Thesis for the degree of Licentiate of Engineering

On Improving Validity of Deep Neural Networks in Safety Critical Applications

Jens Henriksson

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2020

On Improving Validity of Deep Neural Networks in Safety Critical Applications
Jens Henriksson

Licentiate thesis at Chalmers University of Technology
Technical report No. 211L, ISSN 1652-876X

Department of Computer Science and Engineering
Chalmers University of Technology
SE–412 96 Göteborg, Sweden
Telephone + 46 (0) 31 – 772 1000

Printed by Chalmers Digitaltryck
Göteborg, Sweden 2020

Abstract

Context: Deep learning has proven to be a valuable component in object detection and classification, as the technique has shown an increased performance throughput compared to traditional software algorithms. Deep learning refers to the process, in which an optimisation process learns an algorithm through a set of labeled data, where the researcher defines an architecture rather than the algorithm itself. As the resulting model contains abstract features retrieved through the optimisation process, new unsolved challenges emerge that need to be resolved before deploying these models in safety critical applications.

Aim: The aim of this Licentiate thesis has been to study what extensions are necessary to verify deep neural networks. Furthermore, the thesis studies one challenge in detail: how out-of-distribution samples can be detected and excluded.

Method: A comparative framework has been constructed to evaluate performance of out-of-distribution detection methods on common ground. To achieve this, the top performing candidates from recent publications were used as a reference snowballing baseline, from which a set of candidates were studied. From the study, common features were studied and included in the comparative framework. Furthermore, the thesis conducted semi-structured interviews to understand the challenges of deploying deep neural networks in industrial safety critical applications.

Results: The thesis found that the main issue with deployment is traceability and quality quantification, in the form that deep learning lacks proper descriptions of how to design test cases, training datasets and robustness of the model itself. While deep learning performance is commendable, error tracing is challenging as the abstract features in the do not have any direct connection to the training samples. In addition, the training phase lacks proper measures to quantify diversity within the dataset, especially for the vastly different scenarios that exist in the real world.

One safety method studied in this thesis is to utilize an out-of-distribution detector as a safety measure. The benefit of this measure is that it can both identify and mitigate potential hazards. From our literature review it became apparent that each detector was compared with different techniques, hence a framework was constructed that allowed for extensive and fair comparison. In addition, when utilizing the framework, robustness issues of the detector were found, where performance could drastically change depending on small variations in the deep neural network.

Future work: Future works recommend testing the outlier detectors on real world scenarios, and show how the detector can be part of a safety strategy argumentation.

Keywords: Safety critical applications, deep neural networks, out-of-distribution, outlier detection.

Acknowledgments

During the past couple of years, I have been on a research journey with lots of ups and downs. As the research field is still in its infancy, my research has needed to change its coarse as numerous new discoveries occurred during my Licentiate period. This research change would not have been possible without the support of several individuals with expertise in different domains.

First and foremost, my deepest gratitude to my supervisors. *Christian Berger*, my academic supervisor, who have guided me through the academic scene and helped me avoid pitfalls, but also allowed for self exploration with the right amount of challenges and support. And *Stig Ursing*, my industrial supervisor, who have always taken time from his other duties to answer my open ended questions. Even though we often end up with more questions after our discussions, it has been an absolute blast working and discussing research topics with you, I hope we can continue working together.

I also want to thank my connected research hubs. I want to thank the Wallenberg AI, Autonomous Systems and Software Program (*WASP*) for creating courses, research arenas and events where researchers can present and discuss ideas. Additionally, I want to thank *Cristofer, Markus, Sankar* and *Lars* from the *SMILE* research consortium for the discussions, collaborations and the publications we have written together. It has been a pleasure exploring unfamiliar areas with you, and I look forward to further collaboration.

My colleagues at *Semcon* - Thank you for supporting me and making sure I keep on challenging myself. A big Thank you to *Magnus* and *Mats* who both believe in this project. Also, a special thanks to my friend *Axel*, who have always been by my side, always got time to discuss crazy ideas over a coffee break. Additionally, my department colleagues at *Chalmers* - Even though I am seldomly there, you always make sure to make me feel welcome.

Finally, I want to thank my family for always supporting me. To my parents *Lars-Olof* and *Regina* for their encouragement. To my brother *Lars-Henrik* and *Gerda*, who supports whenever they can. Last, but the most important, my love *Anna*, who have supported me through all hardships. Without you, this would not have been possible.

Jens Henriksson
Göteborg, May 2020

List of Publications

This thesis is based on the following appended papers:

Paper I: Jens Henriksson, Markus Borg and Cristofer Englund. *Automotive safety and machine learning: Initial results from a study on how to adapt ISO 26262 safety standard.* Published in IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS), 2018

Contribution: Designed and conducted the interviews, lead and summarized the workshop discussions, structured and wrote a majority of the paper.

Paper II: Jens Henriksson, Christian Berger, Markus Borg, Lars Tornberg, Cristofer Englund, Sankar Raman Sathyamoorthy and Stig Ursing. *Towards Structured Evaluation of Deep Neural Network Supervisors.* Published in IEEE International Conference On Artificial Intelligence Testing (AITest), 2019

Contribution: Designed the comparison framework including datasets and metrics, conducted the baseline performance evaluation, structured and wrote a majority of the paper.

Paper III: Jens Henriksson, Christian Berger, Markus Borg, Lars Tornberg, Sankar Raman Sathyamoorthy and Cristofer Englund. *Performance Analysis of Out-of-Distribution Detection on Trained Neural Networks.* Under review in Journal of Information and Software Technology, 2020

Contribution: Designed the experimental training and method comparison setup, conducted the comparisons, structured and wrote a majority of the paper.

Other relevant publications:

Jens Henriksson, Christian Berger, Markus Borg, Lars Tornberg, Sankar Raman Sathyamoorthy and Cristofer Englund. *Performance Analysis of Out-of-Distribution Detection on Various Trained Neural Networks.* Published in 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2019

Dhasarathy Parthasarathy, Karl Bäckström, **Jens Henriksson** and Sólrún Einarsdóttir. *Controlled time series generation for automotive software-in-the-loop testing using GANs.* Published in IEEE International Conference On Artificial Intelligence Testing (AITest), 2020

Contents

Abstract **i**

Acknowledgments **iii**

List of Publications **v**

1 Introduction **1**

1.1 Background 1

1.2 Research aim 3

1.3 Limitations 4

1.4 Outline 4

2 Frame of Reference **5**

2.1 Deep learning and perception 5

2.2 Safety verification and validation 7

2.3 Out-of-distribution detection 10

3 Research Approach **13**

3.1 Research philosophy 13

3.2 Research activities 14

3.3 Safety verification and validation 15

3.4 Out-of-distribution detection 16

3.4.1 Comparison metrics 18

3.5 Research validity 18

4 Summary of Appended Papers **21**

4.1 Paper I 21

4.2 Paper II 22

4.3 Paper III 24

5 Discussions **27**

5.1 Testing extensions for deep learning 27

5.2 Using deep learning for testing extensions 28

5.3 Research quality 29

5.4 Future Research 29

6 Conclusions **31**

Bibliography **33**

Chapter 1

Introduction

In the past decade, automated vehicles have transitioned from an industry vision to fleet tests with small or limited deployment. This transition has been possible thanks to advances in the field of computer vision, aided by advances with deep learning. However, incorporating deep learning into safety critical applications comes with inherent challenges that need to be addressed before large scale deployment can be achieved. This thesis has investigated what additional measures of testing are needed for deep neural networks, and studied the challenge of detecting out-of-distribution samples more in depth. In this initial chapter the background and research goal will be presented as well as highlight the gap in testing and verification of deep learning models.

1.1 Background

Perception has always been a main pillar of autonomy (Pendleton et al. 2017). Perception refers to processing sensory inputs and aligning it with beliefs or concepts of knowledge to perceive the surroundings of the system. Perceiving the surroundings enables the autonomous system to plan a path ahead without interfering with surrounding objects. Without reliable perception, the system would not be able to move without the risk of causing accidents. For the automotive industry, improved perception systems have enabled collision mitigation systems such as City Safety by Volvo, which reduced the rear-end frontal collisions with 28% (Isaksson-Hellman and Lindman 2015). In addition, it has enabled additional intelligent functions like lane departure warning and adaptive cruise control.

Perception systems are commonly created by a combination of sensory inputs, including radars, ultrasonic, LiDARs and cameras (Rosique et al. 2019). In these systems, radars and LiDARs are commonly used for short and long range distance mapping of objects. However, these sensors struggle to determine the validity

or object type of their detections. This is mitigated by fusing together distance estimations with camera imagery, as object detection is more straight forward using cameras. If the object cannot be categorized, then the detections are considered of unknown type, which suggests that the system proceeds with additional care. Knowing the type of object detected allows the perception system to operate more securely, as knowing an object type will indicate a specific type of motion. For example, a tree waving in the wind can be perceived to have speed, but the object type identification will allow the system to accurately determine that the tree will not move. Furthermore, object classification allows for excluding false detections, plan around static objects and create a better predicted motion of dynamic objects.

