

Aufgabenblatt 3.1

Praktikum Computer Vision
WiSe 2023/24

Christian Wilms

Auf diesem Aufgabenblatt bietet es sich an, eine ipynb-Datei, also ein Notebook, für die Aufgaben 1 und 2 zu verwenden. Die Ergebnisse der Teilaufgaben sollten in Kommentaren oder Markdown-Zellen notiert werden.

Aufgabe 1 — Einfache Bildklassifikation

1. Ladet die Trainings- und Testdaten aus dem Moodle herunter und folgt den Anweisungen zum Einbinden.
2. Berechnet zu jedem Trainings- und Testbild jeweils den Mittelwert als Merkmal.
3. Findet nun zu jedem Testbild das ähnlichste Trainingsbild mittels der vorher berechneten Mittelwerte und der euklidischen Distanz. Ermittelt dazu die Distanz eines Testbilds zu jedem Trainingsbild.
Tipp: Wenn das Merkmal nur ein Skalar ist wie hier, reicht als euklidische Distanz der Betrag der Differenz.
4. Speichert je Testbild das Label des ähnlichsten Trainingsbildes (kleinste Distanz) als Vorhersage.
5. Evaluert euer System, indem ihr die Vorhersagen zu den Testbildern mit den eigentlichen Labels der Testbilder aus dem Datensatz vergleicht. Wie hoch ist die Trefferquote und ab wann ist das Ergebnis als akzeptabel einzustufen?
Tipp: Im Moodle findet ihr eine Spoiler-Datei mit den richtigen Ergebnissen.
6. **Zusatzaufgabe:** Nutzt nun anstelle des Mittelwert das Histogramm des Bildes als Merkmal. Spätestens nun müsst ihr zur Berechnung des Abstands die euklidische Distanz verwenden. Welche Anzahl an Behältern führt zu den besten Ergebnissen in diesem Beispiel? Nutzt nur 2er-Potenzen für die Anzahl an Behältern, um eine Überanpassung zu vermeiden.
7. **Zusatzaufgabe:** Nutzt nun die Standardabweichung je Bild und kombiniert sie mit dem Mittelwert zu einem Merkmalsvektor. Wie verändern sich die Ergebnisse? Ist die einfache Konkatenation der Merkmale sinnvoll?

Tabelle 1: Confusion Matrix bei der Verwendung des Mittelwerts als einzigem Merkmal.

	Auto vorhergesagt	Hirsch vgs.	Schiff vgs.
eigtl. Auto	1	7	2
eigtl. Hirsch	1	2	7
eigtl. Schiff	4	3	3

Aufgabe 2 — Zusatzaufgabe: Ergebnisse genauer auswerten

Erstellt eine sog. Confusion Matrix, wie in Tabelle 1. Die Confusion Matrix dient der genaueren Auswertung von Fehlern, indem diese den falschen Klassen zugeordnet werden. In jeder Zeile wird dabei dargestellt, welcher Klasse die Testbilder einer Klasse zugeordnet wurden (bspw. welches Label haben die Autobilder erhalten). Im Beispiel in Tabelle 1 wurden Schiffe von den drei Klassen am häufigsten als solche klassifiziert wurden (3 mal). Autos hingegen wurden recht häufig für Hirsche (7 mal) und gelegentlich für Schiffe (2 mal) gehalten. Hinweis: Die Darstellung der Confusion Matrix ist für diese Aufgabe irrelevant.

Aufgabe 3 — Zusatzaufgabe: Schleifenloser Code

Schreibt den Code aus Aufgabe 1 für das Merkmal Mittelwert völlig ohne Schleifen, Rekursionen, `map`, `np.vectorize` etc. Nutzt dazu die Funktionen von Numpy. Wird die Bildklassifikation dadurch schneller? Skaliert eure Lösung auch für große Datensätze?

Hinweis: Für die Zeitmessung in Sekunden reicht uns die Verwendung der Funktion `time.time()`. Stoppt die Zeit über 1000 Durchläufe.