UNIVERSITÄT
LEIPZIG

Master‘s Thesis

END-TO-END REINFORCEMENT LEARNING OF A CNN TO ACHIEVE AN AUTONOMOUS DRIVING AGENT RESILIENT TO LIGHT CHANGES

Leipzig, 05.12.2023

Georg Schneeberger

Supervisor:
Dr. Thomas Burghardt

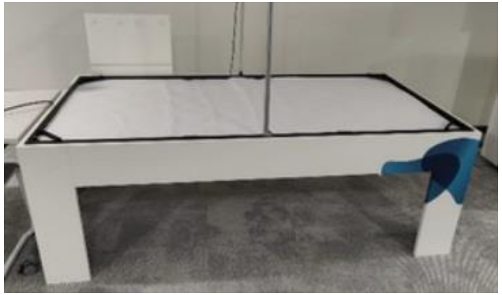
OUTLINE

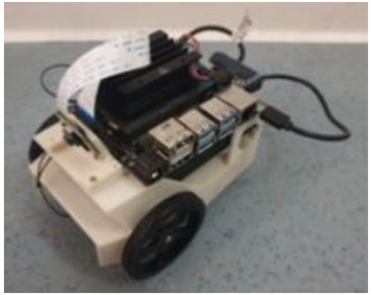
- Previous Work at the ScaDS.AI
- Autonomous Driving Task
- Research Questions
- Related Work
- Implementation
- Experimentation
- Evaluation
- Conclusion

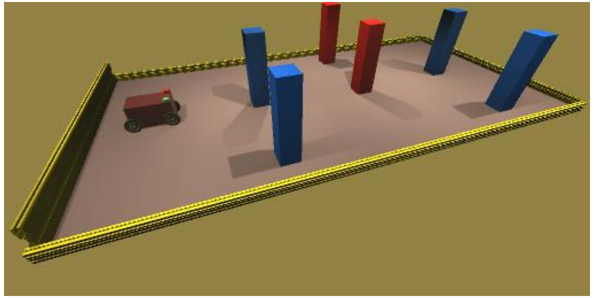
Master‘s Thesis

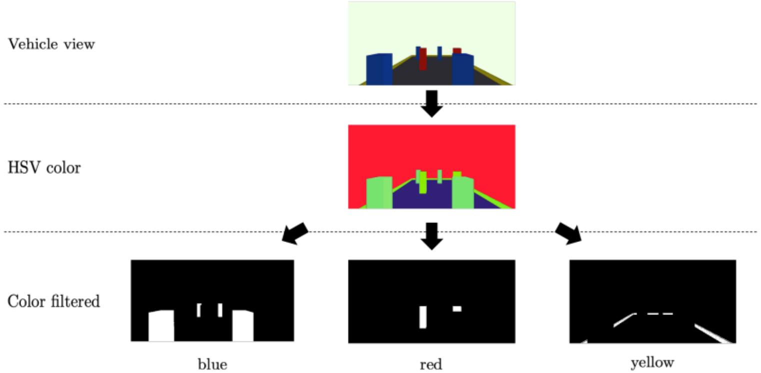
PREVIOUS WORK AT THE SCADS

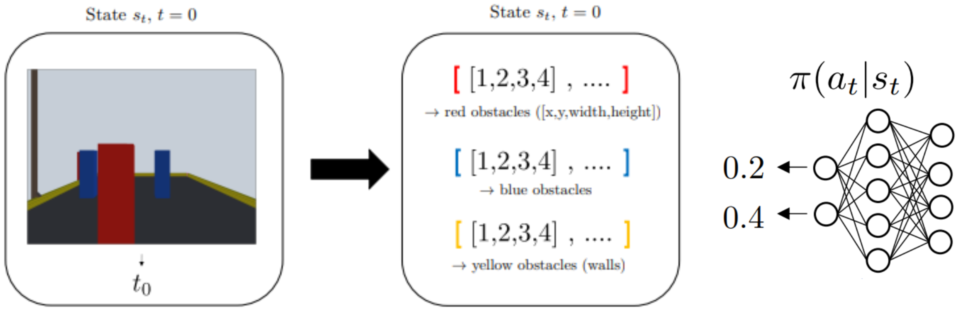
TRAIN AN AGENT TO DRIVE A VEHICLE IN A SIMULATED ENVIRONMENT USING REINFORCEMENT LEARNING











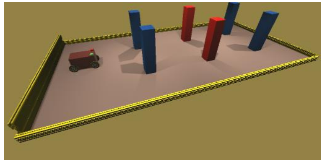
TRAIN AN AGENT TO DRIVE A VEHICLE IN A SIMULATED ENVIRONMENT USING REINFORCEMENT LEARNING

- A hand-crafted preprocessing pipeline extracts features from images.
- Extracted features of the last n frames are stacked together.
- Extracted features are used by a neural network to produce two outputs.
  - Left and right wheel acceleration
- Agent struggled with difficult parcours.
- Agent was not resilient to light changes.

