

# Deep Learning in Automotive: Challenges and Opportunities

Fabio Falcini and Giuseppe Lami<sup>(✉)</sup>

Consiglio Nazionale delle Ricerche,  
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo",  
Via Moruzzi 1, 56124 Pisa, Italy  
{fabio.falcini, giuseppe.lami}@isti.cnr.it

**Abstract.** The interest of the automotive industry in deep-learning-based technology is growing and related applications are going to be pervasively used in the modern automobiles. Automotive is a domain where different standards addressing the software development process apply, as Automotive SPICE and, for functional safety relevant products, ISO 26262. So, in the automotive software engineering community, the awareness of the need to integrate deep-learning-based development with development approaches derived from these standards is growing, at the technical, methodological, and cultural levels. This paper starts from a lifecycle for deep-learning-based development defined by the authors, called W-model, and addresses the issue of the applicability of Automotive SPICE to deep-learning-based developments. A conceptual mapping between Automotive SPICE and the deep learning lifecycles phases is provided in this paper with the aim of highlighting the open issues related to the applicability of automotive software development standards to deep learning.

**Keywords:** Deep learning · Automotive SPICE · Software development lifecycle · ADAS (advanced driver assistance systems) · W model

## 1 Introduction

In the last 20 years automotive witnessed a continuous trend to innovation. In such a period of time, cars moved from being basically mechanical/electromechanical devices, to being very complex vehicles where electronics and software are playing a predominant role. Electronics is so pervasive in today's cars that almost all the main features and functionalities are controlled by software.

Because software, car manufacturers today have to face several challenges. Mastering functional safety and cyber-security issues are the today's hottest challenges for car manufacturers (as well as the daily work for engineers and managers). Nevertheless, while the functional safety and cyber-security are still two open battlefronts, a new one is going to be open and it will determine in the next future a revolutionary change in the way cars are designed, developed and used: the autonomous driving.

Autonomous driving relies on advanced driver assistance systems (ADAS) [2], these technologies often are based on Artificial Intelligence (AI). Injecting AI in the

automotive industry will require deep changes in terms of skills, technologies, and development paradigms.

Today Deep Learning, a special type of AI technology, is considered the most suitable approach to face the complexity of autonomous driving applications. Actually, deep learning is considered by automotive companies as a viable technology not only for implementing autonomous driving, but also for improving other functional domains such as engine management [3] and vehicle cyber-security assurance [4].

Several standards are applicable in automotive. In particular, in the last years standards addressing software development have been released and applied, the most relevant and impacting are Automotive SPICE [10] and ISO 26262 [11] standards. The Automotive SPICE standard provides a framework at process-level that disciplines, at high level of abstraction, the software development activities and allows their capability assessment in matching pre-defined sets of process requirements. ISO 26262, titled “Road vehicles – Functional safety” and released in late 2011, target safety critical development and its scope is expectedly not limited to software engineering. Both standards, as far as the software is concerned, rely conceptually on the traditional development paradigm: the V-model. The V-model is far away from being applicable in the case of development of Artificial Neural Networks for AI applications. That represents an open issue for developers of AI-based automotive applications and devices and for car manufacturers too.

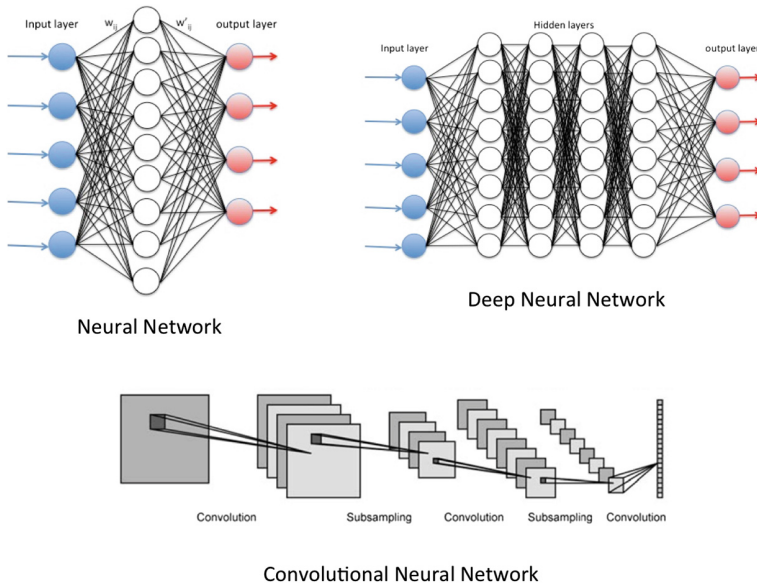
In this paper the authors address the topic of the applicability of the existing reference standards for automotive (in particular Automotive SPICE) to the development of AI-based applications (with special reference to the Deep Learning ones).

