INF250

Hyperspectral imaging 2

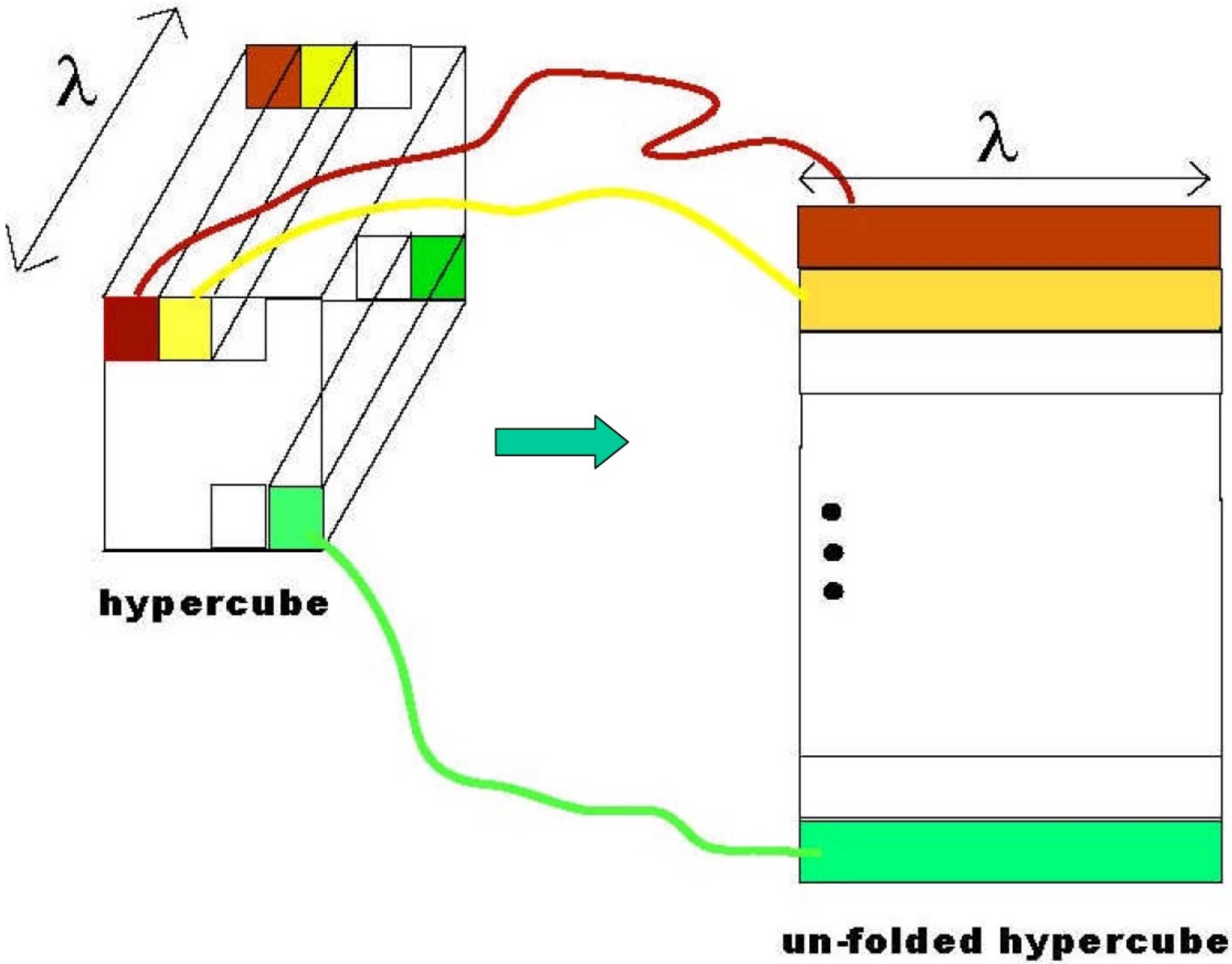
Learning goals

- What is multispectral and hyperspectral imaging ?
- What kind of cameras are used for hyperspectral imaging
- Give som examples of research one can do with hyperspectral imaging
- Describe how PCA works on a hyperspectral image
- Describe how k-means clustering works
- Describe supervised classification

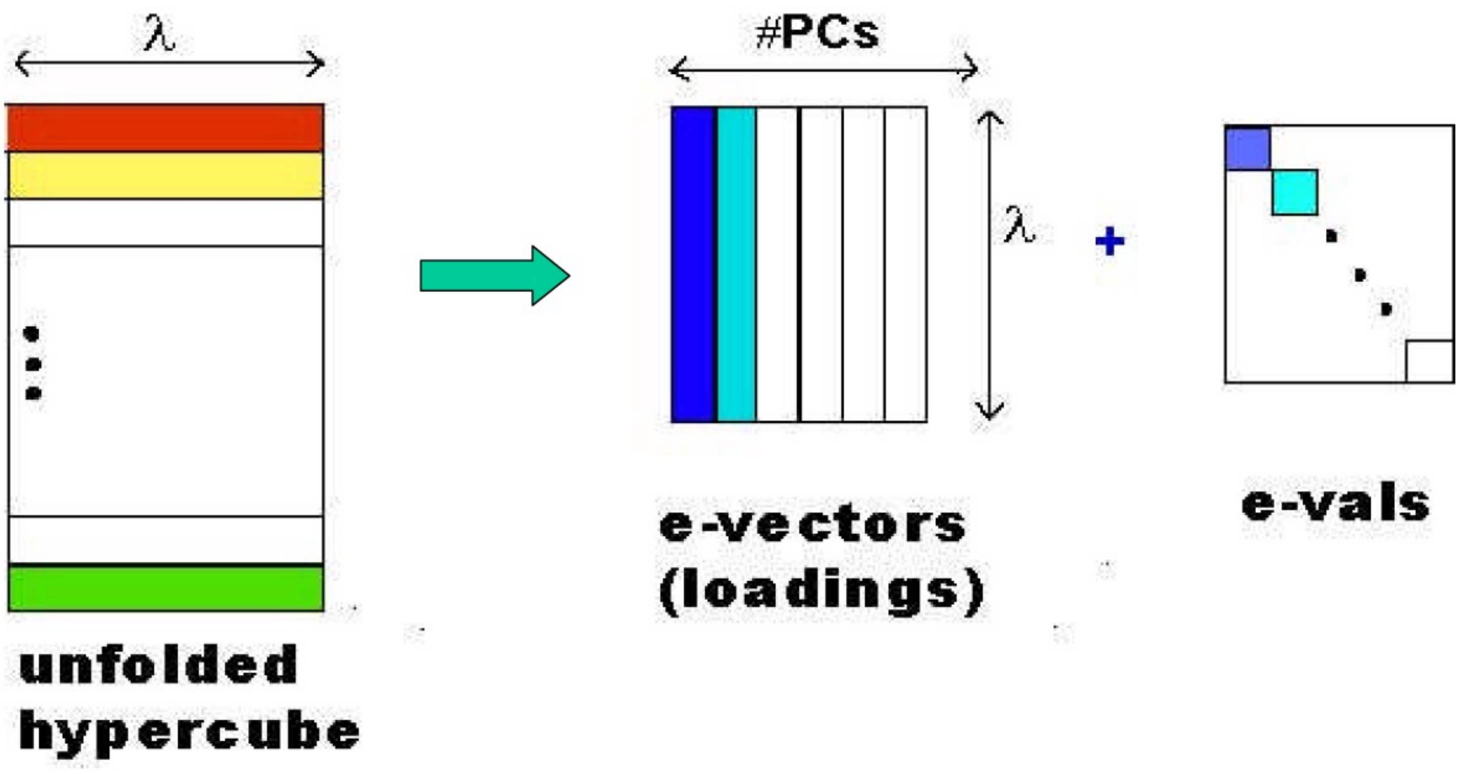
Multivariate image analysis on spectral images

