

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

Nils-Ole Bickel, Michel Brüger

26. Februar 2020

## Inhaltsverzeichnis

<b>1</b>	<b>Hardware</b>	<b>2</b>
1.1	Raspberry Pi 4 . . . . .	2
1.2	Google Coral . . . . .	2
1.3	Das Auto . . . . .	2
1.4	Der Trainingsrechner . . . . .	2
<b>2</b>	<b>Software</b>	<b>2</b>
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 . . . . .	2
2.3.1	Anaconda . . . . .	2
<b>3</b>	<b>Das Netz</b>	<b>3</b>
3.1	Implementierung . . . . .	3
3.2	Training . . . . .	3
3.2.1	Trainingsdaten . . . . .	3
3.2.2	Anzahl Epochen . . . . .	3
3.2.3	... . . . .	3
<b>4</b>	<b>Ergebnisse/Fazit</b>	<b>3</b>

- 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 Miniaturwunderland "getarnt" werden kann.

## 1.2 Google Coral

## 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...