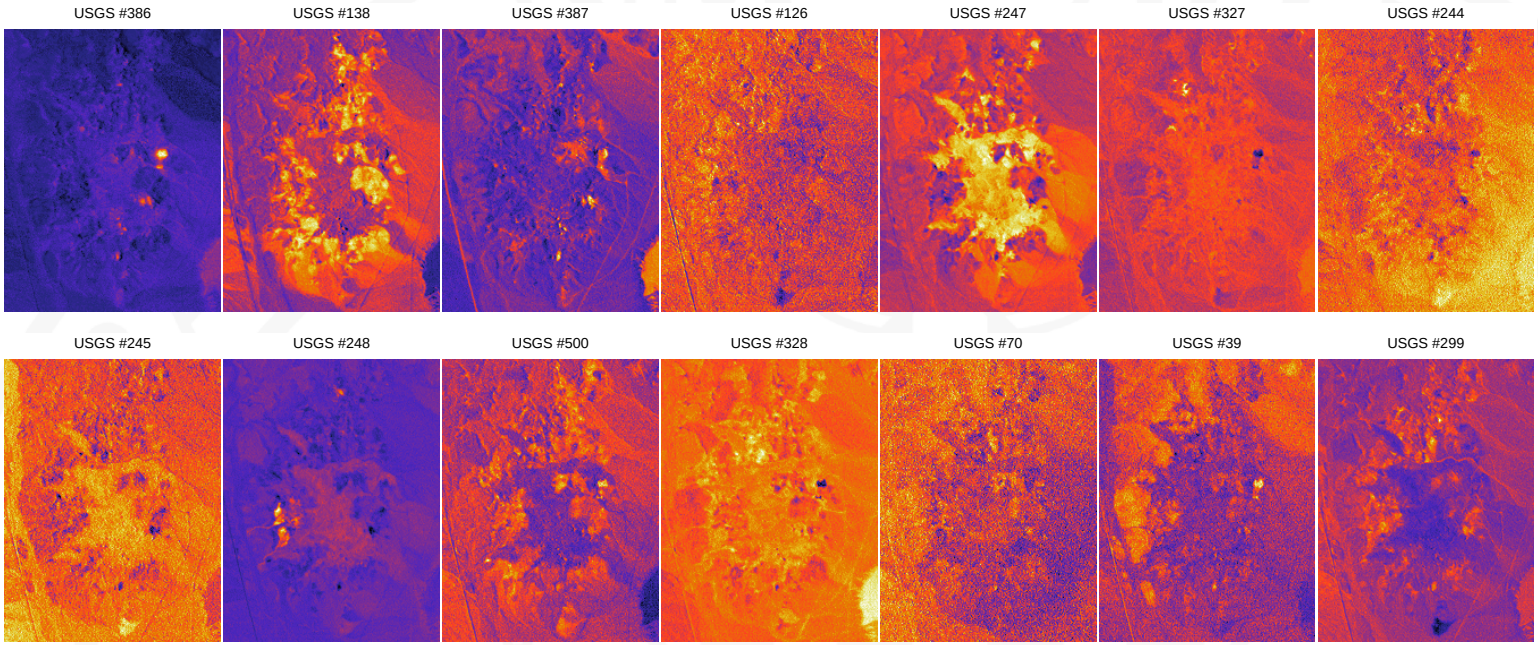
- An easy way to compensante the (strong) assumption of linear mixture models
- Several regression approaches available

  Atkinson97 proposes multilayer perceptrons
  Schowengerdt97 introduces nearest neighbor classifiers
  Brown00 includes support vector machines for unmixing
  Broadwater09 extends the NNCLS method to kernel space
- A simple algorithm for doing nonlinear regression with kernels consists of iterating the equations:

$$\hat{\boldsymbol{\alpha}} = \left(K(\mathbf{M},\mathbf{M})\right)^{-1}\left[K(\mathbf{M},\mathbf{r}) - \boldsymbol{\lambda}\right]$$
$$\boldsymbol{\lambda} = K(\mathbf{M},\mathbf{r}) - K(\mathbf{M},\mathbf{M})\hat{\boldsymbol{\alpha}},$$

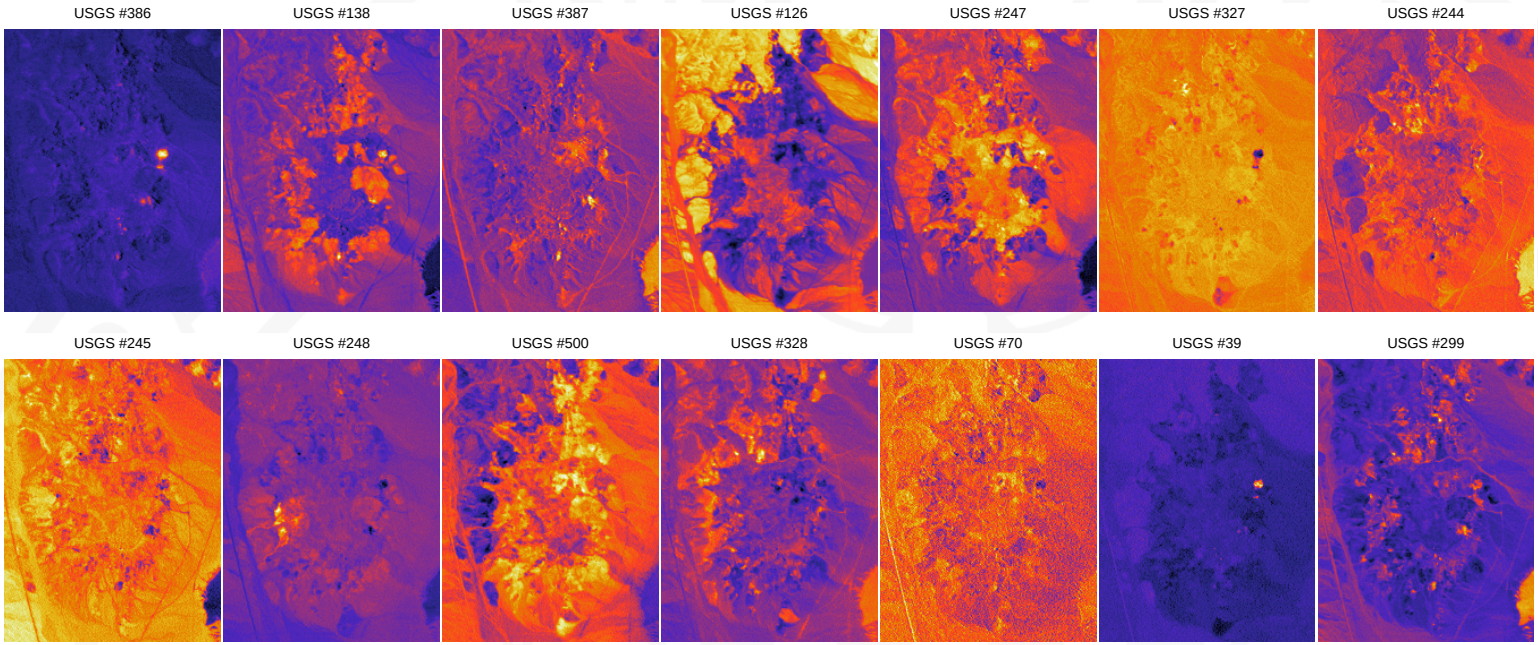
  where $\boldsymbol{\lambda}$ is the Lagrange multiplier vector used to impose the non-negativity constraints of the estimated abundances. Nevertheless, the method does not incorporate the sum-to-one constraint. We refer to this method as the Kernel NonNegativity Least Squares (KNNLS).

Least Squares Abundance Estimation



- The obtained maps nicely resemble the available geological maps.
- Linear and nonlinear methods yield similar results
- The use of the KNNLS achieves more detailed description of the spatial coverage (see e.g. minerals ♯386 and ♯245) or less noisy maps (see e.g. minerals ♯126, ♯139, and ♯299)
- KNNLS has two problems: 1) tuning of the σ parameter for the kernels, and 2) the sum-to-one constraint is met in a trivial way.

Kernel Least Squares Abundance Estimation



- The obtained maps nicely resemble the available geological maps.
- Linear and nonlinear methods yield similar results
- The use of the KNNLS achieves more detailed description of the spatial coverage (see e.g. minerals ♯386 and ♯245) or less noisy maps (see e.g. minerals ♯126, ♯139, and ♯299)
- KNNLS has two problems: 1) tuning of the σ parameter for the kernels, and 2) the sum-to-one constraint is met in a trivial way.

Recent years have witnessed advances in three main directions

Sparse models Spectral vectors can be expressed as linear combinations of a *very few* pure spectral signatures obtained from a (potentially very large) spectral library

Contextual information Inclusion of spatial information helps regularize the solution as close-by pixels in the image should correspond/identify similar elements

Nonlinear models more complex models of the mixture process are assumed: nonlinear mixing holds when the light suffers multiple scattering or interfering paths, which implies that the acquired energy by the sensor results from the interaction with many different materials at different levels or layers

Sparse models:

Intuition/Motivation Spectral vectors can be expressed as linear combinations of a *very few* pure spectral signatures obtained from a (potentially very large) spectral library

Candes06,Donoho06,Blumensath09 Sparse reconstruction/compressive sensing: A sparse signal is exactly recoverable from an underdetermined linear system of equations in a computationally efficient manner via convex/non-convex programming

The linear sparse mixing model:

- A standard linear mixing model

$$\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n}$$

- Only some components of $\boldsymbol{\alpha}$ are active, many shrink to zero!
- The *sparse unmixing* problem: given $\mathbf{r}$ and $\mathbf{M}$, find the sparsest solution:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \qquad s.t. \qquad \mathbf{r} = \mathbf{M}\boldsymbol{\alpha}$$

- Nice idea: interpretable and compact solutions!
- Problem: this is an NP-hard problem!

