

Übung 03

CNN mit **Keras** in **Python**: Erkennung mit eigenen Bildern

INFI-IS

5xHWII

October 9, 2025

Abgabetermin: lt. Moodle
Übungsleiter: Albert Greinöcker



Ziel der Übung:

- Kennenlernen CNN in Keras
- Aufnahme von Bildern mit OpenCV in Python
- Exportieren und Einbauen des trainierten Modells
- Wichtigkeit der Anzahl und Qualität von Bildern für ein Machine Learning Projekt
- Anwenden von bereits vortrainierten Modellen (z.B. ResNet)

1 Vorbereitung

Installation

- KERAS sollte schon für Übung 02 installiert sein
- Die Installation von OpenCV (Open Source Computer Vision Library) kann unter Windows Probleme machen, aber eigentlich müsste diese Installationsanweisung reichen: `pip install opencv-python`. Der Import sieht allerdings anders als bei der Installation aus, nämlich: `import cv2`.

Die Skripte beinhalten Angaben zu lokalen Verzeichnissen die z.B. für das Speichern der Testdaten benötigt werden. Diese müssen natürlich angepasst werden.

2 Aufnehmen von Testbildern

In unseren gemeinsamen Beispielen zu Machine Learning befindet sich ein Programm `capture_images.py`, dass für die Erzeugung von Testdaten verwendet werden kann. Hier einfach die oberen Parameter ändern, dann sollten automatisch Fotos gemacht und in den entsprechenden Ordnern abgelegt werden.

Es besteht natürlich - abhängig vom Anwendungsfall - die Möglichkeit auch bereits bestehende Bildsammlungen zum Trainieren zu verwenden (eine gute Quelle für Bilddaten ist z.B. <https://kaggle.com>)

3 Austausch und Vorbereiten von Bildern

Es besteht auch die Möglichkeit gemeinsam Bilder zu generieren und diese für das Training eines einzigen Modells zu verwenden

Um ein wenig Speicherplatz zu sparen kann man die Images kleinerrechnen, dafür kann man diese Anweisungen in einer Schleife laufen lassen sodass alle konvertiert werden:

```
1 img_small = np.array(Image.open('img/htl-logo.png').resize((180,180)))  
2 Image.fromarray(img_small).save('img/htl-logo-small.png')
```

4 Einlesen und Lernen aus den Daten

Ein weiteres Programm zum Lernen mit den aufgenommenen Bildern liegt auch bei den Beispielen unter: `ex_04_cnn_loaded_images.py`

Hier einfach das Verzeichnis (und ggf. Parameter am Model) ändern, dann sollte das Laden der Bilder, Lernen und Vorhersage eigentlich schon klappen.

Das Modell soll so wie in den Kommentaren am Ende gespeichert werden.

5 Einbauen der Vorhersage

Das oben angeführte Beispiel beinhaltet auch die Vorhersage. Diese soll in ein weiteres Programm integriert werden das folgendes macht:

- Laden des erstellten und trainierten Modells
- Ein Programm wie beim Erstellen der Testbilder, dass die Kamera-Frames einliest soll erstellt werden
- Die eingelesenen Frames sollen mittels KI-Modell überprüft werden ob die gelernten Gesten (z.B. Schere, Stein oder Papier) in die Kamera gedeutet wird (das checkt man indem man sieht ob entsprechende Wahrscheinlichkeiten für die einzelnen Kategorien berechnet wurden)
- Falls eine Geste (z.B. Schere, Stein oder Papier) erkannt wurde soll dies auf der Konsole ausgegeben werden.

6 Bestehende Modelle

Für diese Anforderungen gibt es natürlich schon vorberechnete Modelle. Eines das gut funktioniert ist ResNet (siehe dazu Beispiel `ex_04_existing_cnns`). In diesem Beispiel sind nur die predictions drin, man muss allerdings auch dieses Modell mit den eigenen Daten neu trainieren. Diesen Teil kann man aber aus dem Teil mit dem eigenen CNN nehmen.