

EX2_1 Ball Detection and Segmentation

Froschauer Michael - S2410454009

5. April 2025

1 Recherche der Bilddatenbanken für Sportarten

Für die Umsetzung der Ballerkennung und Ballsegmentierung wurde die Sportart Fußball gewählt. Zur Auswahl entsprechender Testbilder wurden die folgenden öffentlichen und frei verwendbaren Bilddatenbanken durchsucht:

- COCO Dataset – Eine umfangreiche Sammlung von Bildern mit Annotationsdaten für verschiedene Anwendungen, einschließlich Bildsegmentierung und Objekterkennung.
- Open Images Dataset – Eine Open-Source-Bilddatenbank von Google, die eine große Sammlung von Bildern mit Segmentierungs- und Klassifikationsannotationsdaten bietet.
- PA Images – Eine Bilddatenbank, die speziell auf Sportbilder und -ereignisse fokussiert ist, darunter auch Fußballbilder.

Mithilfe dieser Datenbanken wurden einige Testbilder ausgewählt, die für dieses Projekt verwendet werden. Diese Testbilder sind in der ZIP-Datei, die mit der Abgabe übermittelt wird, enthalten.

2 Lokalisierung von Bällen im Bild

Für die Lokalisierung von Bällen wurde YOLO11 von Ultralytics als Python-Bibliothek verwendet. YOLO11 (You Only Look Once) ist ein Modell zur Objekterkennung, das in der Lage ist, Objekte in Bildern zu klassifizieren und zu lokalisieren. Es bietet eine einfache Implementierung und ermöglicht sowohl die Nutzung vortrainierter Modelle als auch das Training eigener Modelle. In unserem Fall wurde das vortrainierte Modell verwendet, das bereits in der Lage ist, den Klassentyp `sports ball` zu erkennen, was für unsere Anwendung ausreichend ist.

Die Verwendung sieht wie folgt aus:

```
1 yolo_results = model(image_path)
```

Dabei gibt `model(image_path)` die Objekte im Bild zurück, zusammen mit den Koordinaten der Rechtecke (Bounding-Boxes), die die erkannten Objekte umschließen. Da YOLO11 jedoch eine gewisse Rechenzeit benötigt, wurden die Ergebnisse für jedes Bild in einer Ergebnisdatei serialisiert. Diese serialisierten Ergebnisse können später erneut verwendet werden, ohne dass das Modell die Objekterkennung für dasselbe Bild wiederholen muss.

Um die Ergebnisse für ein Bild zu verarbeiten und die Bälle zu lokalisieren, wird folgender Aufruf verwendet:

```
1 image_bounding_box, ball_boxes = process_detections(image_bounding_box, yolo_results,  
↪ class_names)
```

In dieser selbst implementierten Funktion wird das Bild zusammen mit den YOLO Ergebnissen übergeben. Es wird ein Bild zurückgegeben, auf dem die erkannten Objekte markiert sind. Zusätzlich wird eine Liste erzeugt, die nur die Rechtecke enthält, in denen Bälle erkannt wurden.

Mit dieser Grundlage kann nun die Segmentierung der Bälle in den erkannten Bereichen beginnen.

3 Segmentierung der Bälle

Nach der Klassifizierung der Bälle folgt die Segmentierung. Die Segmentierung basiert hauptsächlich auf der Farbe der Bälle. Im Rahmen dieses Projekts konzentrieren werden Fußbälle segmentiert.

Es treten jedoch einige Herausforderungen auf:

- Fußbälle haben nicht immer die gleiche Farbe. Sie sind meistens weiß, jedoch variiert dies.
- Fußbälle sind nicht einfärbig, sondern bestehen oft aus mehreren Farbflächen.
- Bälle werfen Schatten, besonders wenn sie auf dem Boden liegen. Diese Schatten führen dazu, dass etwa ein Drittel des Balls deutlich dunkler erscheint, was die Segmentierung erschwert.