Convex relaxation optimization strategies

Chen01 Basis Pursuit (BP)

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \mathbf{r} = \mathbf{M}\boldsymbol{\alpha}$$

Chen01 BPDN - BP denoising

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \|\mathbf{r} - \mathbf{M}\boldsymbol{\alpha}\|_2 \leq \delta$$

Tibshirani96 (LASSO)

$$\min_{\boldsymbol{\alpha}} \|\mathbf{r} - \mathbf{M}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

Approximation strategies

Ji08 Bayesian CS

Needell09 Matching Pursuit

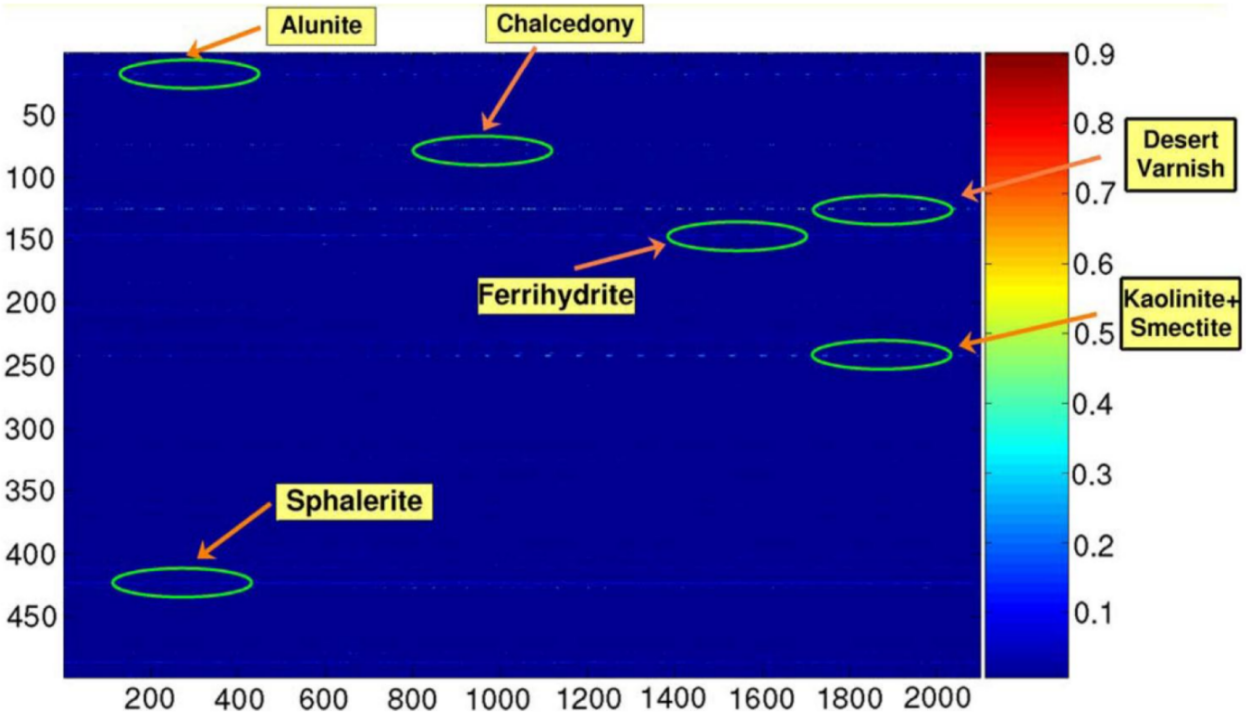
Blumensath09 Iterative Hard Thresholding (IHT)

Garg09 Gradient Descent Sparsification (GDS)

Foucart10 Hard Thresholding Pursuit (HTP)

Villa12 Message Passing (MP)

Results on the USGS Cuprite dataset



- Bad news: Hyperspectral libraries have poor theoretical bounds of recovery, i.e. low restricted isometric property (RIP)
- Good news: Hyperspectral mixtures are highly sparse, very often $p \leq 5$
- Surprising fact: Convex programs (BP, BPDN, LASSO, ...) yield much better empirical performance than non-convex state-of-the-art competitors

Credits: Figure from Bioucas-Dias12.

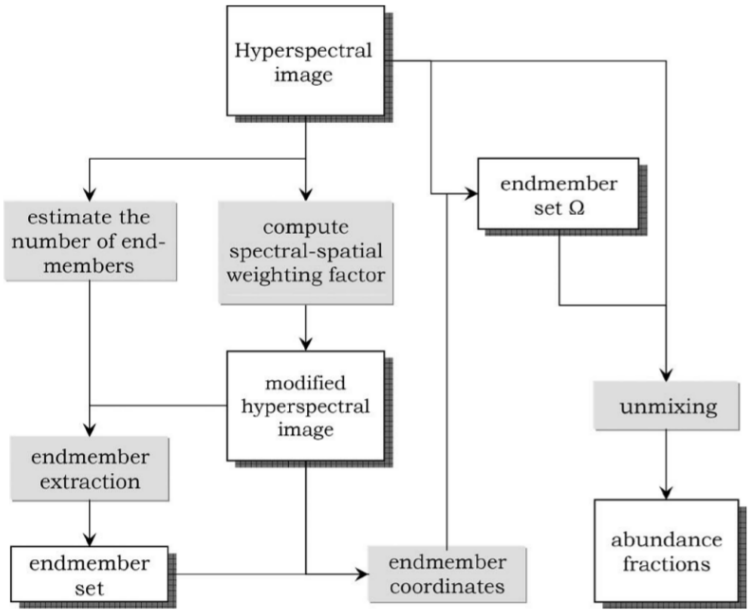
Spatial information: Inclusion of spatial information helps regularize the solution as closeby pixels in the image should correspond/identify similar elements

Main approaches:

Zortea09 Spatial preprocessing (SPP) estimates for each pixel a spatially-derived factor to weight relevance

Plaza02 Automatic morphological endmember extraction (AMEE) algorithm for spatial-spectral endmember extraction

Rogge07 spatial spectral endmember extraction (SSEE) uses a spatial averaging of spectrally similar endmember candidates found via singular value decomposition (SVD)



Credits: Figure from Plaza13.

All methods improve performance with spatial information, and there is an optimal window

Algorithm	Alunite	Buddigntonite	Calcite	Kaolinite	Muscovite	Mean
N-FINDR	9.96°	7.71°	12.08°	13.27°	5.24°	9.65°
OSP	4.81°	4.16°	9.62°	11.14°	5.41°	7.03°
VCA	10.73°	9.04°	6.36°	14.05°	5.41°	9.12°
AMEE	4.81°	4.21°	9.54°	8.74°	4.61°	6.38°
SSEE	4.81°	4.16°	8.48°	11.14°	4.62°	6.64°
SPP+N-FINDR	12.81°	8.33°	9.83°	10.43°	5.28°	9.34°
SPP+OSP	4.95°	4.16°	9.96°	10.90°	4.62°	6.92°
SPP+VCA	12.42°	4.04°	9.37°	7.87°	6.18°	7.98°

Algorithm	$ws = 0$	$ws = 3$	$ws = 5$	$ws = 7$	$ws = 9$	$ws = 11$
OSP	4.791	1.862	1.836	2.656	2.656	3.538
N-FINDR	0.652	0.570	0.548	0.645	0.725	0.545
VCA	0.744	0.608	0.385	0.451	0.768	0.785

Most of the elements are better detected

	OSP				NFINDR				VCA			
USGS Signature	$ws = 0$	$ws = 3$	$ws = 5$	$ws = 9$	$ws = 0$	$ws = 3$	$ws = 5$	$ws = 9$	$ws = 0$	$ws = 3$	$ws = 5$	$ws = 9$
Alunite GDS84	7.67	11.94	–	–	–	9.45	–	–	–	10.45	10.45	–
Alunite GDS82	8.00	6.52	7.26	7.26	6.87	10.90	7.21	8.00	6.47	–	–	6.47
Alunite AL706	19.49	–	16.40	–	19.49	–	14.73	15.51	–	16.40	–	18.59
Buddingtonite GDS85	10.17	10.17	10.17	10.17	7.21	8.33	10.17	10.17	10.17	10.17	10.17	10.17
Calcite WS272	10.03	10.03	10.03	10.03	9.48	10.03	9.99	–	9.48	10.44	10.44	–
Kaolinite KGa-1	10.22	10.22	10.22	10.22	10.22	10.22	10.22	10.22	19.70	17.03	22.83	17.78
Muscovite GDS107	11.06	10.40	12.55	9.80	11.18	12.86	12.82	16.02	10.08	12.38	13.38	12.38
Muscovite GDS108	10.07	10.07	10.22	9.90	9.79	9.79	9.41	9.91	12.78	9.29	13.04	9.80
Muscovite GDS111	21.58	21.58	14.32	21.58	18.06	21.58	16.71	14.71	15.53	13.44	13.44	15.39
Jarosite GDS99	19.22	18.37	20.31	16.22	18.95	16.22	20.31	14.79	16.22	19.53	16.22	15.48
Montmorillonite SWy-1	10.68	8.28	6.95	6.95	11.39	12.60	9.53	7.28	11.97	17.43	11.41	18.48
Pyrophyllite PYS1A	–	–	–	13.79	–	–	–	22.18	21.42	–	21.42	24.24
Chalcedony CU91-6A	7.77	8.05	7.77	9.94	11.40	8.05	9.94	11.40	12.39	17.24	11.19	3.53
Andradite GDS12	18.60	18.04	13.93	18.04	13.26	13.06	7.22	11.83	7.31	10.55	7.73	7.31
Dumortierite HS190.3B	11.95	10.19	11.27	8.96	11.28	9.33	11.27	13.18	11.25	11.25	11.25	11.28
Sphene HS189.3B	–	5.20	8.44	8.44	9.05	7.34	8.59	12.05	8.13	5.20	5.07	6.79
Average	12.61	11.36	11.42	11.52	11.97	11.41	11.29	12.66	12.35	12.91	12.72	12.69

Credits: Figure from Plaza13.

