

End-to-end-Learning Ansatz für autonomes Fahren im Miniatur Wunderland

Nils-Ole Bickel, Michel Brüger

26. Februar 2020

Inhaltsverzeichnis

1	Hardware	2
1.1	Raspberry Pi 4	2
1.2	Google Coral USB Accelerator	2
1.3	Das Auto	2
1.4	Der Trainingsrechner	2
2	Software	2
2.1	Auf dem Raspberry Pi verwendete Software	2
2.1.1	Python	2
2.1.2	Tensorflow lite	2
2.2	Auf dem Auto verwendete Software	2
2.3	Auf dem Trainingsrechner verwendete Software	3
2.3.1	Anaconda	3
3	Das Netz	3
3.1	Implementierung	3
3.2	Training	3
3.2.1	Trainingsdaten	3
3.2.2	Anzahl Epochen	3
3.2.3	3
4	Ergebnisse/Fazit	3

ToDo:

evtl Abstract schreiben

- end-to-end learning erklären

- netzt sagt für übertragene Bilder lenkwinkel voraus

- übermittelte Lenkwinkel als Label, Netz sagt für übertragene Bilder

Lenkwinkel voraus

1 Hardware

1.1 Raspberry Pi 4

Die Hardware in dem Fahrzeug selbst ist nicht stark genug um aus den Kamerabildern den entsprechenden Lenkwinkel zu berechnen. Daher ist das Ziel, die Berechnung der Lenkwinkel auszulagern, auf einen Rechner, der im besten Fall unauffällig im *Miniaturland* "getarnt" werden kann.

Der *Raspberry Pi 4* hat im Verhältnis zu seiner Größe potente Hardware und verfügt vor allem auch über zwei USB-3-Anschlüsse. Diese sind wichtig um das volle Potential aus dem *Google Coral USB Accelerator* herauszuholen. Außerdem bietet er die Möglichkeit vollwertige Linux-Betriebssysteme zu installieren, was die Verwendung der für die AI-Lenkwinkel-Voraussagung notwendigen Software ermöglicht.

Die Möglichkeit, den *Raspberry Pi* über eine Powerbank per USB-C zu laden, macht es noch leichter ihn in der Kulisse zu verstecken, da man nicht noch ein Kabel zu einem Stromanschluss verlegen muss.

1.2 Google Coral USB Accelerator

Der *Google Coral USB Accelerator* ist ein Tensor-Prozessor in Form eines USB-Sticks, der an einen Rechner wie den *Raspberry Pi* angeschlossen werden kann, um die Machine Learning Operationen schnell und besonders energieeffizient durchzuführen. Dies kann der *Raspberry Pi 4* zwar auch alleine bewerkstelligen, der *Google Coral USB Accelerator* ist dabei aber deutlich schneller. Bei spontanen Vergleichen konnten wir eine 14-fache Geschwindigkeit bei der Objekterkennung in Bildern feststellen.

1.3 Das Auto

1.4 Der Trainingsrechner

2 Software

2.1 Auf dem Raspberry Pi verwendete Software

2.1.1 Python

2.1.2 Tensorflow lite

2.2 Auf dem Auto verwendete Software

- irgendwie schickt es einen Videostream...

2.3 Auf dem Trainingsrechner verwendete Software

2.3.1 Anaconda

- Virtual Environments
- Python
- Tensorflow
- Tensorflow lite converter (heisst der so?)

3 Das Netz

3.1 Implementierung

3.2 Training

3.2.1 Trainingsdaten

3.2.2 Anzahl Epochen

3.2.3 ...

4 Ergebnisse/Fazit

- Genauigkeit der Berechneten Lenkwinkel noch nicht erprobt...