The paper is structured as follows: in Sect. 2 an overview of the Deep Learning is provided; in Sect. 3 an authors’ defined development paradigm for Deep Learning called W-model is introduced and discussed. In Sect. 4 the applicability of Automotive SPICE to Deep Learning development is systematically discussed in order to point out the existing lacks and open issues. In Sect. 5 conclusions are finally provided.

## 2 Deep Learning Overview

The principal characteristic of artificial neural networks is the ability to improve their problem-solving capabilities through a “learning” process triggered by exemplary input [5]. This feature makes the use of artificial neural networks particularly suitable in scenarios in which there is no detailed, complete or predictable information about the problem as usually in automotive driving situations. Another relevant feature is their parallel structure, which benefits from the use of powerful hardware to obtain timely and thus usable computation results.

Deep learning, which is synonymous of deep neural network (DNN), is a specific kind of artificial neural network able to model complex non-linear relationships using multiple hidden layers of units between the input and output layer (Fig. 1 – right). Deep learning excels on finding patterns when input are massive analog data – this means not a few numbers in a tabular format, but instead images of pixel data or audio data. Until 2006 advances, DNNs were outperformed by shallow neural network that relied on feature engineering which is the embedding of the domain knowledge in the solution design.



**Fig. 1.** Examples of neural networks.

The structure of a deep neural network is flexible and can be customized by selecting attributes such as the number of hidden layers, number of units (also called “nodes”) per layer, number of connections per unit etc. These attributes, known as hyper-parameters, define the structure as well as the behavior of a deep learning-based system.

Summarizing, the fundamental characteristics of deep learning are:

- Input-output mapping through the learning process
- Nonlinearity - DNNs are composed of an interconnection of nonlinear computational elements (a.k.a. neurons or nodes)
- Adaptation capability - DNNs have a built-in adaptation ability to the changes in the environment
- Fault tolerance - due to DNN distributed nature, localized faults in hidden layers leads to a degradation of performance rather than a system failure.

Their processing capability is stored in the inter-unit connection weight obtained by a process of adaptation to a set of training patterns.

Convolutional neural networks (CNN), also known as ConvNets, are a type of deep neural network (Fig. 1 – bottom) conceived to manage data in form of arrays with some degree of spatial structure [6, 7].

They are designed to emulate the behavior of a visual cortex and perform very well on visual recognition tasks because the convolution operation (in shape of matrix products) itself is capable of capturing the features of images. Convolutional neural networks have special layers called convolutional layers and sampling layers that allow the encoding of the images properties [8].

Convolutional Neural Networks, for their characteristics such as input data segmentation and a high degree of parametrization (up to hundreds of thousands), are of special interest for visual applications in the automotive context such as object detection, vehicle detection, road marking detection and more.

## 2.1 Existing Solutions

Google is making remarkable and highly visible investments in the development of autonomous vehicles. Its prototypal self-driving vehicles embed deep learning based technology already able to detect pedestrians in various and challenging scenarios. Google deep learning systems have achieved outstanding performance, making the error rate for machine vision lower than the one of a human being (5% error rate is human benchmark). This achievement, also due to new hardware architectures using multiple Graphics Processing Units (GPU), is pushing the migration of features based on traditional image processing technology to deep learning-based solutions.

This is just the beginning and, even so, remarkable elements of artificial intelligence are already available in circulating vehicles. In the ADAS domain, Tesla is reported to feature onboard the implementation of a neural network functionality for vision, sonar and radar processing that runs on the powerful NVIDIA DPX2 processor in the driving control unit [13].