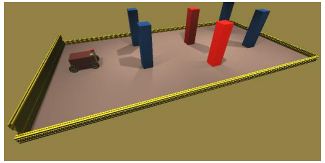
Master‘s Thesis

AUTONOMOUS DRIVING TASK

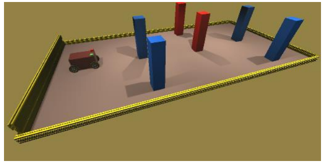
AUTONOMOUS DRIVING TASK



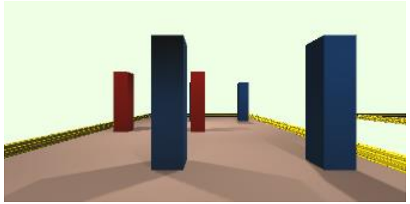
Easy



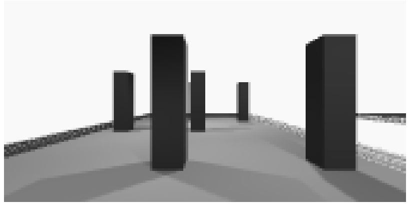
Medium



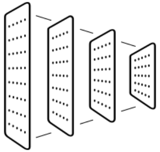
Hard



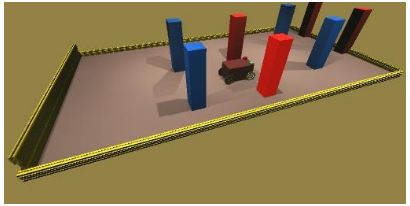
1. Agent Camera PoV



2. Preprocessing



3. CNN produces Wheel Acceleration values



4. Agent moves

Master‘s Thesis

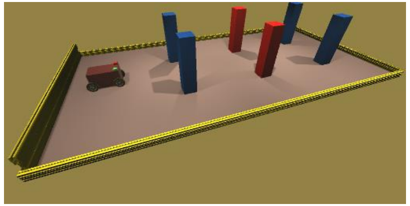
RESEARCH QUESTIONS

RESEARCH QUESTION 1

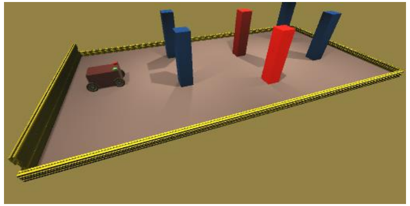
IS IT POSSIBLE TO TRAIN AN AGENT TO RELIABLY TRAVERSE THE PARCOURS OF ALL DIFFICULTY LEVELS?

Evaluation:

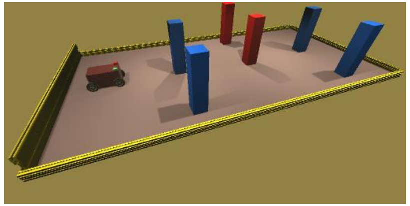
- Pass through the 3 goals without collisions
- Fixed lighting



Easy



Medium



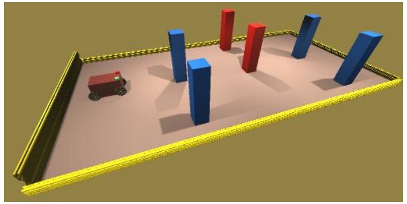
Difficult

RESEARCH QUESTION 2

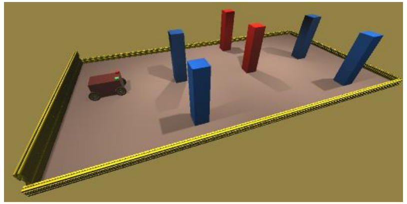
IS IT POSSIBLE TO USE AN END-TO-END TRAINED CNN TO MAKE THE AGENT ROBUST TO CHANGING LIGHT CONDITIONS?