- PCA (principal component analysis)
- K-means clustering
- Supervised classification

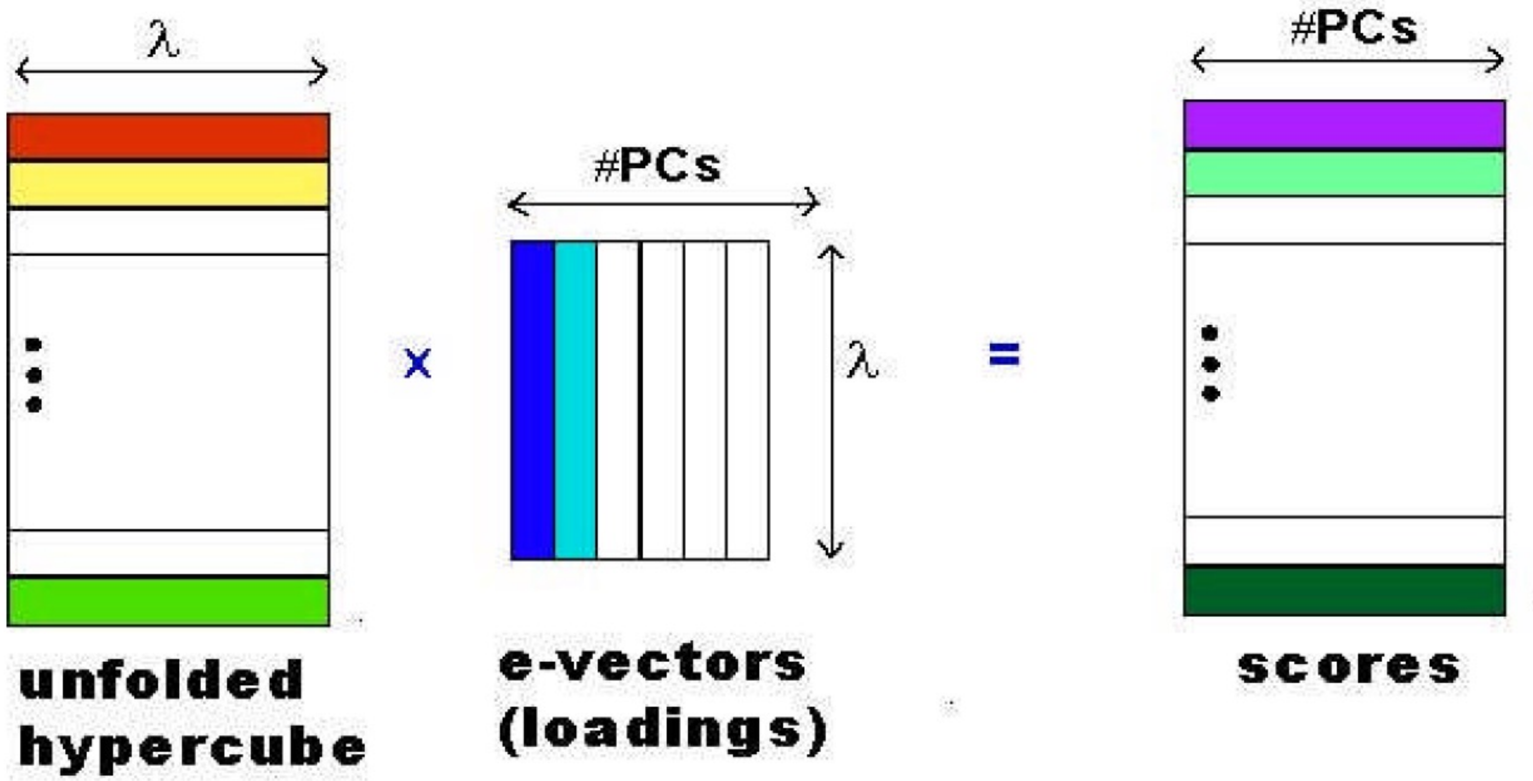
Unfolding the Hypercube



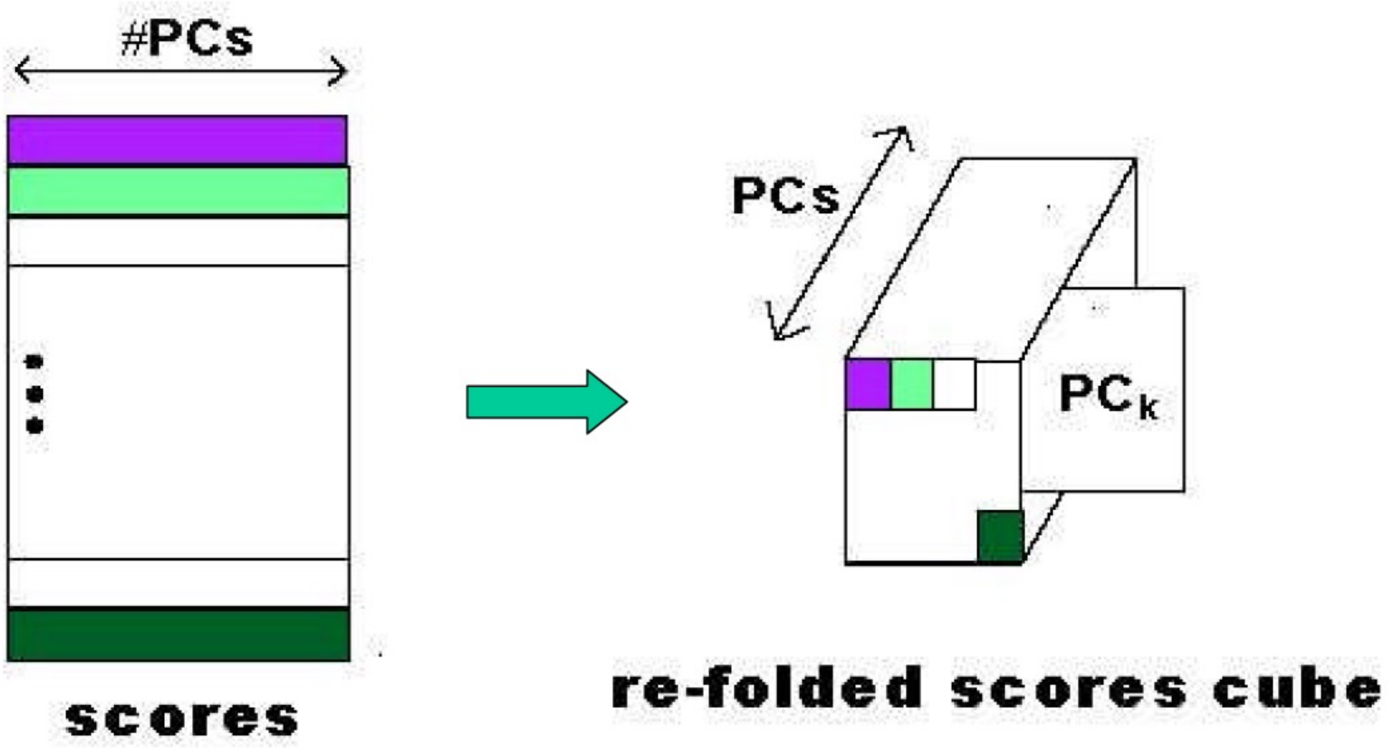
PCA on unfolded hypercube



Calculating Scores

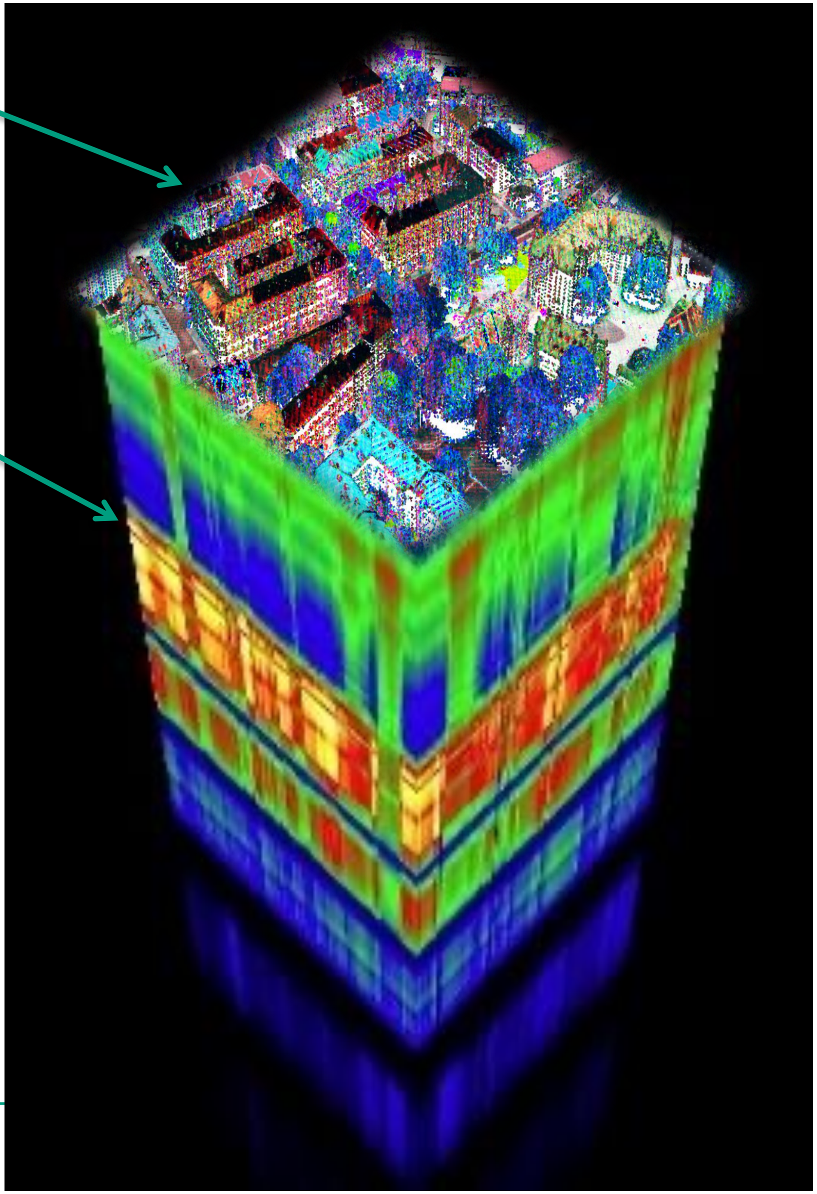


Re-folding Scores



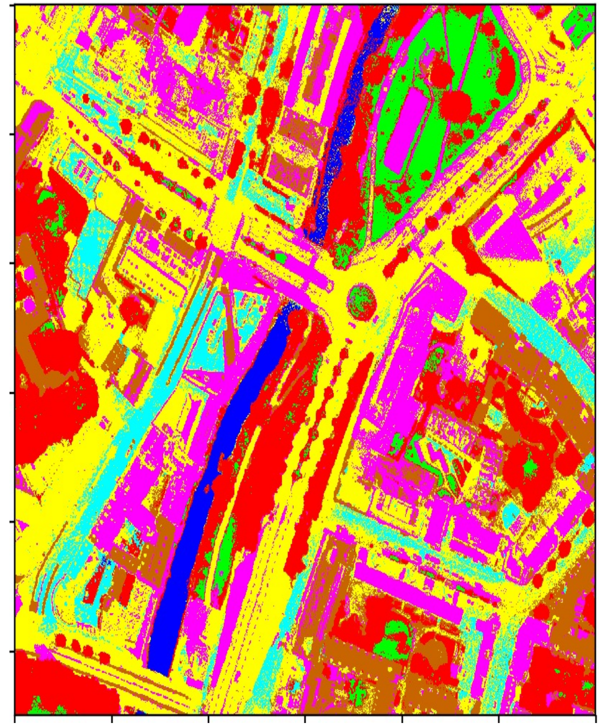
Laser data (LiDAR) yields a 3D model of the surface in the form of a point cloud

Hyperspectral data gives a detailed spectrum of channels in the visible and the near