

WiSe 2025/26

02.12.2025

Praktische Übung: Computer Vision für medizinische und industrielle Anwendungen

Versuch 6: Shape Context

Aufgabe 1 - Shape Context als Formmerkmal

- a) Vervollständigen Sie die Funktion

```
scCompute(P, X, nBinsTheta, nBinsR, rMin, rMax),
```

die bei einer Menge X von gegebenen Punkten $x_i \in \mathbb{R}^2$ den Shape Context Deskriptor im Punkt $P \in \mathbb{R}^2$ bestimmt. Hierbei wird ein zweidimensionales (r, θ) -Histogramm aufgebaut, das die relative Lage der Punkte X zu P beschreibt.

Der Winkel $\theta_i \in [-\pi, \pi]$ beschreibt die Richtung des Vektors $x_i - P$. Der Bereich von $[-\pi, \pi]$ wird dazu in `nBinsTheta` gleichmäßig verteilte Bins unterteilt und Punkte mit ähnlicher Richtung entsprechend zusammengefasst.

Der Abstand $r_i = \|x_i - P\|_2$ wird in `nBinsR` logarithmisch verteilte Bins von `rMin` bis `rMax` aufgeteilt, d.h. die Logarithmen der Grenzen der Bins sind gleichmäßig verteilt. Punkte mit Abstand r kleiner als `rMin` oder größer als `rMax` werden ignoriert.

- b) Testen Sie Ihre Funktion an Bildern von handgeschriebenen Ziffern aus der MNIST-Datenbank (<http://yann.lecun.com/exdb/mnist/>). Die Ziffern sind in Bezug auf Größe und Lage im Bild normiert. Einige Beispiele aus dieser Datenbank können aus der Datei `ShapeContextData.mat` geladen werden. Die gegebenen Daten enthalten jeweils 50 Trainings- und Testbilder der Größe 28×28 Pixel und die zugehörigen Label. Die Bilder sind in der dritten Dimension “gestapelt”, d.h. für das i -te Bild gilt `img = train_data(:, :, i)`.

Die Bilder können mit der gegebenen Funktion `getEdgePoints.m` in eine Menge von normierten (Rand-)Punkten umgewandelt werden. Für P kann z.B. der Mittelpunkt (0.5, 0.5) verwendet werden.

Stellen Sie das Bild einer Ziffer, die Menge der bestimmten Randpunkte und den Shape Context Deskriptor (das Histogramm) in jeweils einem eigenen `subplot` dar.

- c) Vervollständigen Sie die Funktion `histCost(sc1, sc2)`, die zwei Histogramme p und q per χ^2 -Test vergleicht. Es gilt

$$\chi^2(p, q) = \frac{1}{2} \sum_{i,j} \frac{(p_{ij} - q_{ij})^2}{p_{ij} + q_{ij}}$$

wobei p_{ij} und q_{ij} die Bins der entsprechenden Histogramme angeben. Achten Sie darauf nicht durch Null zu teilen, in diesem Fall erhalten Sie in MATLAB das Ergebnis `NaN` (not a number).

Vergleichen Sie die Histogramme verschiedener Beispiele miteinander, um die Funktionalität Ihrer Funktion zu verifizieren.

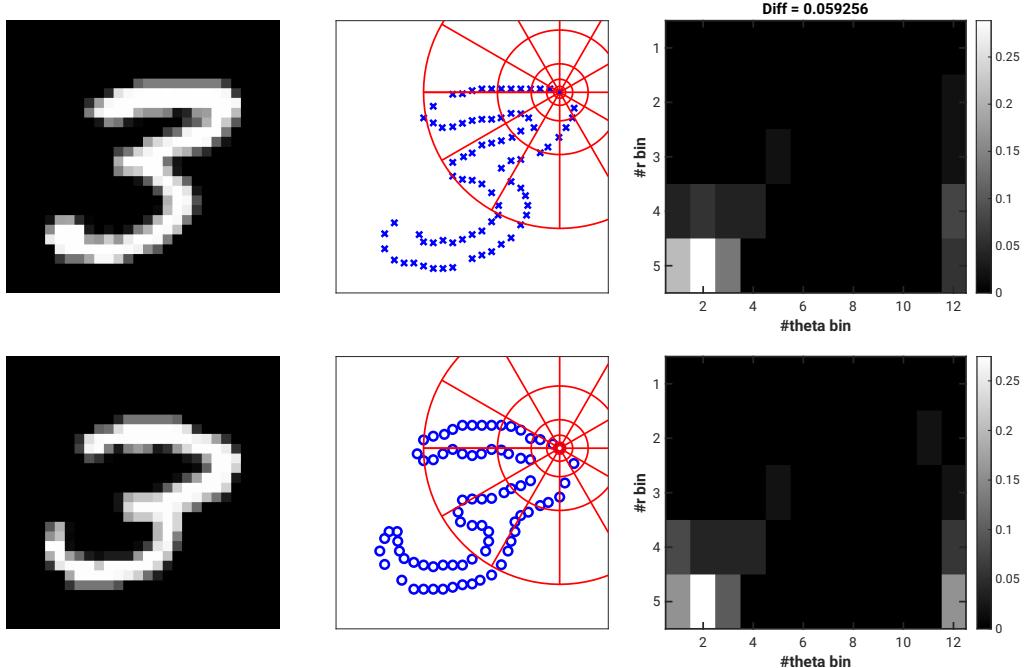


Abbildung 1: Beispielhafte Visualisierung aus Aufgabenteil b) mit dem Vergleich in c)