Evaluation:

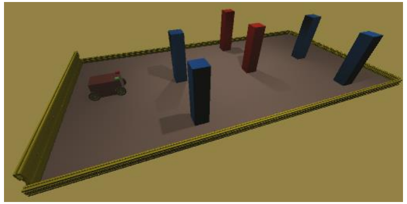
- Pass through the 3 goals without collisions
- Varying lighting



Bright



Standard



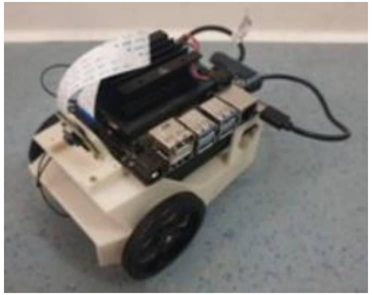
Dark

RESEARCH QUESTION 3

IS IT POSSIBLE TO USE A CNN WHICH IS SMALL ENOUGH TO BE USED IN THE JETBOT?

Evaluation:

- Create a replay of a successfully completed parcour
  - Replay consists of input-output pairs
- Give inputs to JetBot
- Is Output reproduced quick enough by the JetBot?

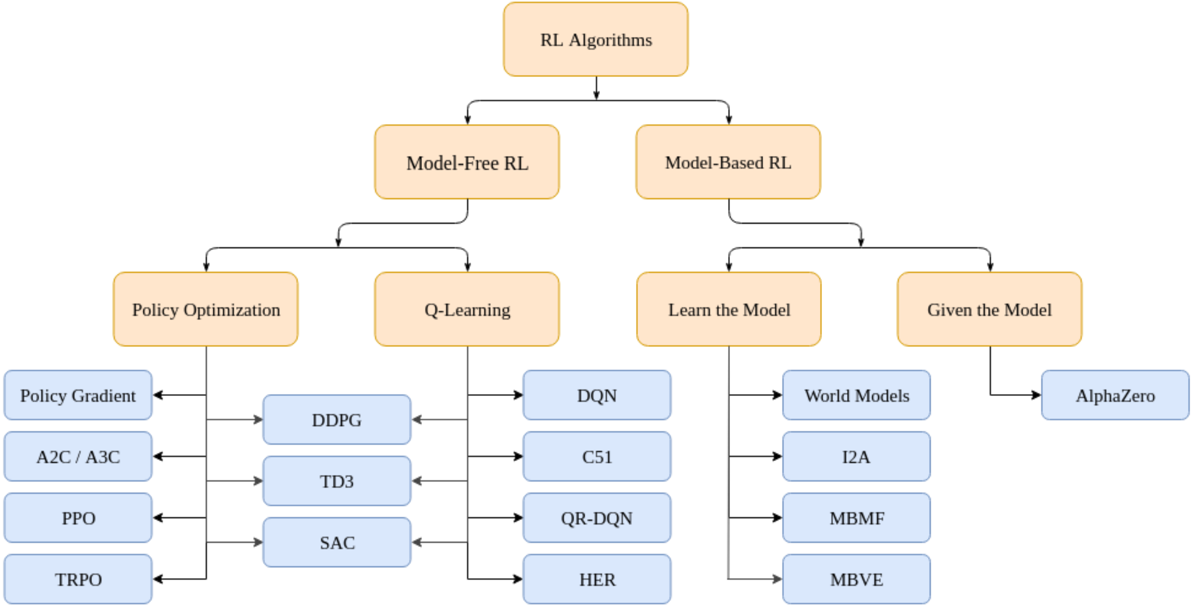


Master‘s Thesis

RELATED WORK

RELATED WORK

RL ALGORITHM CLASSIFICATION



RELATED WORK

PLAYING ATARI WITH DEEP REINFORCEMENT LEARNING

- Introduces DQN, an extension of Q-learning
- Plays Atari games using convolutional neural networks
- Utilized greyscaled and stacked images

Figure 1: Screen shots from five Atari 2600 Games: (*Left-to-right*) Pong, Breakout, Space Invaders, Seaquest, Beam Rider

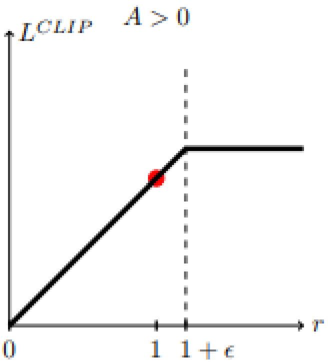
RELATED WORK

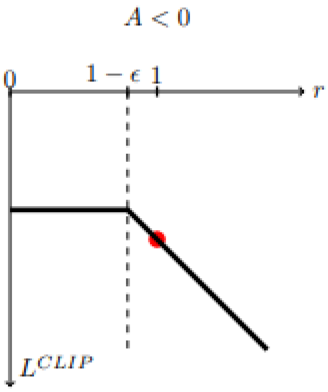
PROXIMAL POLICY OPTIMIZATION ALGORITHMS

- Introduces PPO, a stable and sample efficient policy based approach
- Great performance on continuous control tasks
- PPO restricts the size of policy updates

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right]$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$





RELATED WORK

DEEP REINFORCEMENT LEARNING FOR AUTONOMOUS DRIVING: A SURVEY

- Gives a good overview of RL for autonomous driving
- Describe modern autonomous driving system pipelines
- Describe extensions to RL such as Reward shaping

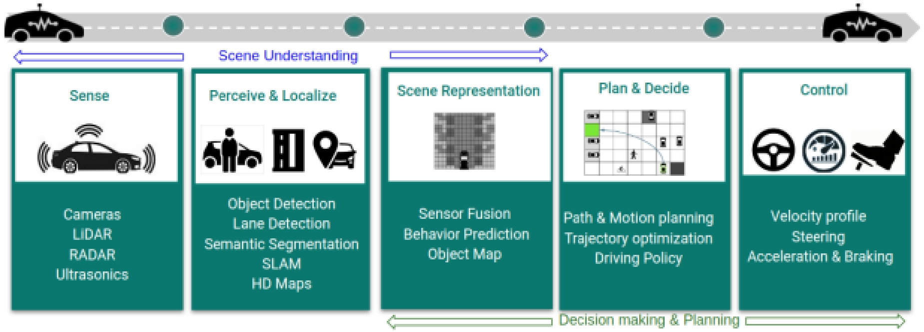


Fig. 1. Standard components in a modern autonomous driving systems pipeline listing the various tasks. The key problems addressed by these modules are Scene Understanding, Decision and Planning.

RELATED WORK

END-TO-END DRIVING VIA CONDITIONAL IMITATION LEARNING

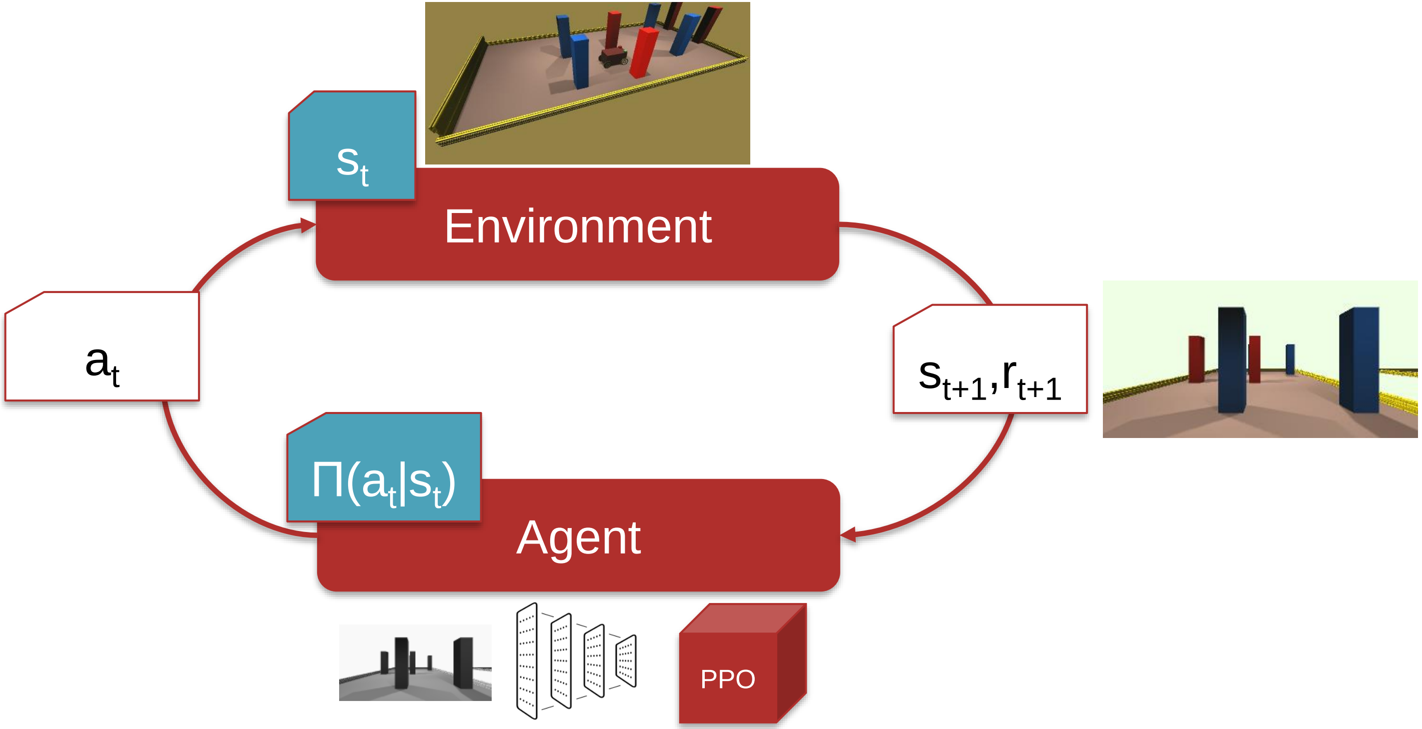
- Imitation learning with a convolutional neural network
- Extensive use of Data Augmentation
  - Transformations like changes in contrast, brightness and tone
  - Gaussian blur, Gaussian noise, salt-and-pepper noise