infrared and shortwave infrared, bringing out «invisible» differences in the surface material

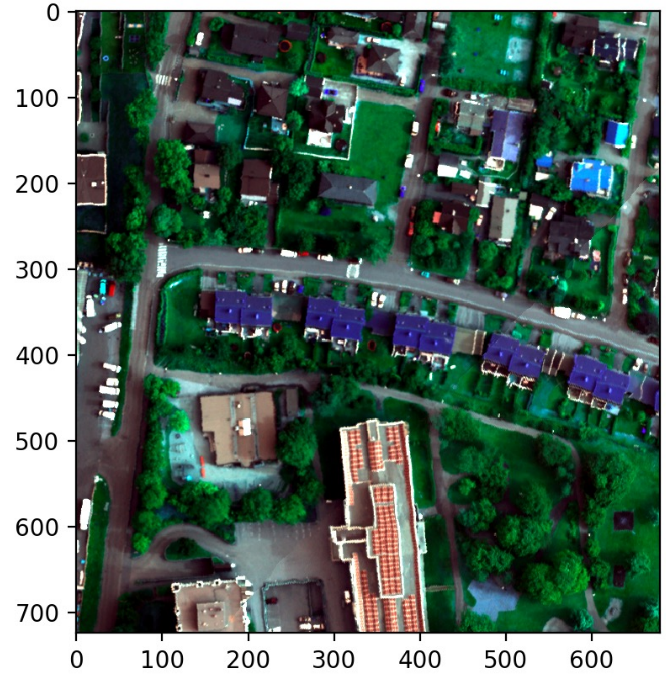


Collaboration with Terratec

Maximum Likelihood Classification



- Asphalt
- Grass
- Trees
- Dark rooftops + sand
- Red rooftops
- Shadowed areas (asphalt)




```python
from spectral import *
import numpy as np
import matplotlib.pyplot as plt
import skimage

hyperim = np.load("Oslo_hsi.npy")

imshow(hyperim, (10, 48, 70), stretch=((0.02,0.98),(0.02,0.98),(0.02,0.98)))
```

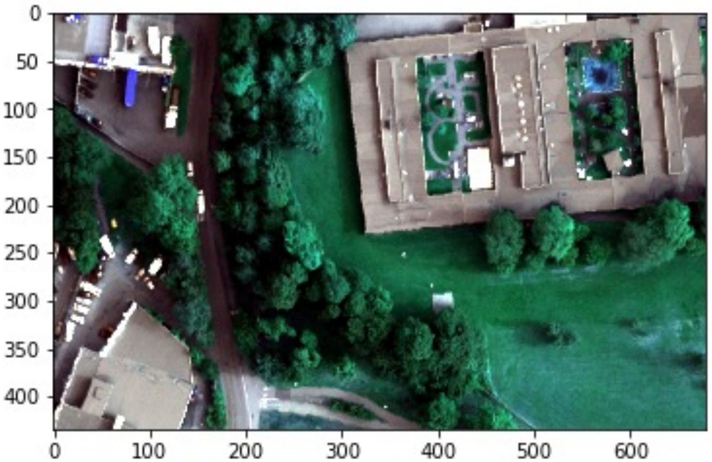
```
z = np.array(hyperim[521,520,:].reshape(-1,1))
z2 = np.array(hyperim[635,214,:].reshape(-1,1))
z3 = np.array(hyperim[441,36,:].reshape(-1,1))
plt.figure()
plt.plot(ww,z)
plt.plot(ww,z2)
plt.plot(ww,z3)
plt.show()

#plot as fct of pixel values
plt.figure()
plt.plot(z)
plt.show()

#compute mean of all spectra
m1 = hyperim.mean(axis=0)
m2 = m1.mean(axis=0).reshape(-1,1)
plt.figure()
plt.plot(ww,m2)
plt.show()

#ndvi

ndvi_ima = (hyperim[:,:,108]-hyperim[:,:,94])/(hyperim[:,:,108]+hyperim[:,:,94])
plt.imshow(ndvi_ima,vmin=0,vmax=0.7)

# ndvi from spectral python
vi = ndvi(hyperim,94, 108)
plt.figure()
plt.imshow(vi,vmin=-0.3,vmax=0.6)
```

```
# pca

pc = principal_components(hyperim)

pc_0999 = pc.reduce(fraction=0.999)

loading = pc_0999.eigenvectors

#loading of pc1
plt.figure()
plt.plot(loading[:,0])

#score of pc1
img_pc = pc_0999.transform(hyperim)
plt.figure()
plt.imshow(img_pc[:,:,0], vmin=-0.1, vmax=0.15)
```

```
# kmeans

(m, c) = kmeans(hyperim, 7, 20)
plt.imshow(m, 'spectral')

plt.figure()
for i in range(c.shape[0]):
    plt.plot(c[i])
```

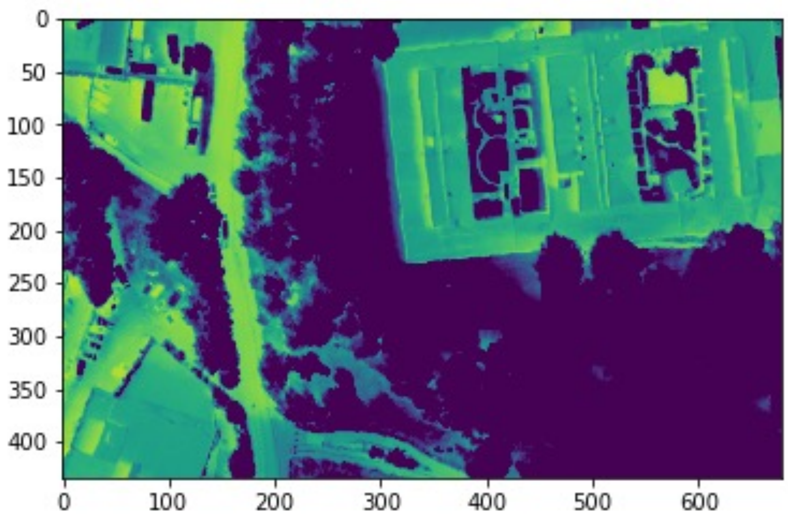
```python
shape = hyperim.shape
groundtruth = np.zeros([shape[0],shape[1]])

groundtruth[128:168,336:366] = 1.0   #grass
groundtruth[623:650, 200:250] = 2.0 # asphalt
groundtruth[460:480, 150:170] = 3.0 # roof1

groundtruth[350:370, 300:320] = 3.0 # roof1

plt.figure()
plt.imshow(groundtruth)
```

```python
# Gaussian Maximum Likelihood classification

classes = create_training_classes(hyperim, groundtruth)
gmlc = GaussianClassifier(classes)
clmap = gmlc.classify_image(hyperim)
imshow(classes=clmap)
```

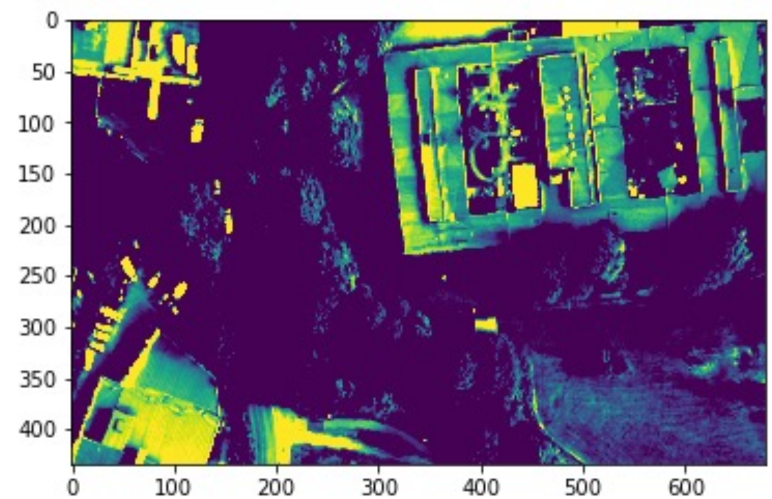



```
# pca

pc = principal_components(hyperim)

pc_0999 = pc.reduce(fraction=0.999)
len(pc_0999.eigenvalues)
img_pc = pc_0999.transform(img)
plt.imshow(img_pc[:,:,0],vmin=0.0, vmax=0.3)
plt.imshow(img_pc[:,:,1],vmin=0.0, vmax=0.1)
```

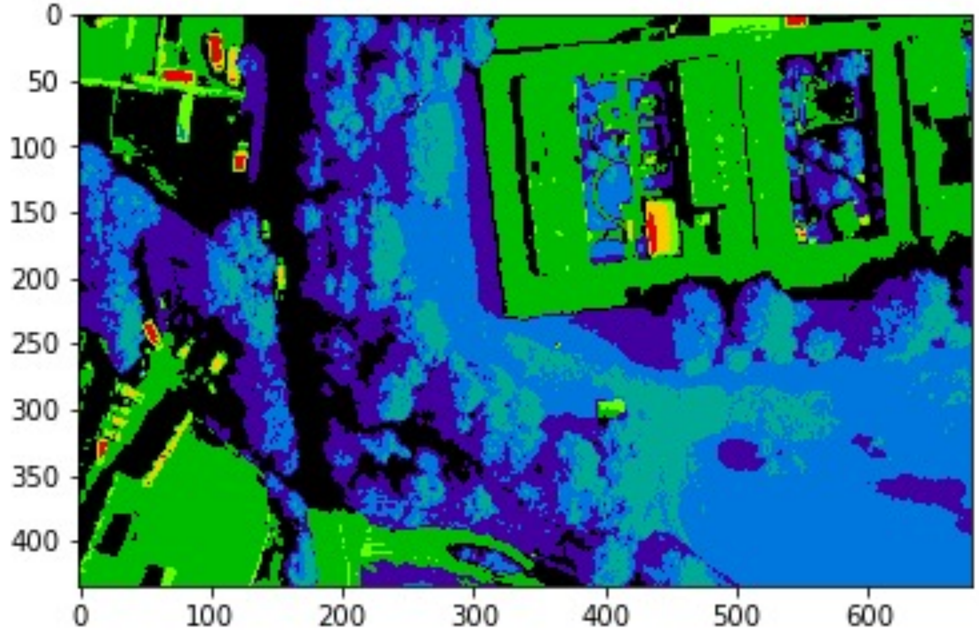
Hyperspectral image composite of band 20,40 and 60



