RWTH Aachen University
Software Engineering Group

Comparison of Deep Learning Architectures on Simulated Environments

Seminar Paper

presented by
Bergerbusch, Timo

1st Examiner: Prof. Dr. B. Rumpe

2nd Examiner:

Advisor: Evgeny Kusmenko

The present work was submitted to the Chair of Software Engineering

Aachen, April 27, 2018

The present translation is for your convenience only.
Only the German version is legally binding.

Statutory Declaration in Lieu of an Oath

Last Name, First Name _____ Matriculation No. (optional) _____

I hereby declare in lieu of an oath that I have completed the present Bachelor's thesis/Master's thesis* entitled

_____

_____

_____

independently and without illegitimate assistance from third parties. I have use no other than the specified sources and aids. In case that the thesis is additionally submitted in an electronic format, I declare that the written and electronic versions are fully identical. The thesis has not been submitted to any examination body in this, or similar, form.

Location/City, Date _____ The German version has to be signed. _____ Signature

*Please delete as appropriate

Official Notification:

Para. 156 StGB (German Criminal Code): False Statutory Declarations
Whosoever before a public authority competent to administer statutory declarations falsely makes such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment not exceeding three years or a fine.

Para. 161 StGB (German Criminal Code): False Statutory Declarations Due to Negligence
(1) If a person commits one of the offences listed in sections 154 to 156 negligently the penalty shall be imprisonment not exceeding one year or a fine.

(2) The offender shall be exempt from liability if he or she corrects their false testimony in time. The provisions of section 158 (2) and (3) shall apply accordingly.

I have read and understood the above official notification:

City, Date _____ The German version has to be signed. _____ Signature

Abstract

The topic of autonomous driving using artificial intelligence increases in importance with the overwhelming amount of software usage within vehicles. For that *Convolutional Neural Networks* (CNNs), which try to figure out the importance of special areas of a single picture, have been shown to be promising.
In this paper we will give a general introduction to the topic of CNNs. We distinguish between the three main *deep learning languages* (DLLs) currently used and researched for autonomous driving agents: mediated perception, behaviour reflex and direct perception. Further we will compare different languages, which can be used to implement the different DLLs, based on the factors of usability, scope of functionality and the integration on a subject.
As a proof of concept we will train a CNN using the language *CNNArch* on the famous KITTI dataset in order to create a trained model. This model will then be tested on a test set created using either the simulation tool `MontiSim` or the open source racing game TORCS, containing multiple different challenging scenarios the agent has to manage.
Finally we evaluate the trained model on it's performance and try to reason, why it performed particularly good/bad, and give an overview based on the implemented test in order to state the similarities and differences of the languages.

Contents

Chapter 1

Introduction

General introduction to the topic

1.1 Available Deep Learning Approaches

The things one needs to know in order to understand everything that follows based on [CSKX15]

1.1.1 Mediated Perception

State the approach of a Mediated Perception based CNN

1.1.2 Behavior Reflex

State the approach of a Behavior Reflex CNN

1.1.3 Direct Perception

1.2 Deep Learning Languages

1.2.1 CNNArch

general and some more in depth information about CNNArch based on [TvWH17]

1.2.2 MxNet

general information from [CLL+15]

Chapter 2

Running Example

The example net, AlexNet [KSH12] implemented as a Direct Perception approach

2.1 Implementation

Implementation of the net using CNNArch or MxNet (maybe discuss already implemented approaches to cover more details)

2.2 Training

The training of the implemented net based on the KITTI dataset [GLSU13]

Chapter 3

Evaluation

Test the trained set in a simulation environment

3.1 MontiSim

Short introduction of the tool used to evaluate the net in [Rea17]

3.2 Results

Evaluating the results of the test of the net in MontiSim

Chapter 4

Conclusion

Conclusion of differences and similarities between the frameworks

Also a general conclusion based on results and [Grz17]

Bibliography

[CLL+15] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

[CSKX15] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2722–2730. IEEE, 2015.

[GLSU13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[Grz17] Adam Grzywaczewski. Training ai for self-driving vehicles: the challenge of scale. Technical report, Tech. rep., NVIDIA Corporation. URL https://devblogs. nvidia. com/parallelforall/training-self-driving-vehicles-challenge-scale, 2017.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[Rea17] Bernhard Rumpe and et. al. Montisim. https://github.com/MontiSim, 2017.

[TvWH17] Thomas Timmermann, Michael von Wenckstern, and Malte Heithoff. Cnnarchlang. https://github.com/EmbeddedMontiArc/CNNArchLang, 2017.

[ZF14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.