(a) Aerial view of test environment (b) Vision-based driving, view from onboard camera (c) Side view of vehicle

Master‘s Thesis

IMPLEMENTATION / METHODS

RL TRAINING ALGORITHM - PPO



RL TRAINING REWARD FUNCTION

$$\begin{aligned}
R(s_t, a_t) &= c_1 \cdot DistanceReward(s_t, a_t) + c_2 \cdot OrientationReward(s_t, a_t) \\
&\quad + c_3 \cdot VelocityReward(s_t, a_t) + c_4 \cdot EventReward(s_t, a_t) \\
DistanceReward(s_t, a_t) &= \Delta distance(Agent, NextGoalPosition) \cdot \Delta T \\
OrientationReward(s_t, a_t) &= S_C(NextGoalPosition - AgentPosition, agentDirection) \cdot \Delta T \\
VelocityReward(s_t, a_t) &= v \cdot \Delta T \\
EventReward(s_t, a_t) &= \begin{cases} 100, \text{completed the parcour} \\ 1, \text{passed a goal} \\ -1, \text{missed a goal} \\ -1, \text{collision with wall or obstacle} \\ -1, \text{timeout} \\ 0, \text{otherwise} \end{cases}
\end{aligned}$$

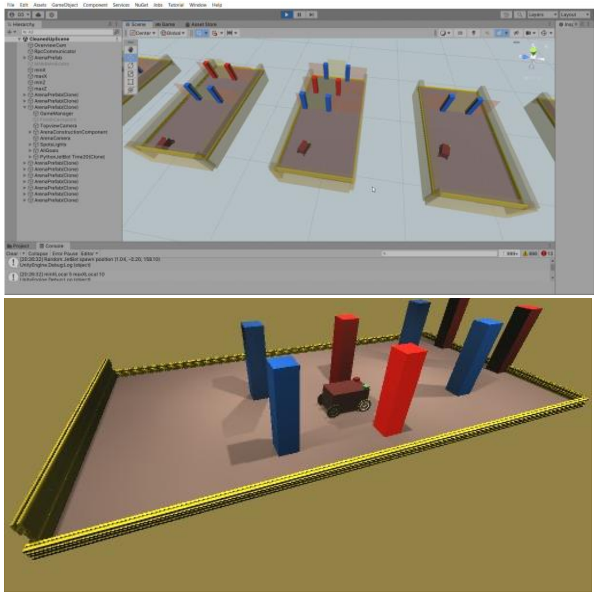
Abbildung 4.3.: Complete reward function R with all its components