Utilizing image processing to detect and classify objects in the scene has previously consisted of finding gradients, color patterns or other constructed features to detect specific objects in the scenery. For example, a simple way to detect road lanes has been to detect bright colors on the surface that indicate lane markings, followed by fitting polygons to find the lane curvatures. While these algorithms have proven sufficient for driving support functions, to enable full autonomous drive the vehicle perception needs to be far more rigorous.

In 2012 a big breakthrough in computer vision occurred, when AlexNet utilized a deep neural network to beat traditional image classification algorithms by a large margin (Krizhevsky et al. 2012) on the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al. 2015). The network highlighted the potential of letting an algorithm learn directly from data, which features represent best the training set, and create its own features based on an optimization process. Deep neural networks, or *deep learning* as the field is named, refers to a computational graph that is designed with several connected layers illustrated in Figure 1.1, where each layer consists of a set of nodes connected to the preceding layer, therefore allowing for a functional approximation with a large multitude of parameters. Each node in a layer represent a feature that is used to create more abstract features in the proceeding layer.

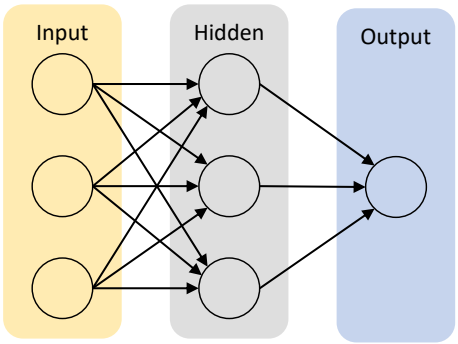

Figure 1.1: An example of a deep neural network with an input layer with three inputs, one hidden layer with three neurons and one output neuron. Each neuron consist of inputs, weights, a bias and an activation function, which the neuron translates into a single output.

Deployment of a system that includes deep learning models is in general not an issue for non-safety applications, as long as performance improvements outweigh potential new unwanted behavior. Recommendation systems and face tagging of social media images are examples of these types of systems that are beneficial for the user as the performance is better than for previous algorithms. However, for safety-critical applications the potential of unwanted behavior may be far more severe than the potential gain of deployment (Russakovsky et al. 2015). Autonomous driving is one topic where this debate shows, as flaws in these systems can lead to fatal events, safety regulations with regards to deployment have to be designed to ensure deployment only occurs when the system has been proven reasonably safe.

One inherent problem with deep learning models stems from the optimization process when training the parameters on gathered datasets. As the optimization consists of minimizing a loss function, the parameters converge with regards to the existing data, therefore the algorithm can create biases towards specific events in the scenery (Szegedy et al. 2014; Subramanya et al. 2017). In addition, there might be undiscovered events that incorrectly trigger the optimization criteria that can make the algorithm perform undesirable in certain scenarios. While some of these inconsistencies can be caught by traditional black box testing (Nidhra and Dondeti 2012), additional measures have to be developed, tested and evaluated that mitigate the impact of undesired or unforeseeable events in systems that include deep neural networks.

1.2 Research aim

The research of this thesis aims to support systematical testing of deep neural networks and initial establishment of the inherent challenges of deep learning. As deep neural networks can consist of a magnitude of parameters received from an optimization process, researchers need tools that analyze the symbiosis of parameters rather than the parameters themselves.

One of the desired outcomes of deep learning is to generalize knowledge and thereby be able to operate outside of the boundaries of its training domain. To achieve this, the model needs to identify when operates on unfamiliar data, and adjust accordingly. This is the basis from which the following research questions are drawn:

- **RQ1: What testing extension is required for deep neural networks in safety-critical applications?** Safety standard ISO 26262 describes how functional development is to be conducted with safety in mind. Additionally, ISO/PAS 21448 were introduced in 2019 to cover faults in the intended functionality of all algorithms, including machine learning. How does this translate and infer requirements of the deep learning model?

- **RQ2: How can deep learning handle unfamiliar data that lie outside of the training domain?** During training, deep neural networks are generally only trained on inlier samples, i.e., samples with a desirable outcome. However, when deployed, these networks can be exposed to previously unseen scenarios for which the model still has to predict with high certainty. How can these scenarios be identified?

1.3 Limitations

This thesis only considers convolutional neural networks and image processing. Even though outlier detection and safety can be an issue in several applications, if the model robustness can be argued for a convolutional neural network, it can be extrapolated to a traditional feed forward network, as the former is more complex. The extrapolation can be motivated as image processing suffers from curse of dimensionality, a phenomenon that arise in high-dimensional data where the volume of parameter space rises so fast, that even the largest dataset is considered sparse (Erfani et al. 2016). In addition, as the research field is still in an infant stage, research is conducted on publicly available datasets so results can be shared and replicated by open sourcing code.

1.4 Outline

The rest of this thesis is structured as follows:

Section II: Frame of Reference presents related work in out-of-distribution detection and inherent flaws of deep neural networks. Furthermore, conventional testing and safety standards are explored to understand and how these interact with deep learning.

Section III: Research Approach summarizes the initial thought behind experiments that have been conducted and ties the experiments to the research questions.

Section IV: Summary of Appended Papers gives a short description and summary of the results of the published papers during the research.

Section V: Discussions puts the results in perspective and compares it to related research activities and the research questions stated in Section 1.2. Future directions, limitations and drawbacks of the thesis is also reflected upon.

Section VI: Conclusions is the last chapter and summarizes the key findings in the results and gives the final remarks of this thesis.

Chapter 2

Frame of Reference

This section covers the related research for the thesis. It aims to explore research with regards to deep neural network development and how the inherent flaws can be abused to fool deep nets. In addition, it covers safety from the machine learning point of view, by identifying verification measures that can be applied in parallel to development of the actual model.

2.1 Deep learning and perception

To enable autonomous solutions, perception is one key ingredient (Pendleton et al. 2017). Perception is the ability to see, hear or become aware of one's surrounding through sensors. For an autonomous vehicle, perception is built with sensors including radar, cameras and LiDARs to identify objects, road lanes and environmental descriptions. Rosique et al. (2019) conducted a systematic review of perception systems and concluded the strength and weaknesses of different types of sensors and simulators. Regarding object detection and classification, it was concluded that LiDARs and radars being optimal for detection objects in short and long range with high accuracy, whilst to determine the object type, the camera is the most promising sensor. Object classification is a critical task in computer vision that has been studied for several decades (Karami et al. 2017), where gradient methods have exhibited good results. In 2012 the field completely changed as Krizhevsky et al. (2012) utilized deep neural networks to beat its competitors with a large margin for object classification on the the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al. 2015)

In its simplest form, a deep neural network is a computational graph with nodes connected in parallel, series and loops. The goal is to mimic an unknown function f, by exposing it to observed inputs and desired outputs from which the graph can learn what direction the information should flow (Ghahramani 2015). The resulting

deep network is a probabilistic approximation of the true function f and is considered adequate if the resulting graph can predict outcomes on unobserved data. The optimization process tries to maximize the distance of a given sample to the most nearby hyper-plane, thus the further away the sample lies, the more confident the model can be that the sample belongs to a specific class (Platt 1999).

Even though advances in image classification have sky-rocketed past years, initially through Krizhevsky et al. (2012) and then by a multitude of successors, it is still a controversy on how to properly report uncertainties in these networks. Instead, focus has been on achieving more accurate approximations of the underlying function f, by larger networks, new forms of layers and different optimization techniques (Alom et al. 2018). While advances in network structures are needed to achieve smaller, better or faster networks, the advances lack proper disclosures. A majority of publications lack proper insight into how to reproduce the results. According to Gundersen and Kjensmo (2018) only between 20% and 30% of publications fully document all factors that are required to recreate the results. Even with open sourcing of evaluation code, there exist cases where the resulting evaluation may vary depending on the model, which also lacks documentation of how to achieve said model.

As deep neural networks are achieved through an iterative optimization process, some prominent flaws exist as the optimization process cannot guarantee a global optimum, nor that the resulting network is a good representation of the training data. Amodei et al. (2016) summarizes five possible error modes from a reinforcement point of view, but applies to a majority of deep learning problems:

- **Avoiding negative side effects:** How can it be ensured that the system does not cause harm by pursuing its goals.
- **Avoiding reward hacking:** How can it be ensured that the system does not abuse glitches in the remaining systems to maximize its performance.
- **Scalable oversight:** How can the system learn to generalize towards events that are too expensive or infrequent to evaluate
- **Safe exploration:** How can the system explore surrounding events without causing bad repercussions.
- **Robustness to distributional shift:** How can we ensure that the system operates robust, when it operates outside of the training environment.

Since a trained model is in general focusing on one task, and one task alone, given an autonomous agent with its goal of reaching point A, it may take dangerous routes to fulfil this goal. The issue lies in how this kind of unwanted behavior can be excluded, without specifying everything the agent may or may not do (as the possible scenarios to handle quickly grows out of proportion). Furthermore, as it is improbable that the training set will contain all data variations, how does one allow the agent to explore surrounding states to the training distribution, even when the given state has a clear difference in behavior.

2.2 Safety verification and validation

A general definition of safety is the absence of harmful events that can have catastrophic consequences for the user (Avizienis et al. 2004). To achieve safety, the system goes through minimization of risk and epistemic uncertainty that are related to unwanted scenarios or events (Möller 2012). Epistemic uncertainty refers to the systematic uncertainty that is due to lack of complete data. This includes errors and not enough accurate sensory readings, as well as insufficient training sets unable to cover the full spectrum of scenarios. A broader definition of safety and safety concepts for components is given by Grunske et al. 2005. Their research define concepts such as risk, failure and hazards and discusses established techniques such a failure modes and effect analysis, and fault tree analysis for safety critical components. Furthermore, (Grunske et al. 2005) discusses safety analysis on a system-level, which allows for methods to consider the component as a black-box and only study its properties and effects on failures occurring on a system-level. Deep learning falls within this definition, thus robustness can be partly tested with the system-level approach, but still lacks error traceability and uncertainty estimations within the component.

How to properly handle uncertainty in deep neural networks is still under debate, as well as how a certification process can be established for these networks. Uncertainty in deep learning remains a controversy, as it is rarely covered in state of the art publications (Bertail et al. 2009). In addition, with regards to certification for large-scale deployment of autonomous systems, inherent issues have to be solved that incorporate infrequent failures, which will require a safety strategy that addresses multi disciplinary concerns including safety engineering, hardware reliability, testing and more (Koopman and Wagner 2017). Due to this, no starting point exists for a test oracle, therefore systems that include machine learning are a risk, as no testing requirements exist for the model nor for the training and validation dataset. Even though model accuracy is considered a statistical representation that hold over the test data, it does not guarantee that it holds against data processed during inference. Hence any claims that a system is completely safe have to argue that the training dataset contain data for every safety-critical situation (Nguyen et al. 2015).

It is important to stress that the testing challenges are not due to deep neural networks being considered as black boxes. There exist several studies on testing techniques, both for white and black box software testing (Nidhra and Dondeti 2012), and which to use at different stages of the software development. However, there is an inherent ambiguity when translating a product specification to a graph computation or dataset definition that are sufficiently detailed to act as the requirements specification. For machine learning applications, this translation needs to be elaborated upon. Varshney (2016) suggests four categories that need to be solved on an application-by-application basis to conduct engineering with safety in mind when it comes to machine learning. The categories include:

- **Inherently safe design:** Design the system such that the potential hazards are excluded rather than controlled.
- **Including safety reserves:** Allow for safety margins within the system, such as redundancy.
- **Safe fails:** Procedures such that the system remains safe as it fails.
- **Procedural safeguards:** Measures beyond the core functionality, such as operator training, audits or warnings.

With most of the suggestions, risk reduction is a key element. Additional frameworks or suggested methods have been proposed that cover specific cases of deep learning testing, such as Ribeiro et al. (2016) who proposed a novel explanation technique with the purpose that each prediction of any classifier has to have an interpretable and faithful design. The technique revolved around learning a spatial locality around the prediction by a linear model-agnostic explanation from which features and stimuli were matched. This can be further developed with backpropagation to find which feature is responsible for the classification.

Safe deployment refers to a solution or product being deployed in such a manner that it operates within safety margins and will not cause any safety related issues. Safety refers to the freedom from unacceptable risk of harmful events that can lead to physical injury or damage (*ISO 26262 Road vehicles — Functional safety* 2011). Additionally, the product has to act in a robust way, such that the function can persevere through stressful conditions as well as act in a reliable way and perform the required function for a desired period of time without failure ("IEEE Standard Glossary of Software Engineering Terminology" 1990).

ISO 26262 is the functional safety standard for road vehicles, consisting of 10 comprehensive parts that covers the full life cycle of safety related automotive functionality, including development, production and maintenance. The standard covers verification and validation in *Part 4: product development at the system level* and *Part 6: product development at the software level*. Salay et al. (2017) analysed Part 6 which consists of 75 software development techniques, that are applied at different stages of development. Their conclusion was that out of these, 34 applied at the unit level and the rest at architectural level. Furthermore, they conclude that the software development techniques are not suitable for solutions that incorporate deep neural networks or other computational graphs where little insight exists that supports traceability. A majority of the suggested techniques focus on enlightening the transparency of the written code through documentation, branch coverage, code reviews and similar. While these may have an effect on documentation of the deep neural network design, they do not manage to secure the actual model. In summary, Salay et al. (2017) suggest safety related systems that include deep neural networks will have to combat new hazards, new fault and failure modes, and new errors propagated through the training set.

In the beginning of 2019, the standard SOTIF (*ISO/PAS 21448:2019 - Road vehicles*

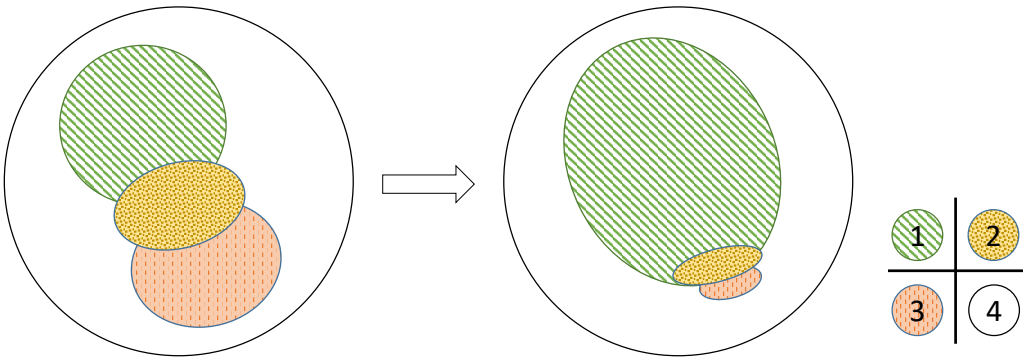

Figure 2.1: Visualization of the four categories representing all scenarios as defined in (*ISO/PAS 21448:2019 - Road vehicles — Safety of the intended functionality* 2019). The goal of the SOTIF process is to minimize unsafe states (2 and 3) by utilizing hazard identification and hazard mitigation techniques.

— *Safety of the intended functionality* 2019) was released to operate in parallel to ISO 26262. The goal of the standard is to meet the increasing need of securing the intended functionality by addressing insufficiencies in the intended functionality and to foresee potential misuse by users. In contrast to the traditional V-model that is applied in ISO 26262, SOTIF instead operates by defining four states that describe all possible states: 1) known safe states, 2) known unsafe states, 3) unknown unsafe states, and 4) unknown safe states. Thus, the goal of the standard is to apply hazard identification and hazard mitigation techniques to move states from 3) $\rightarrow$ 2) and 2) $\rightarrow$ 1) respectively, see Figure 2.1. Deep neural networks come with potential new hazards (Salay et al. 2017) as the method contains a multitude of uncontrolled parameters. However, what exactly constitutes good measures for hazard identification and mitigation for deep neural networks has yet to be defined.

Borg et al. (2019) conducted a review of verification and validation methods for machine learning in the automotive domain. Their review found several initial contributions both on the challenge specification domain as well as proposed solutions. The challenge fields were found to consist of state-space explosion, robustness, system engineering, model transparency, requirements specification, test specification and adversarial attacks. Furthermore, their review also included aspects from interviews and workshops, thereby enabling the discovery of open questions from the industrial domain of how to properly classify a system that includes deep neural networks. The open questions included whether or not deep neural networks should be considered a software unit or not, if the network errors should be considered hardware failures, how test coverage from training set comply with ISO 26262 and what key performance indicators should be used to quantify the quality of the training process to express the success of fulfilling requirement specifications and verification.

One reoccurring topic is verifying artificial intelligence through formal verification. Formal verification is the act of proving or disproving an algorithms correctness by analysing a property for a given system and environment. If a system is disproven, it is typically provided with a counterexample. Seshia et al. (2016) lists five major

challenges for achieving formal verification of artificial intelligence. These include environmental modelling, lack of formal specification, modelling systems that learn, computational engines for training, testing and verification, and lastly, correct-by-construction intelligent systems. To mitigate any of these challenges Seshia et al. (2016) suggest that each challenge is met with a set of principles that include introspect of the system to gather data of the environment, specify end-to-end behavior with quantitative metrics to formalize, develop abstractions for explanations of machine learning components, construct randomized and quantitative formal methods that can be applied on data generation, testing and verification, and lastly develop techniques for formal inductive synthesis for artificial intelligence systems.

2.3 Out-of-distribution detection

Out-of-distribution refers to the merged set of anomaly and novelty detection, of which both topics have been frequently discussed. Anomaly detection refers to detecting patterns in data that are not coherent with what which the algorithm was trained for (Chandola et al. 2009). These non-coherent occurrences are commonly referred to as anomalies, outliers, exceptions, or surprises depending on domain. Novelty detection act similarly to anomaly detection, with the addition that the samples are marked and used for further improvements of the algorithm. A common evaluation technique for detecting outliers has been one-class classification networks, where the networks sole target is to learn whether or not a sample is of interest (Khan and Madden 2010).

Chandola et al. (2009) conducted a literature survey on anomaly detection techniques that spanned multiple research areas and application domains. Regarding image processing, their findings comprise seven techniques: Mixture of models, regression, Bayesian networks, support vector machines, neural networks, clustering and nearest neighbour techniques. Furthermore, they conclude that one major challenge in the field is the large input size, which causes delays, especially when operating on video.