Several other suppliers are already active players. The Intelligent Surround View (ISV) system, by AdasWorks, is an example of an ADAS neural network-based implementation, which processes the environment around the car using the visual information coming from several cameras [14]. In addition, DENSO R&D labs together with other important companies R&D labs are actively researching in this direction.

In the infotainment domain, the 2015 BMW 7 Series is reported to be the first car to feature an innovative voice recognition solution based on deep learning technology that works also in absence of the wireless car connectivity [15].

Other automotive applications of deep learning, currently under development, range from engine fault diagnosis and emissions management to detection of vehicle network intrusion.

From an hardware perspective, the level of electronic support required to embed deep learning in high-performance and safety-related automotive applications is so demanding that companies are aggressively developing new generations of chips: an example is the Mobileye's upcoming cutting-edge EyeQ 5 proprietary chip. In regard of commercially available electronics components, Intel is positioning with the new Xeon Phi chip to compete in this market, which has been so far ruled by Nvidia Tegra chip.

## 3 The W Model for Deep Learning Software Development

Software construction (i.e. the mere activity of software coding) is relatively a simple job in the case of Deep Learning. Basically it consists in developing the code of the nodes belonging to the different layers the neural network is composed of. In fact, the sophisticated features implemented by the neural network are not the result of a coding activity performed by software engineers, more appropriately they can be considered as

the result of the learning process of the network performed using a general purpose learning procedure. That represent a novelty with respect the software development practice in automotive today.

The software side of DNN development is a highly iterative activity composed by a stream of steps in an end-to-end fashion [9], as shown in Fig. 2.



**Fig. 2.** DNN development workflow.

Compared with the traditional approaches, deep learning development process needs the support of empirical design choices driven by heuristics. Development often start from well-known learning algorithms, which have been proven effective in comparable problems or domains, since the understanding of the result of learning process is difficult to be grasped and thus managed.

Expected DNN requirements also include performance demands, expressed in terms of statistical benchmarking of the DNN functional behavior (i.e. error rate), that are carefully targeted during the DNN validation phase.

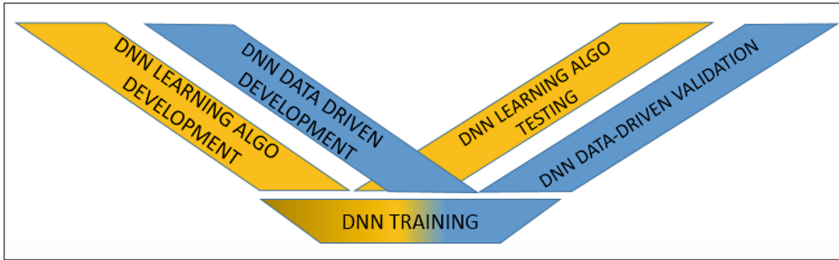
Automotive software engineering, while welcomes innovation and outstanding functional performances, is strict in its request for a robust and predictable development cycle coming from the demand of compliance with standards as Automotive SPICE. The authors introduced in [1] a new lifecycle called W-model with the aim of providing a contribution in placing deep learning in a more controlled V-model perspective to address a lengthy list of challenges, such as requirements criteria for training, validation and test data sets, criteria for the training data pre-processing, management of very large sets of parameters and much more.

The introduction of a more structured conception of the deep learning lifecycle is instrumental to reach a controlled development approach that cannot be addressed by the mere functional benchmarking. However, deep learning intrinsically introduces its specific features (not completely fitting with the V-model) for software development.

The introduction of the W-model is essential because the central role played by data in this context (e.g. for DNN training and training validation). To support it we introduce the term “programming by example” to highlight the importance of data in developing systems based on deep learning technology.

The deep learning W-model is a framework lifecycle that conceptually integrates a V model for data development in the ‘traditional’ V-like perspective (Fig. 3).

The W model acknowledges that deep learning is driven by software development as well as data development. The design and the creation of training/validation/test data sets, together with their exploitation, are crucial development phases because the DNN’s functional behavior is the combined result of its architectural structure and of its automatic adaptation through a training process.