S_C: cosine similarity c_i: weights

s_t: state t a_t: action in state t

REINFORCEMENT LEARNING TRAINING SETUP

Unity

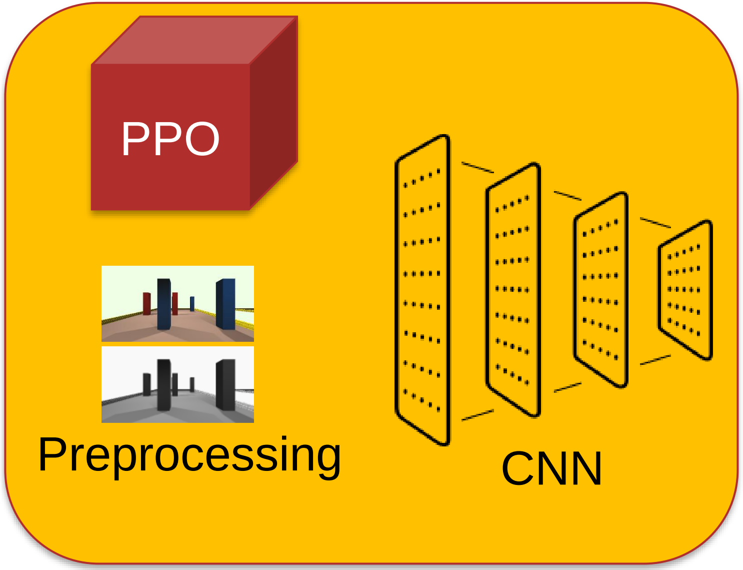


-Agent Behaviour

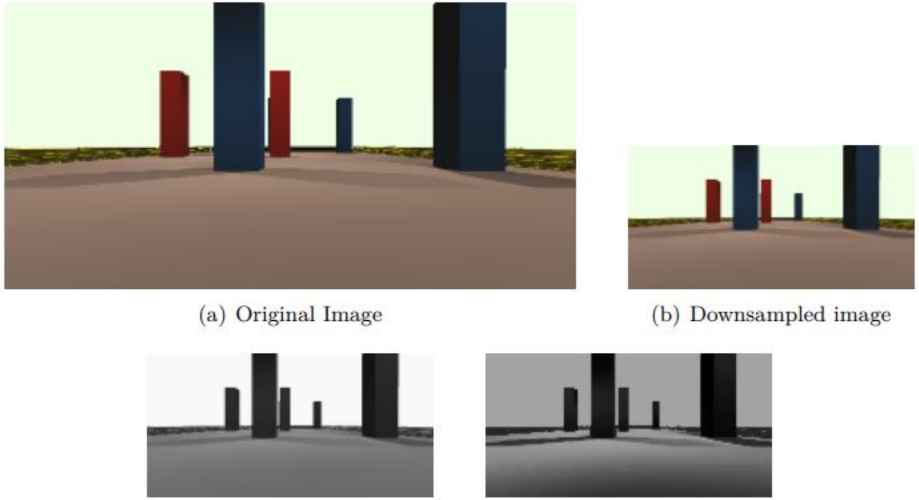
Communication

-Observations
-Rewards

Python



AGENT DESIGN: PREPROCESSING STEPS



(a) Original Image

(b) Downsampled image

(c) greyscale

(d) histogram equalized image

Abbildung 4.4.: 4 Stages of preprocessing images for the CNN

AGENT DESIGN: CONVOLUTIONAL NEURAL NETWORK

- Agent recieves an image at each iteration
- Apply image preprocessing steps
- concatenate with last n images
- Image stack is fed to a Convolutional Neural Network

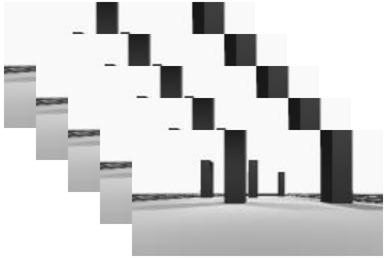
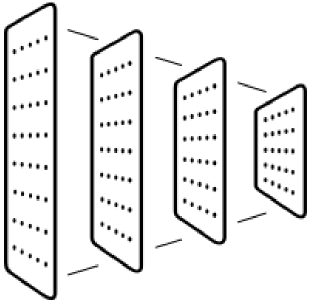


