Moving towards explainable AI by utilizing a holistic lifecycle assurance framework

Lukas Klose M.Sc., Dr.-Ing. Lutz Kelch

Keysight Technologies Deutschland GmbH
Automotive & Energy Solutions
Mittelweg 7
D-38106 Braunschweig
lukas.klose@keysight.com
lutz.kelch@keysight.com

Abstract: As a result of the exponential increase in complexity of rule-based systems processing a large amount of data, the move to AI and data driven processes is inevitable.

This process requires a shift of testing and assurance paradigms.

Implementing AI models in automotive systems requires a novel approach of testing and documenting the process of decision making. Depending on the level of safety, different levels of interpretability and test coverage, as well as documentation are required. Fueled by the proposed Acts on AI, the need for explanations of the system is imminent.

Keysight Technologies, a market leader in test and measurement, proposes a novel holistic framework combining tests covering the complete lifecycle of an AI system, from initial conception to the last day of deployment, standardizing test executions and providing insights into the decision-making process of the Model under Test (MuT). While aligned with requirements stated by governmental institutions, the proposed solution enables AI Engineers as well as Domain Experts to join forces and improve their models based on insights and recommendations offered by the framework.

Separating the lifecycle into five steps (Problem understanding and Data Analysis, Feature Engineering, Model Training, Deep Model Evaluation, Model Inference) allows integration of the framework into different software engineering processes, like V-Model or Waterfall, necessary for safety critical systems, and platforms.

1 Introduction

The recent achievements in Artificial Intelligence (AI), especially in the realm of Machine Learning (ML) will accelerate the already ongoing inclusion of AI driven Advanced Driver Assistance Systems (ADAS) and fully Automated Driving (AD) systems.

With the adoption of AI into high-risk applications, a novel approach for testing those components must be implemented to ensure safe and secure operation, as well as strengthen the confidence of users in the reliability of the system.

Other than classic rule-based systems or decision trees, Neural Networks (NN) are black boxes by design. During the training process, a NN learns relations between the input data and the desired output by adjusting the weights between the different neurons. Depending on the design of the network (influenced by the complexity of the task which shall be solved using NNs), this number of weights can vary between several hundred and up to billions of parameters. The greater the number of parameters, the more difficult it is to interpret them and draw conclusions about the model behavior. However, the understanding of the inner working of a NN is crucial when relying on the correctness of its decision with human life at stake, especially in but not limited to corner case scenarios.

Therefore, the goal is to achieve interpretability and explainability of NNs to ensure their desired behavior even in critical situations. Moving towards general explainability, one will have to answer the following questions:

1. Is the available data relevant and sufficient to train the network?
2. At which point is the network fully and optimally trained?
3. How does the network deal with corner cases and scenarios not included in the training data?
4. And after deployment: Is the network still fitted to the current domain or did the domain change, rendering the current network unable to handle its task?

In the following sections, we will present a framework for answering those questions mentioned above and helping to speed up the development of high-quality AI solutions while cutting costs by providing support for engineers during the entire lifecycle of the ML model.

2 Problems of current explainability methods

Explainability and interpretability are necessary to achieve a deep understanding of the behavior, reliability and limitations of AI systems. During operation in ADAS/AD systems, the AI models will have to deal with unknown, e.g. not trained, scenarios. Especially in this domain, corner cases and extreme situations will be safety critical. Like in classical testing, it is not possible to evaluate every possible situation. Therefore, one must understand how and why the AI model decides the way it does. Utilizing local explainability methods can help to determine how different features influenced the decision of the AI model and to understand the cause of failures in training and deployment.

However, explaining faults after the fact is not sufficient. The goal must be to understand the AI system during the training and evaluation process and learn its limitations. This requires an understanding of the quality of the network, the training state and adaptation to the training data, as well as thorough tests on high quality test data unknown to the network.

Currently, no general method for achieving complete explainability and interpretability of common Neural Networks exists. If explainability is necessary, one must utilize inherently explainable AI models like Decision Trees.

If those models are not feasible to use, a combination of different interpretations, evaluations and test metrics can be utilized to get better insight into the behavior of the ML model.

Each metric can provide information for a specific part of the process of training a NN, e.g. the quality and distribution of data used for training and testing, the certainty of the network or the robustness to adversarial attacks, etc.

The downside of this approach is that it doesn't combine the results of different metrics to improve the understanding of the model's behavior, as well as the differences in implementation of the available metrics, forcing the developers to constantly write wrappers to enable the use with their specific model and or data format.

3 xpl[AI]ned: Moving towards explainability

To help customers create reliable AI faster, Keysight proposes a novel framework for testing, validating and assuring AI models alongside the complete lifecycle, while offering insights into the model along with documentation and enabling AI Experts and Domain Experts to join forces more effectively.