Nonlinear unmixing approaches consider either:

- A fully physically-based model requires inferring the spectral signatures and material densities based on the radiative transfer theory
- Alternative machine learning (statistical) approaches (plus prior physical constraints)

Main approaches:

Borel-Gerst94 A multilayer model that gives rise to an infinite sequence of powers of products of reflectances

A second-order (bilinear approximation) is typically enough

Hapke81 Microscopic mixing model at the albedo level and not at the reflectance level

Broadwater09 proposed alternatives with (physically-inspired) kernel methods

Halimi11 Generalized bilinear models to handle scattering effects, e.g., occurring in the multilayered scene

Guilfoyle01,Liu04,Altmann11,Licciardi11 Neural networks to nonlinearly reduce dimensionality and find a sparse basis

Altmann12 Supervised nonlinear spectral unmixing using a post-nonlinear mixing model

Heylen11,Heylen12 follows a similar approach to NFINDR: maximize the simplex volume computed with geodesic measures on the data manifold

- Moderate spatial resolution in hyperspectral images pose the mixing problem
- Pixels are no longer pure, but a mixture of endmembers
- Linear and nonlinear mixture models can be adopted, yet the LMM dominates
- Many algorithmical approaches to find the purest/extreme pixels in the image
  - Geometrical (pure pixel or min-volume)
  - Statistical (information theory or machine)
- Three main steps to solve the problem
  - Determine/estimate how many endmembers are there
  - Find them
  - Use them for prediction
- Very active research topic, many novel approaches out there:
  - Sparse regression models
  - Structured and collaborative regression
  - Spatial-spectral information
  - Prior knowledge and physics in the statistical models
  - Parallelization of algorithms for fast unmixing

Part 8: Retrieval of biophysical parameters

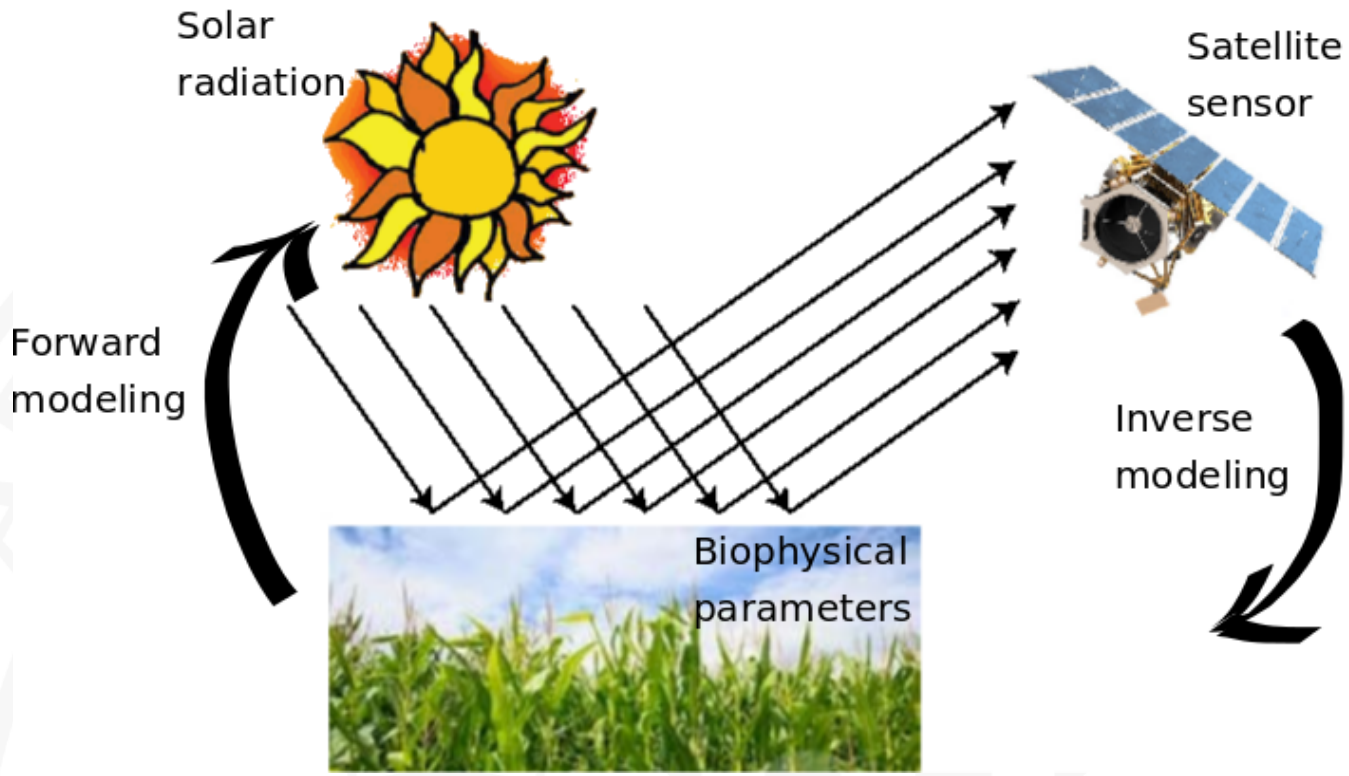
The problem:

- Biophysical parameter retrieval is an essential step in modeling the processes occurring on Earth and the interactions with the atmosphere
- The analysis can be done at local or global scales by looking at bio-geo-chemical cycles, atmospheric situations, ocean/river/ice states, and vegetation dynamics [Lillesand08, Liang08, Rodgers00]
- Land/vegetation parameters are difficult to estimate [Liang04, Liang08]
- Main parameters: temperature, crop yield, biomass, leaf area coverage, chlorophyll content [Liang04, Liang08]

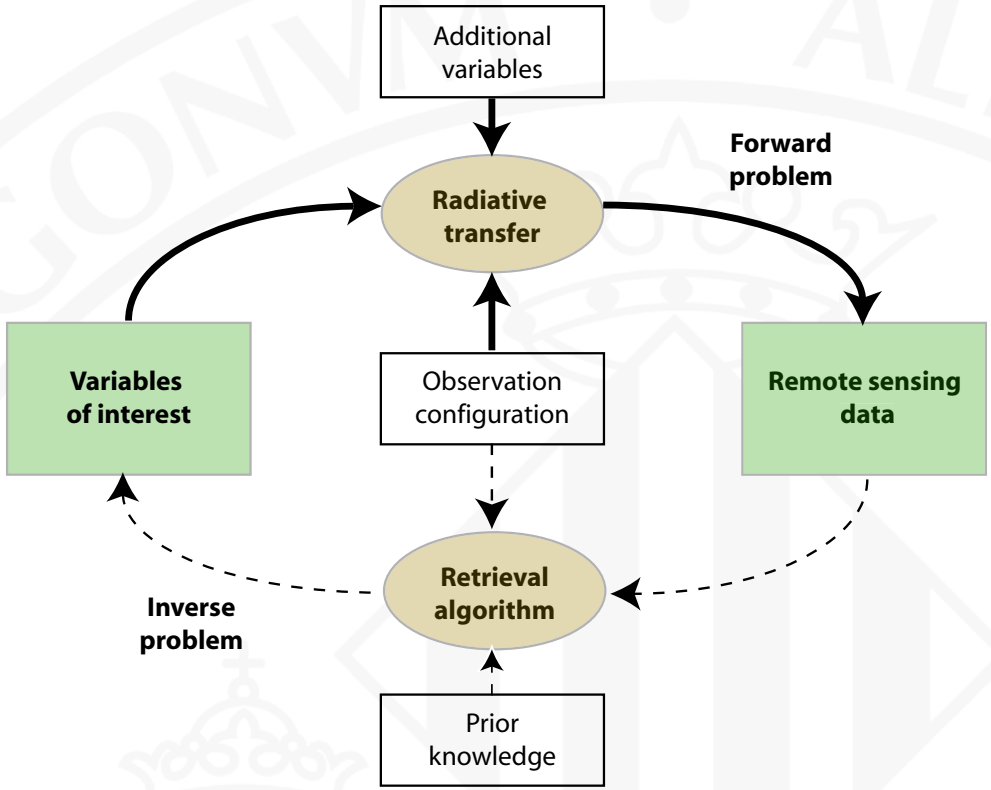
The objective: Transform measurements into biophysical parameter estimates

The data:

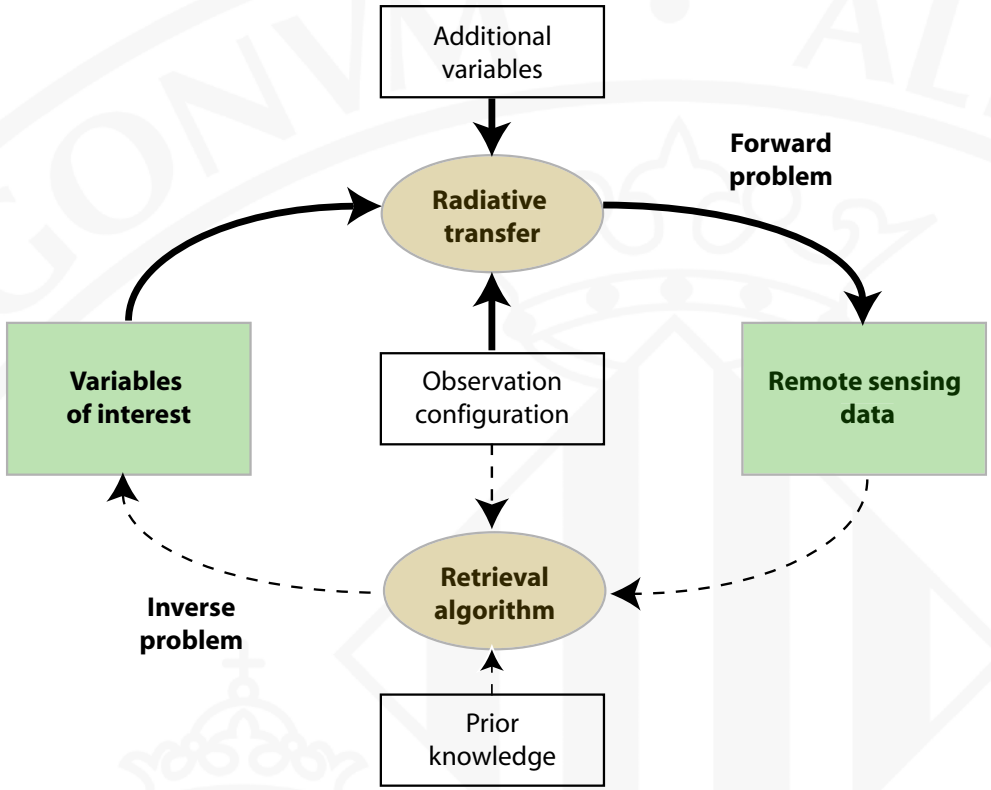
- **Input data:** satellite/airborne spectra, *in situ* (field) radiometers, or simulated spectra by RTMs
- **Output results:** estimation of a bio/geo-physical parameter