Um eine präzise Segmentierung zu erzielen, müssen diese Herausforderungen berücksichtigt werden. Dafür werden verschiedene Methoden getestet, die unterschiedliche Ansätze zur Farbsegmentierung nutzen.

3.1 Information zu den Tests

In dieser Dokumentation wird für Vergleichszwecke dasselbe Bild verwendet. Es wurde jedoch auch mit mehreren anderen Bildern getestet, die im Anhang der Abgabe enthalten sind.

3.2 Aufteilung der Implementierung

Die Implementierung der Ballsegmentierung erfolgt in mehreren Schritten:

1. Einlesen oder Erstellen der YOLO-Ergebnisse (Sportball-Bounding-Boxes und Klassifizierungsbild).
2. Erstellung der Ball-Maske (verschiedene Methoden zur Segmentierung werden getestet).
3. Overlay der Ball-Maske auf das Originalbild.
4. Anzeige und Speicherung der bearbeiteten Bilder.

3.3 Erstellung der Ball-Maske

Zur Erstellung der Ball-Maske wird der vom YOLO-Modell erkannte Bereich des Balls genauer untersucht. Dieser Bereich wird um 10 Pixel in alle Richtungen erweitert, um eventuelle Fehler durch abgeschnittene Bälle zu kompensieren.

Es werden mehrere Methoden zur Segmentierung getestet:

3.3.1 Farbsegmentierung

Diese Methode nutzt ausschließlich die Farbe des Balls zur Segmentierung. Dabei wird ein Schwellenwert für die Farbe des Balls festgelegt, wobei in dieser Methode nur weiße Bälle segmentiert werden können. Andere Farben werden nicht berücksichtigt.

- Filterung:
 1. Festlegung der Farbgrenzen für den weißen Bereich des Balls.
 2. Weichzeichnung des Bildes, um Rauschen zu reduzieren.
 3. Erzeugung eines Farb-Schwellenwerts, der idealerweise nur die Ballfarbe (weiß) enthält.
 4. Anwendung eines Schwellenwerts auf das Bild, um eine binäre Schwarz-Weiß-Maske zu erstellen.
 5. Optional: Einsatz morphologischer Filter, um Artefakte und Rauschen zu entfernen.



Abbildung 1: Ballmaske mit Farbsegmentierung

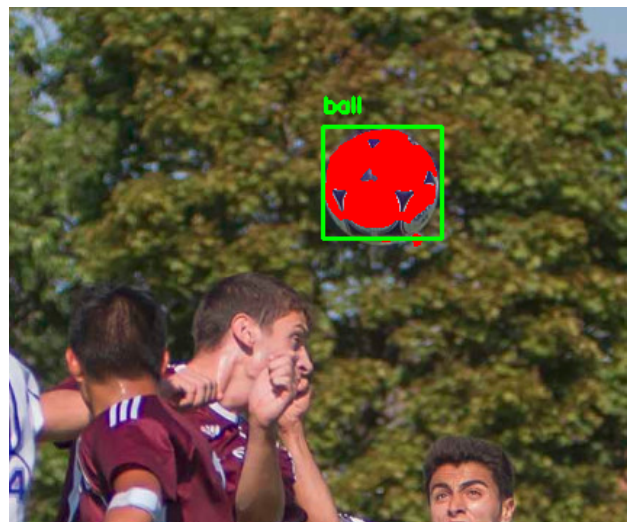


Abbildung 2: Ball Overlay mit Farbsegmentierung

3.3.2 KMeans-Segmentierung

Bei dieser Methode wird das Bild zunächst mit Filtern vorbereitet, bevor der KMeans-Algorithmus auf die Bilddaten angewendet wird, um die Farben zu segmentieren. Diese Methode liefert gute Ergebnisse, solange die Filter und Parameter für das jeweilige Bild passend gewählt werden. Sie ist jedoch nicht universell anwendbar, da sie stark von der Farbgebung des Balls abhängt.