**Fig. 3.** The W-model for deep learning.

Deep learning moves away from feature engineering by definition and this element makes the W-model an appropriate and useful representation of such a sophisticated paradigm.

## 4 Deep Learning vs. Automotive SPICE 3.0

The software development process for on-board automotive ECUs is subject to proprietary OEMs norms as well as several international standards. Among them, the most relevant and influential standards for deep learning are Automotive SPICE [10] and ISO 26262 [11]. Needless to say that these standards are still far from addressing it with dedicated statements.

The Automotive SPICE standard - SPICE stands for Software Process Improvement and Capability dEtermination - provides a process framework that disciplines, at high level of abstraction, the software development activities and allows their capability assessment in matching pre-defined sets of numerous process requirements. ISO 26262 targets safety-related development and its scope expectedly includes system, hardware and software engineering. It is important to remark that the ISO 26262 standard already addresses configuration and calibration data, even though this aspect is an order of magnitude simpler and plainer than the development of DNN data sets.

Both standards, as far as the software is concerned, rely conceptually on the traditional development lifecycle: the V-model.

It is also very relevant for deep learning the ISO PAS “Safety of the Intended Functionality (SotIF)” [12] that is currently in advanced development stage. This ISO document addresses the fact that for some ADAS applications there can be safety violations with a system free from faults - for example a false-positive detection by a radar of an obstacle for the vehicle – because it is extremely problematic to develop systems able to address every possible scenarios.

In the following, we intend deepening the correlation between software engineering processes of Automotive SPICE and the W model for deep learning in order to contribute in the harmonization between deep learning development and the state-of-the-practice in automotive software development. To do that, Automotive SPICE processes related to software development are compared with the Deep Learning development phases with the aim of discussing the applicability of the Automotive SPICE process reference model

in the case of Deep Learning as well as highlighting possible gaps and open issues between the two schemes.

The processes we consider are those related to software development that, in the Automotive SPICE standard, are identified by the prefix SWE:

- SWE.1: Software Requirements Analysis process
- SWE.2: Software Architectural Design
- SWE.3: Software Detailed Design and Unit Construction
- SWE.4: Software Unit Verification
- SWE.5: Software Integration and Integration Test
- SWE.6: Software Qualification Test

To additional details in terms of process purpose, base practices and work products, refer to [10]. According to the W-model discussed in Sect. 3, the development of software of Neural Networks (and in particular of Deep Neural Networks) can be seen as decomposed into two different branches:

1. the DNN Learning Algorithm Development that addresses the development of the code associated to the nodes belonging to the different layers the network is composed of;
2. the DNN Data Driven Development that addresses the data-driven developments consisting in the training of the network by data.

Figure 4 provides, in a graphical way, the description of DNN development phases divided into the two development branches (i.e. the DNN Learning Algorithm Development and the DNN Data Driven Development). It is to be noticed that the “Execution of DNN Training” and “DNN Validation” phases are common to both branches. In fact, their execution and the related effects affect both the learning algorithm and the training and validation data.

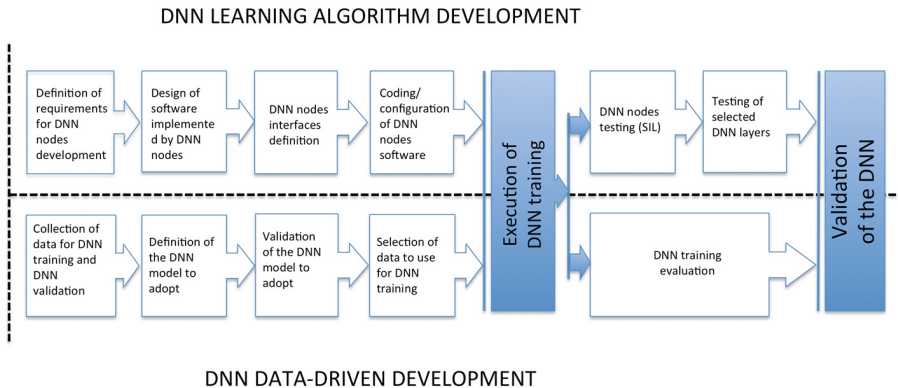


Fig. 4. DNN learning algorithm and DNN data-driven development phases.

