MVA - Discrete Inference and Learning
Lecture 9
-
Learning - Final remarks

Yuliya Tarabalka
Inria Sophia Antipolis-Méditerranée - TITANE team
Université Côte d'Azur - France

Overview

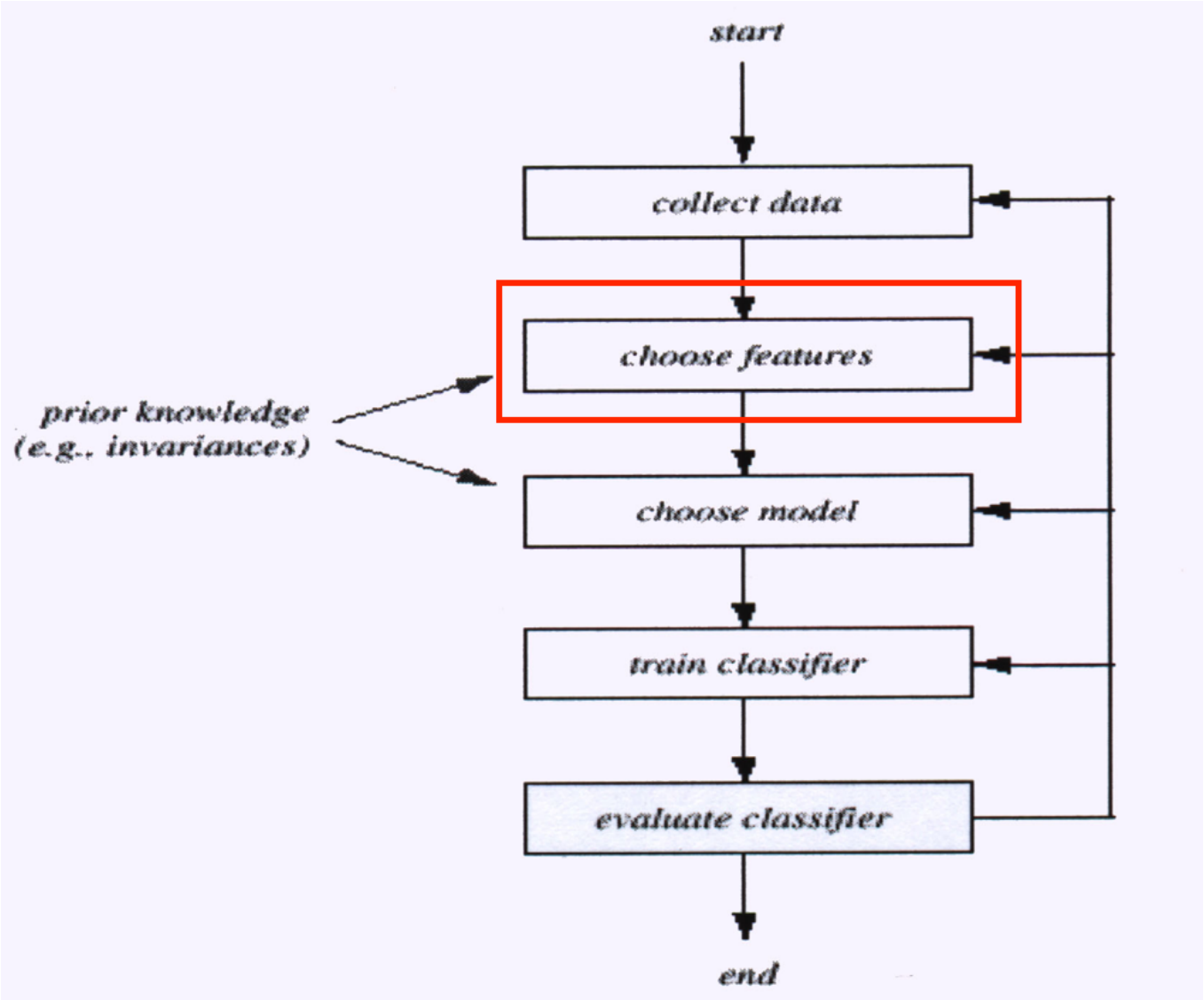
1. Feature extraction

2. Performance evaluation

Overview

1. Feature extraction
2. Performance evaluation

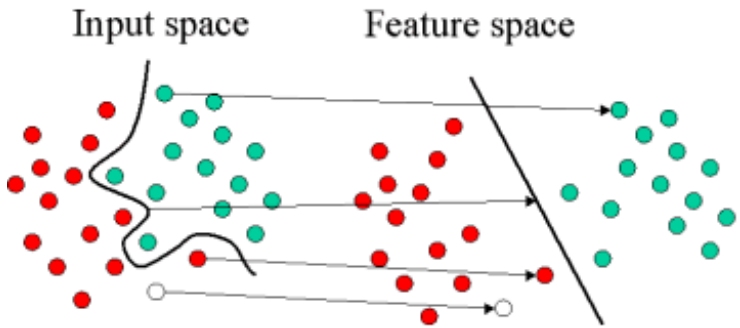
Why do we need feature extraction?



Motivation

- **Main motivation:** get out most of the data

- For classification task: find a space where samples from different classes are well separable



Objectives:

- Reduce computational load of the classifier
- Increase data consistency
- Incorporate different sources of information into a feature vector: spectral, spatial, multisource, ...

Feature extraction to reduce dimensionality

- PCA seeks directions that are efficient for **representation**
  - *Unsupervised technique*
- Discriminant analysis seeks directions that are efficient for **discrimination**
  - *Supervised technique*
- Many other techniques
  - Independent component analysis, manifold learning algorithms, autoencoders, ...

Extract features from text data

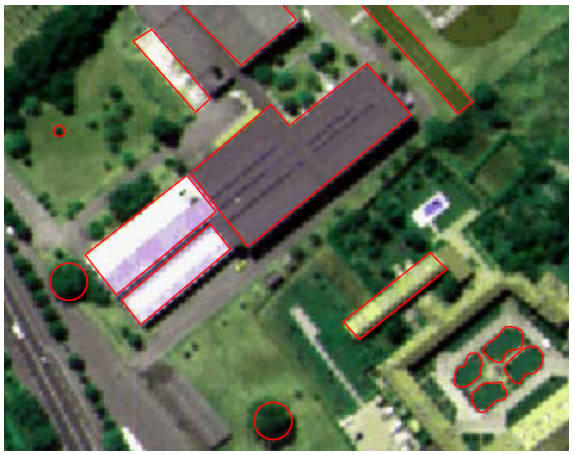
1. **Trimming Vocabulary:** remove "non-content" words (very frequent "stop words" such as "the", "and", ... or very rare words e.g. , that just occur 10 times in 100000 words)

2. **Stemming:** Reduce all variants of a word to a single term (e.g., see, saw, seen $\rightarrow$ "see")

3. **Define Classes:** Define how many classes do you have?
   - Eg. Spam/not spam, News categorization: Sport, Science, Politics
   - Categorize and assign number to the words
   - Count the frequency (number of occurrence) of each word

Features for image data: extract spatial info

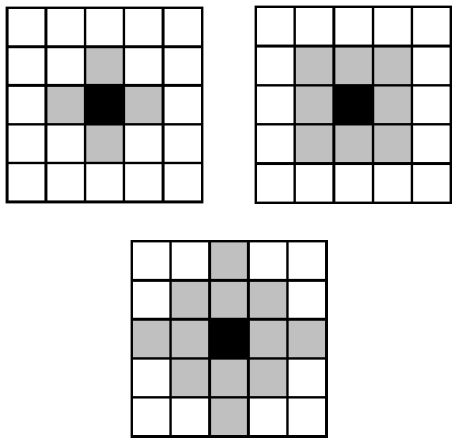
- By simply looking at a grey pixel, we cannot say if it belongs to a *building* or a *road*
- We guess a category by considering spatial/contextual information
- How can a classifier consider this rich source of information?