- Filterung:
 1. Das Bild wird mit einem Gaussian-Blur verschwommen gemacht.
 2. Der KMeans-Algorithmus wird zur Segmentierung angewendet.
 3. Das Bild wird in Graustufen umgewandelt.
 4. Ein Schwellenwert wird auf das Graustufenbild angewendet, um es in ein binäres Schwarz-Weiß-Bild umzuwandeln.
 5. Morphologische Filter (Open/Close) werden eingesetzt, um Artefakte zu entfernen.



Abbildung 3: Ballmaske mit KMeans-Segmentierung

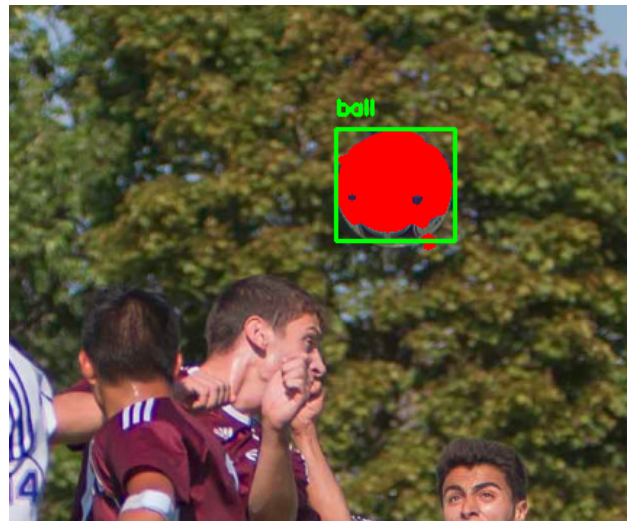


Abbildung 4: Ball Overlay KMeans-Segmentierung

3.3.3 Kantenerkennung mit Canny- und Sobel-Edge-Detektoren

Diese Methode liefert keine brauchbaren Ergebnisse und wird daher nicht weiter verfolgt. Die Kantenerkennung konnte keine zufriedenstellende Segmentierung des Balls erreichen.

3.3.4 Kreiserkennung mit HoughCircles

Diese Methode liefert ausgezeichnete Ergebnisse. Die Bälle werden unabhängig von ihrer Farbe oder etwaigen Spiegelungen zuverlässig erkannt. Dies wird durch die Anwendung der Hough-Transformation zur Kreiserkennung erreicht.

- Filterung:
 1. Das Bild wird in Graustufen umgewandelt.
 2. Die `HoughCircles`-Funktion erkennt Bälle im Bild.
 3. Eine binäre Weiß-Schwarz-Maske wird für die erkannten Kreise erstellt.



Abbildung 5: Ballmaske mit HoughCircles

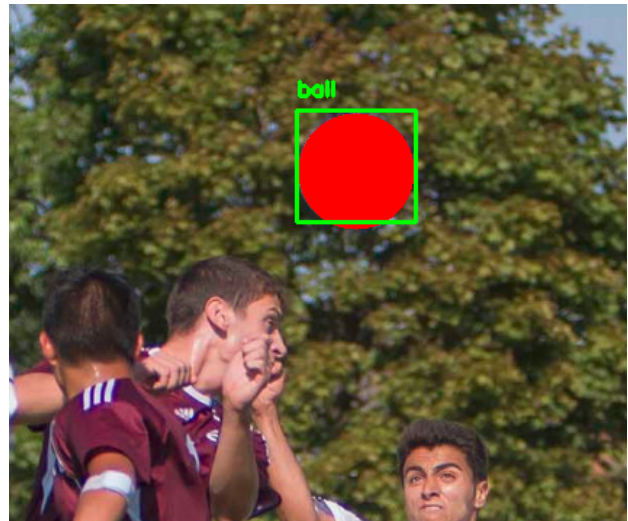


Abbildung 6: Ball Overlay mit HoughCircles

3.4 Ball im Bild segmentieren

Nach der Erstellung der Ball-Maske wird diese auf das Originalbild angewendet, um den Ballbereich zu kennzeichnen. Dies geschieht durch ein Overlay der Maske mit einer roten Farbe, um den Ball visuell hervorzuheben.

```
1 image_ball_box = image[y_min:y_max, x_min:x_max].copy()
2 overlay_color = np.full_like(image_ball_box, (0, 0, 255), dtype=np.uint8)
3
4 mask_indices = ball_mask > 0
5 image_ball_box[mask_indices] = ((1 - alpha) * image_ball_box[mask_indices] + alpha *
  ↳ overlay_color[mask_indices]).astype(np.uint8)
```