- **Forward modeling**: simulation of a database of reflectance spectra and parmaters pairs
- **Inverse modeling**: numerical/statistical inversion of the models from remote sensing data to estimate the parameters



- **Forward modeling**: simulation of a database of pairs of reflectance spectra and parameters
- **Inverse modeling**: numerical/statistical inversion of the models from remote sensing data to estimate the parameters



- The *forward* (or direct) problem involves *radiative transfer models* (RTMs)
- Solving the *inversion* problem implies the design of algorithms that, starting from the radiation acquired by the sensor, can give accurate estimates of the variables of interest, thus 'inverting' the RTM

The discrete forward model can be expressed as:

$$\mathbf{y} = f(\mathbf{X}, \boldsymbol{\theta}) + \mathbf{n}$$

- $\mathbf{y}$ is a set of measurements (e.g. expected radiance)
- $\mathbf{X}$ is a matrix of state vectors that describe the system (e.g. the parameters such as temperature or moisture)
- $\boldsymbol{\theta}$ contains a set of controllable measurement conditions (e.g. combinations of wavelength, viewing direction, time, Sun position, and polarization)
- $\mathbf{n}$ is an additive noise vector
- $f(\cdot)$ is a function which relates $\mathbf{X}$ with $\mathbf{y}$
- f is typically considered to be nonlinear, smooth and continuous

The discrete inverse model is defined as:

$$\hat{\mathbf{X}} = g(\mathbf{y}, \boldsymbol{\omega})$$

where $g(\cdot)$ is a nonlinear function, parametrized by weights $\boldsymbol{\omega}$ that approximates the measurement conditions, $\mathbf{X}$, using a set of observations as inputs, $\mathbf{y}$

Taxonomy of model inversion methods, three main families:

1. The *statistical* inversion models: parametric and non-parametric.
   - Parametric models rely on physical knowledge of the problem and build explicit parametrized expressions that relate a few spectral channels with the bio-geo-physical parameter(s) of interest.
   - Non-parametric models are adjusted to predict a variable of interest using a training dataset of input-output data pairs.
2. *Physical* inversion models: try to reverse RTMs.
   - After generating input-output (parameter-radiance) datasets, the problem reduces to, given new spectra, searching for similar spectra in the dataset and assigning the most plausible ('closest') parameter.
3. *Hybrid* inversion models try to combine the previous approaches.

Two main approaches:

1. **Parametric regression**: assume an explicit model for retrieval

Discrete band approaches (VIs)	Quasi-continuous spectral bands
2-band: SR, NDVI, PRI, OSAVI	*Red-edge position (REP)*
3-band: TVI, MCARI, SIPI	*Integral/Derivative indices*
$\geq 4-$ band: TCARI/OSAVI	*Continuum removal*

2. **Non-parametric regression**: do not assume explicit feature relations

Linear nonparametric models
Stepwise multiple linear regression (SMLR)
Partial least squares regression (PLSR)
Ridge regression (RR)
Least Absolute Shrinkage and Selection Operator (LASSO)

Nonlinear nonparametric models
Decision trees, bagging and random forests
Neural networks
Kernel methods: SVR, RVM, KRR, GPR
Bayesian networks

Literature review of parametric approaches, VI:

Jordan69,Liang04,Liang08 simple ratios

Rouse74 Normalized difference vegetation index (NDVI)

Gamon92 Photochemical reflectance index (PRI)

Rondeaux96 Optimized soil adjusted vegetation index (OSAVI)

Broge01 Triangular vegetation index (TVI)

Daughtry00 Modified *Cab*Absorption in Reflectance Index (MCARI)

Haboudane02 Transformed CARI (TCARI)

Penuelas95 Structure Insensitive Pigment Index (SIPI)

Haboudane02 Combination of indices, TCARI/OSAVI

Thenkabail00,LeMaire04,LeMaire08,Mariotto13 Quality assessment

Literature review of parametric approaches, quasi-continuous bands:

Baret92,Broge01,Clevers02 High-order curve fitting of the first derivative in the red-edge

Miller90 Inverted Gaussian models

Guyot88 Linear interpolation and extrapolation

Dawson98 Lagrangian interpolation

Baranoski05 Rational function

Broge01,Oppelt04,Mutanga05,Malenovsky06,Delegido10 Integral-based indices

Sims02,Penuelas94,Elvidge95,Zarco-Tejada02,LeMaire04 Derivative-based indices

Clark84 Continuum removal for absorption features comparison

Parametric approaches:

Weaknesses	Strengths
• Makes only poorly use of the available information within the spectral observation; at most a spectral subset is used. Therefore, they tend to be more noise-sensitive as compared to full-spectrum methods • Parametric regression puts boundary conditions at level of chosen bands, formulations and regression function. • Statistical function accounts for one variable at the time. • A limited portability to different measurement conditions or sensor characteristics • No uncertainty estimates are provided. Hence the quality of the output maps remain unknown.	• Simple and comprehensive regression models; little knowledge of user required • Fast in processing • Computationally inexpensive

Literature review of linear nonparametric approaches

Yoder95,Fourty97,Bartholomeus12 Stepwise multiple linear regression (LR)

Liang08 Principal component regression (PCR)

Hansen02,Cho07,Darvishzadeh08,Ye08,Im09 Partial least squares regression (PLSR)

Addink07 ridge regression (RR)

Lazaridis11 Least Absolute Shrinkage and Selection Operator (LASSO)

Literature review of nonlinear nonparametric approaches

Im09,Im12,leMaire11,Viedma12,Hansen02 decision trees

CampsValls14 bagging and random forests

Jin97,Paruelo97,Francl97,Kimes99,Kavzoglu03,Huang04,Jensen12,CampsValls13 artificial neural networks

Arenas12,Arenas13,Izquierdo14 kernel feature extraction (KPLS, KOPLS)

CampsValls06,CampsValls10 relevance vector machines (RVM)

Yang01,CampsValls06 support vector regression (SVR)

Peng11,Wang11,CampsValls12 kernel ridge regression (KRR)

Verrelst11,CampsValls12,Verrelst12,Lazaro13 Gaussian processes (GP) on Sentinel-2 data (KRR, GP)

Non-parametric approaches:

Weaknesses	Strengths
• Training can be computational expensive. • They can create over-complex models that do not generalize well from the training data (overfitting). • Therefore, several regressors cannot be trained with high number of samples. • Expert knowledge required, e.g. for tuning. However, toolboxes exist that automate some steps. • Most of them act as a black box. • Some regressors behave rather unstable when applied to data that deviate from statistically different from those used for training.	• Can make use of all bands (full spectral information). • Build advanced, adaptive (nonlinear) models. • Enables accurate and robust performances. • Some methods cope well with redundancy and noisy data. • Once trained, fast processing images. • Some of them (e.g. NN, decision trees) can be trained with high numbers of samples (e.g. $> 10^6$). • Some methods provide insight in model development (e.g. GPR: relevant bands; decision trees: model structure). • Some methods provide uncertainty intervals (e.g. GPR, KRR).

How to measure goodness of a model?

Given two variables y_i and $\hat{y}_i$, $i = 1, \ldots, N$

- *Error (residuals)*: $e_i = y_i - \hat{y}_i$
- *Bias*: mean error (ME):

$$ME = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)$$

- *Accuracy*:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \qquad MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

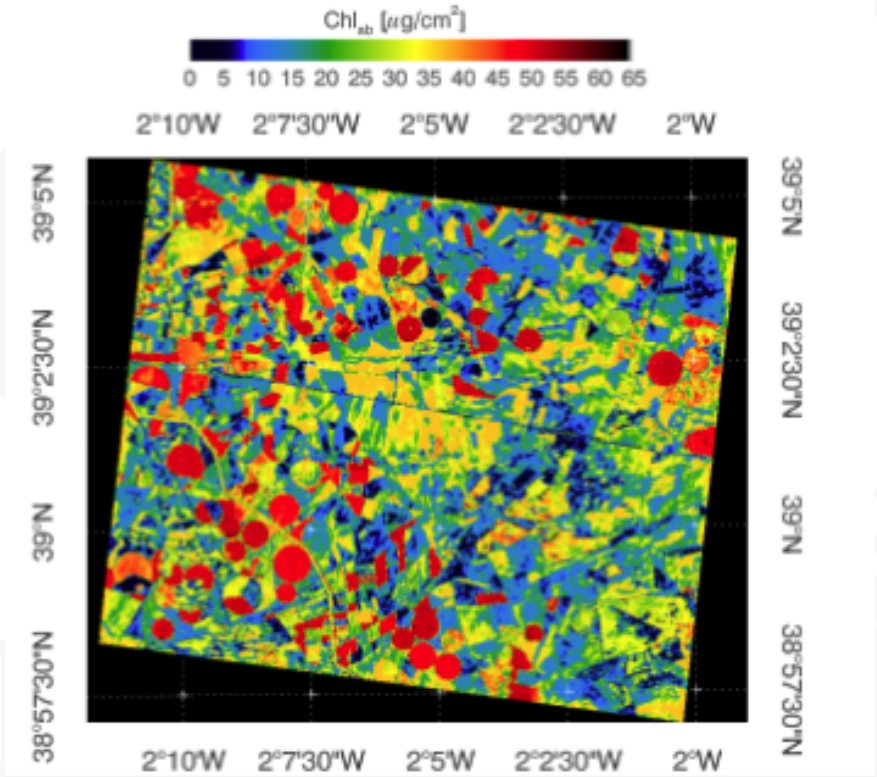
- *Goodness-of-fit*: Pearson's correlation coefficient
- *Matlab*:

```
>> ME = mean(Labels-PreLabels);
>> RMSE = sqrt(mean((Labels-PreLabels).^2));
>> MAE = mean(abs(Labels-PreLabels));
>> r = corrcoef(Labels,PreLabels); R = r(1,2);
>> RESULTS = assessment(y,yhat,'regress')
```