Fundamentally, we divide the AI lifecycle into five different stages:

1. Problem Understanding and Data Analysis
2. Feature Engineering
3. Model Training

4. Model Evaluation
5. Model Inference

As can be seen in Figure 1, each stage is directly connected with the previous and the next stage, creating a circular lifecycle model, where, after a model inference, a new iteration starts. With each iteration of the lifecycle, the quality of the model should improve, mitigating shortcomings detected in the inference stage. Due to the nature of ML development, changes in the model architecture or data structure might be necessary, as the model's behavior differs from expected values.

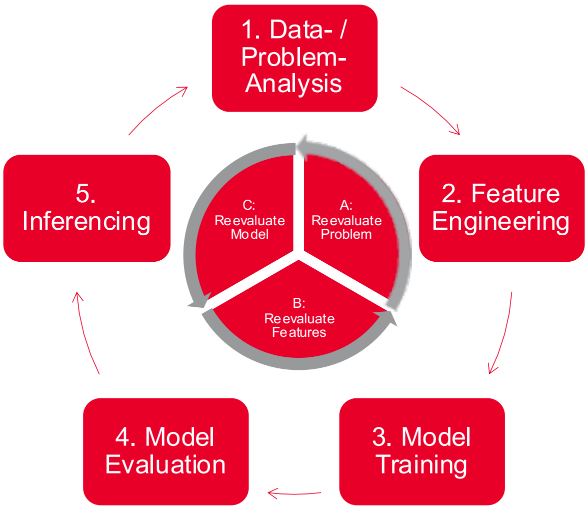

Figure 1: Proposed connected Lifecycle Model

Keysight’s solution provides a framework for developing ML models within this lifecycle, allowing seamless changes between stages in both directions. The framework, as the process itself, is divided into five different test categories, aligned with the stage names. Each test category, by default, has several reference tests implemented by Keysight, which can be used out of the box. Each test will automatically generate a report and can be cross-referenced with other metrics. For example, when analyzing the pixel-wise accuracy for specific objects within a semantic segmentation model, the accuracy might be correlated to the overall number of pixels of this class within the dataset and the percentage of pixels belonging to this class on this specific location.

The AI model solving complex driving tasks will still need to interact with classic system and software development. Data busses, sensor / actor interactions and data processing are just a few examples of components which need to be developed alongside the AI component. This poses the challenge of aligning the two different approaches. With the proposed methodology it is possible to align the iterative process of ML development to the more static approach of the V-Model, maintaining the well-established process for software development while enabling a test-driven AI development from the beginning, as can be seen in Figure 2.

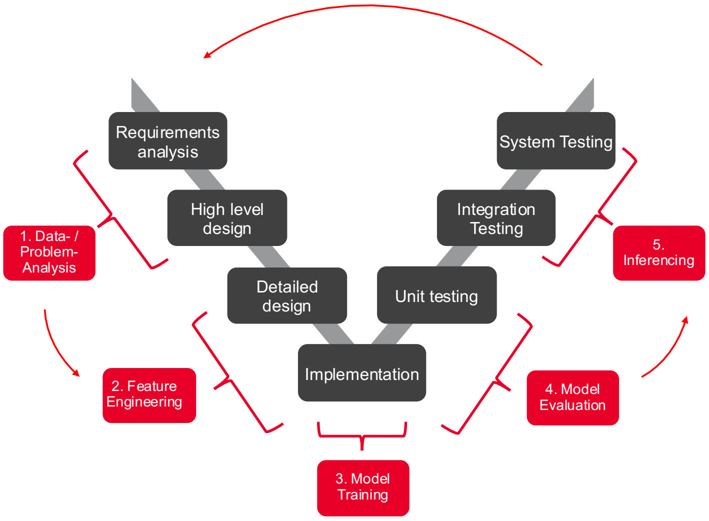

Figure 2: Alignment of Lifecycle Model to V-Model

While a general goal is to achieve total explainability and interpretability of those black box models, with the current state of technology only a partial explainability, e.g. a feature's influence on the outcome of a specific prediction, is possible. However, the field of AI research is moving at an extraordinary speed, so that one key feature of the platform is the easy extendibility by custom plugins. This allows on the one hand to quickly implement new tests and make them available, on the other hand it enables custom plugins for specific use cases, which might not be generally applicable to Deep Learning (DL) models but are required by stakeholders. The easy to implement plugin solution allows for an addition of custom metrics into the software, further accelerating the development lifecycle and eliminating the need for another solution.

4 Foundation for recommendations

By combining results from different tests from multiple stages, Keysight's solution can derive recommendations for improvement. These recommendations can be based on norms and guidelines, governmental requirements, or known good data (golden datasets).