Abbildung 7: Ergebnis mit Farbsegmentierung Abbildung 8: Ergebnis mit KMeans-Segmentierung Abbildung 9: Ergebnis mit HoughCircles

3.5 Zusammenfassung Implementierung

Die Segmentierung der Bälle wurde erfolgreich umgesetzt, indem verschiedene Ansätze zur Farbbasierenden Segmentierung getestet und miteinander verglichen wurden. Die Kreiserkennung mittels `HoughCircles` hat sich als die zuverlässigste Methode erwiesen, um die Bälle präzise zu segmentieren, unabhängig von ihrer Farbe und Position im Bild.

4 Statistische Analyse

In dieser Analyse werden die YOLO-basierten Bounding Boxes und die durch Segmentierung ermittelten Ball-Zentren verglichen. Für beide Methoden werden die Schwerpunkte berechnet und die Abweichung zwischen den Ergebnissen statistisch ausgewertet.

4.1 Berechnung des Schwerpunkts

Für die YOLO-Bounding-Box wird der Schwerpunkt als Zentrum der Box berechnet:

$$x_{\text{YOLO}} = \frac{\text{width}_{\text{box}}}{2}, \quad y_{\text{YOLO}} = \frac{\text{height}_{\text{box}}}{2}$$

Die Hough-Transformation liefert bereits den Schwerpunkt des erkannten Kreises direkt, sodass hier keine weitere Berechnung erforderlich ist.

4.2 Statistische Auswertung

Die Euclidische Distanz zwischen den Schwerpunkten der YOLO-Bounding-Box und den Hough-Zentren wird berechnet und für jede Testbildpaarung gespeichert. Die statistische Analyse umfasst die Berechnung des Mittelwerts, der Standardabweichung und des Medians der Distanzen:

- Mittelwert: 1.9946 Pixel
- Standardabweichung: 1.0312 Pixel
- Median: 1.4977 Pixel

Die statistische Auswertung zeigt eine mittlere Abweichung von etwa 2 Pixeln zwischen den Schwerpunkten der YOLO Bounding Boxen und den Schwerpunkten aus der Hough-Transformation.

Die Ergebnisse der Analyse werden durch Bilder veranschaulicht, auf denen die Schwerpunkte eingezeichnet sind. Der rote Punkt stellt den YOLO-Schwerpunkt dar, der grüne Punkt den Schwerpunkt aus der Hough-Transformation.

Ball	YOLO Centroid	Hough Centroid	Distanz	Abbildung
1	(62.5, 61.0)	(62.0, 62.0)	1.12	10
2	(61.5, 61.5)	(61.0, 60.0)	1.58	11
3	(42.5, 43.0)	(41.0, 40.0)	3.35	12
4	(61.0, 61.0)	(60.0, 60.0)	1.41	13
5	(51.0, 48.5)	(51.0, 52.0)	3.50	14
6	(78.0, 76.0)	(78.0, 75.0)	1.00	15

Tabelle 1: Ergebnisse der Analyse der Schwerpunkte und Distanzen



Abbildung
Schwerpunkte Ball 1



10: Abbildung
Schwerpunkte Ball 2



11: Abbildung
Schwerpunkte Ball 3

12:



Abbildung
Schwerpunkte Ball 4



13: Abbildung
Schwerpunkte Ball 5



14: Abbildung
Schwerpunkte Ball 6

15: