Nummernschilderkennung mit Python

Anne-Sophie Bollmann, Susanne Klöcker, Pia von Kolken, Christian Peters

19. Januar 2021

Inhalt

Einleitung

Einleitung

Ziel: Erkennen von Nummernschildern auf Fotos und Auslesen der Nummernschilder

Arafat et al. (2019):
Systematic review on vehicular licence plate recognition framework in intelligent transport systems [1]

Herausforderungen:

- Vielfältigkeit der Nummernschilder
- Rahmenbedingungen der Bildaufnahme (Beleuchtung)

Beispiel

Abbildung 1: Beschreibung [1]

[1] Bildquelle: `https://github.com/theAIGuysCode/yolov4-custom-functions`

Yolo

Auffassung der Objekterkennung als Regressionsproblem

Charakteristika:

- System zur Objekterkennung
- Generierung potenzieller Bounding Boxes in einem Bild
- Klassifizierung

Extraktion des Nummernschildes

Convolutional Neural Networks

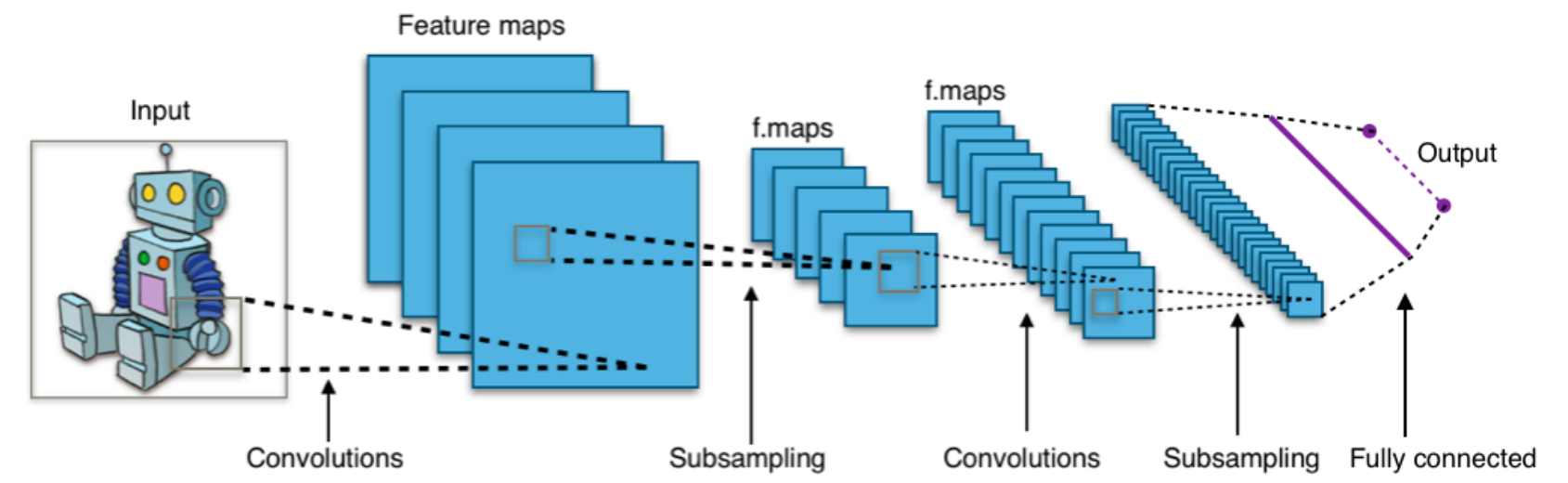

Abbildung 2: Convolutional Neural Network. [2]

Input: Bild mit Auto $\longmapsto$ **Output:** Bounding Box

[2] Bildquelle: `https://de.wikipedia.org/wiki/Convolutional_Neural_Network`

Implementierung

Netzarchitektur:

- Inspiriert durch YOLO (**Y**ou **O**nly **L**ook **O**nce) [3]
- Kann sowohl Klassen als auch Bounding Boxes vorhersagen
  - $\rightarrow$ Wir brauchen nur Bounding Boxes von Nummernschildern, also Vereinfachung nötig

Implementierung:

- Open Source Deep-Learning Bibliothek Keras
- Geschrieben in Python

Nummernschild auslesen

Nummernschild auslesen

Wie können wir Text auf Bildern auslesen?

- Tesseract: freie Software zur Texterkennung mit vielen vorimplementierten Sprachen [5]
- Problem: Text wird größtenteils noch nicht richtig auf den unbearbeiteten Nummernschildern erkannt
- Lösung: das erkannte Nummernschild derart vorverarbeiten (preprocessing), dass das richtige Auslesen der einzelnen Elemente möglichst gut unterstützt wird

Nummernschild vorverarbeiten

- Geeignetes Werkzeug: OpenCV [2]
- OpenCV ist eine plattformübergreifende Bibliothek, für Echtzeit-Computer-Vision-Anwendungen
- beinhaltet Algorithmen für die Bildverarbeitung und im Rahmen von Computer Vision (CV) auch für maschinelles Lernen
- Nutzung für die Verarbeitung des erkannten Nummernschildes (z.B. Thresholding), um die Zeichen besser zu erkennen und richtig auszulesen

Beispiel für die Anwendung von OpenCV

OpenCV wurde bereits auf Nummernschildverarbeitung verwendet:

Abbildung 3: Original [3]

Abbildung 4: Graustufen

[3] Bildquelle: `https://github.com/theAIGuysCode/yolov4-custom-functions`

Beispiel für die Anwendung von OpenCV

Abbildung 5: Thresholding

Abbildung 6: Konturen

Beispiel für die Anwendung von OpenCV

Abbildung 7: Aussortierung

Abbildung 8: Schwarze Schrift auf weißem Hintergrund

Auf das finale Bild (Abbildung 7) wird anschließend Tesseract angewendet, das die Nummern und Buchstaben ausgibt

Validierung der Texterkennung

Validierung:

- Rastersuche über Parametereinstellungen für OpenCV
- Validierung über *character accuracy* [4]:

$$\frac{n - \#errors}{n},$$

  wobei n Anzahl der Zeichen im Datensatz und $\#errors$ Anzahl der fehlerhaft erkannten Zeichen

→ Parametereinstellungen mit höchstem *character accuracy* werden für Texterkennung verwendet

Ausblick:

Definieren der Funktionen

Zusammenführung der Teilmodule

Literatur i

A. et al.
Systematic review on vehicular licence plate recognition framework in intelligent transport systems.
The Institution of Engineering and Technology, 2019.

T. P. I. P. Ltd.
Learn opencv, 2021.

J. Redmon and A. Farhadi.
Yolov3: An incremental improvement.
arXiv, 2018.

J. F. R. Rice, Stephen R. and T. A. Nartker.
The fifth annual test of ocr accuracy.
Information Science Research Institute, 1993.

Literatur ii

R. Smith.

An overview of the tesseract ocr engine.

Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 2:629–633, 2007.