In the following tables the singles phases the DNN Learning Algorithm Development and DNN Data Driven Development are composed of are mapped on the six SWE processes from Automotive SPICE.

**Table 1.** Mapping between automotive SPICE processes and DNN learning algorithm development with related addressability indication.

Automotive SPICE process	DNN learning algorithm development phase	Addressability
SWE.1	- Definition of requirements for DNN nodes development	Yes
SWE.2	- Selection or design of the algorithms implemented by DNN nodes	Yes
SWE.3	- Specification of DNN nodes Interfaces - Coding of nodes' software - Configuration of the DNN (i.e. setup of parameters/weights of the DNN) - Execution of DNN training	Yes Yes Yes No
SWE.4	- DNN nodes testing	Yes
SWE.5	- Testing of selected DNN layers	Yes
SWE.6	- Validation of DNN	Yes

In Table 1 the mapping between the SWE processes and the DNN learning algorithm development phases is provided along with a two-values column (Yes or No) indicating the applicability of the current contents of the SWE process (in terms of Base Practices and Work Products) to those phases. If the value on the column “Applicable” is “Yes” it means that the current version of the related process is applicable for DNN learning algorithm development without significant changes, if the value is “No” it means that the current version is not suitable to address the peculiarities of that phase.

In Table 2, the same is reproduced for the mapping between SWE process of Automotive SPICE and the Data Driven Development phases.

While it is possible the make a conceptual mapping between the W model and the SWE-class processes of Automotive SPICE (as shown above), the content, in terms of Base Practices and Work Products, of the current version Automotive SPICE (ver. 3.0) is far from being applicable in DNN development.

The Automotive SPICE model has been conceived taking into account the V-model. As a consequence of that the Base Practices and Work Products of the processes belonging to the SWE process group reflect the ‘traditional’ way to develop software.

To make Automotive SPICE applicable to projects developing Deep Learning applications, the current set of Base Practices and Work Products simply needs to be re-defined to make it meaningful and aligned with Deep Learning development.

While making the current version of Automotive SPICE applicable to the DNN Learning Algorithm Development can be considered a relatively light job, on the contrary, making it suitable for the DNN Data Driven Development needs a re-definition of the performance indicators (i.e. Base Practices and Work Products).



**Table 2.** Mapping between automotive SPICE processes and DNN data driven development with related addressability indication

Automotive SPICE process	DNN data driven development phase	Addressability
SWE.1	- Identification and storing of data to be used for the DNN training and DNN validation	No
SWE.2	- Selection or Design of the prediction model (i.e. definition of the DNN model to be adopted) - Validation of the DNN model to be adopted - Selection of data to be used for DNN training	No No No
SWE.3	- Execution of DNN training	No
SWE.4	- none	No
SWE.5	- Execution of DNN training - DNN training evaluation	No No
SWE.6	- Validation of DNN	Yes

## 5 Conclusions

The technology growth in computation capability and the availability of huge amount of data are pushing Artificial Intelligence applications, and in particular Deep Learning applications, to be massively adopted in automotive.

Automotive is a domain that in the last two decades witnessed impressive technological advancements accompanied by a progressive introduction of standards and norms that brought discipline and uniformity in development paradigms of software-intensive systems. Those paradigms have been, so far, based on the so-called V-model.

Automotive deep-learning-based applications and components are developed according to paradigms and models different that the V-model. That introduces new challenges in automotive in terms of applicability of the existing standards.

The penetration of Electronic Control Unit (ECU) hosting artificial intelligence is supposed to grow in a steady and substantial way according to endorsed market researches. The expected volumes are so important to reinforce the need to analyze the peculiarities and to integrate deep learning in the development lifecycle of automotive electronics systems.

This paper addressed the issue of the applicability of Automotive SPICE standard to deep-learning-based developments. The authors defined in a previous work the so-called W-model, a reference development model for deep learning able to represent that deep learning development is driven by software development as well as data development. A conceptual mapping between the W-model and Automotive SPICE is provided in this paper, with the aim of systematically highlight possible inconsistencies. The evidences of such a conceptual mapping show that Automotive SPICE can be considered generally applicable only for the part of deep learning development dealing with the algorithmic implementation of nodes. The part dealing with data driven development is far from being mapped to the Automotive SPICE process model.

