

Tutorium 4

Tags	
Erstellt	@11/11/2024

Kriterien zur Klassifizierung

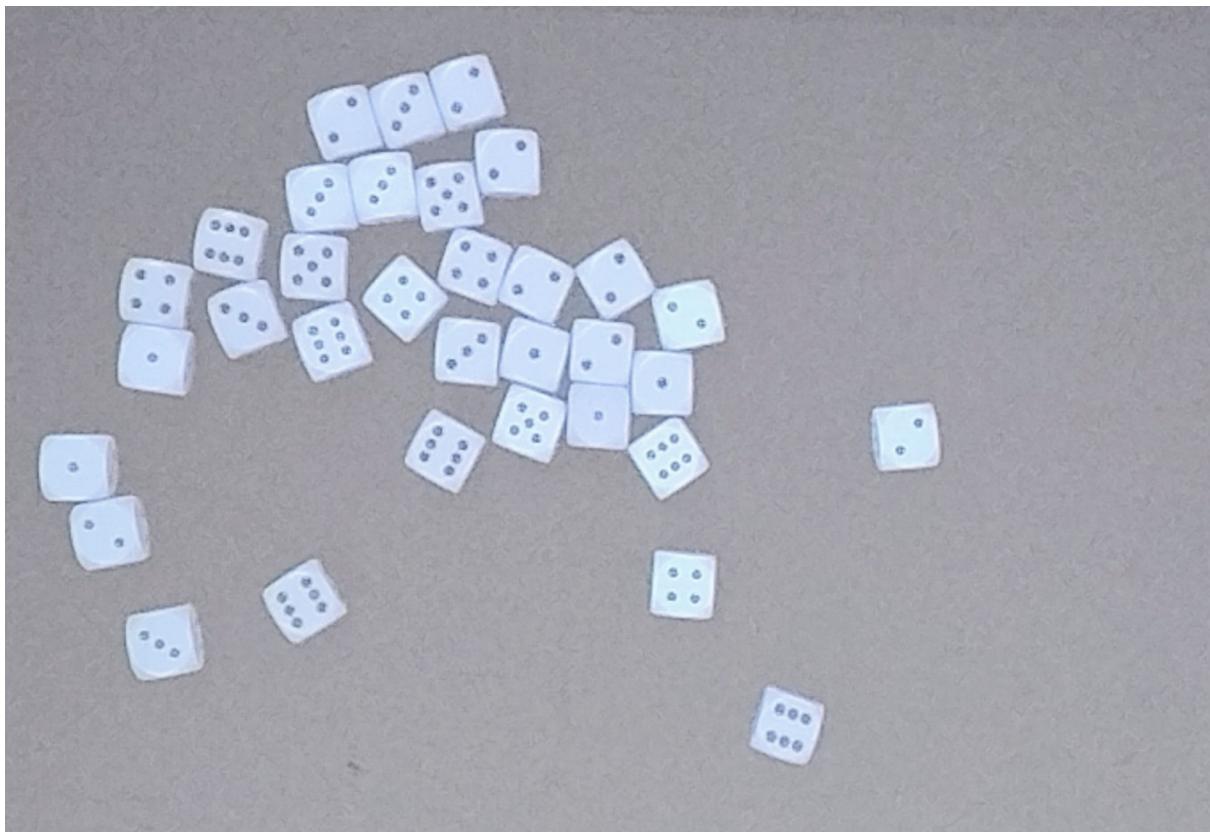
- Rechteck drum herum liegen → Länge und Breite
- Messer und Gabel unterscheiden: Gabel hat Zacken und Messer ist gleichförmig
 - Möglichkeit 1: Pixelweise gegen eine Vorlage vergleichen, muss dabei dann aber exakt passen und funktioniert nicht gut generalisiert
 - Möglichkeit 2: Sonderregel für die Zinken
 - Linie über Zinken/Messerspitze legen, bei häufigem schneiden Gabel, ansonsten Messer
 - Außerdem: Kriterien um Fehlerregionen rauszuschmeißen



Segmentierungsbasierte Bildverarbeitung

- Musterlösungen der Beispiele
 - ...
- ...

