Lab course

Image Analysis I

ST 2023

Hubert Kanyamahanga
(kanyamahanga@ipi.uni-hannover.de)

Lab 1: Bayesian Classifiers

Content of the 1st Lab

Goal:

- Automatic extraction of **features** from images for **classification** task
  - By **probabilistic** image interpretation → requires models/classifiers

Tasks:

1. Implement your methods to extract features from images
2. Implement and train Bayesian classifiers to
   - classify real images using extracted features
3. Generate synthetic datasets drawn from Gaussian distribution and
   - apply the implemented classifiers on that toy dataset
4. Discussion and evaluation (**Visually** and **Quantitively**)

Overview: Lab 1

- Hand-crafted Features
- Bayesian Classification
- Generative Probabilistic Classifiers:
  - Normal Distribution (Gaussian) Classifier
  - Gaussian Mixture Model Classifier
- Quality Metrics

Feature Extraction

- Two patches (training / testing) with:
  - Channels: Near infrared, red and green (NIR,R,G)
  - Normalized Digital Surface Model (NDSM)
  - Reference labels (per pixel, 5 classes)
    - Classes are: 'STREET', 'HOUSE', 'LOW VEG.', 'HIGH VEG.', 'CAR'

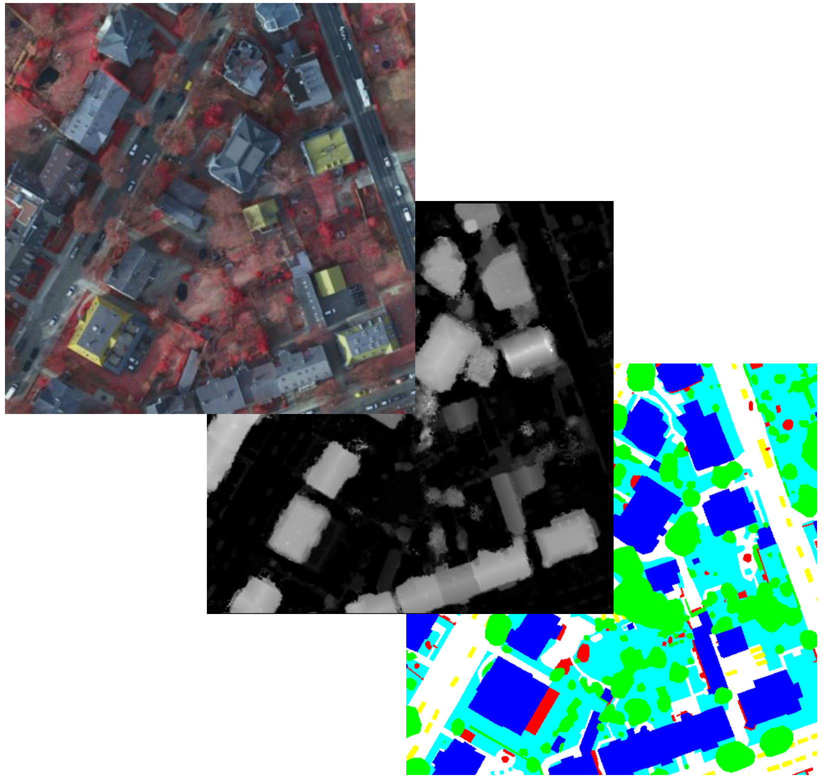
- Additional feature: $NDVI = \frac{NIR-R}{NIR+R}$ (if NIR = R = 0 → NDVI = 0) Note that **-1 < NDVI > +1**

- We will not use class ‚clutter' → merge with class ‚low vegetation'