Approaches to extract spatial info

1. Closest fixed neighborhoods
   - Markov Random Field [Pony00, Jackson02, Farag05]
   - Contextual features [Camps-Valls06]
     - Spectral content +
     - Spatial content (e.g. mean or standard deviation per spectral band)

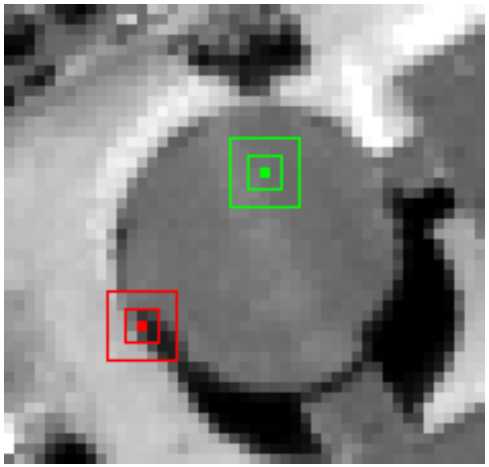
   + Simplicity
   − Imprecision at the border of regions



Approaches to extract spatial info

1. Closest fixed neighborhoods
   - Markov Random Field [Pony00, Jackson02, Farag05]
   - Contextual features [Camps-Valls06]
     - Spectral content +
     - Spatial content (e.g. mean or standard deviation per spectral band)

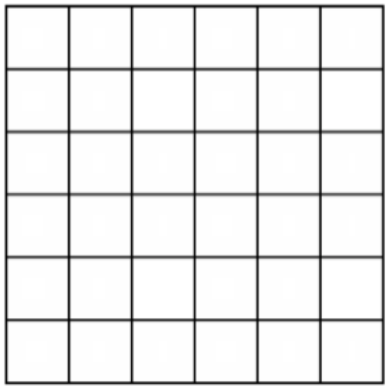
   \+ Simplicity
   − Imprecision at the border of regions

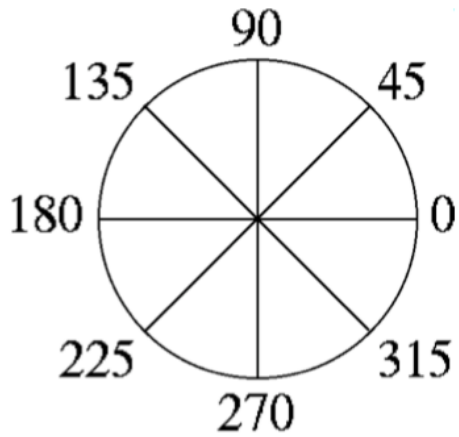


Histograms of oriented gradients

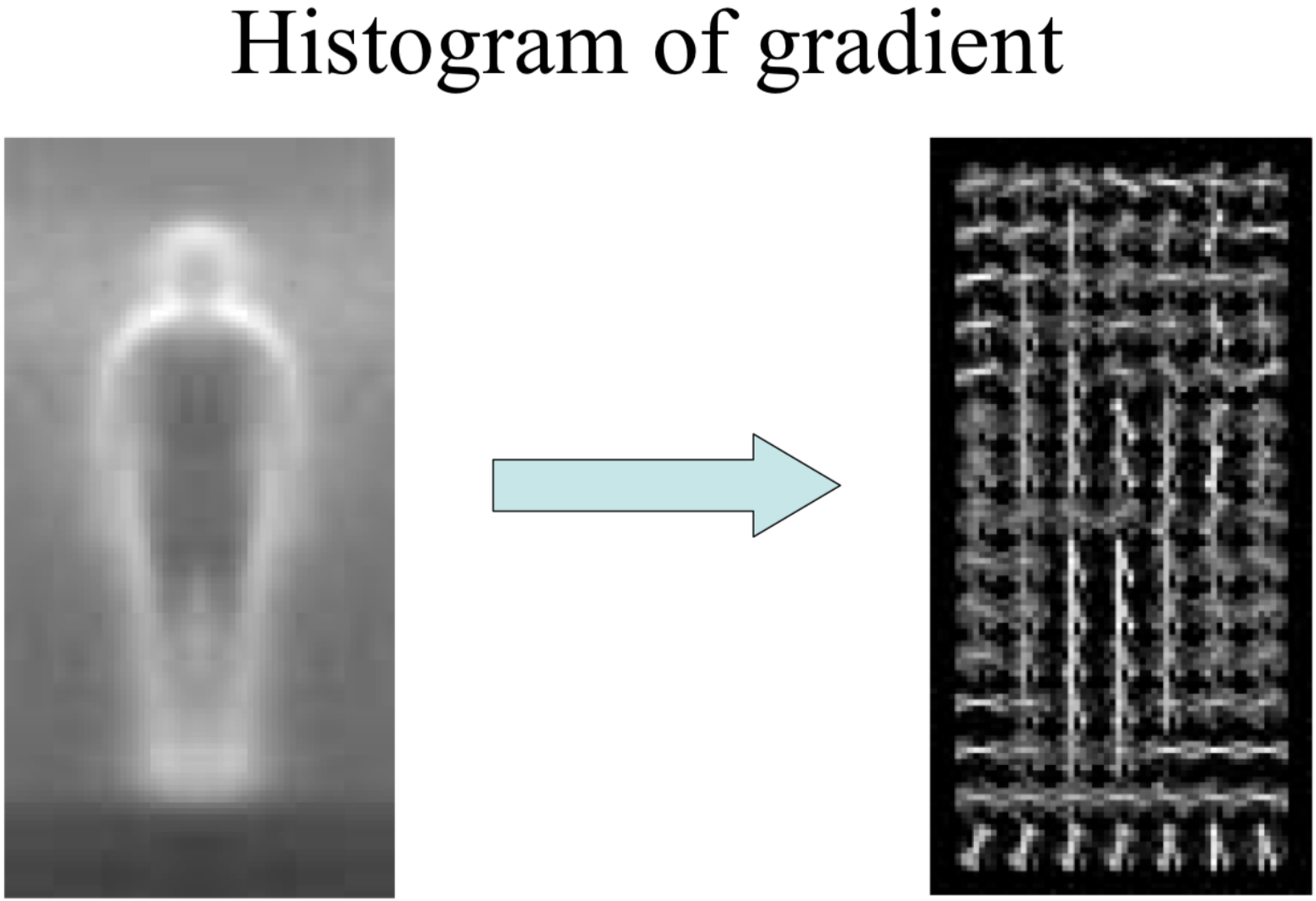
Histogram of Gradient

- Dividing the image window into small spatial regions (*cells*)
- Cells can be either rectangle or radial.
- Each cell accumulating a weighted local 1-D histogram of gradient directions over the pixels of the cell.