Beispiel Würfel



- Wie kann man Würfel zählen?
- Kantenerkennung funktioniert nicht so gut, da viele Phantomkanten
- Nur die Punkte anschauen, man merkt sich welches Muster die Augen haben um davon wieder auf die Augenzahlen zurückzuschließen
- Schwarze Flächen rausfiltern und einen Schwellwert um die weißen Würfel vom schwarzen Hintergrund zu trennen und dann einen Parameter festlegen um die großen schwarzen Flächen rauszukriegen und nur die Augen überbleiben.
- Flächen Filtern und in Binärbaum speichern

- Wenn jetzt die Punkte der Augen vorliegen:

Für $a_1 \in Auge$:

Für $a_2 \in Auge$:

Für $d \in \{1, \dots, 6\}$:

x, y, Θ :

$anpassen(a_1, a_2, form[d])$

;

Für f in $form[d]$:

Suche

Transformation

(x, y, Θ, f) in Auge:

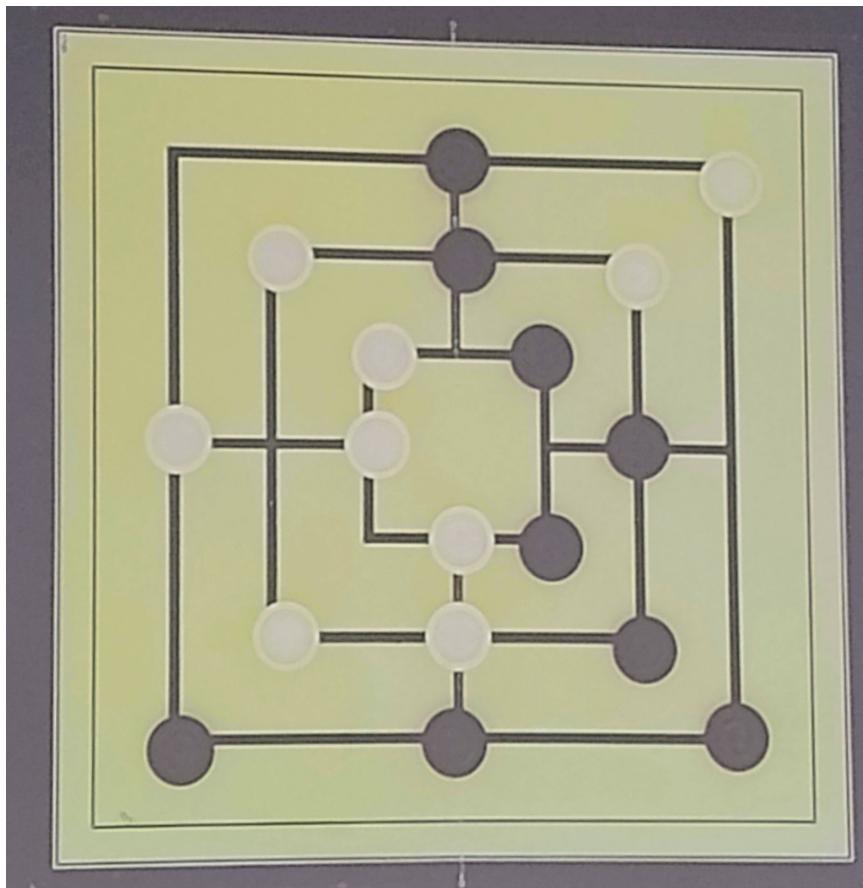
Wenn nicht: siehe
weiter

Wenn gefunden
aus Auge löschen



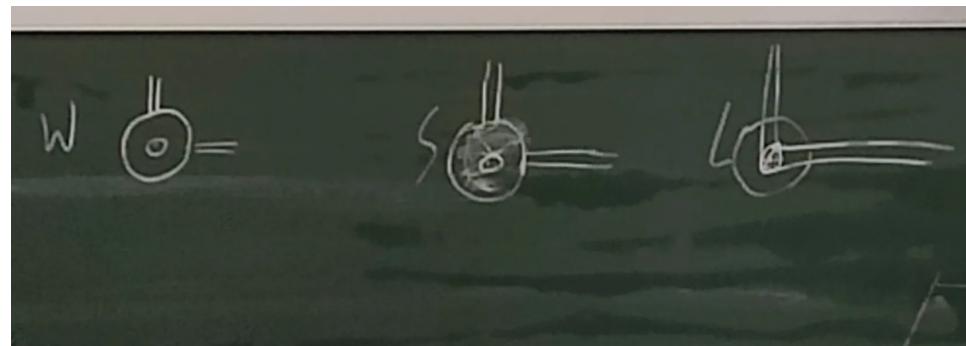
Die Kreuze sind jeweils die Würfelaugen mit
 a_1, a_2, \dots den Nummerierungen der Augen