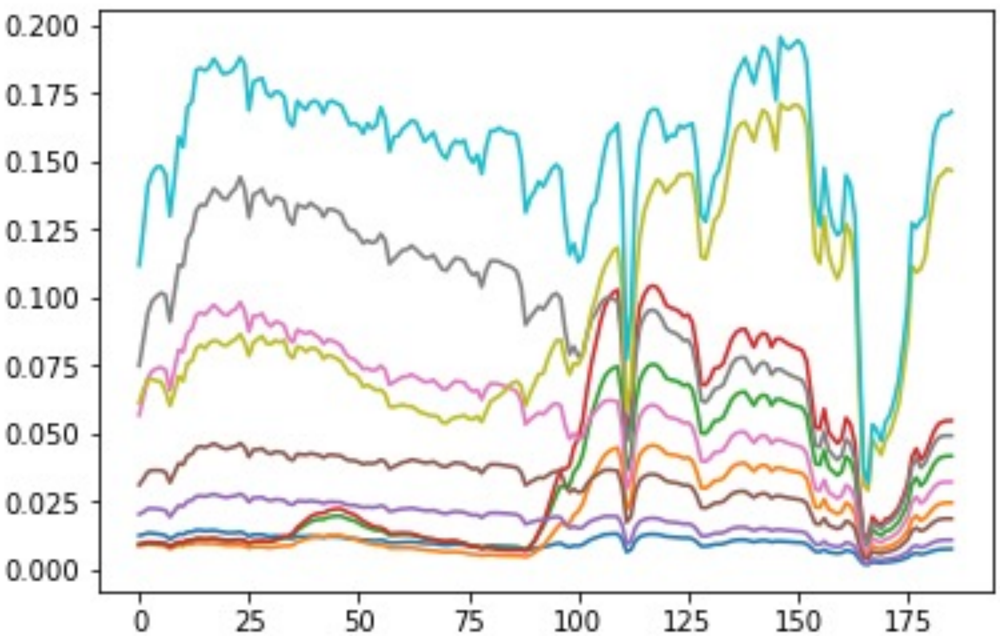
PC 1



PC 2




```python
# kmeans
(m, c) = kmeans(hyperim, 10, 30)
plt.figure()
plt.imshow(m,'spectral')
plt.show()


for i in range(c.shape[0]):
    plt.plot(c[i])
```



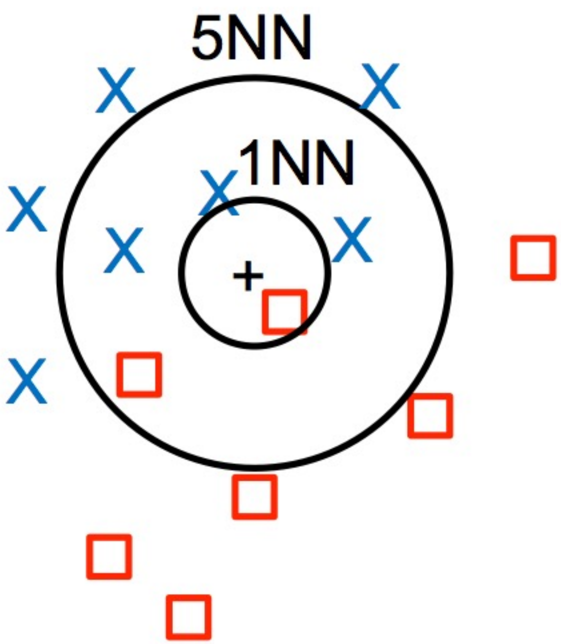
Classification

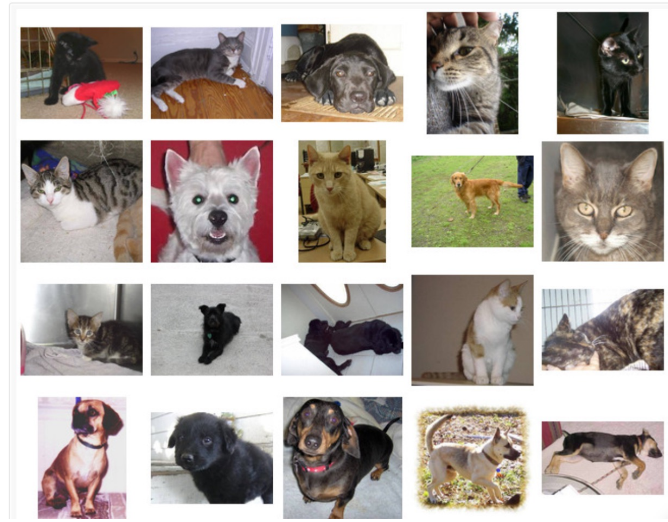
Assigning predefined labels to objects based on their features

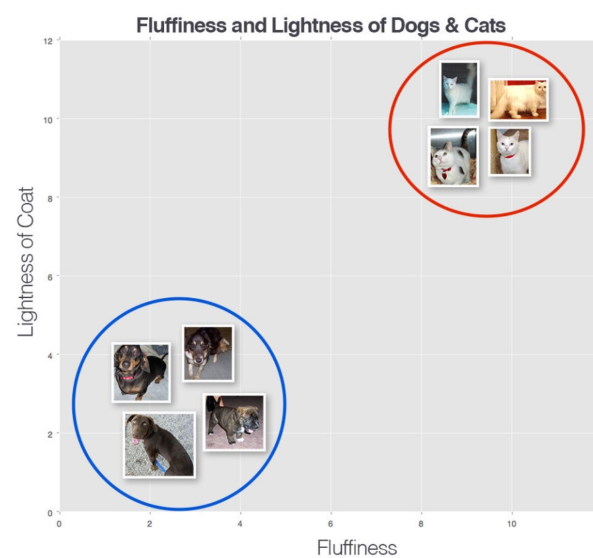
Clustering

Grouping a set of objects in such a way that objects in the same group are more similar in some sense than objects in other groups

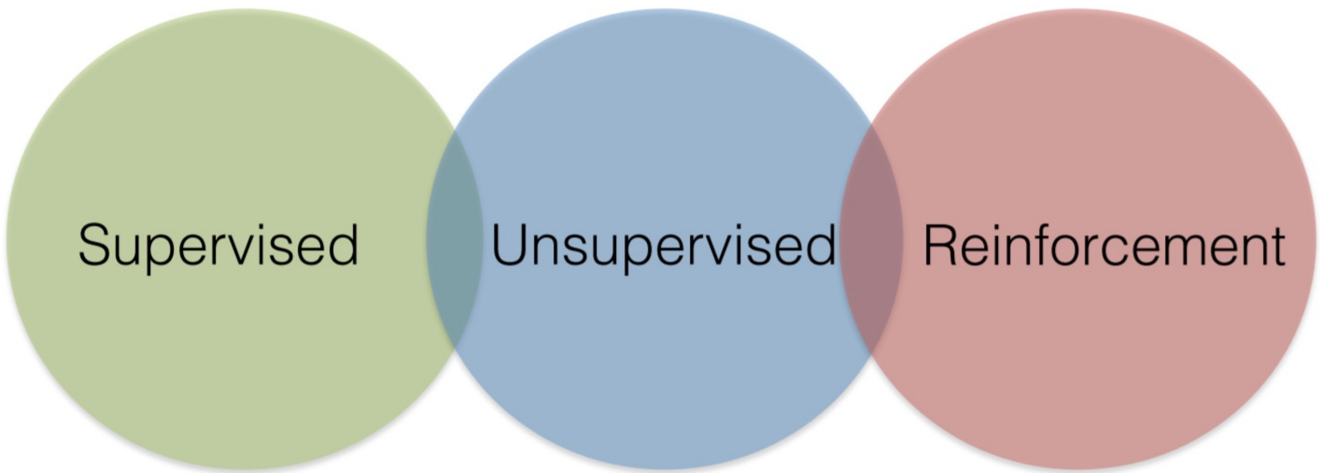
KNN – K nearest neighbours







Classification in images



Learning from
"labelled" data

Discover structure
in "unlabelled" data

Learning by "doing"
with delayed reward

Non supervised vs supervised classification

Unsupervised classification is where the outcomes (groupings of pixels with common characteristics) are based on the software analysis of an image without the user providing sample classes. The computer uses techniques to determine which pixels are related and groups them into classes. The user can specify which algorism the software will use and the desired number of output classes but otherwise does not aid in the classification process. However, the user must have knowledge of the area being classified when the groupings of pixels with common characteristics produced by the computer have to be related to actual features on the ground (such as wetlands, developed areas, coniferous forests, etc.).