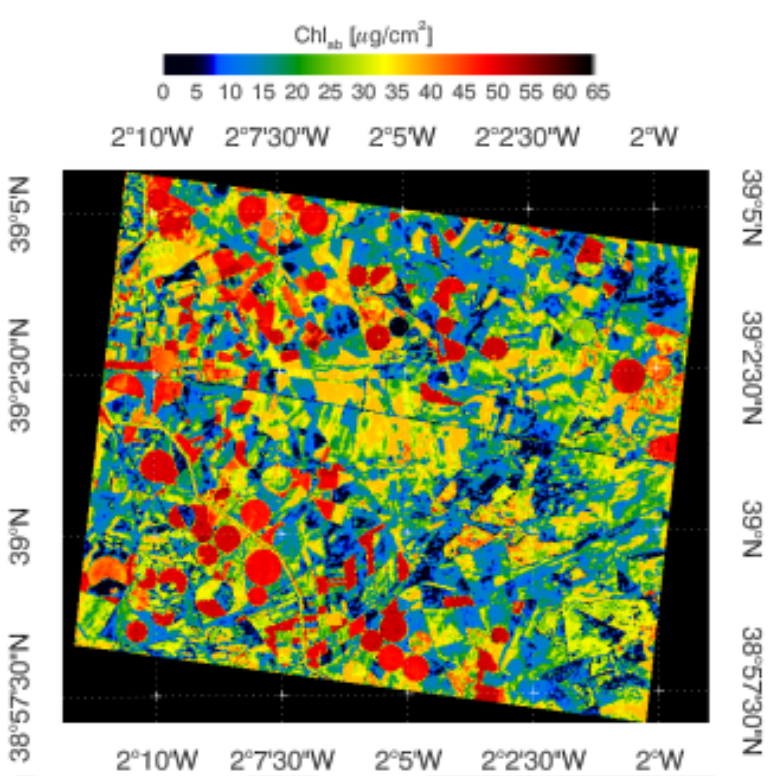
- **Data: SPARC data set (2003, 2004; Barrax, Spain)**
  - Field data: Chl measured with CCM-200
  - 30 additional bare soil samples
  - CHRIS mode 1 (62 bands; 34m) nadir spectra
- **Kernel ridge regression (and GPs) excel in predicting Chla-LAI-fCover over many parametric indices**

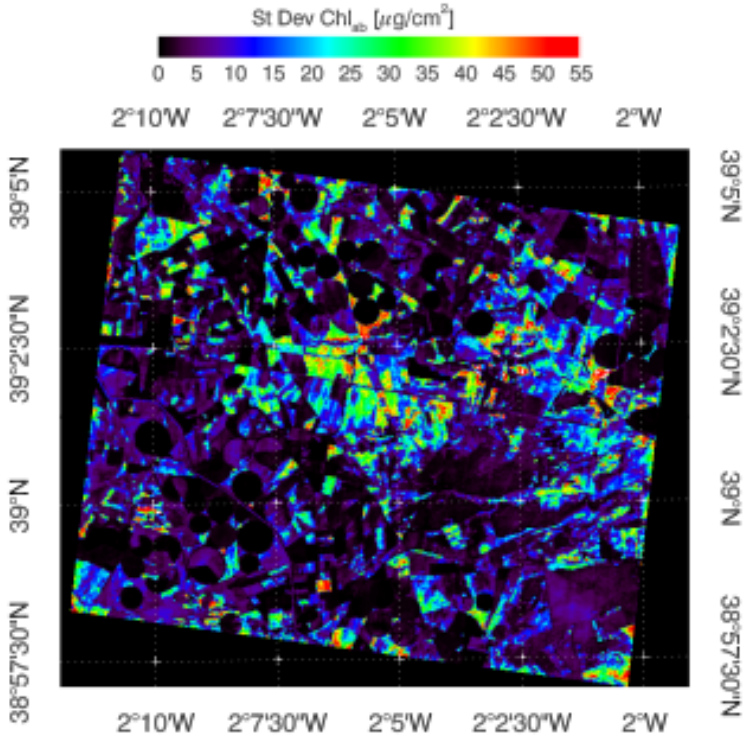
Table 6.1: Correlation coefficient R results of narrowband and broadband indices proposed in relevant literature tested in the present study along with recent non-parametric models. See Verrelst et al. 2011 and references therein.

Method	Formulation	R
GI	R_{672}/R_{550}	0.52 (0.09)
GVI	$(R_{682}\text{-}R_{553})/(R_{682}+R_{553})$	0.66 (0.07)
Macc	$(R_{780}\text{-}R_{710})/(R_{780}+R_{680})$	0.20 (0.29)
MCARI	$[(R_{700}\text{-}R_{670})\text{-}0.2(R_{700}\text{-}R_{550})]/(R_{700}/R_{670})$	0.35 (0.14)
MCARI2	$1.2[2.5(R_{800}\text{-}R_{670})\text{-}1.3(R_{800}\text{-}R_{550})]$	0.71 (0.12)
mNDVI	$(R_{800}\text{-}R_{680})/(R_{800}+R_{680}\text{-}2R_{445})$	0.77 (0.12)
mNDVI_{705}	$(R_{750}\text{-}R_{705})/(R_{750}+R_{705}\text{-}2R_{445})$	0.80 (0.07)
mSR_{705}	$(R_{750}\text{-}R_{445})/(R_{705}+R_{445})$	0.72 (0.07)
MTCI	$(R_{754}\text{-}R_{709})/(R_{709}+R_{681})$	0.19 (0.26)
mTVI	$1.2[1.2(R_{800}\text{-}R_{550})\text{-}2.5(R_{670}\text{-}R_{550})]$	0.73 (0.07)
NDVI	$(R_{800}\text{-}R_{670})/(R_{800}+R_{670})$	0.77 (0.08)
NDVI2	$(R_{750}\text{-}R_{705})/(R_{750}+R_{705})$	0.81 (0.06)
NPCI	$(R_{680}\text{-}R_{430})/(R_{680}+R_{430})$	0.72 (0.08)
NPQI	$(R_{415}\text{-}R_{435})/(R_{415}+R_{435})$	0.61 (0.15)
OSAVI	$1.16(R_{800}\text{-}R_{670})/(R_{800}+R_{670}+0.16)$	0.79 (0.09)
PRI	$(R_{531}\text{-}R_{570})/(R_{531}+R_{570})$	0.77 (0.07)
PRI2	$(R_{570}\text{-}R_{539})/(R_{570}+R_{539})$	0.76 (0.07)
PSRI	$(R_{680}\text{-}R_{500})/R_{750}$	0.79 (0.08)
RDVI	$(R_{800}-R_{670})/\sqrt{(R_{800}+R_{670})}$	0.76 (0.08)
SIPI	$(R_{800}\text{-}R_{445})/(R_{800}\text{-}R_{680})$	0.78 (0.08)
SPVI	$0.4[3.7(R_{800}\text{-}R_{670})\text{-}1.2(R_{530}\text{-}R_{670})]$	0.70 (0.08)
SR	R_{800}/R_{680}	0.63 (0.12)
SR1	R_{750}/R_{700}	0.74 (0.07)
SR2	R_{752}/R_{690}	0.68 (0.09)
SR3	R_{750}/R_{550}	0.75 (0.07)
SR4	R_{672}/R_{550}	0.76 (0.10)
SRPI	R_{430}/R_{680}	0.76 (0.09)
TCARI	$3[R_{700}\text{-}R_{670})\text{-}0.2(R_{700}\text{-}R_{550})(R_{700}/R_{670})]$	0.53 (0.13)
TVI	$0.5[120R_{750}\text{-}R_{550})\text{-}200(R_{670}\text{-}R_{550})]$	0.70 (0.10)
VOG	$R_{740}/(R_{720}$	0.76 (0.06)
VOG2	$(R_{734}\text{-}R_{747})/(R_{715}+R_{726})$	0.72 (0.09)
NAOC	Area in [643, 795]	0.79 (0.09)
LR	Least squares	0.88 (0.06)
SVR Smola and Schölkopf 2004	RBF kernel	0.98 (0.03)
MSVR Tuia et al. 2011b	RBF kernel	0.98 (0.03)
GP Verrelst et al. 2011	Anisotropic RBF kernel	0.99 (0.02)



- **Data: SPARC data set (2003, 2004; Barrax, Spain)**
  - Field data: Chl measured with CCM-200
  - 30 additional bare soil samples
  - CHRIS mode 1 (62 bands; 34m) nadir spectra
- **Gaussian Processes also provide confidence intervals for the predictions (e.g. to identify poorly-sampled areas)**





Motivation:

- Statistical approaches may lack transferability, generality, and robustness to new geographical areas
- Physical models can fill in the gap for estimating bio-geo-chemical structural state variables from spectra

Physically-based inversion:

- Rely on well-established physical laws encoded in radiation transfer models (RTMs), and a set of remote sensing measurements
- Physically-sound approach to retrieve biophysical variables over terrestrial surfaces because it is generally applicable [Dorigo07]
- The advantage of physical models is that they can be coupled from lower to higher levels (e.g. canopy level models build upon leaf models), thereby providing a physically-based linkage between optical EO data and biochemical or structural state variables