From such a situation we can argue that, because of a large amount of deep-learning-based applications are expected to be part of vehicles in the next few years, a strict demand of new standards (or extensions of existing ones) exists. To respond to such a demand the automotive community should start to address this challenge in a similar way it has been able to face other technological and methodological challenges in the recent past.

## References

1. Falcini, F., Lami, G., Costanza, A.M.: Deep learning in automotive software. *IEEE Softw.* **34**(3), 56–63 (2017). doi:[10.1109/MS.2017.79](https://doi.org/10.1109/MS.2017.79)
2. Levinson, J., et al.: Towards fully autonomous driving: systems and algorithms. In: Proceedings of 2011 IEEE Intelligent Vehicles Symposium (IV11) (2011). [ieeexplore.ieee.org/document/5940562](http://ieeexplore.ieee.org/document/5940562)
3. Parlak, A., et al.: Application of artificial neural network to predict specific fuel consumption and exhaust temperature for a diesel engine. *Appl. Therm. Eng.* **26**(8–9), 824–828 (2006)
4. Kang, M.-J., Kang, J.-W.: Intrusion detection system using deep neural network for in-vehicle network security. *PLoS ONE*, **11**(6), e0155781 (2016). [journals.plos.org/plosone/article?id=10.1371/journal.pone.0155781](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0155781)
5. Schmidhuber, J.: Deep learning in neural networks: an overview. *Neural Netw. J.* **61**, 85–117 (2015)
6. Haykin, S.: *Neural Networks and Learning Machines*. Prentice-Hall, New York (2009)
7. Credi, J.: Traffic sign classification with deep convolutional neural networks. Master's thesis, Department of Applied Mechanics, Chalmers University of Technology (2016)
8. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
9. Miao, H., et al.: ModelHub: lifecycle management for deep learning. University of Maryland (2015). [www.cs.umd.edu/class/spring2016/cmssc396h/downloads/modelhub.pdf](http://www.cs.umd.edu/class/spring2016/cmssc396h/downloads/modelhub.pdf)
10. Automotive SPICE Process Assessment/Reference Model, ver. 3.0, Verband der Automobilindustrie (2015). [www.automotivespice.com/leadadmin/software-download/Automotive\\_SPICE\\_PAM\\_30.pdf](http://www.automotivespice.com/leadadmin/software-download/Automotive_SPICE_PAM_30.pdf)
11. ISO 26262—Road Vehicles—Functional Safety—Part 1: Vocabulary, Int'l Standard Org. (2011)
12. ISO/AWI PAS 21448—Road Vehicles—Safety of the Intended Functionality, Int'l Standard Org. (2016)
13. Pressman, M.: Inside Nvidia's New Self-Driving Supercomputer Powering Tesla's Autopilot. *CleanTechnica*, 25 October 2016. [cleantechnica.com/2016/10/25/inside-nvidias-new-self-driving-supercomputer-powering-teslas-autopilot](http://cleantechnica.com/2016/10/25/inside-nvidias-new-self-driving-supercomputer-powering-teslas-autopilot)
14. CEVA and AdasWorks to Demonstrate Free Space Detection for Autonomous Driving at AutoSens Conference 2016, CEVA, 15 September 2016. [ceva-dsp.mediaroom.com/2016-09-15-CEVA-and-AdasWorks-to-Demonstrate-Free-Space-Detection-for-Autonomous-Driving-at-AutoSens-Conference-2016](http://ceva-dsp.mediaroom.com/2016-09-15-CEVA-and-AdasWorks-to-Demonstrate-Free-Space-Detection-for-Autonomous-Driving-at-AutoSens-Conference-2016)
15. De Ambroggi, L.: Artificial Intelligence Systems for Autonomous Driving on the Rise, IHS Says, IHS, 13 June 2016. [technology.ihs.com/579746artificial-intelligence-systems-for-autonomous-driving-on-the-rise-ihs-says](http://technology.ihs.com/579746artificial-intelligence-systems-for-autonomous-driving-on-the-rise-ihs-says)