Aufgabe 2 - Shape Context zur Klassifikation

Im Folgenden wird der Shape Context Descriptor zur Bestimmung von handgeschriebenen Ziffern mit einem Nearest-Neighbor-Klassifikator verwendet. Hierzu wird aus den Trainingsbeispielen die ähnlichste Ziffer auf der Basis des Shape Context bestimmt. Die Testziffer wird dann dieser Klasse zugeordnet.

- a) Das Skript `scClassification.m` setzt die Klassifikation der Ziffern bereits um. Sie müssen lediglich geeignete Parameter für den Shape Context wählen. Machen Sie sich mit dem Skript vertraut.
- b) Bisher wurde lediglich der Shape Context für einen Punkt in der Bildmitte bestimmt und zur Klassifikation verwendet. Die Klassifikation soll nun so erweitert werden, dass alle Punkte verwendet werden.
- Erweitern Sie Ihre Funktion `scCompute`, so dass eine beliebige Menge an Punkten $p_i \in \mathbb{R}^2$ übergeben werden kann. In jedem Punkt p_i soll der Shape Context auf der Basis der Punkte in X bestimmt werden.
Hinweis: Die gegebenen Bestandteile in der Funktion verarbeiten bereits beliebige Punktmengen P .
 - Erweitern Sie zudem ihre Funktion `histCost`, um zwei Mengen P und Q an Histogrammen zu vergleichen. Verwenden Sie das Distanzmaß

$$D(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \chi^2(p, q) + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \chi^2(p, q).$$

- c) Nun kann die Klassifikation der handgeschriebenen Ziffern erweitert werden, indem nicht nur der Shape Context im Bildmittelpunkt sondern von allen Randpunkten bestimmt und verglichen wird.

Kopieren und erweitern Sie dazu das Skript `scClassification.m`, indem Sie die entsprechenden Stellen geeignet anpassen, d.h. anstatt `scCompute([0.5, 0.5], X, ...)` verwenden Sie `scCompute(X, X, ...)`. Achten Sie auch darauf, dass Sie die Datenstrukturen zur Speicherung geeignet anpassen.

Klassifizieren Sie die Ziffern mit den gleichen Parametern wie in a).

Fehlklassifikationen: 15 von 50										
	0	1	2	3	4	5	6	7	8	9
erkannte Ziffer	3	0	1	1	0	1	1	0	0	0
0	0	5	0	0	0	0	1	0	1	1
1	0	0	0	0	0	0	0	0	0	0
2	0	0	3	0	0	0	0	0	0	0
3	0	0	1	4	0	2	0	0	0	0
4	0	0	0	0	6	0	0	0	0	0
5	0	0	0	1	0	3	0	0	0	0
6	1	0	0	0	0	0	2	0	0	0
7	0	0	0	0	1	0	0	3	0	0
8	0	0	0	0	1	0	1	0	2	0
9	0	0	0	0	0	0	0	0	0	4

Fehlklassifikationen: 10 von 50										
	0	1	2	3	4	5	6	7	8	9
erkannte Ziffer	3	0	0	1	0	1	1	0	0	0
0	0	5	0	0	0	0	1	0	1	0
1	0	0	4	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	1	4	0	1	0	0	0	0
4	0	0	0	0	7	0	0	0	0	0
5	1	0	0	1	0	4	0	0	0	0
6	0	0	0	0	1	0	3	0	0	0
7	0	0	0	0	0	0	0	3	0	0
8	0	0	0	0	0	0	0	0	2	0
9	0	0	0	0	0	0	0	0	0	5

Abbildung 2: Konfusionsmatrizen der Klassifikationen aus a) und c)

Kontrollfragen

- a) Warum wird der χ^2 -Test zum Vergleich zweier Histogramme verwendet? Wann liefert der Test den Wert 0 bzw. 1?
- b) Was beschreibt die Konfusionsmatrix? Welche Ziffern sind einfach zu klassifizieren, welche Ziffern sind schwieriger zu unterscheiden? Warum?
- c) Beschreiben Sie das Distanzmaß $D(P, Q)$ aus Aufgabe 2 b) ii).
- d) Ist der Shape Deskriptor invariant gegenüber Translation, Rotation und Skalierung, d.h. können Objekte in beliebiger Lage und Größe verglichen werden? Falls nicht, wie kann eine Invarianz erreicht werden?