Supervised classification is based on the idea that a user can select sample pixels in an image that are representative of specific classes and then direct the image processing software to use these training sites as references for the classification of all other pixels in the image. Training sites (also known as testing sets or input classes) are selected based on the knowledge of the user. The user also sets the bounds for how similar other pixels must be to group them together. These bounds are often set based on the spectral characteristics of the training area, plus or minus a certain increment (often based on "brightness" or strength of reflection in specific spectral bands). The user also designates the number of classes that the image is classified into. Many analysts use a combination of supervised and unsupervised classification processes to develop final output analysis and classified maps.

https://articles.extension.org/pages/40214/whats-the-difference-between-a-supervised-and-unsupervised-image-classification

Examples of supervised classification

- Gaussian Maximum Likelihood Classifier (GMLC)
- Spectral Angle Mapper (SAM)
- Discriminant Analysis

Gaussian Maximum Likelihood Classifier (GMLC)

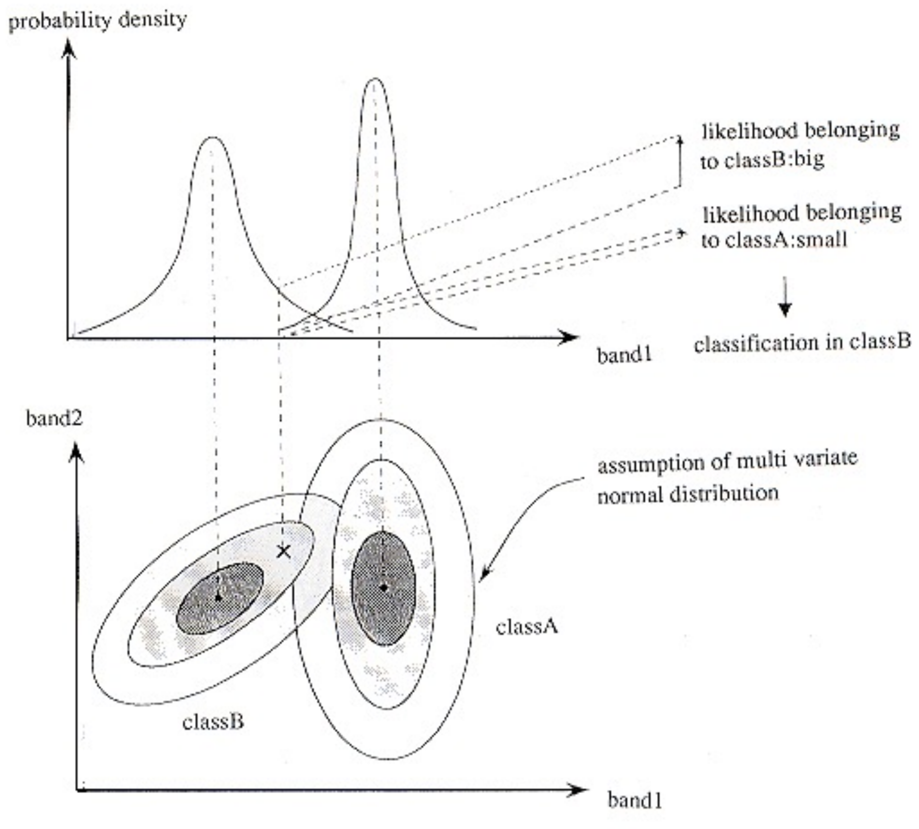
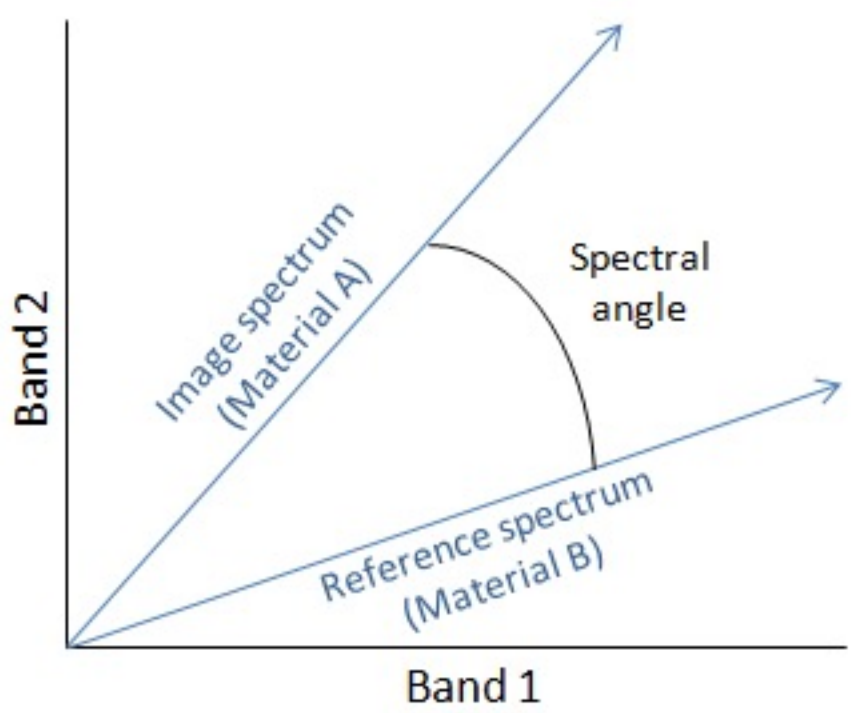