Mühle Beispiel

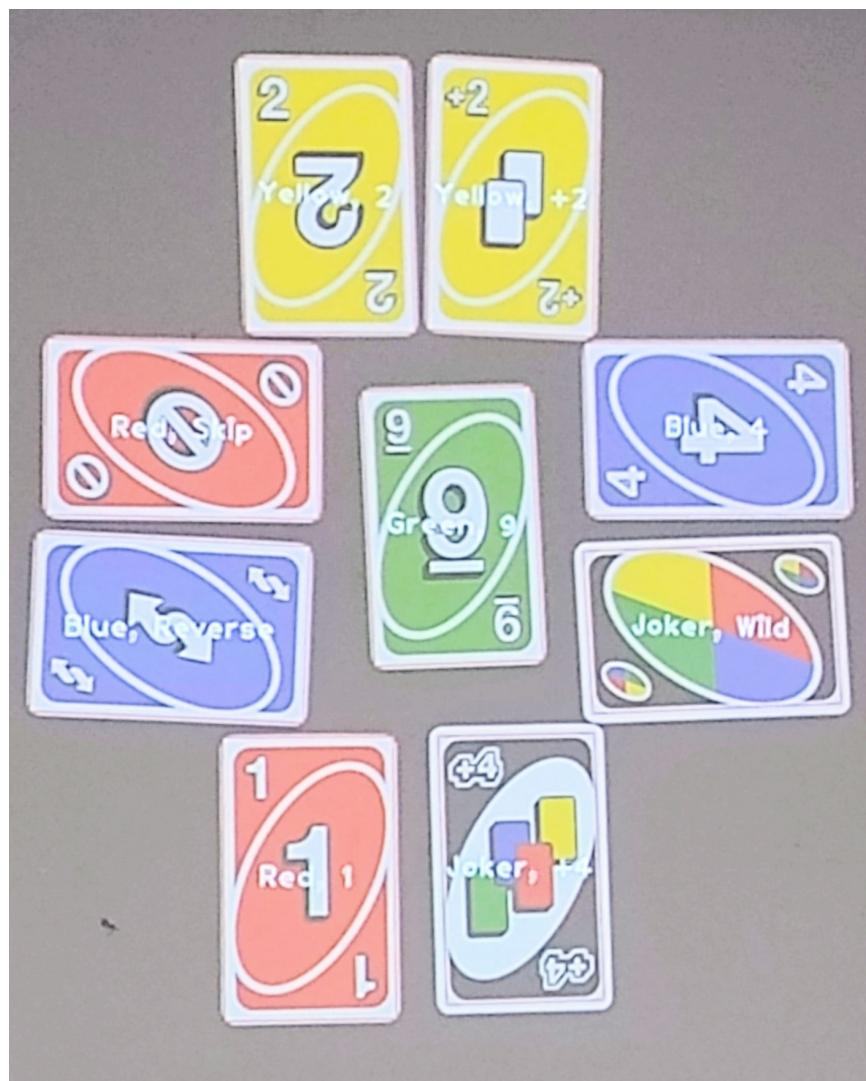


Der aktuelle Spielstand soll erkannt werden.

- Eine Option: Linien anschauen, da wo Linie fehlt liegt der weiße Stein.
- Binärbild erstellen, die Linien und die Koordinaten der Eckpunkte sind bekannt. Wenn in der Ecke weiß ist, liegt da ein weißer Stein. Wenn die Ecke schwarz ist, muss der Durchmesser angesehen werden. Ist er klein ist es bloß die dunkle Farbe der Ecke, wenn er größer ist, liegt dort ein schwarzer Stein.
 - Statt nur an einem Pixel nach Farbe zu suchen: einen größeren Kreis um den Punkt ziehen. Wenn fast nur weiß: Weißer Stein, wenn fast nur schwarz: Schwarzer Stein, wenn relativ gemischt: Linie