Histograms of oriented gradients

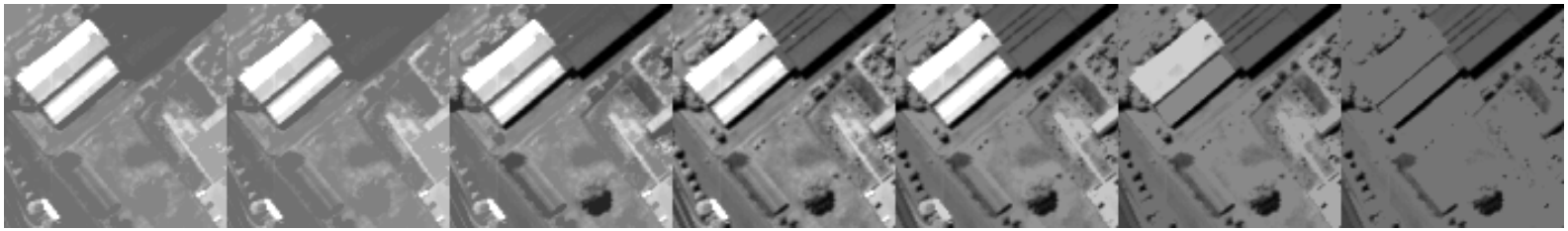


Approaches to extract spatial info

2. Morphological and area filtering

   - Morphological profiles [Pesaresi01, Dell'Acqua04, Benediktsson05]
   - Self-complementary area filtering [Fauvel07]
   - Attribute profiles [Ghamisi15, Cavallaro17]

   \+ Neighborhoods are adapted to the structures
   \+ Non-linear operators $\Rightarrow$ do not blur the edges as convolutions do
   – Neighborhoods are scale dependent



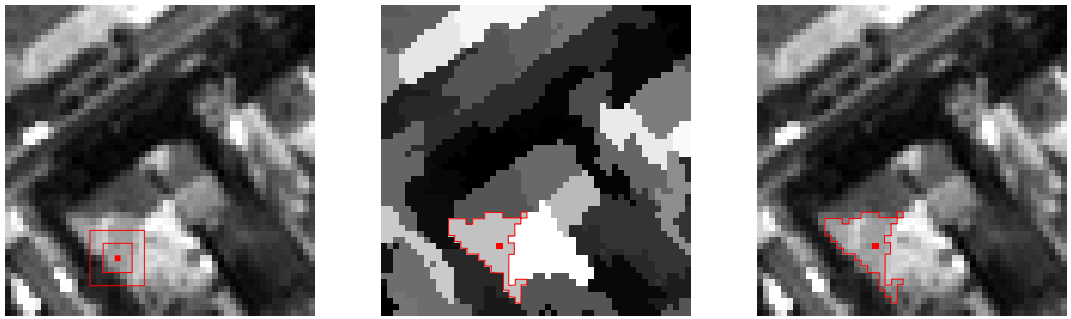
Closing – Original – Opening

Course on mathematical morphology:
`http://www-sop.inria.fr/members/Yuliya.Tarabalka/teaching.htm`

Approaches to extract spatial info

2. Morphological and area filtering
   - Morphological profiles [Pesaresi01, Dell'Acqua04, Benediktsson05]
   - Self-complementary area filtering [Fauvel07]
   - Attribute profiles [Ghamisi15, Cavallaro17]

+ Neighborhoods are adapted to the structures
+ Non-linear operators $\Rightarrow$ do not blur the edges as convolutions do
– Neighborhoods are scale dependent



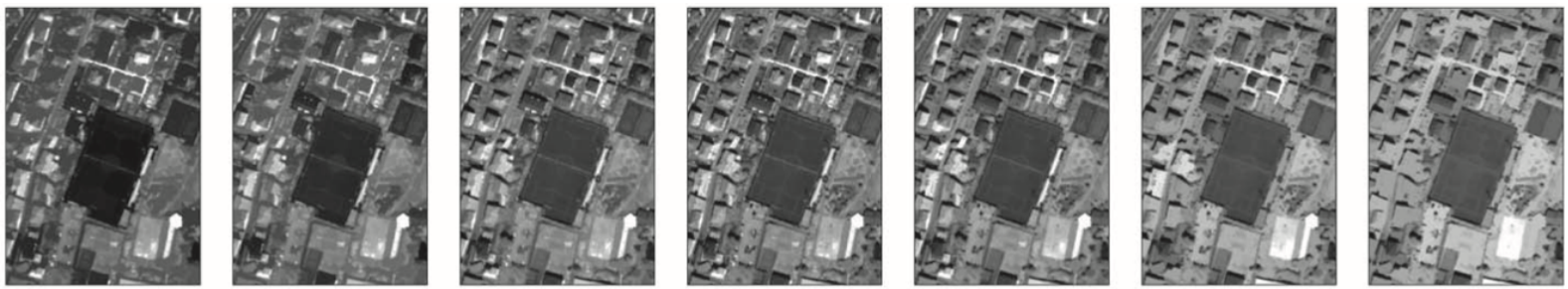
Course on mathematical morphology:
`http://www-sop.inria.fr/members/Yuliya.Tarabalka/teaching.htm`

Approaches to extract spatial info

2. Morphological and area filtering

   - Morphological profiles [Pesaresi01, Dell'Acqua04, Benediktsson05]
   - Self-complementary area filtering [Fauvel07]
   - Attribute profiles [Ghamisi15, Cavallaro17]

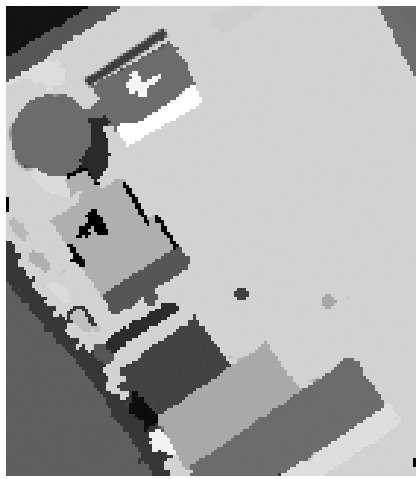
   \+ Neighborhoods are adapted to the structures
   \+ Non-linear operators $\Rightarrow$ do not blur the edges as convolutions do
   – Neighborhoods are scale dependent



Course on mathematical morphology:
`http://www-sop.inria.fr/members/Yuliya.Tarabalka/teaching.htm`