Image stack



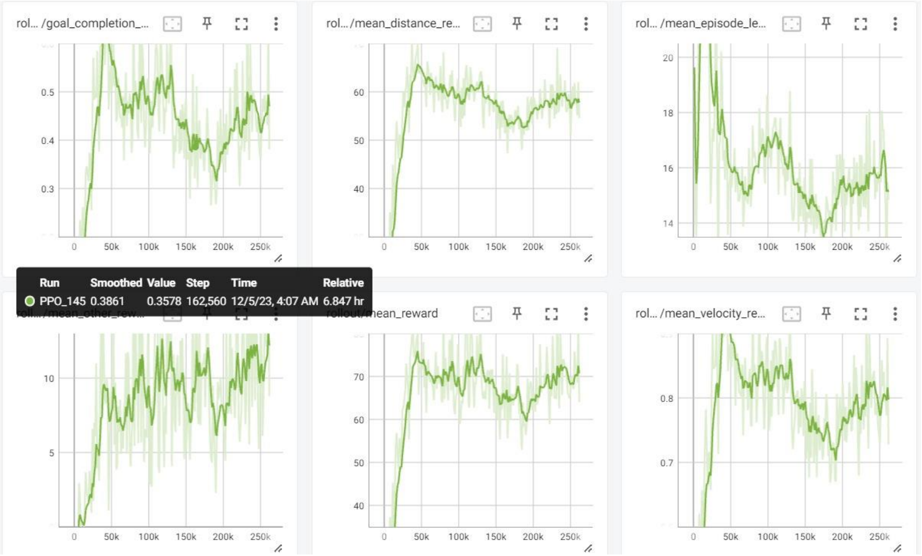
Left and right
wheel acceleration

RL TRAINING MONITORING

LOGGING TO TENSORBOARD

Standard and custom measurements

- Mean reward per episode
- Percentage of passed goals
- Percentage of successfully completed parcours
- Loss
- …



Master‘s Thesis

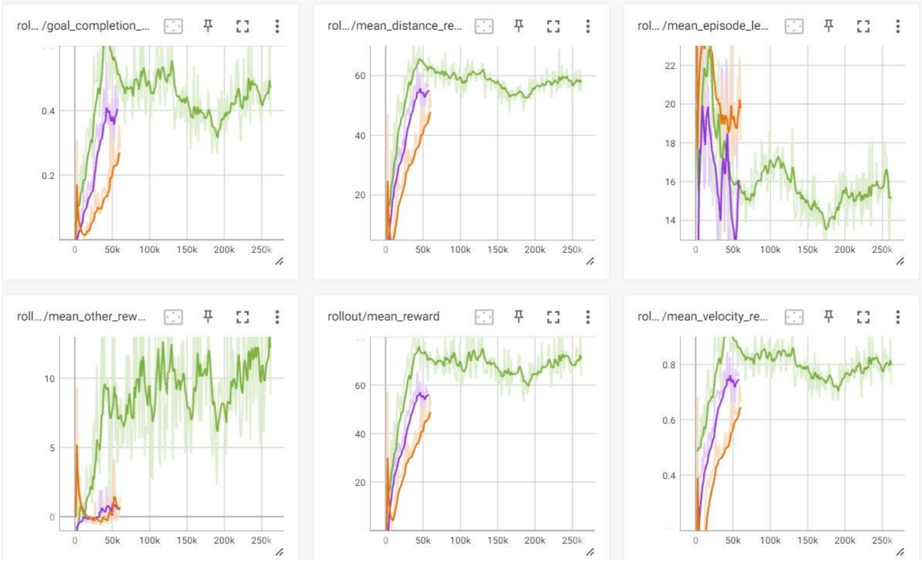
EXPERIMENTATION

UNIVERSITÄT
LEIPZIG

EXPERIMENTATION

CHOICE OF PARAMETERS AND PREPROCESSING STEPS

- Network size
- Preprocessing steps
  - Brightness change
  - Contrast change
  - Histogram equalization
- Amount of frame stacking
- Reward function weights



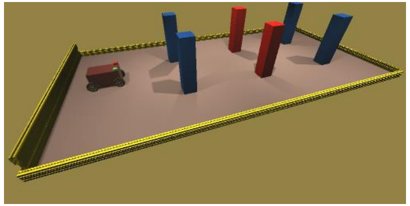
Master‘s Thesis

EVALUATION

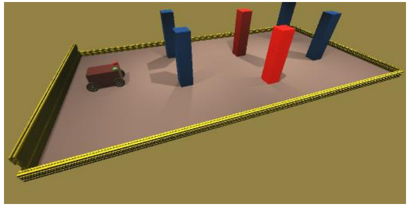
AGENT EVALUATION - Q1

AGENT IS EVALUATED ON THE THREE EVALUATION TRACKS WITH STANDARD LIGHTING

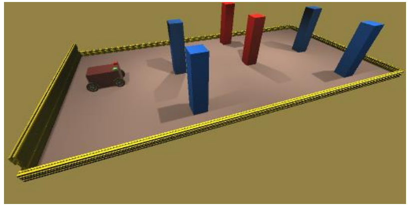
- Rate of successfully completed attempts



