

Grundlagen KI Projekt persönlicher Report

Alina Simon

Technische Universität Clausthal, Clausthal-Zellerfeld 38678, Deutschland

1 Aufgaben

Meine Aufgabe in der Gruppe war es, eine Pipeline zu entwickeln, die aus einem Foto abliest, welche Verkehrszeichen darauf zu sehen sind. Dabei sollten sowohl die svm models für die Lokalisierung eines Verkehrsschildes auf einem Foto, als auch ein CNN für die Erkennung eines einzelnen Verkehrsschildes, die ein Teil der Gruppe bereits entwickelt hat, genutzt werden. Dafür überlegte ich mir zunächst, wie dieser Prozess aussieht. Meine Überlegungen und Schritte pflegte ich als Überschriften in einem Jupyter Notebook ein, um daraufhin den Code für diese Abschnitte zu schreiben. Ich überlegte mir folgende Struktur:

1. Bildimport
2. Lokalisierung (Models laden, Bild laden, Model ausführen, Zeichen ausschneiden)
3. Erkennung (Bilder vorbereiten, Model laden, Verkehrsschild erkennen, Output)

1.1 Bildimport

Bei dem Bildimport entschied ich mich, sowohl das Bild als auch den Ordner als Variable zu deklarieren, um dieses Notebook an beliebiger Stelle zu nutzen. Da ich Teile des von uns als Gruppe ausgewählten Datensatzes nutzte, um die Pipeline zu testen, musste ich zuerst das ppm format in ein jpg umwandeln.

1.2 Lokalisierung

Um das Model zu laden, brauchte ich die Library **dlib** [1]. Hier stoß ich an ein technisches Problem, denn diese Library kann man, zumindest auf Windows, nicht einfach mithilfe von **pip install** installieren, denn sie braucht einen c++ compiler. Nach einer kleinen Recherche fand ich einen Medium Artikel [2], der mir hilf, dieses Problem zu lösen - auch wenn es zuerst an meiner Geduld scheiterte und ich den Buildvorgang des Wheels für dlib mehrfach abgebrochen hatte, weil ich dachte, dass das doch nicht so lange dauern könnte.

Nachdem ich die Models dann laden konnte und ich mir auch noch einmal das Foto angezeigt habe, mit dem ich arbeitete, habe ich die models auf mein Bild ausgeführt. Dabei habe ich sowohl das gesamte Foto mit den eingezeichneten Bounding Boxes ausgegeben, als auch die jeweiligen erkannten Zeichen in eine Liste gelegt. Zuerst bestand dann diese Liste aus den georteten Zeichen, aber

mit eingezeichneter Bounding Box. Dieses Problem löste ich mit einer zweiten Instanz des Fotos, in das keine Bounding Boxes eingezeichnet werden, aus der ich dann die Boxen Ausschnitt. Das Ergebnis dieses Schrittes war also eine Liste an nicht gelabelten aber ausgeschnittenen Verkehrszeichen.

1.3 Erkennung

Die Schilderkennung fiel mir deutlich leichter als die Lokalisierung. Hier mussten die Verkehrszeichen noch passend für das Model auf 64*64 zugeschnitten werden. Danach habe ich das Model angewendet, das Ergebnis auf eine Klasse gemappt und zusammen mit dem Bild ausgegeben.

1.4 Weiterentwicklung

Neben der bereits erwähnten Geduld, ein package zu installieren habe ich mich auch fachlich deutlich weiterentwickelt. So lernte ich beim Arbeiten mit den Daten die library **open cv** ein wenig kennen. Gerade in Bezug auf Bilder lesen, reformatten, und annotieren. Außerdem lernte ich, wie mit fertigen Modellen umgegangen wird, was für mich ein neuer Einblick war.

Literatur

1. dlib Python Dokumentation, <https://dlib.net/python/index.html>, letzter Zugriff 16.02.2025
2. How to install dlib library for Python in Windows 10, <https://medium.com/analytics-vidhya/how-to-install-dlib-library-for-python-in-windows-10-57348ba1117f>, letzter Zugriff 17.02.2025
3. opencv Python Dokumentation, https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html, letzter Zugriff 17.02.2025