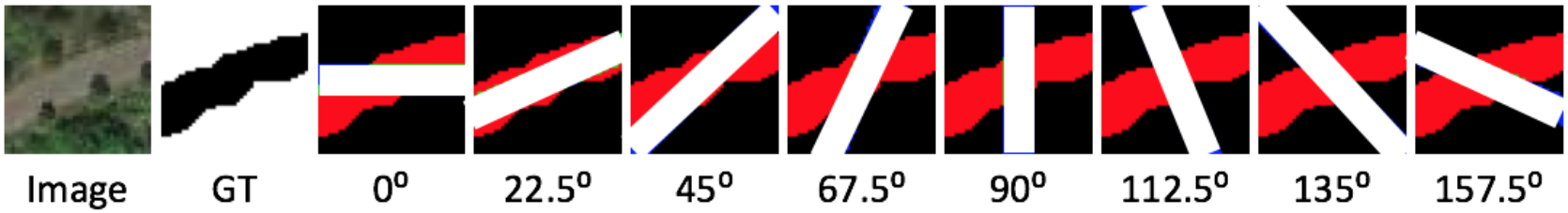
Approaches to extract spatial info

3. Superpixels derived from segmentation
   - Extraction and Classification of Homogeneous Objects [Kettig76]
   - ...
   - Multiscale segmentation, then features are derived from the regions [Linden07, Huang09]
     + Flexible
     – Computationally demanding
     – Difficult to scale/parallelize



Approaches to extract spatial info

4. Features handcrafted for a particular application
   - Example 1: Line templates with different orientations for road detection [Jeong15]
   - Example 2: Rectangular templates for building detection [Garcin01]
     + Can model complex shape
     – Lack of genericity
     – Computationally demanding



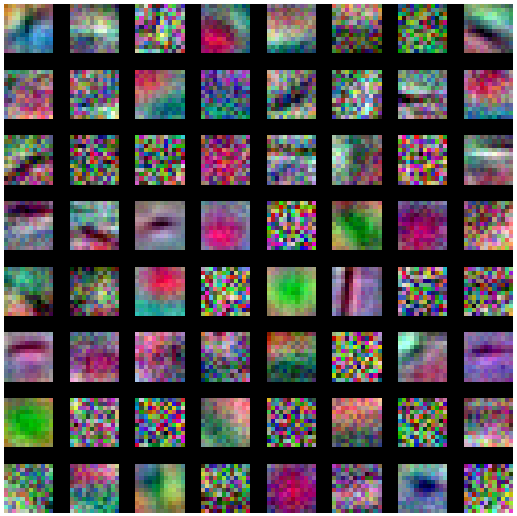
Approaches to extract spatial info

4. Features handcrafted for a particular application
   - Example 1: Line templates with different orientations for road detection [Jeong15]
   - Example 2: Rectangular templates for building detection [Garcin01]
     + Can model complex shape
     – Lack of genericity
     – Computationally demanding

Modern trend & Conclusions

Deep learning:

- Automatically learn features if a lot of training data are available



Advice:

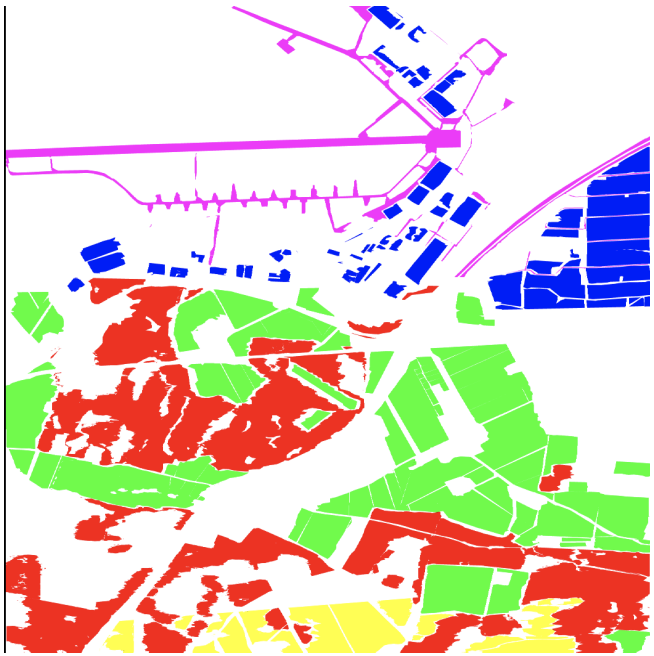
- If for the considered application it is easy to hand-craft class-separable features, no need to learn them
- If it is not easy to discriminate between categories, learning features often helps

Overview

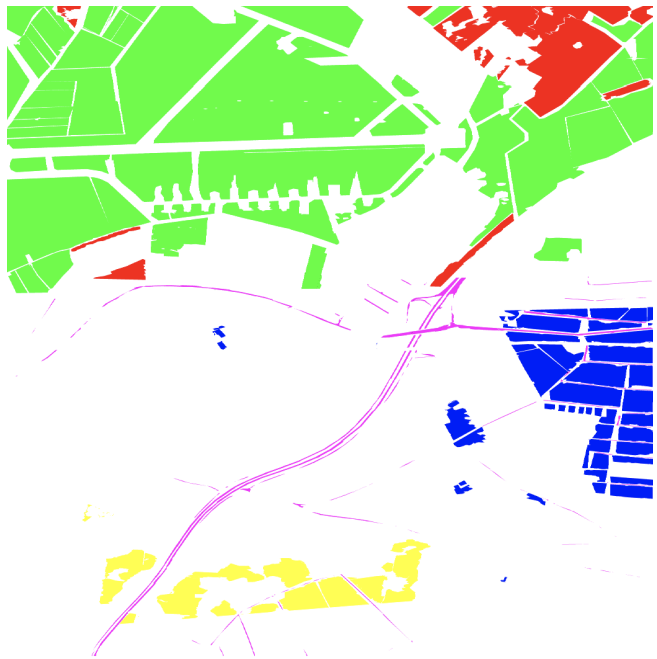
1. Feature extraction
2. Performance evaluation

How to evaluate performance of a classifier?

- **Common strategy:**
  1. Split all available data into training and test sets
  2. Train a classifier using the training data
  3. Compare the obtained results with the test data



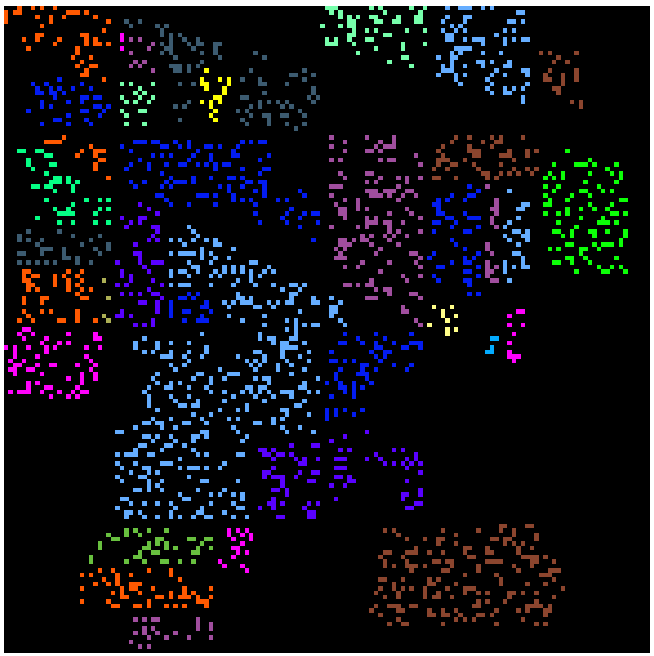
Training set



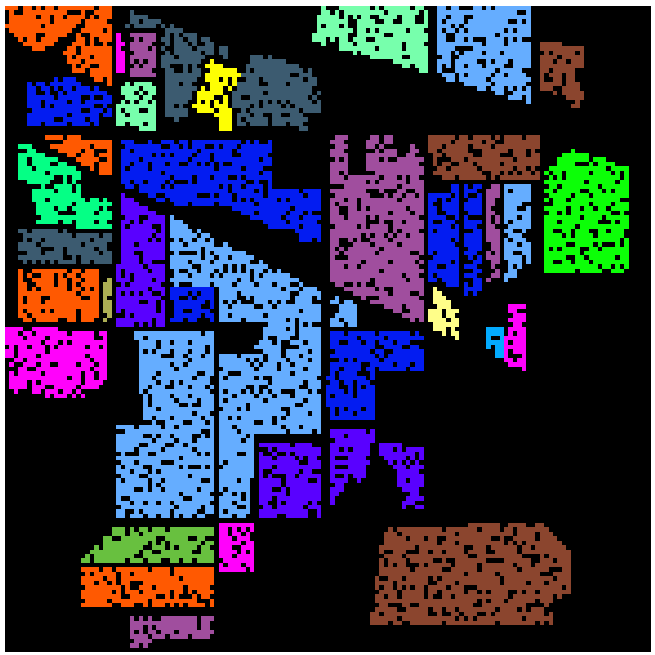
Test set

How to evaluate performance of a classifier?

- **Common strategy:**
  1. Split all available data into training and test sets
  2. Train a classifier using the training data
  3. Compare the obtained results with the test data



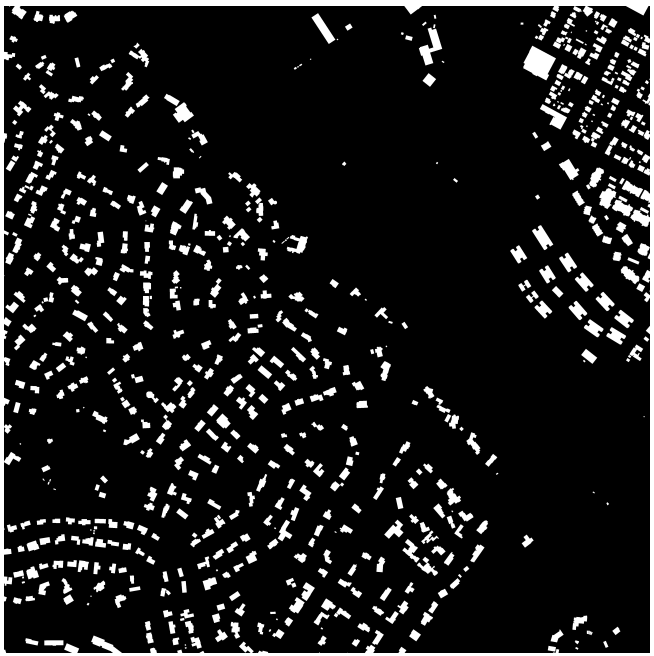
Training set



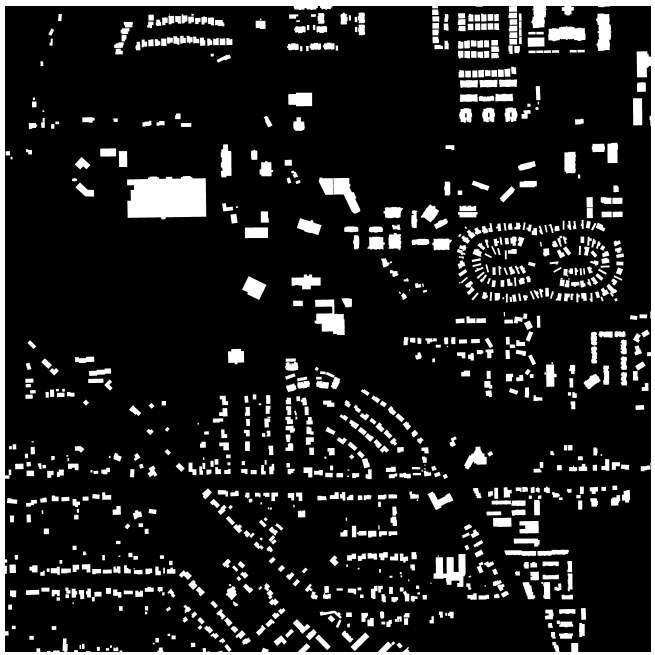
Test set

How to evaluate performance of a classifier?

- **Common strategy:**
  1. Split all available data into training and test sets
  2. Train a classifier using the training data
  3. Compare the obtained results with the test data



Training set



Test set

How to compare the obtained results with the test data?

1. **Visual/huiman comparison** of classification results

How to compare the obtained results with the test data?

1. **Visual/huiman comparison** of classification results
   - Very important, but is possible for a limited data
   - Reveals weaknesses of the classifier
2. **Accuracy measures**: overall / misclassification rate, class-specific, average, kappa coefficient, intersection over union

How to compare the obtained results with the test data?

- **Confusion matrix**: table, where each column represents the instances in a predicted class, while each row represents the instances in an actual class
  - Easy to see where the system mislabels one class as another

Percentage	Classification data				
Reference data	C_1	C_2	C_3	Row total	Class-specific accuracy
C_1	C_{11}	C_{12}	C_{13}	$\sum_i^K C_{1i}$	$\frac{C_{11}}{\sum_i^K C_{1i}}$
C_2	C_{21}	C_{22}	C_{23}	$\sum_i^K C_{2i}$	$\frac{C_{22}}{\sum_i^K C_{2i}}$
C_3	C_{31}	C_{22}	C_{33}	$\sum_i^K C_{3i}$	$\frac{C_{33}}{\sum_i^K C_{3i}}$
Column total	$\sum_i^K C_{i1}$	$\sum_i^K C_{i2}$	$\sum_i^K C_{i3}$	N	
User's accuracy	$\frac{C_{11}}{\sum_i^K C_{i1}}$	$\frac{C_{11}}{\sum_i^K C_{i2}}$	$\frac{C_{33}}{\sum_i^K C_{i3}}$		

  - C_i represents the class i and C_{ij} is the number of pixels classified to the class j and referenced as the class i

How to compare the obtained results with the test data?

Percentage	Classification data				
Reference data	C_1	C_2	C_3	Row total	Class-specific accuracy
C_1	C_{11}	C_{12}	C_{13}	$\sum_i^K C_{1i}$	$\frac{C_{11}}{\sum_i^K C_{1i}}$
C_2	C_{21}	C_{22}	C_{23}	$\sum_i^K C_{2i}$	$\frac{C_{22}}{\sum_i^K C_{2i}}$
C_3	C_{31}	C_{22}	C_{33}	$\sum_i^K C_{3i}$	$\frac{C_{33}}{\sum_i^K C_{3i}}$
Column total	$\sum_i^K C_{i1}$	$\sum_i^K C_{i2}$	$\sum_i^K C_{i3}$	N	
User's accuracy	$\frac{C_{11}}{\sum_i^K C_{i1}}$	$\frac{C_{11}}{\sum_i^K C_{i2}}$	$\frac{C_{33}}{\sum_i^K C_{i3}}$		

- **Overall Accuracy** (OA) is the percentage of correctly classified pixels (K is the number of classes):

$$OA = \frac{\sum_i^K C_{ii}}{\sum_{ij}^K C_{ij}} \times 100\%$$

How to compare the obtained results with the test data?

Percentage	Classification data				
Reference data	C_1	C_2	C_3	Row total	Class-specific accuracy
C_1	C_{11}	C_{12}	C_{13}	$\sum_i^K C_{1i}$	$\frac{C_{11}}{\sum_i^K C_{1i}}$
C_2	C_{21}	C_{22}	C_{23}	$\sum_i^K C_{2i}$	$\frac{C_{22}}{\sum_i^K C_{2i}}$
C_3	C_{31}	C_{22}	C_{33}	$\sum_i^K C_{3i}$	$\frac{C_{33}}{\sum_i^K C_{3i}}$
Column total	$\sum_i^K C_{i1}$	$\sum_i^K C_{i2}$	$\sum_i^K C_{i3}$	N	
User's accuracy	$\frac{C_{11}}{\sum_i^K C_{i1}}$	$\frac{C_{11}}{\sum_i^K C_{i2}}$	$\frac{C_{33}}{\sum_i^K C_{i3}}$		

- **Class Accuracy** (CA, or producer's accuracy) is the percentage of correctly classified pixels for a given class:

$$CA_i = \frac{C_{ii}}{\sum_j^K C_{ij}} \times 100\%$$

How to compare the obtained results with the test data?

Percentage	Classification data				
Reference data	C_1	C_2	C_3	Row total	Class-specific accuracy
C_1	C_{11}	C_{12}	C_{13}	$\sum_i^K C_{1i}$	$\frac{C_{11}}{\sum_i^K C_{1i}}$
C_2	C_{21}	C_{22}	C_{23}	$\sum_i^K C_{2i}$	$\frac{C_{22}}{\sum_i^K C_{2i}}$
C_3	C_{31}	C_{22}	C_{33}	$\sum_i^K C_{3i}$	$\frac{C_{33}}{\sum_i^K C_{3i}}$
Column total	$\sum_i^K C_{i1}$	$\sum_i^K C_{i2}$	$\sum_i^K C_{i3}$	N	
User's accuracy	$\frac{C_{11}}{\sum_i^K C_{i1}}$	$\frac{C_{11}}{\sum_i^K C_{i2}}$	$\frac{C_{33}}{\sum_i^K C_{i3}}$		

- **Average Accuracy** (AA) is the mean of class-specific accuracies for all the classes:

$$AA = \frac{\sum_i^K CA_i}{K} \times 100\%$$

How to compare the obtained results with the test data?

- **Kappa Coefficient** (κ) is the percentage of agreement, *i.e.*, correctly classified pixels, corrected by the number of agreements that would be expected purely by chance:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \times 100\%,$$

$$P_o = OA/100\%,$$

$$P_e = \frac{1}{N^2} \sum_{i}^{K} C_{i\cdot} C_{\cdot i},$$

$$C_{i\cdot} = \sum_{j}^{K} C_{ij}, \quad C_{\cdot i} = \sum_{j}^{K} C_{ji},$$

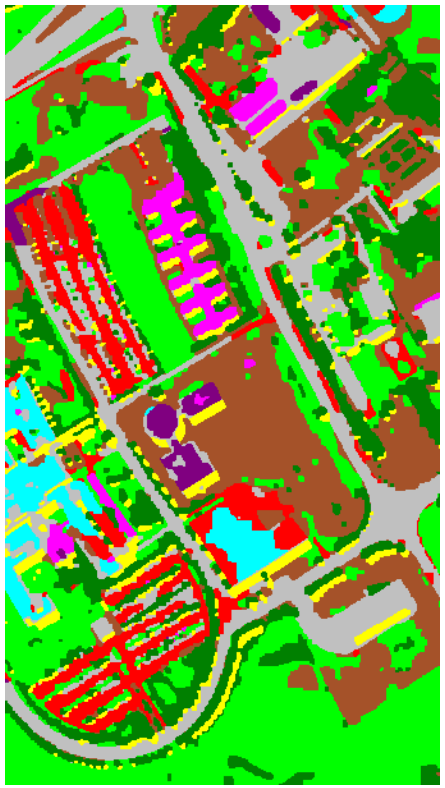
Percentage	Classification data				
Reference data	C_1	C_2	C_3	Row total	Class-specific accuracy
C_1	C_{11}	C_{12}	C_{13}	$\sum_i^K C_{1i}$	$\frac{C_{11}}{\sum_i^K C_{1i}}$
C_2	C_{21}	C_{22}	C_{23}	$\sum_i^K C_{2i}$	$\frac{C_{22}}{\sum_i^K C_{2i}}$
C_3	C_{31}	C_{22}	C_{33}	$\sum_i^K C_{3i}$	$\frac{C_{33}}{\sum_i^K C_{3i}}$
Column total	$\sum_i^K C_{i1}$	$\sum_i^K C_{i2}$	$\sum_i^K C_{i3}$	N	
User's accuracy	$\frac{C_{11}}{\sum_i^K C_{i1}}$	$\frac{C_{11}}{\sum_i^K C_{i2}}$	$\frac{C_{33}}{\sum_i^K C_{i3}}$		

where N is the number of referenced pixels

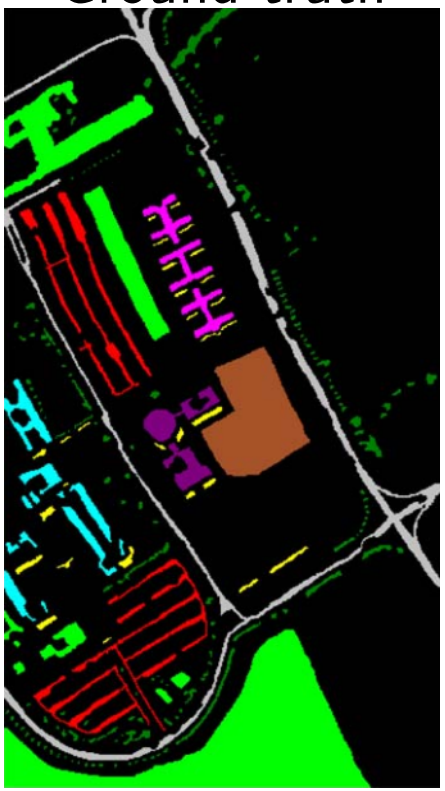
How to compare the obtained results with the test data?

Input image

Classification map



Ground-truth

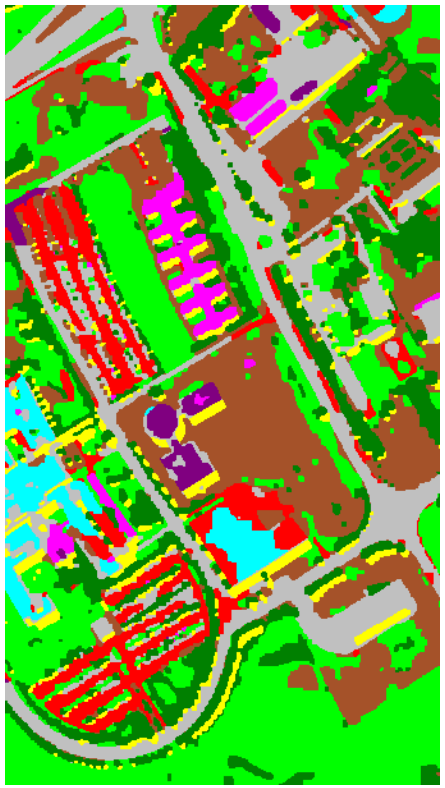


- Overall accuracy $> 90\%$, average accuracy $> 90\%$
- Is this classification result satisfactory?

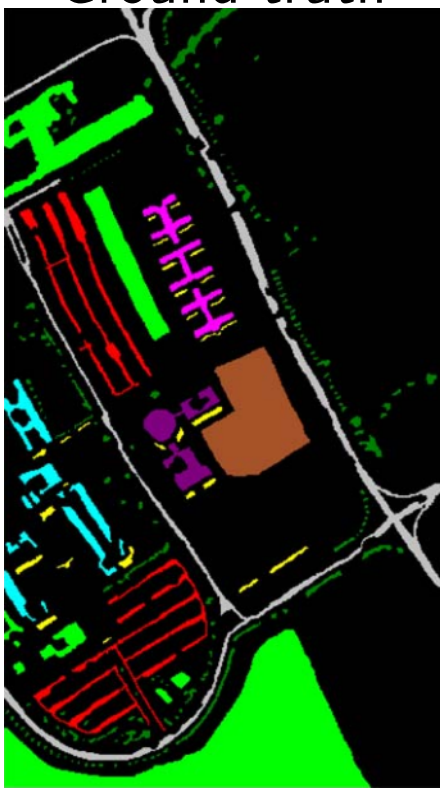
How to compare the obtained results with the test data?

Input image

Classification map



Ground-truth



- Overall accuracy $> 90\%$, average accuracy $> 90\%$
- Is this classification result satisfactory? $\Rightarrow$ Depends on the application

Criterion for object-based classification?

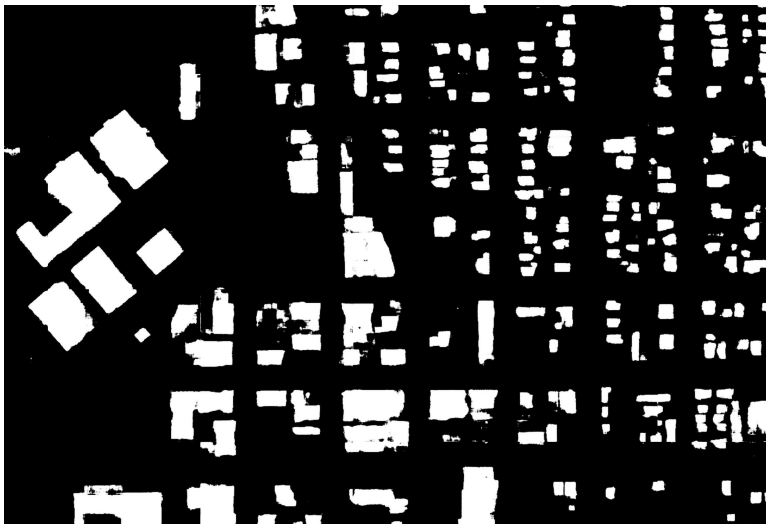
- Intersection over union (IoU) criterion
  - Number of pixels assigned to a particular class both in the classified image and in the ground truth, divided by the total amount of pixels labeled as such in either of them
  - Object-based overlap measure typically used for imbalanced datasets

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Criterion for object-based classification?

- Intersection over union (IoU) criterion
  - Number of pixels assigned to a particular class both in the classified image and in the ground truth, divided by the total amount of pixels labeled as such in either of them
  - Object-based overlap measure typically used for imbalanced datasets

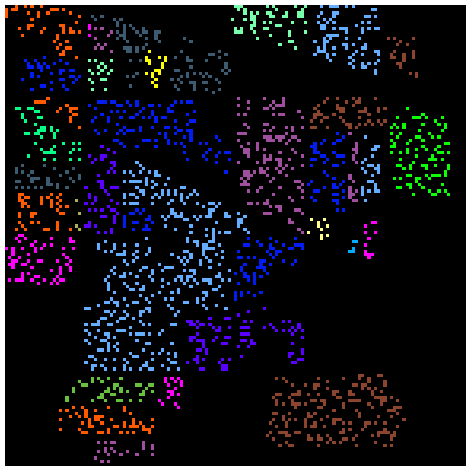
Color image



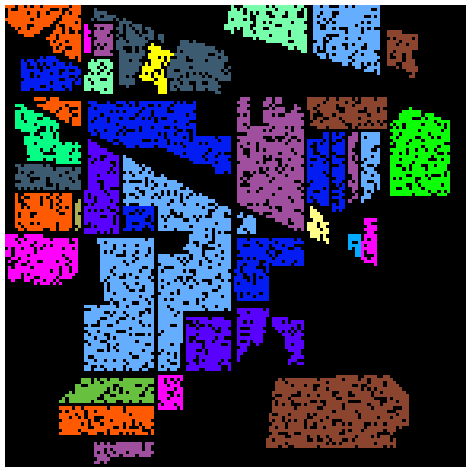
OA = 94%, IoU = 61%

How to split data into training and test sets?

- Sampling strategies
  - Train on one area, test on another area
  - Random sampling: randomly select training samples within the area of each class
  - Patch sampling: image is divided into blocks, test samples are from blocks that haven't been used for training
  - Cluster sampling



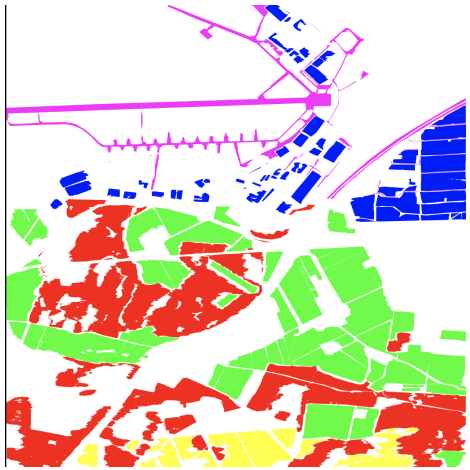
Training set



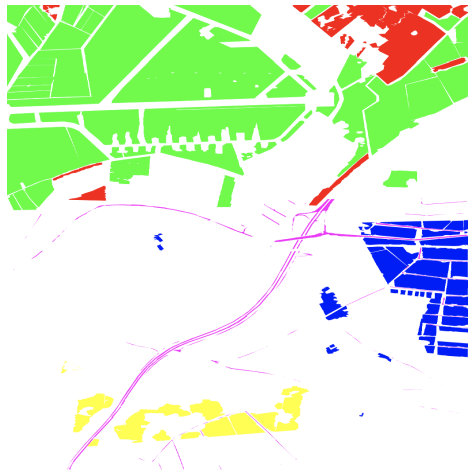
Test set

How to split data into training and test sets?

- Sampling strategies
  - Train on one area, test on another area
  - Random sampling: randomly select training samples within the area of each class
  - Patch sampling: image is divided into blocks, test samples are from blocks that haven't been used for training
  - Cluster sampling



Training set



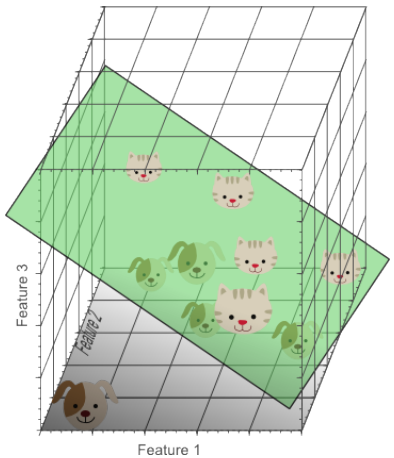
Test set

Choose model / parameters

- **Validation:** if enough training data, we can use $X\%$ for training and $Y\%$ for validaton
  - Test on different parameters
  - Retain parameters that give the highest accuracy on validation set
- If limited training data $\Rightarrow$ **cross-validation**
  - Different types
  - Popular choice: k-fold cross-validation
    - Randomly partition the training data into k equal sized subsets
    - k times: one of k subsets is used for validation, and the rest of data are used for training

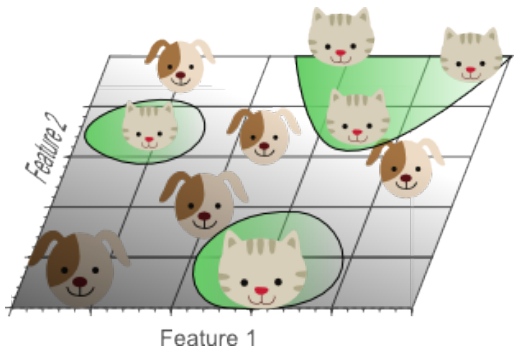
Underfitting versus overfitting

- **Underfitting:** machine learning algorithm cannot capture the underlying trend of the data
  - Often a result of an excessively simple model
- **Overfitting:** Model or the algorithm fits the data too well & captures the noise of the data
  - Often a result of an excessively complicated model
  - Can be prevented by fitting multiple models and using validation or cross-validation

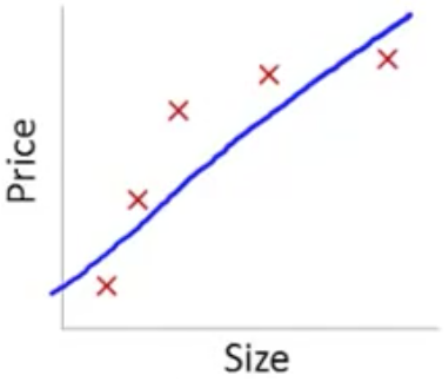


$\Leftarrow$ Good fit

Overfitting $\Rightarrow$



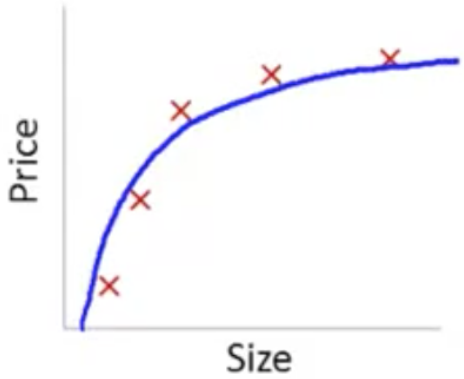
Underfitting versus overfitting



$\theta_0 + \theta_1 x$

High bias
(underfit)

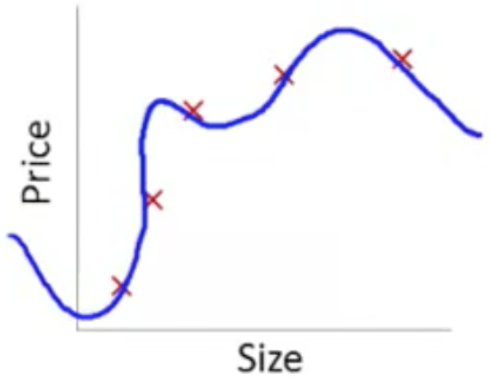
$d = 1$



$\theta_0 + \theta_1 x + \theta_2 x^2$

"Just right"

$d = 2$

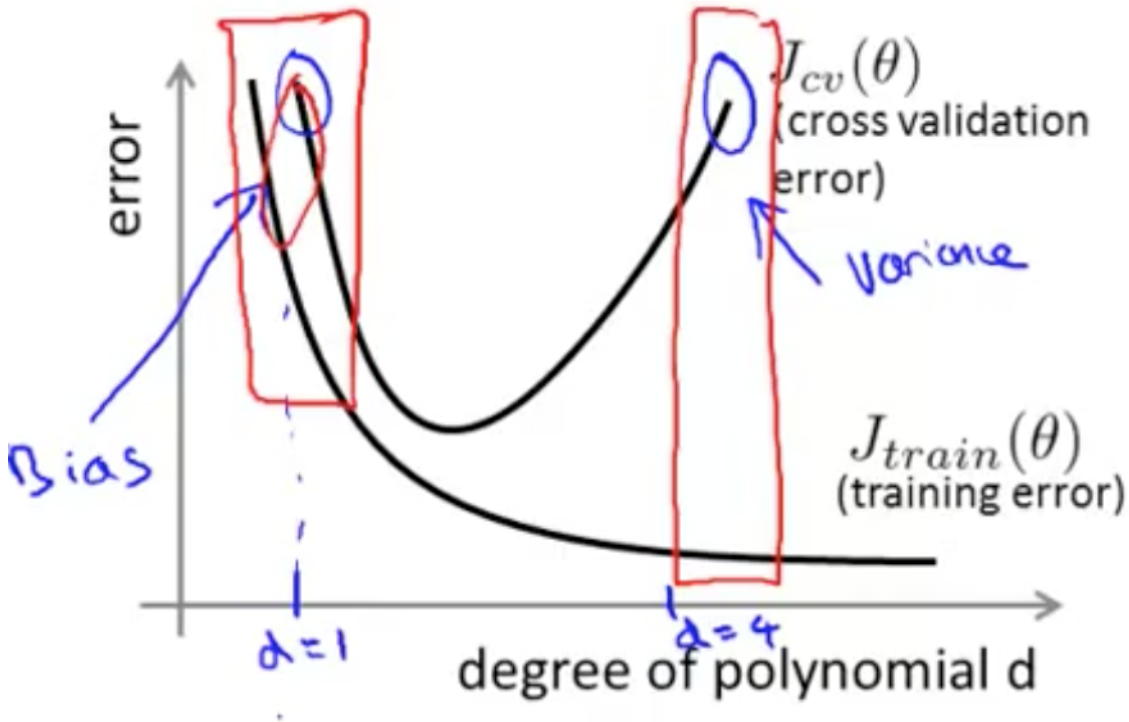


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance
(overfit)

$d = 4$

Diagnosing bias vs. variance



What to try next

- Get more training examples
  - Crowdsourced tools, mobile platforms