Figure 11.7.1 Concept of Maximum Likelihood Method

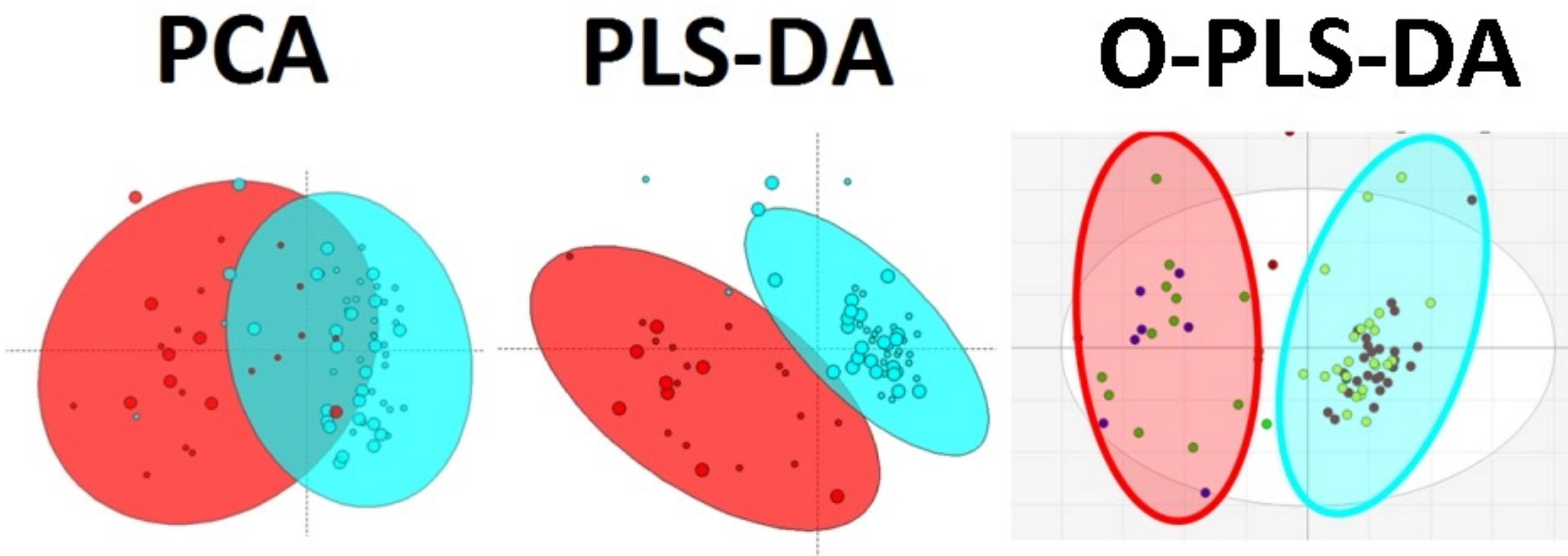
Spectral Angle Mapper



http://ijiset.com/v1s4/IJISET_V1_I4_27.pdf

PLS-DA

Rotation of PCA score plot so as to separate classes

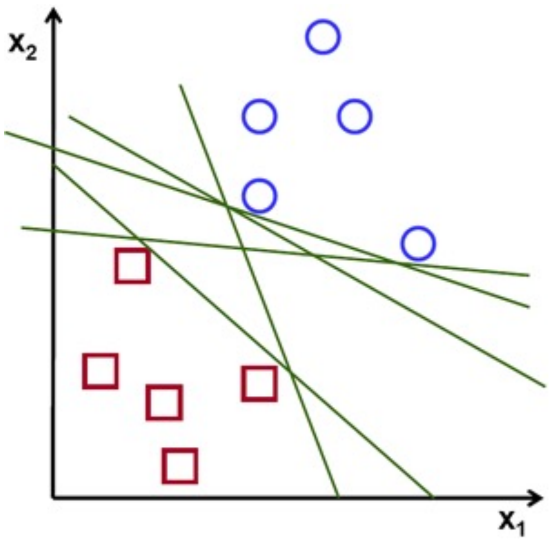


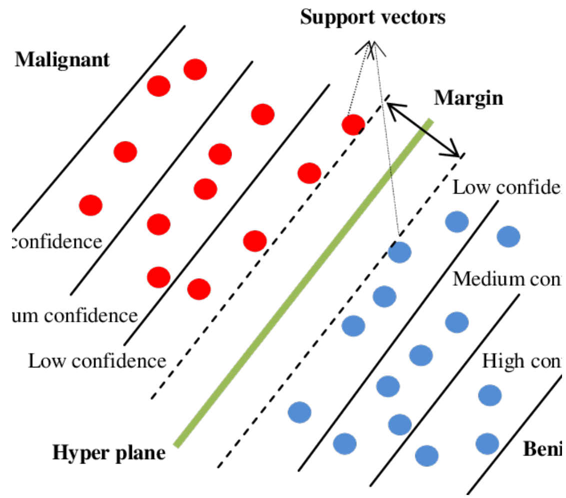
https://www.youtube.com/watch?v=HnmS51Q_LsE

Maximum Likelihood Classification

- Asphalt
- Grass
- Trees
- Dark rooftops + sand
- Red rooftops
- Shadowed areas (asphalt)

Support Vector Machine

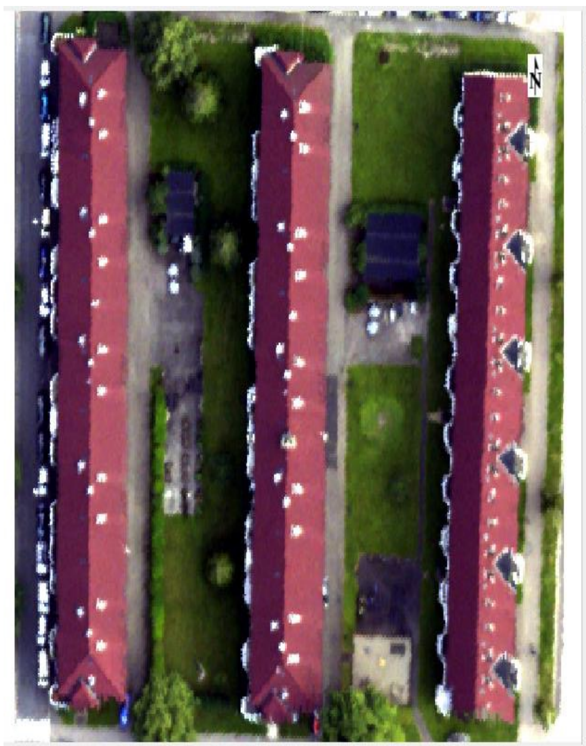


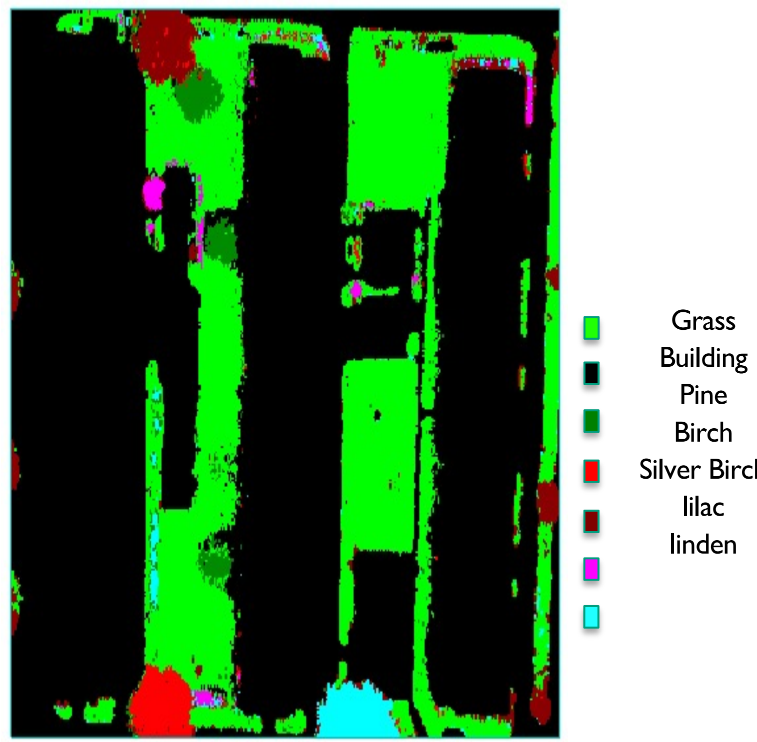


Classification of tree species

Different tree species have different leaf area, which is related to capacity of absorb polluting particles in the air

