

Nummernschilderkennung mit Python

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

19. Januar 2021

1. Einleitung
2. Extraktion des Nummernschildes
3. OpenCV
4. Tesseract

Einleitung

Ziel: Erkennen von Nummernschildern auf Fotos und Auslesen der Nummernschilder

Herausforderungen:

- Vielfältigkeit der Nummernschilder
- Rahmenbedingungen der Bildaufnahme (Beleuchtung)



Abbildung 1: Beschreibung

Auffassung der Objekterkennung als Regressionsproblem

Herausforderungen:

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

Extraktion des Nummernschildes

Convolutional Neural Networks

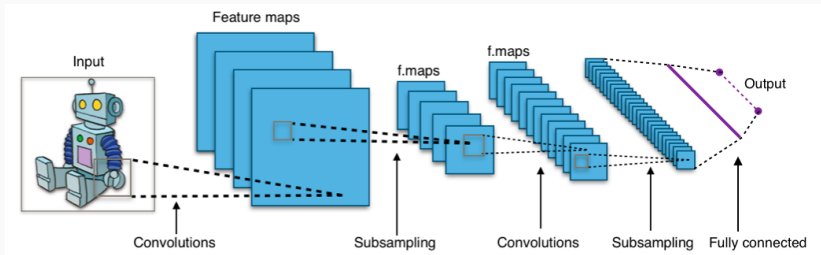


Abbildung 2: Convolutional Neural Network. ¹

Input: Bild mit Auto \mapsto **Output:** Bounding Box

¹Bildquelle: https://de.wikipedia.org/wiki/Convolutional_Neural_Network

Netzarchitektur:

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

Implementierung:

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

OpenCV

Was ist OpenCV?

- 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

Wofür nutzen wir OpenCV?

- 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



Abbildung 4: Graustufen



Abbildung 5: Tresholding



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

Tesseract

- freie Software zur Texterkennung mit vielen vorimplementierten Sprachen
- häufig erprobt für eine Vielzahl von Problemen

Validierung:

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

wobei `#character` Anzahl der Ziffern im Text und `#errors` Anzahl der fehlerhaft erkannten Ziffern

Ausblick:

Definieren der Funktionen

Zusammenführung der Quellcodes