Depending on use case and domain, there will be varying regulatory requirements, like the EU AI Act, defining specific minimum quality standards. These may be extended by industry standards, like DIN SPEC 13266, ISO 26262, ISO/PAS 21448 and the upcoming ISO/PAS 8800.

Keysight continuously monitors changes and extensions to those norms, guidelines and governmental requirements and classifies the results of tests accordingly. Based on this classification, recommendations are generated on how the current stage could be optimized. In addition, a comprehensive report is automatically generated for each test execution, containing all information about the model, used datasets, test configurations and results. This report serves two purposes; it can be used to keep track of different experiments within the organization and inform additional stakeholders like Risk Officers. As the documentation is also a proof of the ongoing endeavor to provide the model with the highest possible quality, it can also be used in case of an accident to work with the prosecution to prove that there was no negligence in the development process of the AI system. Secondly, those reports also contain recommendations for improvement, pointing towards problematic areas and causes, e.g. an uneven data distribution or low information content of the features used for training.

On the way towards explainability the combination of different metrics can enhance the transparency of the AI system, hinting towards limitations and helping to locate faults, not only in the training but also in the data used for training.

By utilizing a “golden dataset” created by Domain Experts as a foundation, the data selection and model training can be automatically checked against these datapoints, extrapolating to the problem areas which need attention. This golden dataset can also be a representation of the Operational Design Domain.

By defining the domain boundaries to match with the boundaries of the golden dataset, the framework can use test results from the golden dataset as a reference for checking the quality of the model.

While the task of creating a golden dataset may be challenging, its usage within xpl[AI]ned enables Domain Experts without a deep knowledge of AI to better apply their domain specific knowledge. This helps to further improve the model by creating a common basis for discussion, understood by AI and Domain Experts, making the black box more transparent. Combining the results of the metrics, a maturity score of the model can be calculated, serving as foundation for deciding whether the model is ready for deployment.

5 Case Study: Automatic analysis of Pedestrian detection

For most ADAS/AD functions, the camera is an important sensor. Other road users, drivable and non-drivable areas, traffic signs, obstacles and other information can be extracted from images. However, processing of the images is necessary to extract and use this information. One possible way of processing is pixelwise semantic segmentation, where each pixel of a given image is assigned to one predefined class, e.g. pedestrian, motor vehicle, drivable road, vegetation, etc.

For this use case, a custom dataset for training a classifier for semantic segmentation was recorded in the CARLA simulator, consisting of 8000 data samples, split into train (6k samples), test (1k samples) and validation (1k samples).

The first analysis for Semantic Segmentation data executed with the xpl[AI]ned framework already shows a significant imbalance in distribution of the different classes, e.g. 2.57 million “car” pixels vs. 0.55 million “pedestrian” pixels. While this is a common problem in semantic segmentation tasks and can be mitigated for example by using an adjusted loss function for unbalanced distributions during training, e.g. focal loss, the clustering of the different classes in certain areas of the image poses a problem, as can be seen in Figure 3.

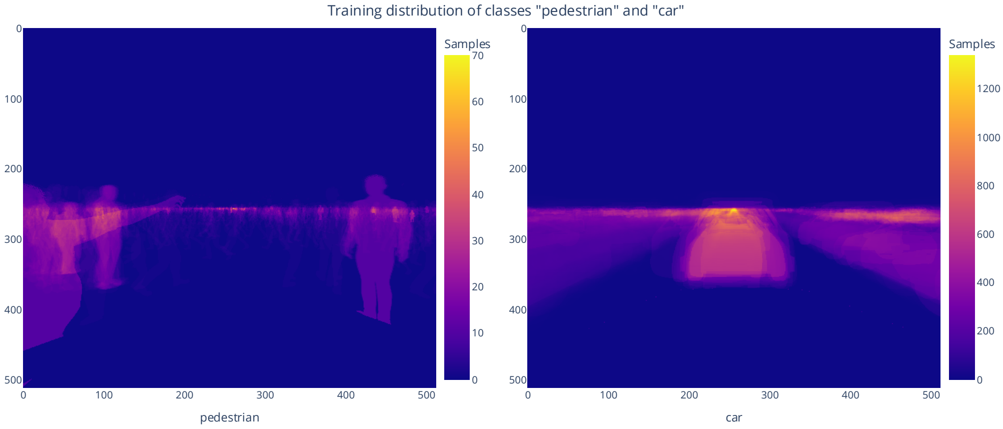

Figure 3: Distribution of classes in the training data

Training a simple segmentation network (approx. 18 million parameters) using a focal loss function, results in an overall accuracy score of 0.952. Despite that, analyzing precision (0.035) and recall (0.0002) for the class “pedestrian” shows that the overall score of the model is influenced by the imbalance in the data distribution, leading to misclassifications of “pedestrians”, as can be seen in Figure 4. However, the question, how exactly the decisions for the class “pedestrian” are influenced, remains.

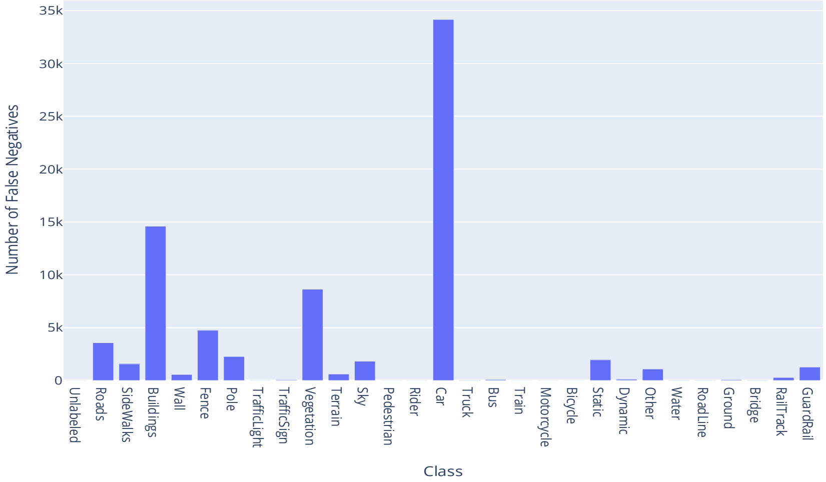

Figure 4: Class wise amount of "False Negatives" for class “pedestrian”

Xpl[AI]ned can automatically analyze and connect results of several metrics. Doing this for the confidence scores resulting from the probabilistic output layer of the network shows that the confidence for “pedestrians” is very low while the most likely class detected by the network for those pixels is “cars”. Comparing this information to the aforementioned data distribution, one can see that pedestrians are predominantly in locations with high proportions of cars or buildings.

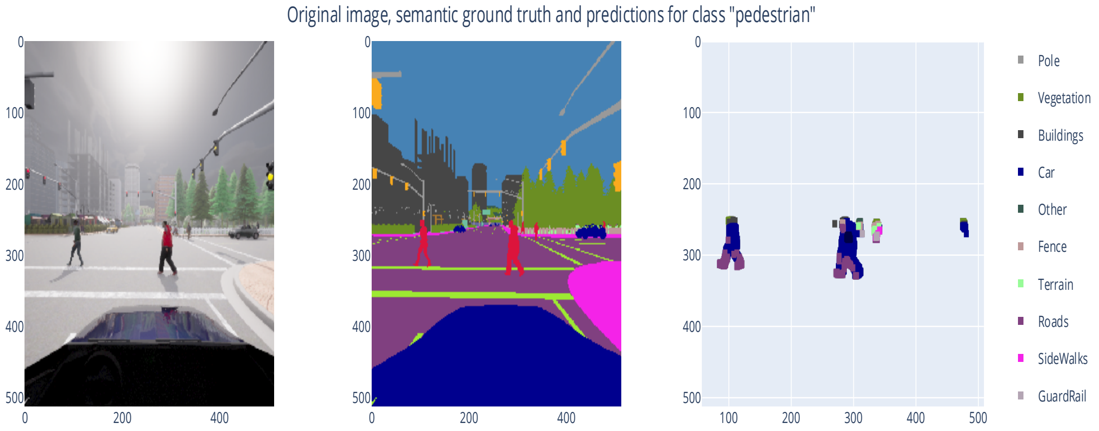
Original image, semantic ground truth and predictions for class "pedestrian"

Figure 5: Misclassification of class "pedestrian"

This leads to the assumption that pedestrians based on the low share and location are misclassified as “cars”. To further confirm this assumption, the framework automatically selects datapoints which are outliers to the data distribution, as well as samples from the data distribution and analyzes the decisions of the model. As can be seen in Figure 5, the model classifies the pedestrian on the road as car while pedestrians in areas with low proportions of cars are not recognized as at all, but are being integrated into the background, thus becoming “invisible”.

Using this automatically gained information, necessary adjustments can be made to improve the network. By automatically analyzing the shortcomings and problems of the network, xpl[AI]ned can help to build your safety argumentation and improve the quality, reliability and trustworthiness of the MuT.

6 Conclusion and Summary

Being able to test AI systems during the complete lifecycle using a comprehensive framework can accelerate the development of safe ADAS/AD functions. While challenging, the understanding of the decision process in AI model is crucial. To achieve this understanding, a combination of different metrics from different stages of the AI lifecycle can help to understand boundaries and shortcomings of the model. With xpl[AI]ned, Keysight offers a comprehensive and easy to use solution for automating the test and validation during the AI development lifecycle, helping engineers to make AI driven ADAS/AD components safe, while saving time and effort.