Easy



Medium

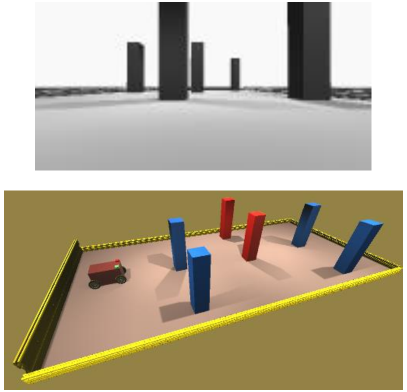


Difficult

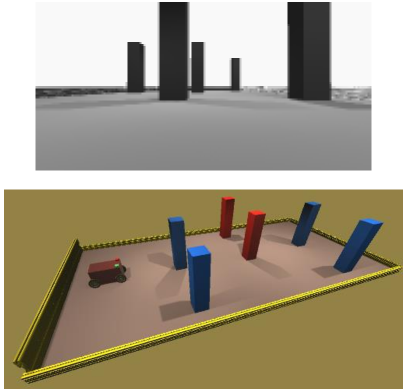
AGENT EVALUATION – Q2

AGENT IS EVALUATED ON THE THREE EVALUATION TRACKS WITH VARYING LIGHTING

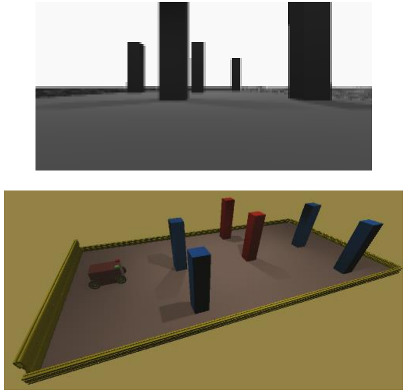
- Rate of successfully completed attempts



Bright



Standard



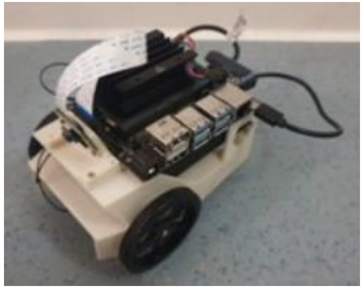
Dark

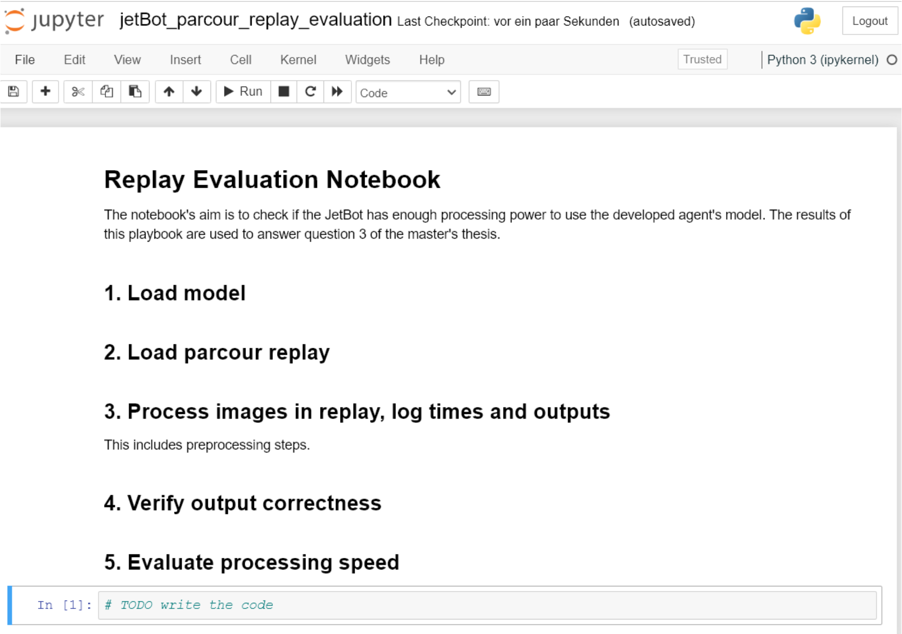
JETBOT PROCESSING POWER EVALUATION – Q3

GENERATE REPLAY OF PARCOUR IN PYTHON AND EVALUATE REPLAY ON JETBOT

Replay consists of

- Input-output pairs
- Processing times
- Neural Network model





Master‘s Thesis

CONCLUSION

CONCLUSION

The thesis investigates training an agent that is resilient to light changes.

- Train an autonomous driving agent with Reinforcement Learning
- The agent uses a convolutional neural network as its policy
  - Input is a preprocessed image
- Agent is evaluated on tracks of 3 difficulty levels with different light settings

THANK YOU!

Georg Schneeberger
Faculty of Mathematics and Computer Science
Database Department

g.m.schneeberger@gmail.de
www.uni-leipzig.de