

Aufgabenblatt 3.1

Praktikum Computer Vision
SoSe 2018

Christian Wilms

19. April 2018

Aufgabe 1 — Einfache Klassifikation

1. Ladet euch zunächst den Datensatz aus dem CommSy herunter und folgt den Anweisungen zum Einbinden und Nutzen.
2. Berechnet zu jedem Bild den Mittelwert als Merkmal.
3. Nutzt das Merkmal/die Merkmale als Deskriptor und findet zu jedem Bild aus den Validierungsdaten das am besten passende Bild aus den Trainingsdaten und speichert das entsprechende Label als Vorhersage. Vergleicht dazu die jeweiligen Deskriptoren und berechnet den Abstand mit der L_2 -Distanz (Euklidische Distanz).
Tipp: Wenn der Deskriptor nur ein Merkmal hat, reicht als Abstand der Absolutbetrag der Differenz.
4. Evaluert eure Ergebnisse, indem ihr die Vorhersagen zu den Validierungsdaten mit den eigentlichen Labels aus dem Datensatz vergleicht. Wie viel Prozent der Zuordnungen stimmen? Ab wann ist das Ergebnis als akzeptabel einzustufen?
5. **Zusatzaufgabe:** Nutzt auch die Standardabweichung und kombiniert beide Merkmale zu einem Deskriptor. Bringt die gewichtete Distanzberechnung einen Vorteil?

Aufgabe 2 — Klassifikation mit Histogrammen

Erweitert die Berechnung aus Aufgabe 1, sodass nur das Histogramm als Merkmal genutzt wird. Welche Anzahl an Behältern ist optimal in diesem Beispiel?

Aufgabe 3 — Ergebnisse genauer auswerten

Erstellt eine sog. Confusion Matrix, wie in Tabelle 1. In jeder Zeile wird dabei dargestellt, welcher Klasse die Bilder einer Klasse zugeordnet wurden (bspw. welches Label haben die Autobilder erhalten). Im Beispiel in Tabelle 1 ist erkennbar, dass Autos recht oft als solche klassifiziert wurden (50%). Schiffe hingegen wurden recht häufig für Hirsche (5 mal) und gelegentlich für Autos (2 mal) gehalten.

Tabelle 1: Confusion Matrix mit Beispielwerten.

	Auto vorhergesagt	Hirsch vgs.	Schiff vgs.
eigtl. Auto	5	3	2
eigtl. Hirsch	1	5	4
eigtl. Schiff	2	5	3

Aufgabe 4 — Zusatzaufgabe: Schleifenloser Code

1. Schreibt den Code für das Merkmal Mittelwert völlig ohne Schleifen, Rekursionen, etc. Nutzt nur die Funktionen von Numpy.
2. Nutzt die in der Bibliothek scikit-learn (sklearn) vorhandene Implementation des NN-Klassifikators `sklearn.neighbors.KNeighborsClassifier`. Achtet auf die korrekte Übergabe der Parameter.
3. Vergleicht die Laufzeiten der Klassifikation der drei Varianten für das Merkmal Mittelwert.