RTMs in forward mode create a database (LUT) covering a wide range of situations and configurations:

- Sensitivity studies of canopy parameters relative to diverse observation specifications
- Improved understanding of the Earth Observation (EO) signal as well as to an optimized instrument design of future EO systems

RTMs in inversion mode enables retrieving particular characteristics from EO data:

- The unique and explicit solution for a model inversion depends on the number of free model parameters relative to the number of available independent observations
- A prerequisite for a successful inversion is therefore the choice of a validated and appropriate RTM, which correctly represents the radiative transfer within the observed target
- When a unique solution is not achieved then more *a priori* information is required to overcome the ill-posed problem

Inversion of radiative transfer models

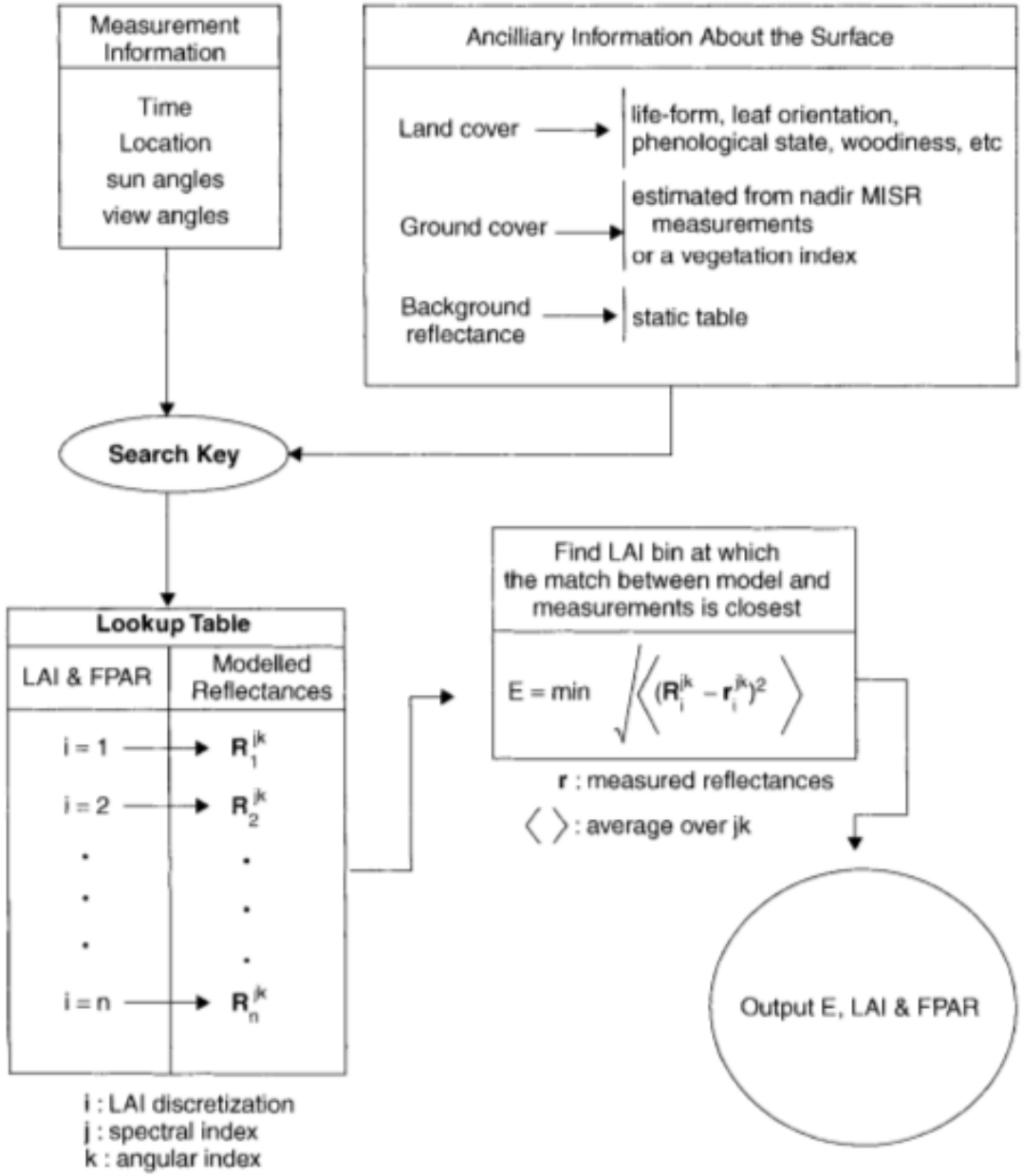
- Inverting an RTMs means: given a spectra, find the closest spectra in the database and return the corresponding parameter
- Given a set of n data pairs generated by an RTM, $\{\mathbf{y}, \mathbf{X}\}_{i=1}^{n}$,

$$\min_{\theta}\{\|\mathbf{y} - f(\mathbf{X};\theta)\|^2\}$$

Two main approaches:

Jacquemoud95,Kuusk98,Zarco-Tejada01 *Numerical optimization* minimizes a function that calculates the RMSE between the measured and estimated quantities by successive input parameter iteration

Liang07 *Look-up tables (LUT)* precompute the model reflectance for a large range of combinations of parameter values, so the problem reduces to searching a LUT for the modeled reflectance that most resembles the measured one


Measurement Information
Time
Location
sun angles
view angles
Ancilliary Information About the Surface
Land cover
life-form, leaf orientation, phenological state, woodiness, etc
Ground cover
estimated from nadir MISR measurements or a vegetation index
Background reflectance
static table
Search Key
Lookup Table
LAI & FPAR
Modelled Reflectances
i = 1
i = 2
i = n
Find LAI bin at which the match between model and measurements is closest
E = min
r : measured reflectances
〈 〉: average over jk
Output E, LAI & FPAR
i : LAI discretization
j : spectral index
k : angular index

Role of regularization:

Dorigo2009,Verrelst12c,Laurent2013 At a local scale, use of prior knowledge to constrain model parameters per land cover class

At a global scale, the MODIS LAI inversion algorithm constrains the structural and optical parameter space per biome

Richter2009,Combal2002,Koetz2005,Richter2011,Darvishzadeh2011 The use of multiple best solutions in the inversion (instead of the single best solution)

Richter2009,Koetz2005,Richter2011 The addition of Gaussian noise to account for uncertainties attached to measurements and models.

Meroni2004,Fang05,Schlerf2005,Darvishzadeh2011 Improved performance when only few well-chosen wavelengths are chosen for model inversion

Atzberger2004,Atzberger2012 Spatial information

Koetz2005,Lauvernet2008 Temporal constraints

At a global scale, the MODIS LAI inversion algorithm constrains the structural and optical parameter space per biome:

	Biome 1		Biome 2		Biome 3		Biome 4		Biome 5		Biome 6	
NDVI	LAI	FPAR	LAI	FPAR	LAI	FPAR	LAI	FPAR	LAI	FPAR	LAI	FPAR
0.025	0	0	0	0	0	0	0	0	0	0	0	0
0.075	0	0	0	0	0	0	0	0	0	0	0	0
0.125	0.3199	0.1552	0.2663	0.1389	0.2452	0.132	0.2246	0.1179	0.1516	0.07028	0.1579	0.08407
0.175	0.431	0.2028	0.3456	0.1741	0.3432	0.1774	0.3035	0.1554	0.1973	0.08922	0.2239	0.1159
0.225	0.5437	0.2457	0.4357	0.2103	0.4451	0.2192	0.4452	0.218	0.2686	0.1187	0.324	0.1618
0.275	0.6574	0.2855	0.5213	0.2453	05463	0.2606	0.574	0.271	0.3732	0.1619	0.4393	0.2121
0.325	0.7827	0.3283	0.6057	0.2795	0.6621	0.3091	0.7378	0.3395	0.5034	0.2141	0.5629	0.2624
0.375	0.931	0.3758	0.6951	0.3166	0.7813	0.3574	0.878	0.393	0.6475	0.2714	0.664	0.3028
0.425	1.084	0.419	0.8028	0.3609	0.8868	0.3977	1.015	0.4425	0.7641	0.32	0.7218	0.333
0.475	1.229	0.4578	0.9313	0.4133	0.9978	0.4357	1.148	0.4839	0.9166	0.3842	0.8812	0.393
0.525	1.43	0.5045	1.102	0.4735	1.124	0.4754	1.338	0.5315	1.091	0.4402	1.086	0.4599
0.575	1.825	0.571	1.31	0.535	1.268	0.5163	1.575	0.5846	1.305	0.4922	1.381	0.5407
0.625	2.692	0.6718	1.598	0.6039	1.474	0.566	1.956	0.6437	1.683	0.568	1.899	0.6458
0.675	4.299	0.8022	1.932	0.666	1.739	0.6157	2.535	0.6991	2.636	0.702	2.575	0.7398
0.725	5.362	0.8601	2.466	0.7388	2.738	0.7197	4.483	0.8336	3.557	0.7852	3.298	0.8107
0.775	5.903	0.8785	3.426	0.822	5.349	0.8852	5.605	0.8913	4.761	0.8431	4.042	0.8566
0.825	6.606	0.9	4.638	0.8722	6.062	0.9081	5.777	0.8972	5.52	0.8697	5.303	0.8964
0.875	6.606	0.9	6.328	0.9074	6.543	0.9196	6.494	0.9169	6.091	0.8853	6.501	0.9195
0.925	6.606	0.9	6.328	0.9074	6.543	0.9196	6.494	0.9169	6.091	0.8853	6.501	0.9195
0.975	6.606	0.9	6.328	0.9074	6.543	0.9196	6.494	0.9169	6.091	0.8853	6.501	0.9195

Source: Myneni et al. (1999).

