

## Modellbasierte Detektionsverfahren

### Überblick

Das Ziel dieser Übung ist es, erste Detektionsverfahren zu implementieren. Die Verkehrsschilder der Kategorie „prohibitory“ sollen modellbasiert gefunden werden und deren Leistungsfähigkeit evaluiert werden. Im Jupyter Notebook UE2\_Aufgaben.ipynb findet ihr Hilfsfunktionen, die Ihr für die Erledigung der Hausaufgabe benutzen könnt. Die in der ersten Übung implementierten Funktionen können ebenfalls als Basis benutzt werden.

### Aufgabe 1 – Aussortieren bestimmter Verkehrsschilder

Der Benchmark-Datensatz unterscheidet mehrere Kategorien von Verkehrsschildern. Die Schilder der Kategorie „prohibitory“ sind rund, haben einen weißen Hintergrund und einen roten Rand.

Es soll eine Funktion implementiert werden, die die `gt_txt` Datei übergeben bekommt und ein Ergebnis in Form eines Python-Dictionary zurückgibt. Das Python-Dictionary enthält die Zeilen der `gt.txt`-Datei, die der Kategorie „prohibitory“ zugeordnet werden, mit

- Dateinamen als `dict.keys()` und
- Listen `[leftCol, topRow, rightCol, bottomRow, ClassID]` als `dict.values()`.

Zum Anzeigen der Verkehrszeichen-Bilder aus der Python-Dictionary soll eine weitere Funktion implementiert werden. Ihr könnt die Funktion `calc_rois` aus der ersten Übungsaufgabe als Basis nehmen und entsprechend anpassen oder eine eigene Funktion implementieren. Jede Gruppe lädt ein gespeichertes Beispiel-Bild in den dafür vorgesehenen Ordner (Screenshots) hoch.



Abbildung 1: Beispiel für ein Screenshot für Aufgabe 1

## Aufgabe 2 – Formbasierter Ansatz

Zunächst wollen wir lediglich die Form als Merkmal zur Detektion verwenden. Ihr könnt die Funktion [HoughCircles\(\)](#) nutzen, um mit Hilfe von OpenCV eine Hough-Transformation zur Kreiserkennung durchzuführen. Diese Funktion nutzt die Gradientenrichtung, um die Mittelpunkte der Kreise zu bestimmen. Daher führt sie auch die Kantendetektion selbstständig durch. Schaut euch die OpenCV-Dokumentation um herauszufinden, welche Vorverarbeitungsschritte notwendig bzw. sinnvoll sind. Beachtet auch, dass ihr die gefundenen Kreise in Rechtecke (siehe Abbildung 3) überführen müsst. Jede Gruppe lädt ein gespeichertes Beispiel-Bild (siehe Abbildung 3) in den dafür vorgesehenen Ordner (Screenshots) hoch.



Abbildung 2: Beispielbild (1) nach der Anwendung von `HoughCircles()`



Abbildung 3: Beispielbild (2) nach der Anwendung von `HoughCircles()` und Überführung der Kreise in Rechtecke

## Aufgabe 3 – Optimierung und Evaluation des formbasierten Ansatzes

Probiert verschiedene Parameterkombinationen für euren Ansatz aus.

Über welche(n) Schwellwert(e) könnt ihr die Empfindlichkeit der Detektion variieren? Nehmt für eine Parameterkombination eurer Wahl einen Precision-Recall-Plot auf, indem ihr ausschließlich den gefundenen Schwellwert verändert. Vergleicht eure Ergebnisse mit den Baseline-Algorithmen von Houben et. al. Jede Gruppe lädt einen Precision-Recall-Plot in den dafür vorgesehenen Ordner (Screenshots) hoch.

## Aufgabe 4 – Form- und farbbasierter Ansatz

Im zweiten Schritt soll nun nicht nur die Form, sondern auch die Farbe der Schilder berücksichtigt werden. Überlegt euch, auf welche Art und Weise ihr die Farbinformationen nutzen könnt und erweitert euer Programm entsprechend!

Hilfreiche OpenCV-Funktionen könnten beispielsweise die folgenden sein:

- [cv2.cvtColor](#) – Konvertieren in einen anderen Farbraum
- [cv2.split](#) – Kanäle eines Bildes trennen
- [cv2.merge](#) – einzelne Kanäle zu einem Bild zusammenfügen
- [cv2.threshold](#) – Umsetzung einer Schwellwertoperation
- [cv2.inRange](#) – Umsetzung einer Schwellwertoperation mit zwei Schwellwerten
- [cv2.bitwise\\_or](#), [cv2.bitwise\\_and](#) – Logische Operatoren in Zusammenhang mit binären Bildern
- [Segmentierung von Farben](#)

## Aufgabe 5 – Optimierung und Evaluation des form- und farbbasierten Ansatzes

Findet heraus welcher Schwellwert entscheidend für die Empfindlichkeit der Detektion ist und nehmt einen Precision-Recall-Plot auf. Vergleicht das Ergebnis mit der rein formbasierten Detektion. Jede Gruppe lädt einen Precision-Recall-Plot in den dafür vorgesehenen Ordner (Screenshots) hoch.

### Bonus:

Die Kategorie „mandatory“ hat dieselben Formmerkmale, jedoch andere Farbmerkmale. Wendet eure beiden Ansätze auf diese Schildkategorie an und nehmt die Precision-Recall-Plots auf. Passt dafür die Verwendung der Farbinformation im zweiten Ansatz entsprechend an.

**Benennung der Bilder:** Gruppe\_xy\_Aufgabe\_z.png (xy – Gruppennummer, z- Aufgabennummer)