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, May 11, 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

The field of autonomous driving agents has rapidly increased in modern car manufacturing. Current research topic rises the agents from parking or lane keeping assistant fully autonomous driving agents obeying the traffic rules and having the ability to react to the volatile environment in a reasonable way.

For that machine learning techniques have proven themselves as an essential part. But in order to fulfil the security standards and create a sophisticated agent it has to be trained on hundred-thousands of scenarios each having a large set of data attached, for example sensor and camera data. The approach of *Convolutional Neural Networks* (CNNs) have been proven to be powerful enough to handle such many training iterations with a huge number of input variables, while maintaining the large learning capacity. [KSH12]

A CNN, as explained in Section 1.1, has a general structure, but can be altered to fit into the approach in various ways influencing the result. Therefore we introduce in Section 1.2 the three main approaches of using a CNN.

Further in Section 1.3 we see that there is the need of specific languages for the design and implementation of such agents. Two languages will be discussed and compared based on their suitability regarding the AlexNet, stated in Section 1.1.1.

1.1 Convolutional Neural Network (CNN)

A CNN is a special class of deep feed-forward neural networks. On of the main design goals of a CNN is that they require a minimal amount of preprocessing. This is an important aspect, because they are often fed with images. Preprocessing high resolution images is very costly in terms of computational time. In the context of autonomous driving the time is even more crucial, since the driving agent needs to be able to react to spontaneous events.

Like most parts of neural networks, also the CNNs are inspired by biological processes. It is mainly based on the connectivity pattern of an animals visual cortex, where special neurons respond only to stimuli of their receptive field. Partially overlapping guarantees a complete coverage of the field of view. [Wik18]

Further CNNs make strong and mostly correct assumptions about the nature of images, like staionarity of statistics nad locality of pixel dependencies. This leads to fewer connections and parameters, compared to a normal feed-forward neural net with similar sized layers, and therefore reduces the time it takes to be trained, while being only slightly worse in their best-performance. [KSH12]

TODO:

1. how detailed?
2. where starting?

1.1.1 AlexNet

The *AlexNet* is on of the best performing CNN architectures currently known. It is trained on the ImageNet subsets of `ILSVRC-2010` and `ILSVRC-2012`[1] and became famous because of its result being way ahead of all other competitors.

A highly optimized GPU implementation of this architecture combined with innovative features is publicly available. Those features lead to improve performance and reduce training time. An important note is that the original test is dated back to 2012 and therefore was used with an overall GPU memory of 6GB, with which training took abound six days. With modern hardware the training can be done faster, or the model can be trained much more. [KSH12]

TODO:maybe calc the possible speed up based on "Hardware for Machine Learning"

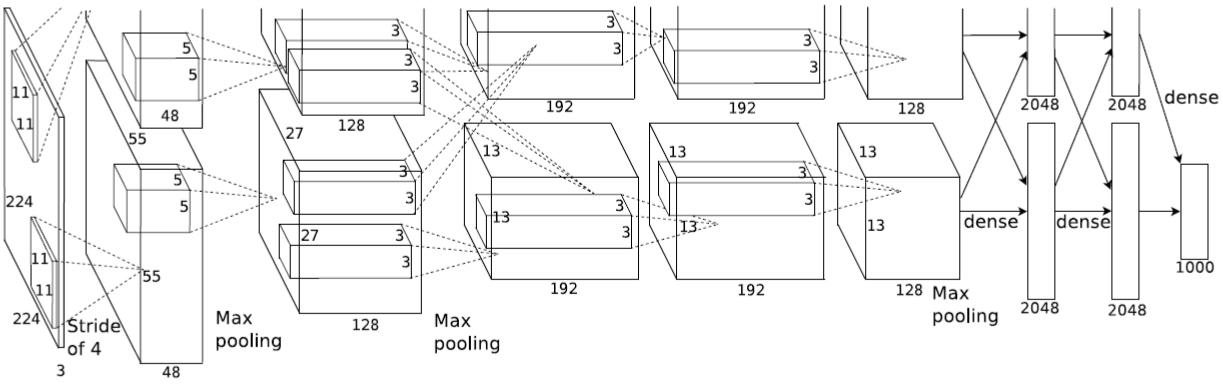

Figure 1.1: The AlexNet-architecture for two GPUs. It consists of 5 convolutional layers (at the beginning) and three fully-connected layers (at the end). The GPUs only communicate between two layers, but never within a layer.[KSH12]

TODO:Vereinfachen der Zeichnung

[1] Further information: http://www.image-net.org/challenges/LSVRC/

1.2 Available Deep Learning Approaches

There are various approaches of autonomous driving agents making a variety of assumptions and differ in numerous options. But they can be mostly categorized into two major groups of approaches: mediated perception approaches and behavior reflex approaches. [CSKX15]

In this paper we further analyze a suggested third group, called direct perception, which can traced to [Gib14] in the late 80's, but was sharply criticized by researchers of the other two groups, i.e. in [Ull80].