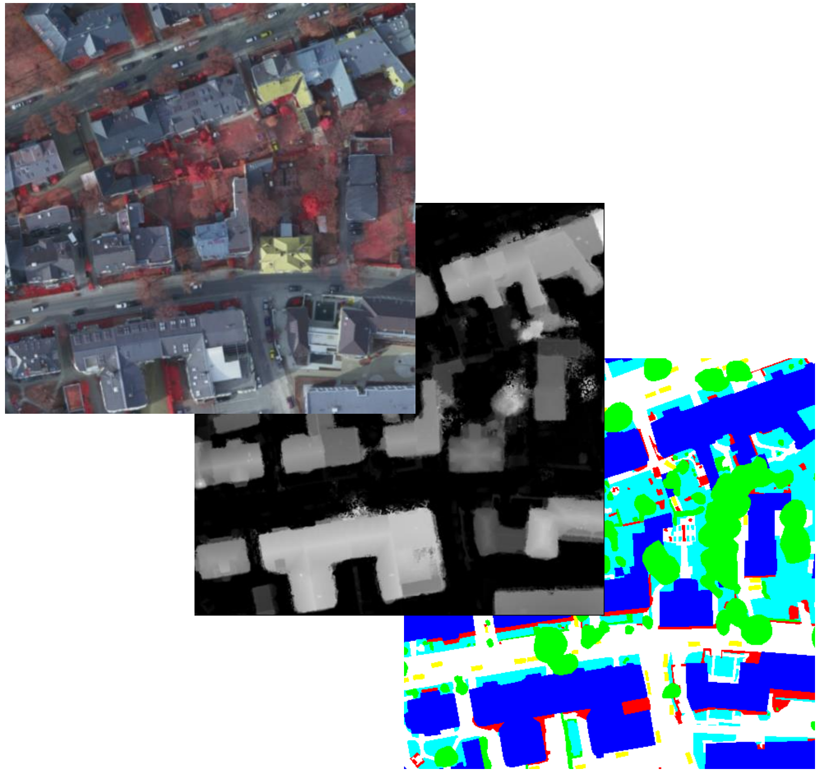
- Normalization of channels required!
  - Shift and scale IR, R and G so that 0 will be mapped to -1.0 and 255 will be mapped to 1.0

Dataset

Training



Testing



Objective: Classification Task: USE extracted features, fit a probabilistic classifier

Overview: Lab 1

- Hand-crafted Features
- **Bayesian Classification**
- Generative Probabilistic Classifiers:
  - Normal Distribution (Gaussian) Classifier
  - Gaussian Mixture Model Classifier
- Quality Metrics

Bayesian Classification

Classification task: What is the class label L for a feature $\mathbf{x}$?

MAP-Criterion: Classification based on maximum of the **posterior probability** $p(C \mid \mathbf{x})$

Generative approach:

- Often easier to model the causal relation between object type and observed features: the observed features are a function of the object type
- $p(C \mid \mathbf{x})$ is modelled indirectly according to the Theorem of Bayes:

$$p(C \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C) \cdot p(C)}{p(\mathbf{x})}$$

- The theorem of Bayes allows inverse reasoning: derive information about the cause (the class) from the **effect** (the observed features).

Bayesian Classification

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C) \cdot p(C)}{p(\mathbf{x})}$$

- $p(\mathbf{x}|C)$: **Likelihood**
  - Probability to observe **x** if it is known to belong to class C
  - For **each class C^k** there is a model for $p(\mathbf{x}|C=C^k)$ that **describes the distribution of the features** for this class
  - **Determined from training data,** but **?????? HOW ??????**
  - ~~Non-parametric Models: direct determination of $p(\mathbf{x} \mid C)$ from the training data~~
  - Parametric Models: Based on the assumption of an analytical model for $p(\mathbf{x} \mid C)$ whose **parameters are estimated from the training data → Machine Learning**