A review of novelty detection was conducted by Pimentel et al. (2014). Their survey summarizes different novelty detection methods into five different categories: probabilistic, distance based, reconstruction based, domain based and information-theoretic based. Furthermore, they conclude that novelty detection has a similar problem definition with one-class classification, and thus can be seen as one (Khan and Madden 2010). A one-class classification problem refers to a learning problem where data only exists for one class, in contrast to traditional classification tasks where the classifier tries to distinguish between two or more classes. In one-class classification, the learning attempts to minimize the boundary that encapsulate the training data.

More recent research has utilized the information output from deep neural networks to obtain a representation of uncertainty inside the network. Hendrycks and Gimpel

(2017) created a baseline analysis of uncertainty by analysing the softmax layer of a given sample. Since the softmax layer normalizes the output vector to a probability distribution, the anomaly score can be seen as the difference between the most likely class and the sum of the distribution. Hendrycks and Gimpel (2017) results show that out-of-distribution samples tend to have a different probability distribution, which allows the baseline algorithm to separate between in and outlier samples. Their tests were conducted on several pre-trained networks on prominent deep learning datasets and fields including computer vision and natural language processing.

The output layer was also used by Bendale and Boult (2016). Their approach consisted of fitting a Weibull distribution through meta-recognition of the output layer prior to the softmax activation. This approach allows for a likelihood estimation of each class, to be created from the training data, that can be used to estimate an outlier score. The computed outlier score is then compared to the most probable class, in which the sample can be excluded if the outlier score rises above the most probable class.

In contrast to only looking at the output layer, Liang et al. (2018) constructed a method hat utilizes backpropagation through the network. By assuming the most likely prediction is the correct one, and backpropagating based on this, a small perturbation can be added to the input image. The findings show that the small perturbation is more harmful for inlier samples, thus if the anomaly score remains the same, it is more likely to be an outlier.

A parallel field to detection of out-of-distribution samples is the research of adversarials and adversarial training. Both fields are similar in many regards, except that out-of-distribution samples refer to natural samples that can occur, while adversarials are custom-made samples purposely created to trick the network. Lee et al. (2018) constructed a method that studied a measure of probability density in the feature space of deep neural networks by utilizing a Mahalanobis distance to the density cluster. Their findings included experiments on state-of-the-art adversarial generators, which also showed dissimilar clusters compared to the true classes. Additional attempts have been focused on only detecting adversarial attacks, such as Zantedeschi et al. (2017) and Shaham et al. (2018), where both extend the deep neural network with a parallel adversarial detector with the goal of improving the robustness of the trained neural network.

Chapter 3

Research Approach

The aim of the research is to aid testing of deep learning models, to support safety verification of these models. This chapter describes the research approach and connects it to safety verification, out-of-distribution detection, and research validity.

3.1 Research philosophy

Several academic and industrial research facilities are following the recent advances in deep learning with interest. The past decade has seen image processing improvements to surpass human performance, opening up new business opportunities. As with most emerging technologies, the question remains how to properly test, verify and utilize this novel field.

Even though black box testing is well-established (Nidhra and Dondeti 2012), deep neural networks come with additional complexities that are not accounted for, such as bias in the gathered training sets and abstractions in the form of neural network parameters (Tommasi et al. 2017). Before deployment of systems that include deep neural networks, it is necessary to argue for safety measures and methods to properly evaluate uncertainties in the system implementation before it being deployed in safety-critical applications. As the field of deep learning is rather novel, the process of how to properly test and verify deep neural networks are yet to be defined.

In order to explain uncertainties of deep neural networks and be able to test them for safety critical applications, one has to perceive a pragmatic view on related test methodologies of deep neural networks and their experimental setup, as research has been studied on low scale examples. These initial results should act as a basis for future experiments to enhance the ability of verification of deep neural networks.

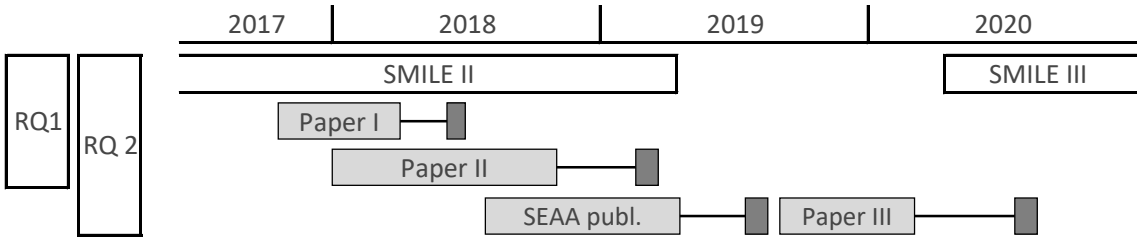

Figure 3.1: Timeline of publications during the research period.

3.2 Research activities

The activities conducted in this thesis are summarized in Figure 3.1. Additional collaboration has been conducted jointly with SMILE II, a research project running in parallel, funded by Sweden's innovation agency Vinnova[1]. The aim of the research project overlaps with this thesis, and thus yields good cooperation for experiments and publications.

The first publication, Paper I, aimed at established an understanding of which hurdles have to be addressed before one can argue for safe deployment of systems including deep neural networks. During the study, it became evident that rigorous changes to development of deep neural networks are required, including changes to training, testing and robustness evaluation of the model itself. The paper conducted two semi-structured interviews to explore the concern with deep learning and supported the claim that the research questions described in Chapter 1.2 were unresolved.

Following the initial paper, it became of interest to study how current outliers were handled with deep neural networks, and furthermore how solid outlier detection can be utilized in motivating integrity of a deep learning model. To further investigate, a study was conducted with the target of doing method comparison of outlier detection methods for deep neural networks. The methods were found through reference snowballing of a set of previously published outlier detection methods (Landgren and Tranheden 2018). The study found a plethora of methods of this kind, however the majority was either ad hoc or incomparable to related methods due to a difference in description, dataset distributions or metrics.

Since comparison of outlier detection performance on deep neural networks is new, no common comparison metrics have been concluded. When reviewing the best performing methods presented at the most prominent conferences, some common denominators could be established, which laid the foundation for Paper II. The paper delves deeper to assess how to conduct a structured evaluation of outlier detection methods. Furthermore, the paper establishes a good setup of datasets, metrics to report and graphs suitable for plotting a balanced comparison of methods. One observation after the paper was published was that variations in the training

[1]SMILE II - Safety analysis and verification/validation of MachIne LEarning based systems - Reference number 2017-03066

process could affect both the models prediction performance along with the ability to separate between inlier and outlier samples. Hence utilizing pre-training or training augmentation has to be reported thoroughly as to not jeopardize the reproducibility of the research. Going forward, the complete training procedure is recommended to be reported, ideally by open sourcing of complete code for reproducibility.

Due to the difficult comparison, especially with variations in training, the decision was made to test the most prominent methods found and compare their performance under controlled circumstances. This was first done in Henriksson et al. (2019) by utilizing the descriptions made in Paper II. The results were compelling, hence yielding the opportunity to extend the conference paper into Paper III, a journal paper currently under review. The paper investigated how alterations in the training process and models affect the possibility of detecting out-of-distribution samples. The experiments highlighted large variations of outlier detection rate, where models with similar accuracy performance had varying outlier detection performance. This highlight that the training process is of great importance, as well as small variations in said training can have large impact on end performance.

In parallel to the third paper, a parallel project ran with the aim to investigate if utilizing different goals in image processing alter the performance of the neural network (Edvardsson and Trieu 2019). The project tested if fusing depth-estimation predictions from a second neural networks would benefit the object classification neural network. Several combinations of fusing was tested and the changes in training performance documented. The results showed that the additional fusion neither improved or decreased the performance.

3.3 Safety verification and validation

Safety is one of the critical properties for automated systems, where testing is one of the measures conducted at different stages to achieve safety. A common scenario is the description of stakeholder targets, resulting in performance specifications. These specifications lay the foundation for the initial functional development to conduct whether or not the product is feasible, consistent and solvable within limitations. This comparison to specification is known as the verification part. The verification part is also referred to as the inner loop in the verification and validation process or the left side of the V-model. The outer loop in the process, or the right side of the V-model refers to the validation process, and is the process of evaluating that the product satisfies the requirements.