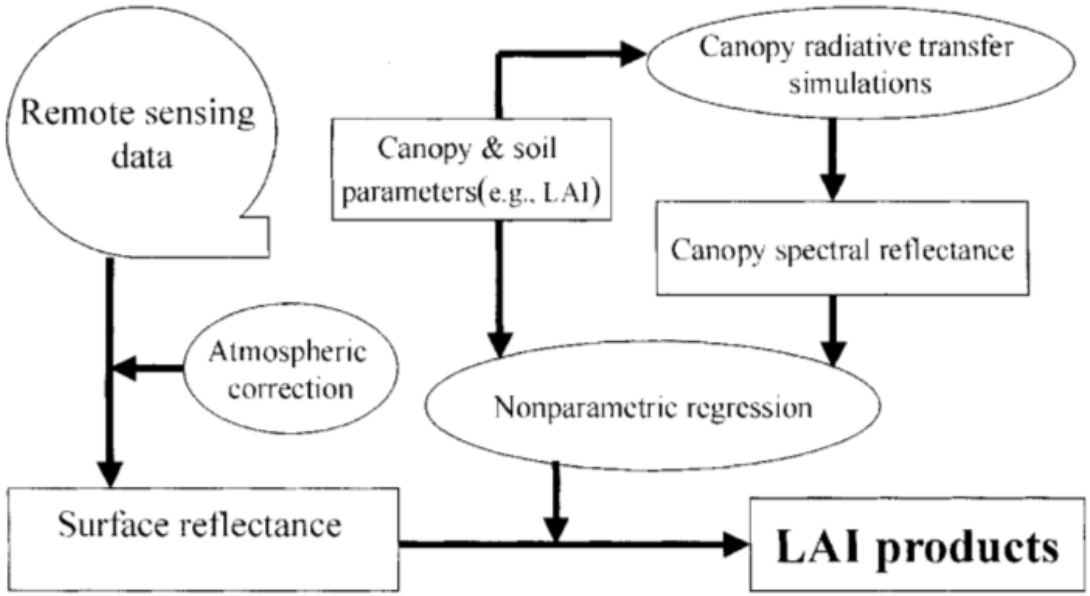
Physical approaches:

Weaknesses	Strengths
• Computational-expensive because per-pixel based and therefore slow (however solutions based on *a priori* info have been developed) • Quality depends on quality RT models, prior knowledge and regularization. • Quite complicated approach: parametrization and optimization required. • The imposed upper/lower boundaries in the LUT had as a logical consequence that estimated parameters could not go beyond the imposed bounds. This contradicts somewhat the physical approach as the prior information has an overwhelming influence • LUT-based inversion methods are often strongly affected by noise and measurement uncertainty	• Reputation of physically-based (however note influence of regularization factors) • Generally and globally applicable (e.g. MODIS) • Additional information about uncertainty of the retrievals (e.g. residuals).

Hybrid inversion method:

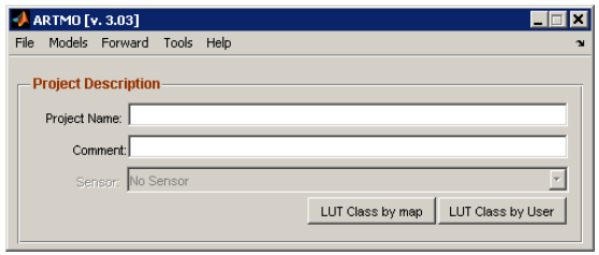
- **The approach:** combination of extensive simulations using a canopy RTM model (physical) and a non-parametric statistical inversion model (statistical)
- **Advantanges:**
  - Exploit the advantages of physically-based models and the flexibility and computational efficiency of nonparametric nonlinear regression methods.
  - Many possible combinations of RTMs and regression models
  - Very efficient approach
- **Shortcomings:**
  - How many parameters?
  - How many radiance-parameters do we need in the database?
  - How to include regularization with noise-fre data?

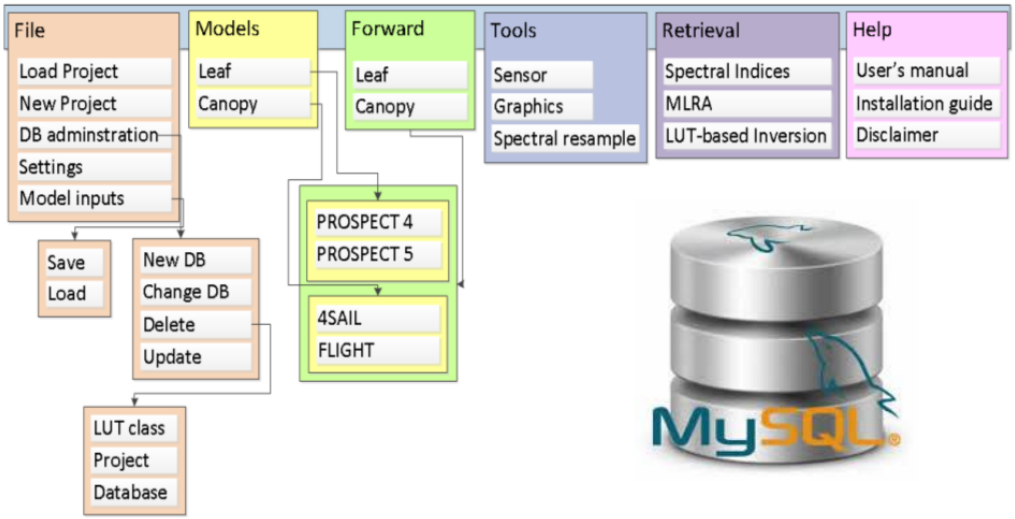
A hybrid approach for LAI estimation [Liang03]



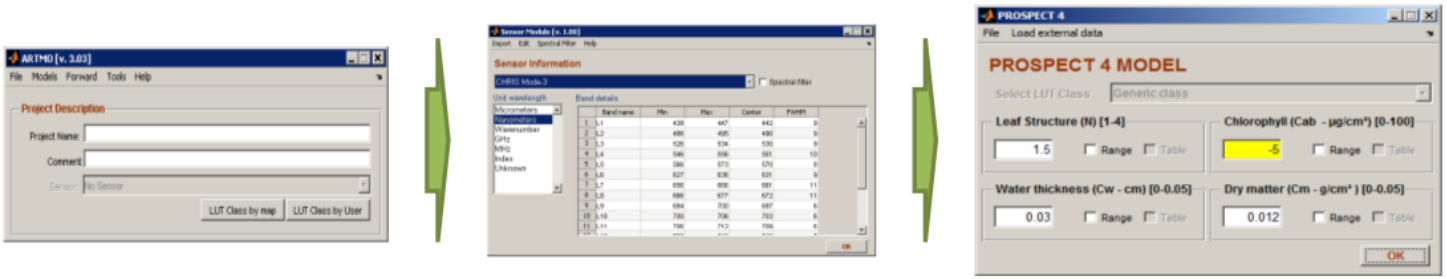
Automated Radiative Transfer Models Operator (ARTMO)

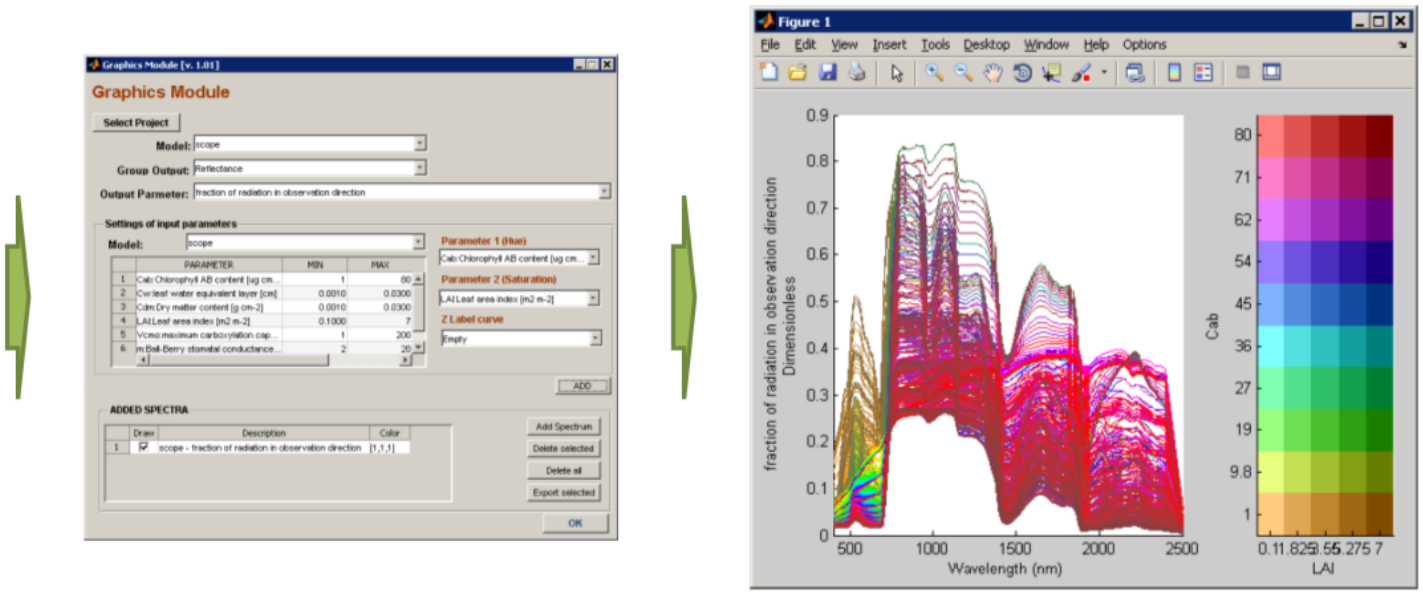
http://ipl.uv.es/artmo/





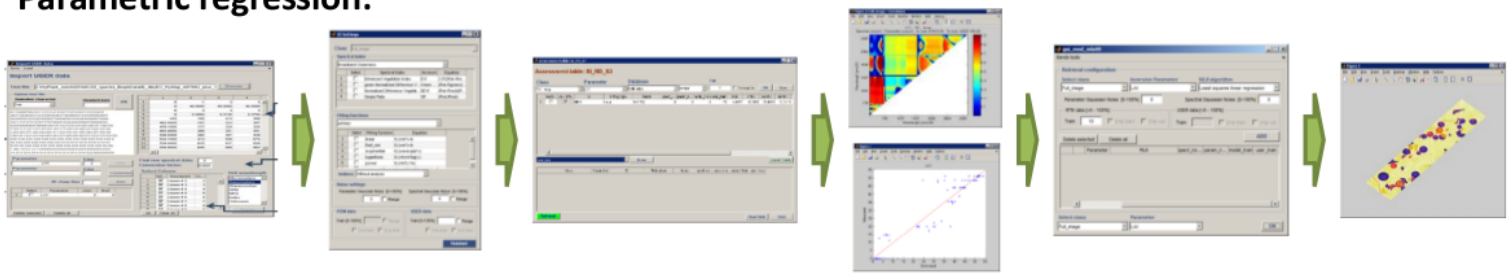
ARTMO can automatize the whole process ...