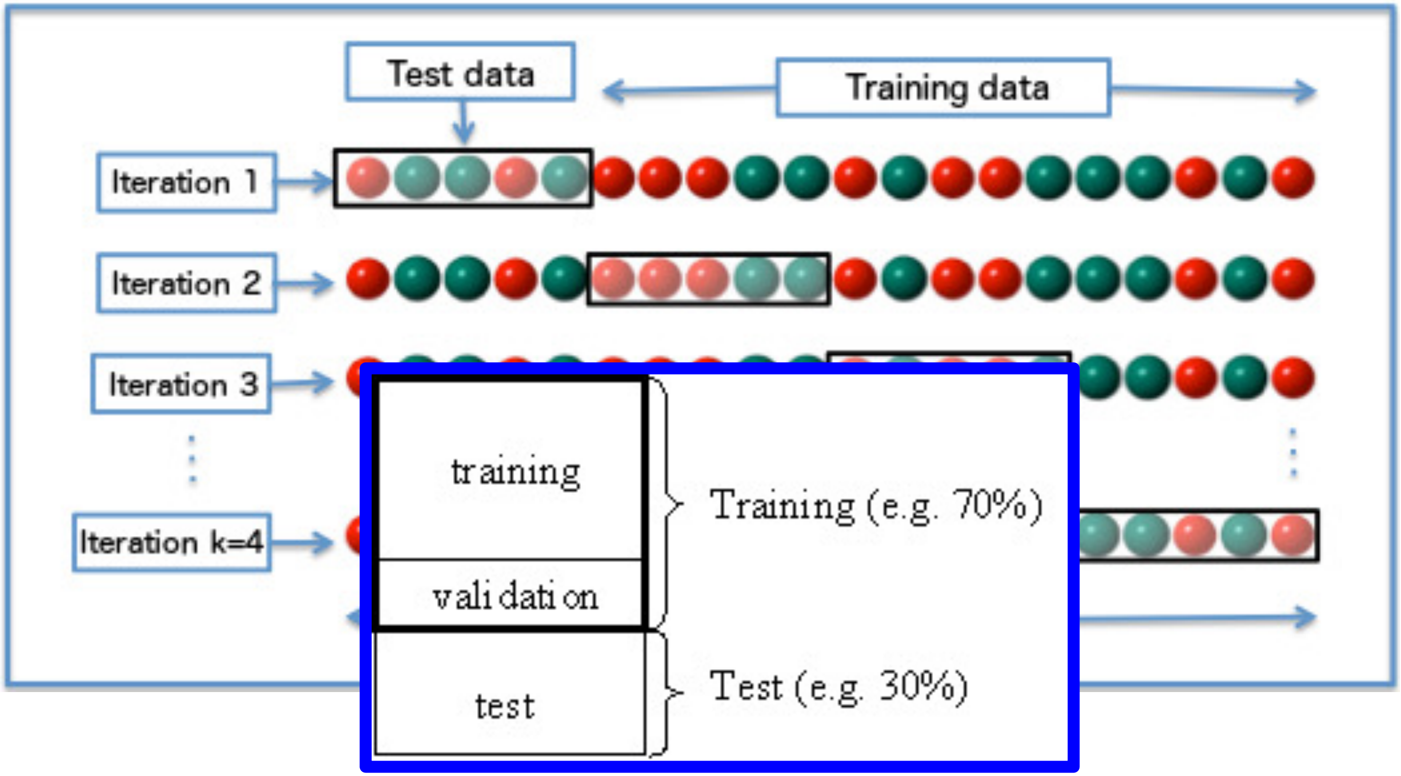
Support Vector machine classification

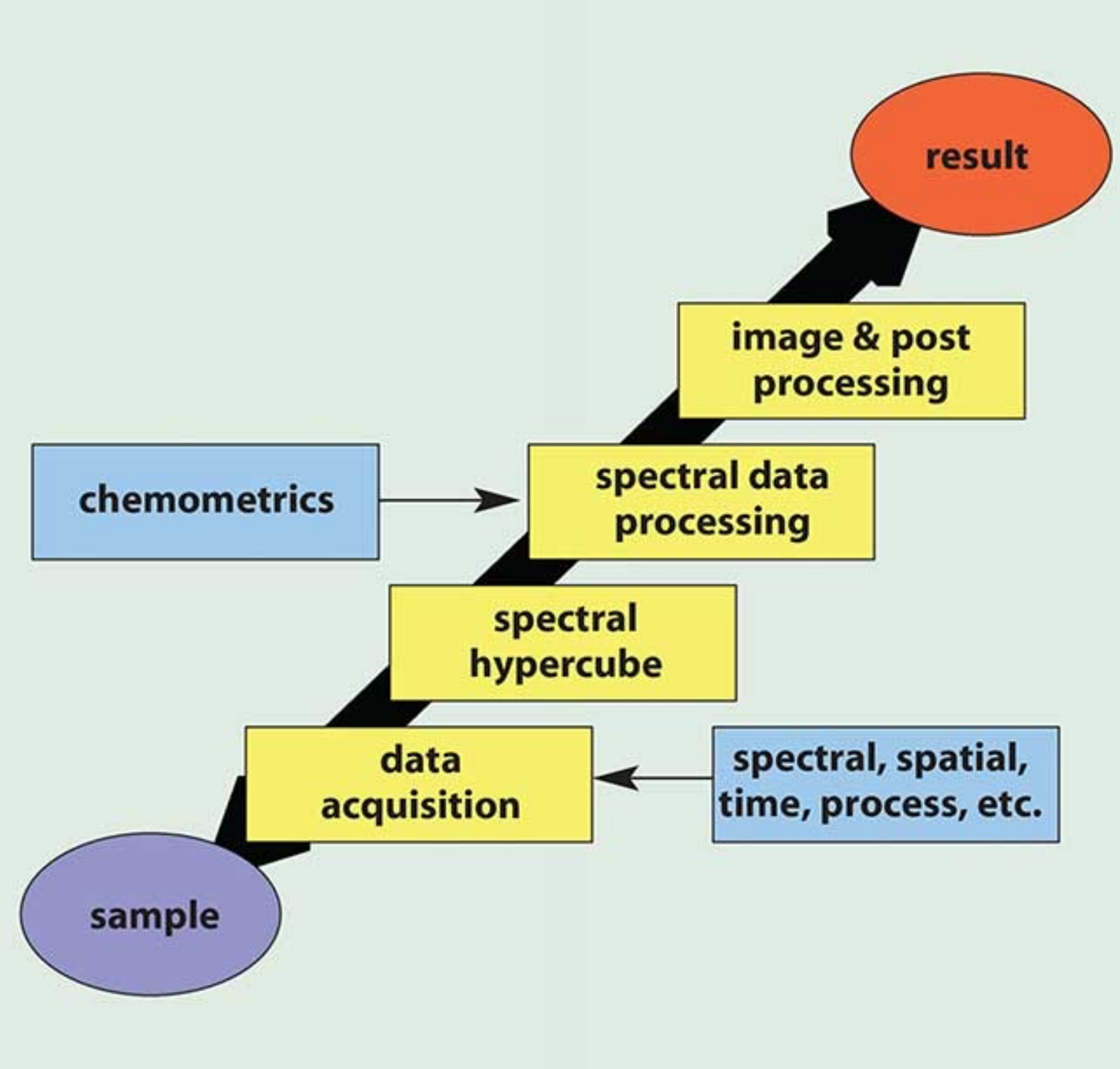




Erik Røstad, Master thesis 2018

Validation


Test data
Training data
Iteration 1
Iteration 2
Iteration 3
Iteration k=4
training
validation
test
Training (e.g. 70%)
Test (e.g. 30%)


result
image & post
processing
chemometrics
spectral data
processing
spectral
hypercube
data
acquisition
spectral, spatial,
time, process, etc.
sample