The common practice is an iterative approach of development and comparison of compatibility with requirements in the validation stage. Typically, requirements are shown as fulfilled by passing constructed tests or by introducing measures applied during the development process. However, there are instances of requirements that are unfeasible to quantify, or suffer from a state explosion in the functionality that

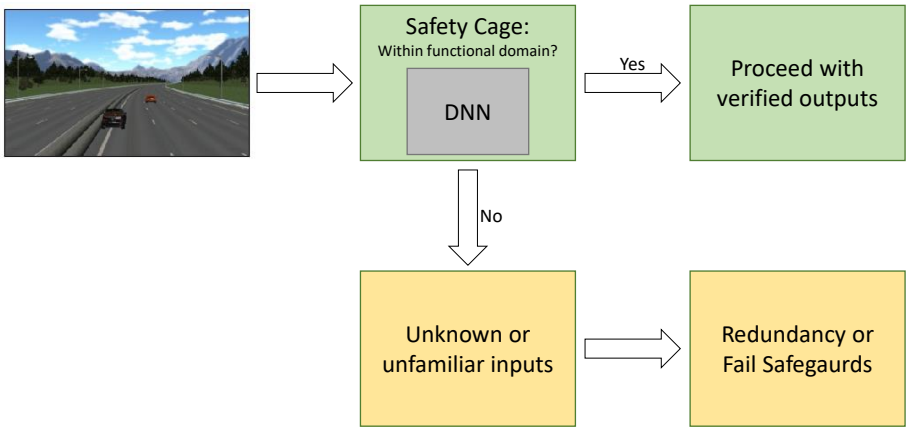

Figure 3.2: Borg et al. (2019) suggest the addition of a safety cage method that determines if the safety critical application needs to enter a fail-safe or safe degradation, as documented in ISO 26262.

yield the possible solution. As per example from the computer vision field, a camera based object detection system can be used with the requirement of detecting all pedestrians on the road. As a pedestrian is not possible to define as a function of image pixels, it is unfeasible to test all combinations of pixels. Additionally, due to the input space of images is so vast, there will always remain a risk of subspace that is not addressed by the testing.

As described in Chapter 3.3 and Figure 2.1, one risk minimization process is to utilize hazard identification and mitigation techniques to transform unknown unsafe states to known safe states. As the main issue described in this thesis relates to identification of unknown unsafe states. Even if the function at hand has gone through rigorous testing and hazard identifications, there are no guarantees that all harmful subsets outside of the functional domain have been found. For deep neural networks, Borg et al. (2019) suggest to extend the network with a safety cage, another term for outlier detector, as risk reduction. The aim of the safety cage is to reduce the amount of outlier scenarios that the system can exhibit. The suggested method, as depicted in Figure 3.2, highlights the idea of out-of-distribution detection as one of the measures to mitigate cases outside of the scope of the model. This kind of measure is suggestively one out of several measures to increase the robustness and transparency of deep neural networks.

3.4 Out-of-distribution detection

Out-of-distribution detection refers to the detection of samples that are not coherent with the data the model was trained on. In literature, this phenomenon is often referred to as anomaly detection or novelty detection, where the former refers to detect stochastic outliers that occur rarely, and the latter refers to outliers that typically follow a pattern which can be utilized in training. As any outlier can

contribute to harmful events, these are all grouped as out-of-distribution samples or outlier samples for short.

To start mitigating the potential risk of samples outside of the desired functional domain, it is important to quantify how far off a given sample is compared to the expected data. In this chapter, out-of-distribution detection is described, and how this measure can support risk mitigation.

A common practice within the field of safety critical development is that testing and verification of a function is not done by the original developer. This process allows testing of the product from a different point of view, thus allowing the testers to find errors in the functions that the developer did not consider. As deep neural networks are inherently challenging to interpret, extending the network with additional measures that support interpretation will improve the process of verifying the model.

Throughout the research period, the experiments have been designed with the assumption that for each sample that goes through a deep model, an *anomaly score* can be retrieved. This score is defined as a measure of uncertainty that is received by a functional approximation of model parameters, inputs or training data. A higher anomaly score indicates a larger probability of the sample being of an unknown distribution. By analyzing the score further, thresholds can be designed and iterated on to study how risk of false activations changes, or coverage in relation to set threshold.

Outlier detection of deep neural networks can be split into three categories, depending on which part of the model it operates on. The rationale for the split is to enable comparisons between methods within the category, as well as for comparing ensembles of methods operating in parallel. The categories to group anomaly scores consist of 1) using network internal information, i.e. methods utilizing information within the network, such as hidden activation vectors and output vectors of a given input, 2) external network information, i.e. methods that learn features from the training set, optimization procedure or design, and 3) adversarials, i.e. methods that allows backpropagation through the network to utilize the gradients for a given desired stimuli.

Outlier samples constitute a major hazard in autonomous systems (Borg et al. 2019); this falls in line with the SOTIF standard as both a hazard identification and mitigation strategy, as receiving an anomaly score for each sample can be used as a rejection criterion, but also as an indication of novel cases. Thus, research on how to properly design a formula of anomaly score needs more development. In addition, it has to be further studied how to use the anomaly score in safety argumentation for safety critical applications.

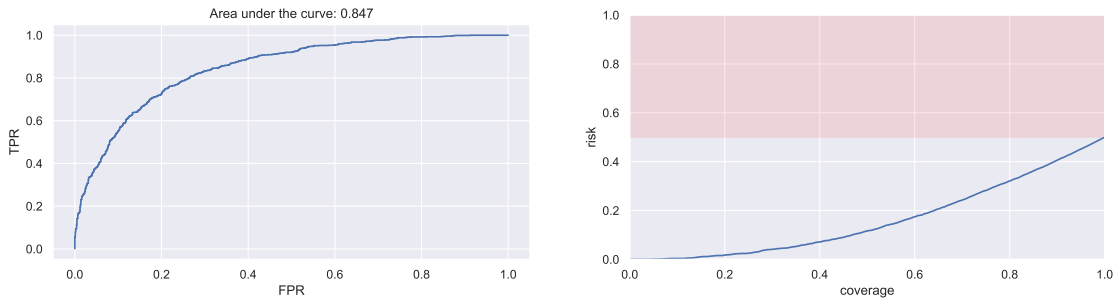

Figure 3.3: Plot visualization of the receiver operating characteristics curve (left) and the risk-coverage curve (right). The ROC-curve illustrates the separation ability of a binary classifier (defined as a supervisor in this thesis) as the discrimination threshold is varied. The risk-coverage curve incorporates the performance of the deep neural network to see how the coverage and risk of false classification changes with the discrimination threshold. The red area in the risk-coverage curve represents the outlier set added for testing purposes.

3.4.1 Comparison metrics

To evaluate the performance of the out-of-distribution detection, a plot visualization of true positive rate as a function of false positive rate is used. This plot is referred to as a receiver operating characteristics curve (ROC-curve) and originated as a prediction of correct radar signals during the Second World War, but has since spread to a plethora of fields, and has been extensively used in medical diagnostic research (Zweig and Campbell 1993). An example of a ROC-curve can be seen to the left in Figure 3.3.

When comparing datasets with similar characteristics, it is expected that the anomaly score will exhibit similarities. To restrict a model from operating on outliers in this case, the accepted anomaly threshold has to be set to a low level. This procedure will cause the system to exclude a majority of true positive cases as well, therefore forcing the system to operate on very few cases. To study this coverage change, the relation between coverage and classification risk can be studied (Geifman and El-Yaniv 2017), see Figure 3.3. By studying the risk-coverage curve, it can be determined how restrictive the out-of-distribution detection has to be to reduce prediction risk when distinguishing inlier from outlier data.

3.5 Research validity

As with all scientific research, we strive towards general understanding or conclusions that are in general drawn from experiments built on hypotheses or from summarizing existing research. However, it is important to understand that the conclusions drawn from the experiment conducted at this initial stage are not defining generic facts applicable to all scenarios, but rather act as a basis for new hypotheses that can be

used as a foundation for further investigation. Hence, the research conducted in this thesis has to be regarded as a cog in a larger machinery that aims at explaining the decision done in deep models trained through an optimization process.

This thesis has focused on computer vision through the input domain of images, experimenting with public datasets that are well known to the scientific community. Moving forward, the discoveries on these generally small-scale public datasets will be utilized for larger datasets but more importantly, data from real-life applications, such as vehicle perception. Furthermore, the thesis has focused on the topic of distinguishing between in- and out-of-distribution samples, for the sake of highlighting the effect when a system encounters novel inputs previously unseen to the model.

Whether out-of-distribution detection is the only extension needed to verify remains to be seen. While this kind of analysis provides well-rounded metrics and thereby enables assessment of input quality, it does not cover potential stability issues in the network. One common way to visualize stability issues are by utilizing gradients in the network, highlighting a specific set of pixel changes that manipulates the result from the model. In addition, an unexplored area is estimation of how *good* the training set really is. As with all compute models, the deep model is only as good as the training data, hence more research has to be conducted to establish a baseline for what constitutes a good training dataset.

One of the major issues recently regarding research in artificial intelligence is recreation of related work. With the surge of additional research in the field of computer vision, advances in method design or datasets selections is published frequently. While most of the related researchers have adopted open source of their code, it is a common problem that results are not easily replicable due to slight modifications in the setup. As the deep learning community is valuing conferences to similar extent as journals, the amount of submissions to top conferences has tenfolded over the past decade, without updating policies on full disclosure of experimental code setups. To give an example from the out-of-distribution field; one paper did not disclose that their model was of a pre-trained sort, thus when attempting to replicate their training approach described in their paper, the results from outlier detection became a magnitude worse while the model accuracy were similar. Due to event like this, this thesis publishes research results, as well as all experimental code for training and evaluation approaches.