Bayesian Classification

$$p(C\,|\,\mathbf{x}) = \frac{p(\mathbf{x}\,|\,C)\cdot p(C)}{p(\mathbf{x})}$$

- p(*C*): **Prior probability**
  - Corresponds to bias for the occurrence of *C*
  - If no information is available: **Uniform Distribution** **(***e.g for 5 classes: p(C1)=1/5, p(C2)=1/5, p(C3)=1/5, p(C4)= /5, p(C5)=1/5***)**
    → MAP becomes **Maximum Likelihood** (ML) criterion
  - *p*(*C*) can be determined iteratively :
    1) Classification under the assumption of a uniform distribution of the occurrence of the individual classes
    2) Determination of p(*C*) from the relative frequencies of occurrence of the individual classes C^k
    3) Classification according to the theorem of Bayes

Bayesian Classification

$$p(C \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C) \cdot p(C)}{p(\mathbf{x})}$$

- p(**x**): **Probability of the data**
  - Equal for all values of *C* because it does not depend on *C*

    $\Rightarrow$ MAP can also be applied without knowing $p(\mathbf{x})$:

    $p(C|\mathbf{x}) \propto p(\mathbf{x}|C) \cdot p(C)$ implies that

    $\max(p(C|\mathbf{x})) = \max(p(\mathbf{x}|C) \cdot p(C))$
  - $p(\mathbf{x})$ ensures that $p(C|\mathbf{x})$ can be interpreted as a probability and can be used as such in further probabilistic processes
  - $p(\mathbf{x})$ can be determined as the marginal distribution of $p(\mathbf{x},C)$: *$p(\mathbf{x})$ independent of C*

$$p(\mathbf{x}) = \sum_k p\left(\mathbf{x} \mid C = L^k\right) \cdot p\left(C = L^k\right)$$

Summary: Bayesian Classification

$$p(C \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C) \cdot p(C)}{p(\mathbf{x})}$$

- Given:
  - Models for the likelihoods $p(\mathbf{x}|C=L^k)$ of all classes L^k
  - Priori probabilities $p(C=L^k)$ of all classes L^k
  - ??Question??: A feature vector $\mathbf{x}$ to be classified
- Wanted: class C_{map} of $\mathbf{x}$ according to the MAP criterion: *Maximize posterior probability p(C | x)*
- Procedure: Compute the posterior and take the class label to be the index of the maximum (ref. slide 9)
  1) For all L^k: calculate $p(\mathbf{x}, C=L^k) = p(\mathbf{x}|C=L^k) \cdot p(C=L^k)$
  2) Calculate $p(\mathbf{x}) = \sum_k p(\mathbf{x} \mid C = L^k) \cdot p(C = L^k)$
  3) For all L^k: calculate $p(C=L^k|\mathbf{x}) = p(\mathbf{x}, C=L^k) / p(\mathbf{x})$
  4) C_{map} is the label L^k for which $\boldsymbol{p(C=L^k|\mathbf{x})}$ **is a maximum**

Overview: Lab 1

- Hand-crafted Features
- Bayesian Classification
- Generative probabilistic classifiers:
  - Normal Distribution (Gaussian) Classifier
  - Gaussian Mixture Model Classifier
- Quality Metrics

Generative Probabilistic Classifiers

How to model the likelihood? → **Learning** from data!

$$p(C \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C) \cdot p(C)}{p(\mathbf{x})}$$

Parametric Methods:

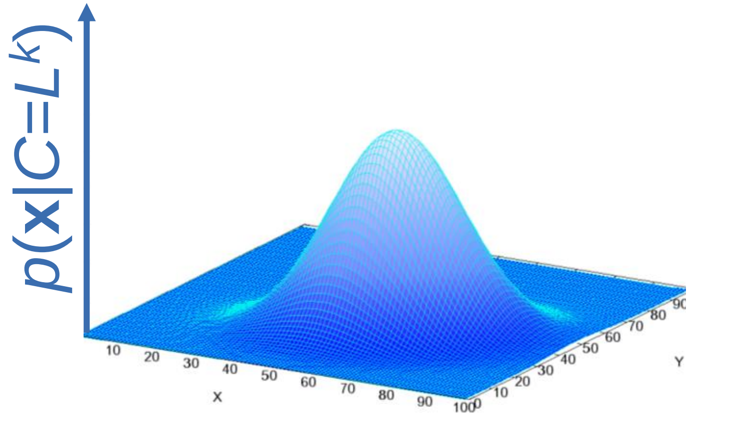
- Analytical model for the probability density $p(\mathbf{x}|C)$ is assumed
- The probability density function $p(\mathbf{x}|C)$ also depends on parameters $\boldsymbol{\theta}$, i.e. $p(\mathbf{x}|C) = p(\mathbf{x}|C,\boldsymbol{\theta})$
- Parameters are learned from training data
  → Training samples are required for each class $C=L^k$ to determine the parameters $\boldsymbol{\theta}_k$ of $p(\mathbf{x}|C=L^k,\boldsymbol{\theta}_k)$
- **BUT which parametric models do we USE?**

~~(Non-parametric methods are another option, but not covered in this lab)~~

1. Single Gaussian Model

- Frequent assumption: **Multivariate normal distribution**

$$p\left(\mathbf{x} \mid C = L^k\right) = \frac{1}{(2\pi)^{D/2} \cdot \left\|\mathbf{\Sigma}_k\right\|^{1/2}} \cdot e^{-1/2(\mathbf{x}-\boldsymbol{\mu}_k)^T \cdot \mathbf{\Sigma}_k^{-1} \cdot (\mathbf{x}-\boldsymbol{\mu}_k)}$$



- **Extension of Univariate case**, **x** here is a feature **Vector**
- **Maximum (Log-) Likelihood estimation of the parameters:**

$$\Sigma_i \ln p(\mathbf{x}_{ik}|\theta_k) \rightarrow \max$$

$\Rightarrow$ Result for $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \cdot \sum_i \mathbf{x}_{ik}$$

$\theta_k ==$ ($\boldsymbol{\mu}_k$ and Σ_k):

$\Rightarrow$ Result for Σ_k :

$$\mathbf{\Sigma}_k = \frac{1}{N_k - 1} \cdot \sum_i (\mathbf{x}_{ik} - \boldsymbol{\mu}_k) \cdot (\mathbf{x}_{ik} - \boldsymbol{\mu}_k)^T$$

Learned parameters for a Single Gaussian Model

- **Prerequisite: L^k must only correspond to one cluster in feature space (c./e.g. Slide 48/47)**

Overview: Lab 1

- Hand-crafted Features
- Bayesian Classification
- Generative Probabilistic Classifiers:
  - Normal Distribution (Gaussian) Classifier
  - Gaussian Mixture Model Classifier
- Quality metrics

2. Gaussian Mixture Model

- In the case of $\boldsymbol{N_j}$ **clusters** for the class $C = L_k$, every cluster is described by a normal distribution

$\theta_k ==$ (π_j , μ_{kj} and Σ_{kj}):

- The total probability density is obtained from the **weighted sum**

Learned parameters for a Mixture of Gaussian Model

of the components:

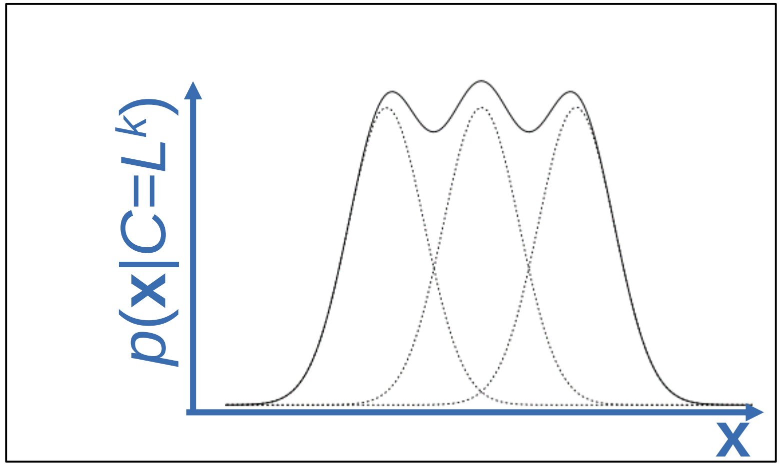
$$p\left(\mathbf{x} \mid C = L^k\right) = \sum_{j=1}^{N_j} \pi_j \cdot N\left(\mathbf{x} \mid \boldsymbol{\mu}_{kj}, \boldsymbol{\Sigma}_{kj}\right)$$

