



# Bildgestützte Automatisierung II

## Übung 2

# 12.05.2022 - Übungsablauf

---

Uhrzeit	Aktivität
14:30 – 15:00 Uhr	Vorbereitungsaufgabe UE 2: Klärung der Fragen UE 2 Durchführung, Aufg. 3 Optimierung und Evaluation: Erläuterung der Funktionen
15:00 – 16:00 Uhr	Arbeiten in Gruppen in Breakout-Rooms oder selbstständig <ul style="list-style-type: none"><li>▪ Mehrere Breakout-Rooms (Konferenzräume) stehen zur Verfügung</li><li>▪ Freie Auswahl des Raums</li></ul>
16:00 – 16:15 Uhr	Pause
15:15 – 17:45 Uhr	Arbeiten in Gruppen in Breakout-Rooms oder selbstständig <ul style="list-style-type: none"><li>▪ Mehrere Breakout-Rooms (Konferenzräume) stehen zur Verfügung</li><li>▪ Freie Auswahl des Raums</li></ul>

# Vorbereitungsaufgabe (1)

---

Paper „Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark”

## Fragen zum Kapitel III. DATASET

- Wie groß ist die Bilddatenbank? Wie ist die Datenbasis strukturiert?
- Was könnte der Grund für die Unterteilung in Training und Evaluation Set sein?
- Welche Kategorien werden betrachtet?

# Vorbereitungsaufgabe (2)

---

Paper „Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark”

## Fragen zum Kapitel IV. EVALUATION PROCEDURE

- Mit welchen Daten wird die Evaluation durchgeführt?
- Was ist ein Precision-Recall-Plot?
- Welcher Score ist ausschlaggebend für das Ranking im Wettbewerb?
- Welcher Score wird genutzt, um die Überlappung von gefundener und tatsächlicher ROI zu ermitteln?

# Vorbereitungsaufgabe (1)

---

Paper „Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark”

Fragen zum Kapitel V. BASELINE ALGORITHMS

- Welche drei Methoden wurden als Baseline vorgestellt?
- Auf welchen Features basieren die Methoden?

# Jupyter Notebook: Funktionen aus Aufgabe 3 (Übung 2)

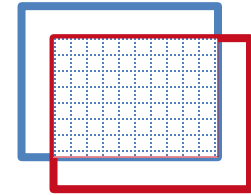
---

- Funktion zur Berechnung des Jaccard-Ähnlichkeitsmaßes
- Funktion zur Evaluation der Detektionen
- Funktion zur Berechnung von Precision und Recall

# Jaccard-Ähnlichkeitsmaß

- Output eines typischen Detektionsalgorithmus ist eine Liste mit ROIs
- Für jedes ROI wird Jaccard-Ähnlichkeitsmaß berechnet:

$$J(S, G) = \frac{|S \cap G|}{|S \cup G|} \in [0,1]$$



- Mit Hilfe einer binären Loss-Funktion  $J_b$ :
  - $J_b = 0$  if  $J < 0.6$  and  $J_b = 1$  otherwise
- Bei Überschneidung von mehreren berechneten ROIs mit dem „groundtruth“-ROI  $G$ :
  - der ROI mit dem größten Überschneidungsgrad wird verwendet
  - Alle anderen ROIs werden verworfen (weder als „hit“ noch als „miss“ gezählt)

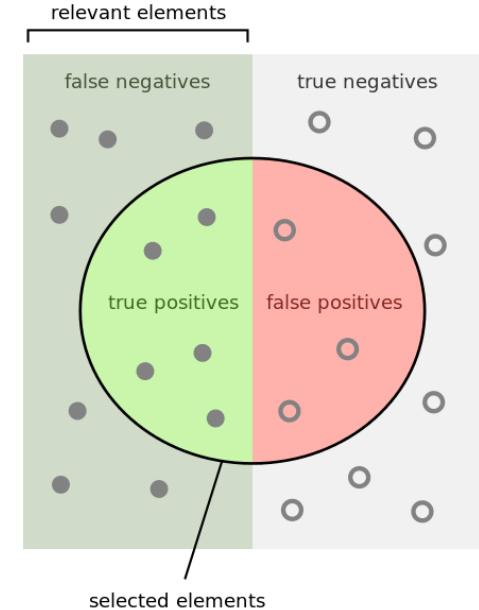
ROI – Region of Interest

# Precision und Recall

$$precision = \frac{true\ positives}{true\ positives + false\ positives} =$$



$$recall = \frac{true\ positives}{true\ positives + false\ negatives} =$$



Quelle: Walber, CC BY-SA 4.0



## Zwischenergebnisse Formbasierter Ansatz

## Form- und farbbasierter Ansatz

- Berücksichtigung der **Form** und der **Farbe** der Schilder

# UE 2 – Optimierungsvorschläge(1)

---

- Vorverarbeitung:
  - Betrachtung der Farbkanäle
  - Einsatz der morphologischen Operationen (Erosion, Dilatation - [Link](#))
  - Rauschenunterdrückung
- Hilfreiche OpenCv-Funktionen:
  - cv2.cvtColor – Konvertieren in einen anderen Farbraum
  - cv2.split – Kanäle eines Bildes trennen
  - cv2.merge – einzelne Kanäle zu einem Bild zusammenfügen
  - cv2.threshold oder cv2.adaptiveThreshold – Umsetzung einer Schwellwertoperation
  - cv2.inRange – Umsetzung einer Schwellwertoperation mit zwei Schwellwerten
  - cv2.bitwise\_or, cv2.bitwise\_and – Logische Operatoren in Zusammenhang mit binären Bildern

## UE 2 – Optimierungsvorschläge (2)

---

- Dynamische Parameteranpassung in HoughCircles-Funktion
- Entfernung kollidierender Kreise mit z.B.: einer Zusatzfunktion
- Einschränkung der Suche nach Kreisen durch folgende Angaben:
  - minimale und maximale Anzahl der Kreise
  - minimale und maximale Größe der Kreise
  - Position der Kreise im Bild
  - Farben der Kreise und des Hintergrunds (z.B.: Bäume)