Finally, while looking at the out-of-distribution detection results of appended and related papers, the results may appear grim. In the experiments, there is as high as a 10% risk of letting an unknown sample through. It is important to keep in mind that these experiments are only for one algorithm, which in general is one of several operating in parallel. The complete system will utilize an ensemble of methods as well as redundancy and fallback systems to increase reliability of the complete system at hand (Leaphart et al. 2005). With that said, risk numbers have to be reduced further, and experiments need to be extended to look at a broader spectrum of datasets and model types.

Chapter 4

Summary of Appended Papers

This chapter summarises the attached publications. Each paper is listed with scope, background, methodology, key results and conclusions in a brief fashion.

4.1 Paper I

This section gives a short summary of the paper *Automotive Safety and Machine Learning: Initial Results from a Study on How to Adapt the ISO 26262 Safety Standard*.

Short description

The paper involves two semi-structured interviews regarding the effect of machine learning in the automotive industry and its compliance with current standards. As machine learning becomes more prominent in various solutions, the importance of testing and verification of models becomes apparent. For safety critical applications, such as automotive development, testing and verifying is conducted in a structured manner by safety standard ISO 26262 (Kafka 2012). The goal of the safety standard is to guide a product through its development cycle, including from specification construction through design, implementation, integration, verification, validation and release.

Chapter 6 of the safety standard: *Product development at the software level* consists of 34 methods to enable identification of safety risks of the system and safety requirements that mitigate the risk of harmful situations. In short, an item (a system product or function) is required to go through rigorous analysis resulting in a set of *hazardous events* that are given an *Automotive Safety Integrity Level* (ASIL) ranging from ASIL A to ASIL D, where ASIL D represents the most severe classification.

From the hazard list, a set of safety goals are constructed to meet the hazardous events, followed by refining the goals to lower level safety requirements. Theses safety requirements are then allocated to components within the item, which then can be developed and validated for compliance with the safety requirements.

Results and conclusions

The vast majority of the 34 methods in ISO 26262 Chapter 6 exists to increase interpretability of the unit at hand. The remaining methods refer to fault injection and coverage metrics. While interpretability, fault injection and (neuron) coverage is relevant for deep learning, it falls outside of ISO 26262.

During the interviews conducted in the research, it became evident that standardized methods are required for three additional fields:

- **Model training phase:** Additional methods are required to ensure model training and generalization is properly tied to the designed functional space. For example, a neural network is a functional mapping for a given input space to a target output space, thus the functional mapping requires more methods that improve the understanding of the mapping itself.

- **Model sensitivity:** Several studies have shown that deep neural networks are prone to errors for small variations in the input space, for example adversarial samples. Similar to fault injection for items in ISO 26262, machine learning needs methods that verify the robustness of the models. One way to do this is to expose the model to known inputs with small perturbations and ensure that the model still performs satisfactory.

- **Test case design:** When applying ML algorithms, it is needed to more thoroughly design test cases. A common pitfall during development is that training and testing are conducted on homogeneous data, which results in false confidence of the model performance. Thus, test cases have to be diverse, and additionally, the testing phase needs methods explaining the uncertainty of the model as well as detecting when scenarios are outside of the function scope.

4.2 Paper II

This section gives a short summary of the paper *Towards Structured Evaluation of Deep Neural Network Supervisors.*

Short description

As stated in Paper I, model robustness and sensitivity towards samples outside of the scope is a major concern for deep neural networks. Samples outside of the scope of the network is considered outliers, which has been a frequent topic throughout the years. This has not been any different for deep learning or computer vision. Several methods attempting to find outliers have been presented in recent years with varying capabilities or restrictions of the model.

From a safety perspective, it is reasonable to consider a supervisor; a system running in parallel to the deep learning model analyzing the inputs and outputs of the model. The supervisors goal would be determining when an input sample does not resemble the training data, thus informing the rest of the system that the output is of lower certainty. By identifying these scenarios, a vehicle could enter a safe-mode, thus following the principles of graceful degradation, a common safety approach in safety critical applications described in ISO 26262. From related papers researched, it became evident that reporting results for a supervisor varies between publications. In general, there is a common ground of certain key performance indicators, but the vast majority of the testing setup differs in one or more settings that affect the supervisor performance. The differences include deep neural network model setups, training processes (either pre-trained or re-trained), training and testing datasets, and evaluation metrics, which all increases the difficulty of comparison.

Results and conclusions

The result of the paper is a summary of how to compare supervisors by describing the most prominent metrics. Furthermore, the paper demonstrates how these are applicable on two use-cases that include image classification datasets and driving scenarios respectively. The single assumption that is made on compared supervisors is that each can provide an anomaly score, a measurement indicating how dissimilar the sample is compared to the training set.

The paper defines in total 7 metrics that are recommended to use when comparing supervisors. From the 7 metrics, 5 are connected to the traditional Receiver Operating Characteristics curve, and 2 to the Risk-Coverage curve. The metrics and the corresponding description is given below:

- **AUROC:** Area under the Receiver Operating Characteristics curve is an overall metric of how well the supervisor distinguish between inlier and outlier samples.
- **AUPRC:** Area under the Precision Recall curve is a metric similar to AUROC, but takes into consideration imbalance in the dataset.

- **TPR05:** True positive rate at 5% of false negative rate is intended to analyse the slope of the of ROC-curve, where higher indicates better separation.
- **P95:** Precision at 95% recall. This metric highlights the precision error in the system when 95% of the true cases are rejected.
- **FNR95:** False negative rate at 95% false positive rate. This metric shows the remaining amount of outliers when the supervisor has rejected 95% of the inliers.
- **CBPL:** Coverage breakpoint at performance level indicates how restrictive the supervisor has to be to return to the same accuracy as achieved during training
- **CBFAD:** Coverage breakpoint at full anomaly detection metric reports at what coverage all outlier samples have been rejected.

In addition to the metrics, the paper suggested dataset combinations for testing as well as 4 plots supporting the metrics: The ROC-curve, PR-Curve, the histogram of the anomaly scores of the inlier and outlier distribution, and the risk-coverage curve.

4.3 Paper III

This section gives a short summary of the paper *Performance Analysis of Out-of-Distribution Detection on Trained Neural Networks*, an extension of Henriksson et al. (2019).

Short description

With the introduction of ISO/PAS 21448 - Safety of the Intended Functionality (SOTIF) it became apparent that functionality requires a structured way of detecting limits and potential hazardous situations caused by insufficiencies of the intended functionality. One common scenario for highlighting limitations of deep learning models is misclassifications occurring in object detectors. To achieve systematic risk reduction of functional insufficiencies, these learning systems require quantifiable methods for risk and uncertainties in the system.

Following the SOTIF process, we identify that out-of-distribution samples constitute a major hazard, i.e. samples that the system is not trained for and that differ significantly from the training distribution. Detecting these samples are referred to out-of-distribution detection. Throughout the paper series, the term supervisor has been used to refer to the method determining if a sample belongs to an outlier distribution or not. This paper compares 3 of these supervisors during the course of training to see how the ability to detect outliers changes as the deep neural network performance increases.

Results and conclusions

To compare supervisors, the experimental setup of this paper consisted of training 4 widely adopted deep neural network architectures on the CIFAR-10 dataset for 300 epochs. Every 10^{th} epoch 3 supervisors were tested from related research: ODIN (Liang et al. 2018), OpenMax (Bendale and Boult 2016) and Baseline (Hendrycks and Gimpel 2017), which were tested on 3 outlier datasets: Tiny ImageNet, SVHN and FakeData.

In contrast to the metrics presented in Paper II, this paper refines them further. Most notable is the replacement of ROC/PR curve metrics by a false positive rate metric (FPR95), which quantifies how many inliers have to be rejected to catch 95% of the outliers. In addition, the coverage breakpoint at full anomaly detection were excluded, due to all dataset combinations will contain minor overlapping, thus rendering the metric unnecessary. Instead, a new metric Cov10 was introduced, referring to coverage at 10% error rate, which was selected arbitrarily.

During the experiments it was found that the overall performance of supervisor, measured by the AUROC metric, increases as the model performance increases. For separation between similar types of images, the performance increases with model accuracy, whereas for Gaussian noise, the results are more scattered. This scattering illustrates the instability of the networks, and that there are combinations of samples and supervisors creating overlapping distributions that cannot be separated.