Kartenspiel Beispiel



- Helligkeitsschwellwert
- Regionen 1. Ebene

- Rechteck
- HSV
 - Hue, Saturation, Value
- Kreis in Karte erkennen (Regionen + Rechteck)
- linke/obere Ecke
 - Kontur + pixelweiser Vergleich mit Template
 - Ziffern als Vorlage einspeichern und dann pixelweise vergleichen welche Ziffer auf der Karte steht
- Durchschnittsfarbe außerhalb des Kreises → Schwellwerte

Musterlösung:

- Annahme: Karten berühren sich nicht
- Karten finden durch
 - Farbsegmentierung auf Schwarz
 - Regionenbildung
 - Filtern
 - Minimal umschließendes Rechteck
- Karten einzeln erkennen
 - Anzahl rote, grüne, blaue, gelbe, weiße Pixel zählen
 - Daran Kartenfarbe bzw. Joker erkennen
 - Farbsegmentierung auf Weiß
 - Regionenbildung
 - Filtern
- Symbole einzeln erkennen
 - Minimal umschließendes Rechteck
 - Pixelweiser Vergleich mit Vorlagebild

Spargelklassifikation

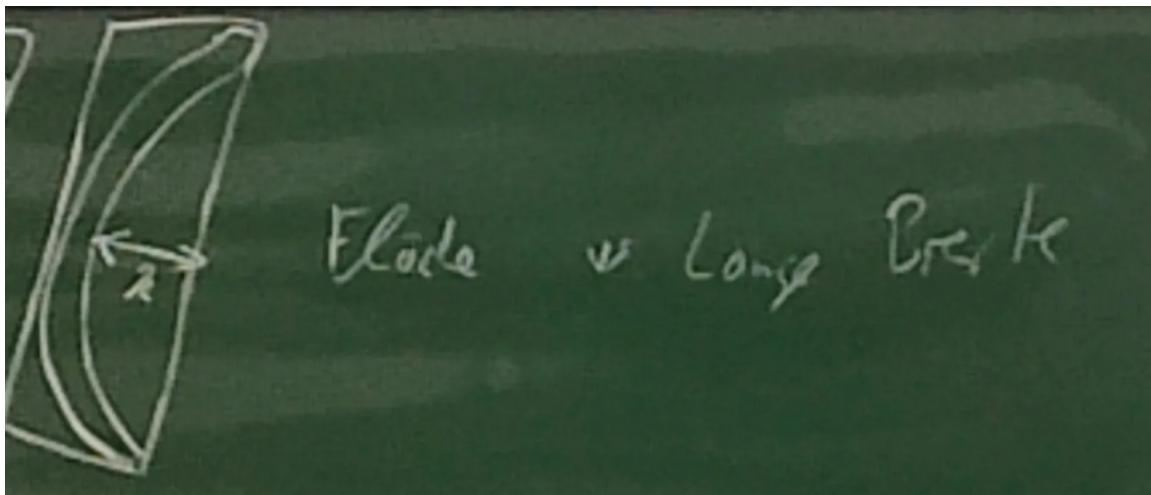
Spargelklassifikation

- Klasse extra
 - gerade (max 5mm)
 - unbeschädigt
 - geschlossene Köpfe
 - 12-16mm, 17-22cm
 - keine Verfärbungen
- Klasse I
 - leicht gebogen (max 10mm)
 - geschlossene Köpfe
 - 10-16mm, 12-22cm
 - Verfärbungen zulässig
- Klasse II
 - stärker gebogen (max 20mm)
 - Köpfe etwas geöffnet
 - >8mm, >12cm
 - stärkere Verfärbung



Wie könnte man den Spargel mit einer Maschine sortieren?

- Möglichst einzeln auf ein schwarzes Band legen
- Rechteck drum herum liegen
 - Vergleich dere Fläche (Anzahl schwarzer Pixel) vs. $Länge \cdot Breite$
 - Länge mal Breite minus Anzahl schwarzer Pixel verrechnen mit breitestem Abstand k zur Rechteckkante vom Spagel um auf Krümmung zu kommen



- Linien über Spargelkopf legen, wenn Linie den Kopf mehrfach Kreuzt muss er geöffnet sein, da er wenn er geschlossen ist nur zwei mal eine Linie gekreuzt werden würde
- Oder eine Linie durch den Spagel legen und an ein Polynom dritten Grades anpassen um die Krümmung herauszufinden