with π_j ... Mixture coefficient for cluster j, corresponding to the prior probability for j

μ_{kj} ... Mean value for cluster j

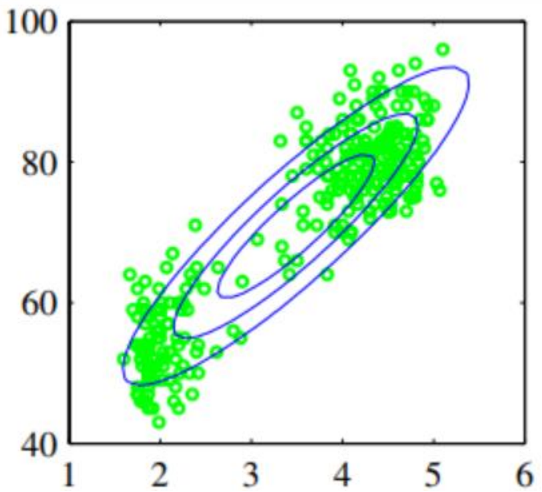
Σ_{kj} ... Covariance matrix for cluster j

$N(\mathbf{x}| \mu_{kj}, \Sigma_{kj})$ ... Probability density of the normal distribution for cluster j

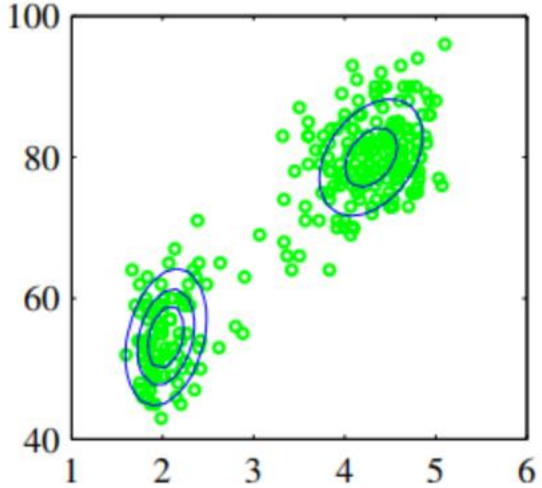


Gaussian Mixture Model: Example

Single Gaussian Model



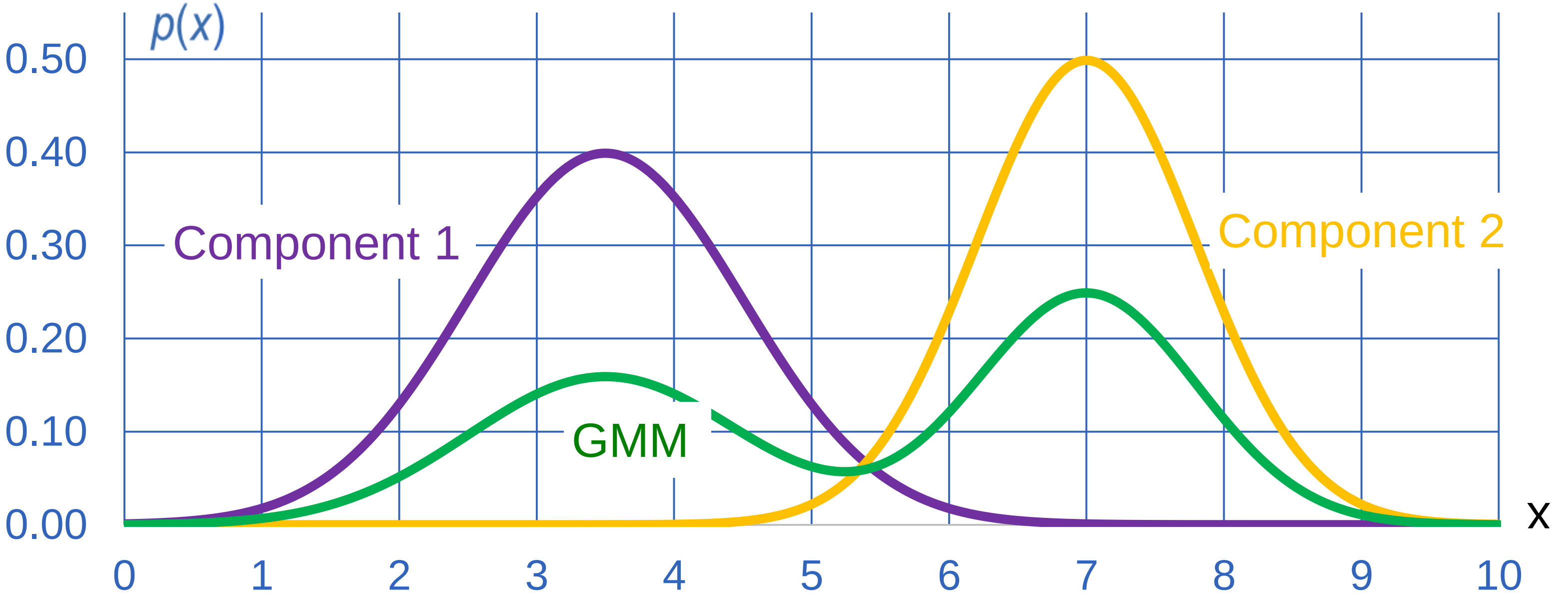
Gaussian Mixture Model



- One feature x, two Gaussian components (class index k is omitted):
  - Component 1 (j = 1): $\mu_1 = 3.5$, $\sigma_1 = 1.0$, $\pi_1 = 0.3$
  - Component 2 (j = 2): $\mu_2 = 7.0$, $\sigma_2 = 0.8$, $\pi_2 = 0.7$
  - GMM: $p(x) = \pi_1 \cdot N(x|\mu_1, \sigma_1) + \pi_2 \cdot N(x|\mu_2, \sigma_2)$

Linear combination of Gaussian distributions

- e.g. Street shadow/ sun



Summary: Gaussian Mixture Model

- Issue: We doesn't know which data points came from which latent component/cluster
- Parameters to be estimated: π_j, $\mathbf{\mu}_{kj}$, Σ_{kj} (on set of these params. per cluster j and class k)
- Training of the mixture model requires cluster analysis of the feature space → unsupervised classification
- Method: "**Expectation Maximisation**" (EM)
  → see lecture "Unsupervised Classification"
  - E-step: For each data point, compute the prob. of being generated by each component
  - M-step: Adjust the parameters to maximize the likelihood of the data given those params.
- EM requires the number of clusters N_j to be known in advance

Overview: Lab 1

- Hand-crafted Features
- Bayesian Classification
- Generative Probabilistic Classifiers:
  - Normal Distribution (Gaussian) Classifier
  - Gaussian Mixture Model Classifier
- Quality metrics

Quality Metrics

- Quality metrics are used to evaluate the performance of a classifier
- Evaluation requires a reference against which to compare the results of a classifier
- Class-wise metrics:
  - Precision (aka User's Accuracy, Correctness)
  - Recall (aka Producer's Accuracy, Completeness)
  - Intersection over Union (aka Quality)
  - F1 score
  - …
- Global metrics:
  - Overall Accuracy
  - Mean F1 score
  - …

[python script]

```python
def calculate_quality_metrics(predicted, true_labels, classes):
    TP = # True Positives
    FP = # False Positives
    FN = # False Negatives

    p = TP/(TP+FP)
    r = TP/(TP+FN)
    f1 = 2*p*r/ (p + r)

    return p, r, f1
```

Quality Metrics

Terminology:

- True Positives (TP_k):
  - Number of features correctly classified as class k
- False Positives (FP_k):
  - Number of features wrongly classified as class k
- False Negatives (FN_k):
  - Number of features wrongly classified as not class k

Quality Metrics

- Precision:

$$P_k = \frac{TP_k}{TP_k + FP_k}$$

All samples classified as k

- Recall:

$$R_k = \frac{TP_k}{TP_k + FN_k}$$

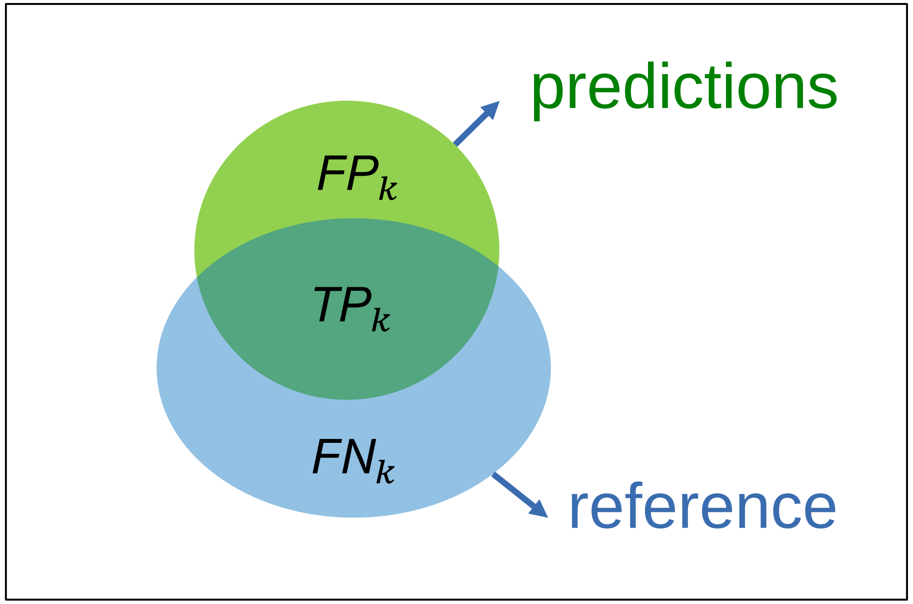
All samples that belong to class k

- I.o.U.:

$$IoU_k = \frac{TP_k}{TP_k + FP_k + FN_k}$$

- F1 score:

$$F_k = \frac{2 \cdot P_k \cdot R_k}{P_k + R_k}$$



Quality Metrics

- Overall accuracy:

$$OA = \frac{1}{N} \cdot \sum_{k=1}^{M} TP_k$$

- Mean F1 score:

$$mF1 = \frac{1}{M} \cdot \sum_{k=1}^{M} F_k$$

With:

- N: Number of samples
- M: Number of classes

Submission of Results

- Assignment

→ Jupyter Notebook (only digital)

  – Run every cell and save the notebook
  – Use meaningful variable names (refer to IAI_23_Lab_Python_Basics.ipynb)

  – Write comments if required
  – Answer concisely but completely

  – Consider acceptance rules
  – Consider IAI_23_Lab_Technical_Details.pdf

Submission of Results

- Submission deadline: **12. June 2023 before 11:00 am**

- There are no resubmissions!
- If you failed one or more labs in the last semester, you have to hand in ALL labs again!

- Introduction to the 2nd Lab: **12. June. 2023 at 1:00pm**