All these three groups differ in the way of interpreting the given sensor data and whether or not to create a some what bigger picture based on consequent data.

1.2.1 Mediated Perception

The mediated perception approach is a multi-component continuous process. Every component recognizes specific aspects for driving. For example traffic signs, lanes, other cars. Those components are then combined into one single world state representing the cars surrounding based on the sensor data. [GLSU13]
These world states are 3D models of the current world. Cars are identified using a classifier and then often surrounded by a 3D bounding box. An example can be seen in Figure 1.2. By comparing different frames generated one can estimate the speed and distance to those objects and derive an A.I. based precedence behavior. [GLSU13][CSKX15]

The often stated problems with such approaches are, that computing such a scene is costly in terms of computation time. Some information is irrelevant, redundant or even misleading due to inaccuracy of sensors. To perform a right turn the sensor information of the distance to a car left behind me is irrelevant, but becomes very important when taking a left turn.
Additionally many of the subtasks are still open research topics themselves. For example a reliable lane detection throughout various weather conditions or worse. Imagine a new road not having any drawn lines yet.
Also mediated perception approaches require very detailed information up front, like up-to-date maps.

The approach of mediated perception is a reasonable and very sturdy way of handling such a complex task, but has its drawbacks regarding computational time and additional knowledge.

1.2.2 Behavior Reflex

State the approach of a Behavior Reflex CNN

1.2.3 Direct Perception

State the approach of direct perception

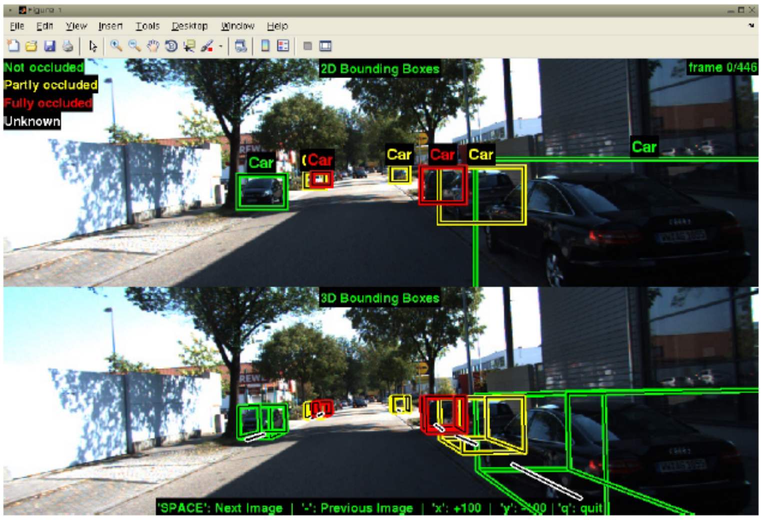
File Edit View Insert Tools Desktop Window Help
Not occluded
Partly occluded
Fully occluded
Unknown
2D Bounding Boxes
frame 0/446
Car
Car
Car
Car
Car
Car
3D Bounding Boxes
Car
'SPACE': Next Image | '-': Previous Image | 'x': +100 | 'y': -100 | 'q': quit

Figure 1.2: An example of a scene using 3D bounding boxes. This image is taken from the MATLAB delvopment kit of [GLSU13]

1.3 Deep Learning Languages

TODO:

1. why need a language
2. what should it in general do?
3. state that different languages ex.

1.3.1 CNNArch

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

1.3.2 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 [**?**]

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.

[Gib14] James J Gibson. *The ecological approach to visual perception: classic edition.* Psychology Press, 2014.

[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.

[Ull80] Shimon Ullman. Against direct perception. *Behavioral and Brain Sciences*, 3(3):373–381, 1980.

[Wik18] Wikipedia contributors. Convolutional neural networks — Wikipedia, the free encyclopedia, 2018. [Online; accessed 11-May-2018].