In contrast to reaching the same accuracy as achieved during training, the supervisor has to be more restrictive to ensure that the model returns to its original accuracy rate. The new metric avoids this penalization by assigning a given target accuracy, which is more in line with what requirement specifications will describe. As all networks achieve 8% error rate or less, the accepted error rate for this comparison is

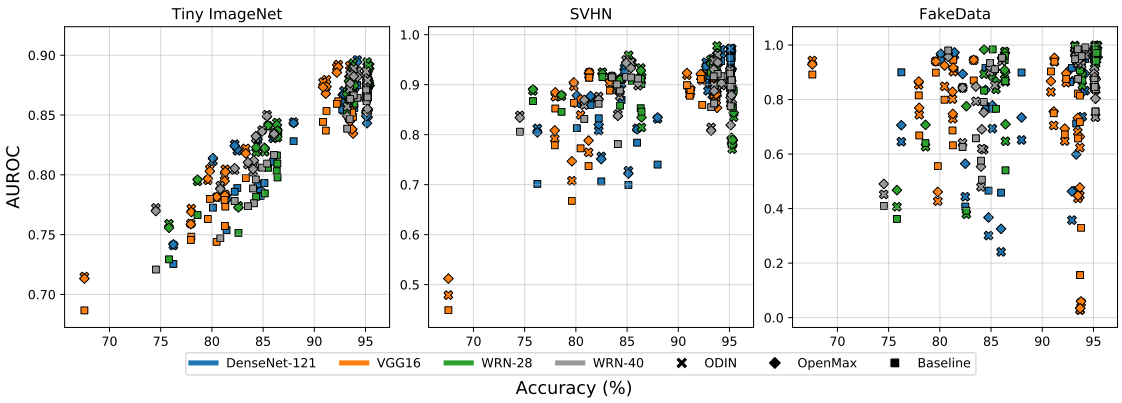

Figure 4.1: AUROC on the y-axis and model prediction accuracy on the x-axis for three experiments on different datasets. Each mark represents a supervisor evaluation. Coloring and marker type represents which model and supervisor were used, i.e an orange square refers to the model VGG16, tested with the Baseline algorithm.

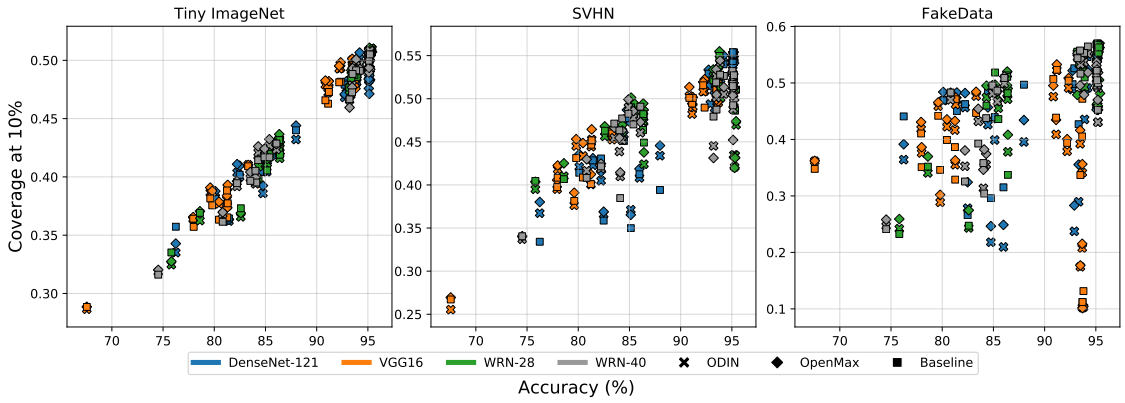

Figure 4.2: Cov10 on the y-axis and model prediction accuracy on the x-axis for three experiments on different datasets. Each mark represents a supervisor evaluation. Coloring and marker type represents which model and supervisor were used, i.e an orange square refers to the model VGG16, tested with the Baseline algorithm.

set to 10%, thus being reachable for all trained models.

In Figure 4.2, the new metric is shown for the epochs. While a positive trend of coverage is maintained,as the model performance increases it becomes apparent that the variation of coverage becomes larger when it is tested on an outlier dataset that is more diverse compared to the training set. For a given accuracy range, the coverage can vary with as much as 20%-points for real image comparison and even larger variations for Gaussian Noise.

One observation during the comparisons done in Figure 4.2 is that several models achieve coverage above 50%, even for some models with less than 90% accuracy. This indicates that the supervisor manages to distinguish between inlier samples that, if processed, would yield a misclassification but instead are rejected.

Chapter 5

Discussions

This chapter elaborates on drawn conclusions from the appended papers, as well as discusses additional measures for safety verification. First, testing extensions are discussed with outlier detection being considered as one extension, followed by a broader look at additional safety measures and research contribution. Last, potential future research directions are presented.

5.1 Testing extensions for deep learning

Even though deep neural networks are outperforming current state-of-the-art methods for several applications, they should still be excluded from safety critical applications, as traceability being a major issue for deep learning. Traceability is an important component in error handling and bug fixes, which could be improved for deep neural networks in several areas. The areas include improving quantification of quality, reliability and robustness, as discussed in the first appended paper. The areas that need improvement include evaluation of quality of the training and validation dataset, how to quantify model robustness and reliability, and how design of test cases have to change for quality control. These topics are interconnected as quantification depends on diversity of training and validation data. If the two datasets are intertwined, looking at model accuracy can be misleading, and thereby lack performance on general cases. Furthermore, how to quantify diversity or dataset coverage is still an open issue, and will most likely be ad hoc at best, as each measure will be dependent on application, deployment and training procedure.

A common safety strategy consist of monitoring inputs and outputs. There are already protocols of how to handle irrational data for trivial safety critical functions, for example through redundancy systems, fail-safes or degradation. Detecting when either inputs or outputs are irrational for camera imagery is more cumbersome, as pre-defined protocols do not exist due to images exhibit a vast magnitude of

additional parameters compared to few or several singular input sensors. Even though lack of protocols, the out-of-distribution detection method attempts this sort of monitoring.

During our experiments conducted in Paper II and Paper III, the outlier detection performance shows promising results, especially when operating on small-scale images, with less than a thousand pixels. For these experiments, rejections could be made even for samples inside of the training set where the classification network would otherwise have predicted wrong. However, when translating the same method to real-life large sized images, the results often resembles a tossup. This indicates that the monitoring principle does find an exclusion criteria, however it is not powerful enough to work for large-scale imagery. We interpret this to be due to the small scale images being smoother, with less noise and heavy variation which often occurs in ordinary pictures.

5.2 Using deep learning for testing extensions

Research of model robustness has resulted in several applications utilizing adversarial samples. For example, adversarial training has shown to improve robustness against outliers and adversarial samples, as it forces the training process to learn adversarial features, which seldom exist in the training set. In addition, similar to adversarial samples utilizing the gradient in the model to create erroneous samples, the technique can also improve outlier detection, as was shown by Liang et al. 2018. Utilizing gradients and additional parameters inside the network can be used for error detection by studying regular behavior of the model.

Another example utilizing adversarials are Parthasarathy et al. 2020, where the generative adversarial network structure was utilized to improve software-in-the-loop testing by adopting a linear interpolation technique that generates stimuli similar to a test case template. This stimuli allows a tester to write a scenario template, and the generative adversarial network creates a plethora of realistic test scenarios for extended test coverage.

Regarding out-of-distribution testing, as conducted in this thesis, deep learning approaches can also be considered as supervisors. In Paper II, an autoencoder model was used, whose sole purpose is to recreate the input data, and makes use of the difference between input and output as anomaly score. An autoencoder creates a functional mapping of the training set, which it utilizes to detect outliers during runtime. An observation regarding this setup is that it can be seen as a catch-22 situation, where one deep learning algorithm is monitoring another. This phenomenon needs to be solved by either verification protocols of deep learning models or argumentation when a model is safe.

5.3 Research quality

Research quality and validity, as described in Section 3.5, is an important aspect to refine and improve ones research. For the research conducted in this thesis, credibility comes from replicability and quality through generalization from the results. This thesis enables replicability as experimental code is open sourced[1].

Regarding the first paper included in this thesis, Paper I, the conclusions drawn are solid and backed by related research. However, the study only conducted two interviews, followed by workshop discussions, which allows for opinions to be seen as facts. The low numbers of participants in the first study can be seen as a threat to validity, hence repeating the study now with more practical results to present and discuss could be a possible extension to the study.

Regarding the remaining papers, Paper II and Paper III, the background research was far rigorous. Through the reference snowballing procedure, a solid literature base could be established, which laid the foundation for both papers, as well as Landgren and Tranheden 2018. Each publication covered a concern found in the literature base and extended upon each other. While the studies could have been more rigorous if a complete systematic literature review would be conducted as described by Kitchenham and Charters 2007, it was deemed unreliable, as the academic field has reused words as verification and validation with a different purpose, thus yielding thousands of unnecessary publications in the inclusion criteria.

5.4 Future Research

One critical aspect of verification is the generalization for wide set of domains. The experiments conducted in this thesis have shown proof-of-concepts on small scale scenarios and would benefit of being tested on real-world application data. Thus suggestively, the supervisor technique to detect outlier samples should be tested on safety critical applications, for example autonomous driving scenarios. Furthermore, more supervision algorithms have to be compared to understand what parts inside of a neural network needs to be controlled, or which parts can be used for analysis to understand when the model operates outside of its comfort zone.

One additional topic that needs to be defined is the complete verification process of deep learning. In this thesis, the out-of-distribution detection has been studied, as it can operate on a hazard identification and mitigation basis. This method needs several additional methods to operate in parallel, all described in a safety strategy of how to achieve a robust model. However, this safety strategy is not as straight-forward as described here. In addition to methods that describe the model,

[1] https://github.com/jenshenriksson/ood-comparison

the strategy needs to cover traceability issues when the error origins from another part of the system, perhaps a limitation in the sensors or other processing units.

Chapter 6

Conclusions

This thesis has investigated deep learning models, and what additions to testing are necessary before these models should be considered for safety critical applications.