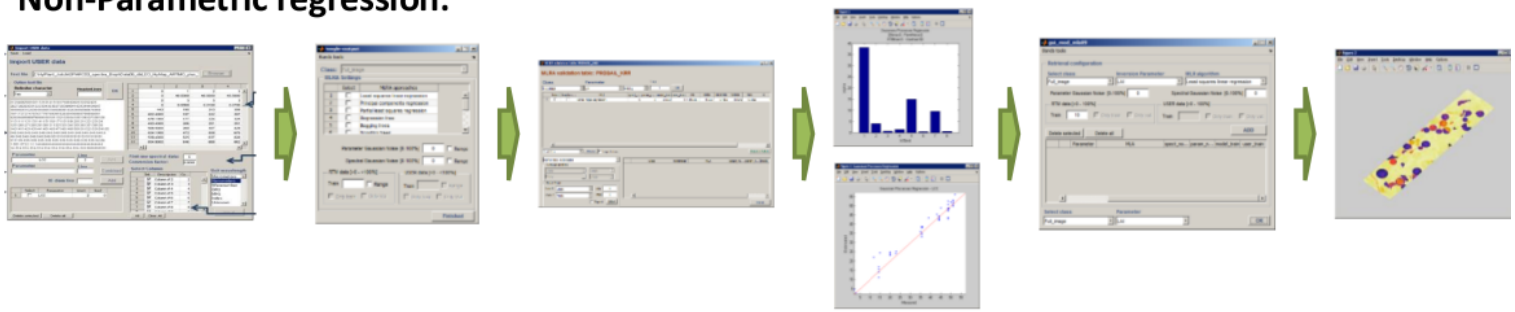


ARTMO can automatize the whole process ...

Parametric regression:



Non-Parametric regression:

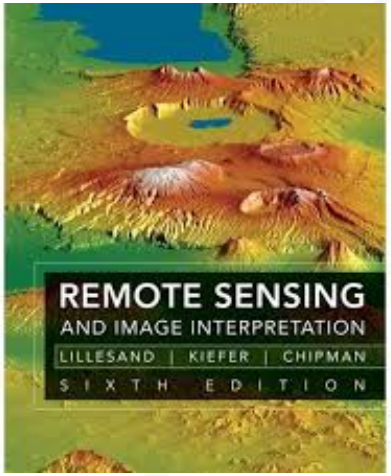


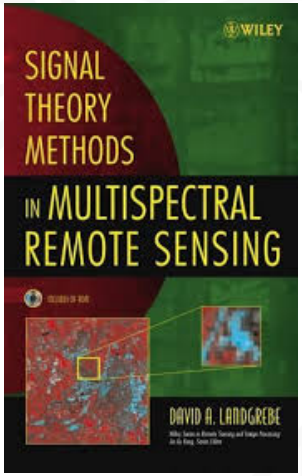
Physically-based inversión:

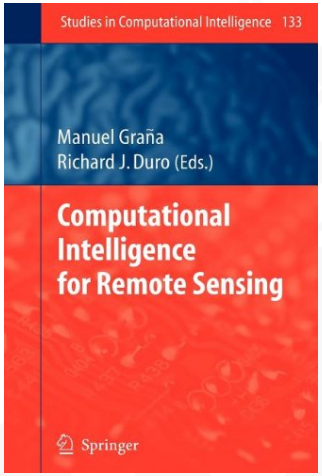
- Biophysical parameter estimation is perharps the most important (and challenging) problem in remote sensing
- Hyperspectral sensors provide an unprecedented piece of information for accurate estimation
- Traditional methods were focused on simplistic approaches using only few spectral bands
- New regression-based approaches alleviate the problems by exploiting the wealth of spectral information
- The common approaches consider:
  - Empirical models (e.g. Vegetation indices) are easy, fast but too general
  - Physical radiative transfer models are flexible but slow and require plant specific information (e.g. geometry, background) which is not always available
  - Non-parametric regression may offer a robust alternative that can be easily implemented in operative processing chains
- Problem: Scalability to many data, high dimensionality!
- Solution: Hybrid approaches + nonparametric sparse learning regression

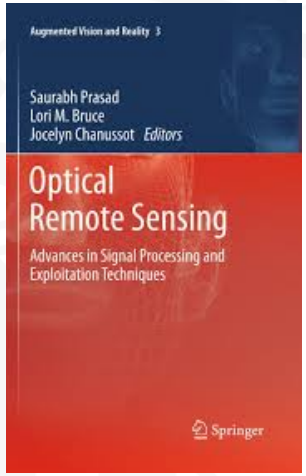
Part 9: Bibliography, source code and resources

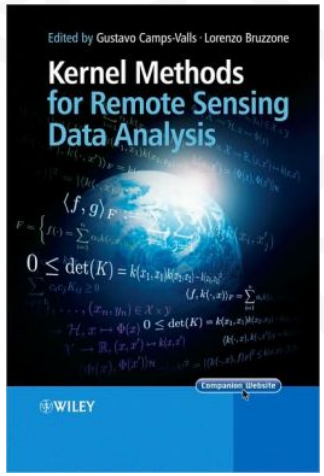
Some relevant books:

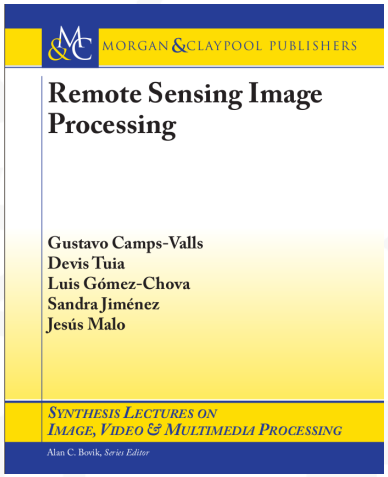


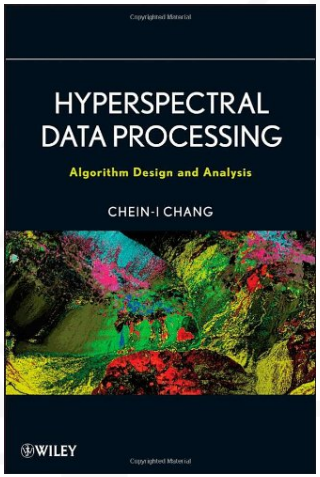


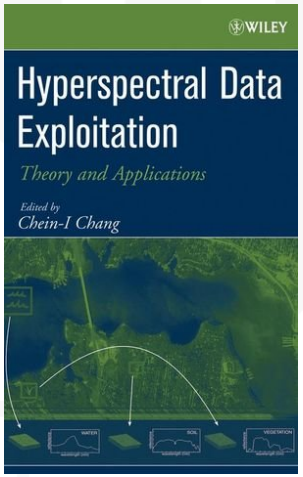


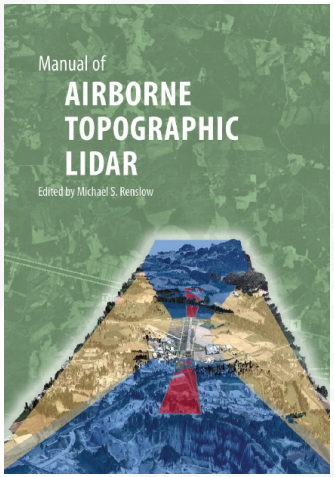


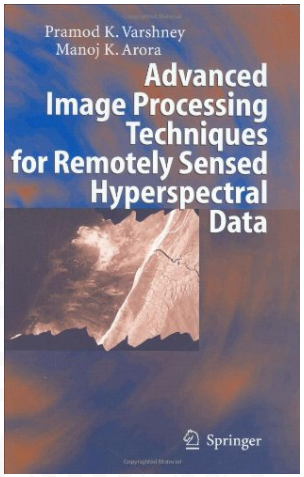


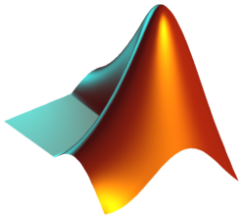












The ISP València Matlab Suite

http://isp.uv.es/soft.htm

HyperLabelMe Coming soon

- 50 *labeled* multi/hyper images
- An automatic system to evaluate classification accuracy

SimFEAT

- 10 state-of-the-art feature extraction methods
- Linear and kernel methods: (k)PCA, (k)MNF, (k)CCA, (k)PLS, (k)OPLS, (k)KECA

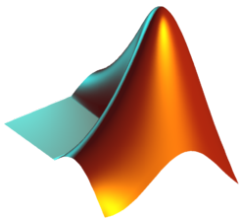
SimpleR

- 10 state-of-the-art nonparametric regression algorithms
- Trees, boosting, bagging, neural nets, kernel methods, Gaussian processes, etc.

SimpleClass

- 10 state-of-the-art supervised classifiers
- Trees, bagging, random forest, neural nets, SVMs, kernel machines, GPC, etc.

Simple to use, open source, re-useable, free!



Automated Radiative Transfer Models Operator (ARTMO)

http://ipl.uv.es/artmo/

ARTMO