The initial paper studied the first research question by describing which fields require additional testing procedures to combat the inherent challenges with deep neural networks. The fields discussed in the paper includes: quantification of quality on training and testing datasets, model robustness in the form of stability against small variations of input samples, and test case design that properly covers the scope of the deep neural network. When quantifiable measures exist for these three topics, as well as a safety strategy for the complete system, argumentation for safe deployment can be prepared and conducted.

The second research question has been shown through the extension of out-of-distribution detection for various cases. It is important to understand that no method is perfect and will reduce performance on inlier cases, as datasets are not disjoint. Nevertheless, the concept of out-of-distribution detection as a safety measure is valid as it can separate out large portions of outlier samples. The measure has shown promise of the public datasets with colored images.

Furthermore, when investigating the separation ability of various trained neural networks, additional problematic topics were found, such that the separation ability is dependent on the training procedure. This forces the safety argumentation to take the datasets, training procedures and model design into consideration when validating the performance of the actual model.

Bibliography

Alom, Md Zahangir et al. (2018). "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches". In: *Arxiv preprint [1803.01164]* (cit. on p. 6).

Amodei, Dario et al. (2016). "Concrete Problems in AI Safety". In: *Arxiv preprint [1606.06565]* (cit. on p. 6).

Avizienis, Algirdas et al. (2004). "Basic concepts and taxonomy of dependable and secure computing". In: *IEEE Transactions on Dependable and Secure Computing* (cit. on p. 7).

Bendale, Abhijit and Terrance E Boult (2016). "Towards open set deep networks". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (cit. on pp. 11, 25).

Bertail, Patrice, Stéphan Clémençon, and Nicolas Vayatis (2009). "On bootstrapping the ROC curve". In: *Advances in Neural Information Processing Systems* (cit. on p. 7).

Borg, Markus et al. (2019). "Safely Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry". In: *Journal of Automotive Software Engineering* (cit. on pp. 9, 16, 17).

Chandola, Varun, Arindam Banerjee, and Vipin Kumar (2009). "Anomaly Detection: A Survey". In: *ACM Computing Surveys (CSUR)* (cit. on p. 10).

Edvardsson, Annie and Martin Trieu (2019). *Deep learning and Fusion for Situational Awareness*. Master Thesis 257223, Chalmers University of Technology, Gothenburg (cit. on p. 15).

Erfani, Sarah M et al. (2016). "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning". In: *Pattern Recognition* (cit. on p. 4).

Geifman, Yonatan and Ran El-Yaniv (2017). "Selective classification for deep neural networks". In: *Advances in Neural Information Processing Systems* (cit. on p. 18).

Ghahramani, Zoubin (2015). "Probabilistic machine learning and artificial intelligence". In: *Nature* (cit. on p. 5).

Grunske, Lars, Bernhard Kaiser, and Ralf H Reussner (2005). "Specification and evaluation of safety properties in a component-based software engineering process". In: *Lecture Notes in Computer Science* (cit. on p. 7).

Gundersen, Odd Erik and Sigbjørn Kjensmo (2018). "State of the art: Reproducibility in artificial intelligence". In: *32nd AAAI Conference on Artificial Intelligence* (cit. on p. 6).

Hendrycks, Dan and Kevin Gimpel (2017). "A baseline for detecting misclassified and out-of-distribution examples in neural networks". In: *5th International Conference on Learning Representations* (cit. on pp. 10, 11, 25).

Henriksson, Jens et al. (2019). "Performance Analysis of Out-of-Distribution Detection on Various Trained Neural Networks". In: *45th Euromicro Conference on Software Engineering and Advanced Applications*, pp. 113–120 (cit. on pp. 15, 24).

"IEEE Standard Glossary of Software Engineering Terminology" (1990). In: *IEEE Std 610.12-1990*, pp. 1–84 (cit. on p. 8).

Isaksson-Hellman, Irene and Magdalena Lindman (2015). "Real-world performance of city safety based on Swedish insurance data". In: *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (cit. on p. 1).

ISO 26262 Road vehicles — Functional safety (2011). Standard. Geneva, CH: International Organization for Standardization (cit. on p. 8).

ISO/PAS 21448:2019 - Road vehicles — Safety of the intended functionality (2019). Standard. Geneva, CH: International Organization for Standardization (cit. on pp. 8, 9).

Kafka, Peter (2012). "The Automotive Standard ISO 26262, the innovative driver for enhanced safety assessment & technology for motor cars". In: *International Symposium on Safety Science and Technology* (cit. on p. 21).

Karami, Ebrahim, Siva Prasad, and Mohamed Shehata (2017). "Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images". In: *Arxiv preprint [1710.02726]* (cit. on p. 5).

Khan, Shehroz S and Michael G Madden (2010). "A survey of recent trends in one class classification". In: *Irish conference on artificial intelligence and cognitive science*, pp. 188–197 (cit. on p. 10).

Kitchenham, Barbara Ann and Stuart Charters (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering* (cit. on p. 29).

Koopman, Philip and Michael Wagner (2017). "Autonomous Vehicle Safety: An Interdisciplinary Challenge". In: *IEEE Intelligent Transportation Systems Magazine*, pp. 90–96 (cit. on p. 7).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems* 25, pp. 1097–1105 (cit. on pp. 2, 5, 6).

Landgren, Mattias and Ludwig Tranheden (2018). *Input Verification for Deep Neural Networks*. Master Thesis 255752, Chalmers University of Technology, Gothenburg (cit. on pp. 14, 29).

Leaphart, Eldon G. et al. (2005). "Survey of software failsafe techniques for safety-critical automotive applications". In: *SAE Technical Papers*, pp. 149–164 (cit. on p. 19).

Lee, Kimin et al. (2018). "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in Neural Information Processing Systems*, pp. 7167–7177 (cit. on p. 11).

Liang, Shiyu, Yixuan Li, and R Srikant (2018). "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *6th International Conference on Learning Representations* (cit. on pp. 11, 25, 28).

Möller, Niklas (2012). "The concepts of risk and safety". In: *Handbook of risk theory: epistemology, decision theory, ethics, and social implications of risk* (cit. on p. 7).

Nguyen, Anh, Jason Yosinski, and Jeff Clune (2015). "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 427–436 (cit. on p. 7).

Nidhra, Srinivas and Jagruthi Dondeti (2012). "Black Box and White Box Testing Techniques - A Literature Review". In: *International Journal of Embedded Systems and Applications*, pp. 29–50 (cit. on pp. 3, 7, 13).

Parthasarathy, Dhasarathy et al. (2020). "Controlled time series generation for automotive software-in-the-loop testing using GANs". In: *IEEE International Conference on Artificial Intelligence Testing* (cit. on p. 28).

Pendleton, Scott Drew et al. (2017). "Perception, planning, control, and coordination for autonomous vehicles". In: *Machines*, pp. 1–54 (cit. on pp. 1, 5).

Pimentel, Marco A.F. et al. (2014). "A review of novelty detection". In: *Signal Processing*, pp. 215–249 (cit. on p. 10).

Platt, John (1999). "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in large margin classifiers*, pp. 61–74 (cit. on p. 6).

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ""Why should I trust you?" Explaining the predictions of any classifier". In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144 (cit. on p. 8).

Rosique, Francisca et al. (Feb. 2019). "A systematic review of perception system and simulators for autonomous vehicles research". In: *Sensors* (cit. on pp. 1, 5).

Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision*, pp. 211–252 (cit. on pp. 2, 3, 5).

Salay, Rick, Rodrigo Queiroz, and Krzysztof Czarnecki (2017). "An Analysis of ISO 26262: Using Machine Learning Safely in Automotive Software". In: *Arxiv preprint [1709.02435.]* (cit. on pp. 8, 9).

Seshia, Sanjit A, Dorsa Sadigh, and S. Shankar Sastry (2016). "Towards Verified Artificial Intelligence". In: *Arxiv preprint [1606.08514]* (cit. on pp. 9, 10).

Shaham, Uri, Yutaro Yamada, and Sahand Negahban (2018). "Understanding adversarial training: Increasing local stability of supervised models through robust optimization". In: *Neurocomputing*, pp. 195–204 (cit. on p. 11).

Subramanya, Akshayvarun, Suraj Srinivas, and R. Venkatesh Babu (2017). "Confidence estimation in Deep Neural networks via density modelling". In: *Arxiv preprint [1707.07013]* (cit. on p. 3).

Szegedy, Christian et al. (2014). "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations* (cit. on p. 3).

Tommasi, Tatiana et al. (2017). "A Deeper Look at Dataset Bias". In: *Advances in Computer Vision and Pattern Recognition* (cit. on p. 13).

Varshney, Kush R (2016). "Engineering safety in machine learning". In: *Information Theory and Applications Workshop* (cit. on p. 7).

Zantedeschi, Valentina, Maria Irina Nicolae, and Ambrish Rawat (2017). "Efficient defenses against adversarial atacks". In: *AISec 2017 - Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, co-located with CCS 2017*, pp. 39–49 (cit. on p. 11).

Zweig, M. H. and G. Campbell (1993). "Receiver-operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine". In: *Clinical Chemistry*, pp. 561–577 (cit. on p. 18).