Towards Engineered Safe AI with Modular Concept Models

Lena Heidemann, Iwo Kurzidem, Maureen Monnet, Karsten Roscher
Fraunhofer IKS
{firstname.lastname}@iks.fraunhofer.de

Stephan Günnemann
Technical University of Munich
guennemann@in.tum.de

Abstract

The inherent complexity and uncertainty of Machine Learning (ML) makes it difficult for ML-based Computer Vision (CV) approaches to become prevalent in safety-critical domains like autonomous driving, despite their high performance. A crucial challenge in these domains is the safety assurance of ML-based systems. To address this, recent safety standardization in the automotive domain has introduced an ML safety lifecycle following an iterative development process. While this approach facilitates safety assurance, its iterative nature requires frequent adaptation and optimization of the ML function, which might include costly retraining of the ML model and is not guaranteed to converge to a safe AI solution. In this paper, we propose a modular ML approach which allows for more efficient and targeted measures to each of the modules and process steps. Each module of the modular concept model represents one visual concept and is aggregated with the other modules' outputs into a task output. The design choices of a modular concept model can be categorized into the selection of the concept modules, the aggregation of their output and the training of the concept modules. Using the example of traffic sign classification, we present each step of the involved design choices and the corresponding targeted measures to take in an iterative development process for engineering safe AI.

1. Introduction

Artificial intelligence (AI) has made significant advancements in recent years, enabling machines to perform complex tasks with remarkable precision. However, as AI systems become more sophisticated, ensuring their safety and reliability poses significant challenges. The inherent complexity and uncertainty of AI requires an innovative approach to safety assurance of AI systems in safety-critical domains like autonomous driving. The upcoming safety standard for the use of AI in road vehicles ISO PAS 8800 [16] provides principles for building an assurance argument for the safety of an AI system. The principles include the use of an iterative approach based on the AI safety lifecycle proposed by Burton et al. [5]. The safety lifecycle includes data specification and collection, selection and design of the Machine Learning (ML) approach, evaluation, and causal analysis.

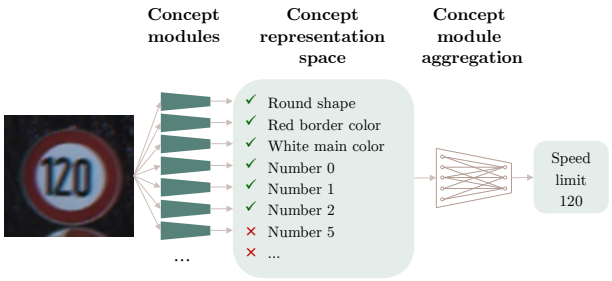


Figure 1. Example of a modular concept model for traffic sign classification (Image: GTSRB [31]).

While an iterative approach to assuring AI safety helps with reducing uncertainty in the safety argument, there is still a risk of not achieving all safety goals. With many, potentially conflicting, safety goals to optimize towards at the same time, it may become difficult for the ML model to converge to a solution that fulfills all safety goals. Adapting and retraining the model might lead to an improvement in one safety property, while deteriorating another, e.g., like a potential trade-off between interpretability and performance [3, 4, 14]. Additionally, it might entail frequent resource-intensive retrainings of the ML model as part of the iterative adaptation and optimization [3]. In this paper, we propose a practical and systematic approach to engineer-

ing safe AI in a more targeted and efficient way. We explore modular concept models and their application in enhancing the safety and reliability of AI systems. By breaking down ML models into independent modular components and establishing clear conceptual boundaries, potential risks and vulnerabilities can be effectively identified and addressed.

In the context of computer vision, concepts can be defined as features in an image that are relevant for the primary task of the model. A modular concept model consists of multiple independent concept modules. Their outputs span the concept representation space which in turn serves as the input to the concept module aggregation for determining the task output (see Figure 1). Each module represents one visual concept, e.g., shape, color, or symbol of a traffic sign. The output of the concept modules is aggregated into a task output, e.g., the traffic sign class. The development process of a modular concept model includes the selection of the concept modules, the aggregation of their output and the training of the concept modules.

The variety in design possibilities and the modularity allow for more flexibility in the model's design and a more targeted optimization. Thus, by using modular concept models, a more resource-efficient and goal-oriented iterative development may be implemented, e.g., as updates can be restricted to only one concept module instead of the entire ML function. Modular concept models additionally offer interpretability by using the detected concepts and their relevance for the task prediction. These insights may facilitate the causal analysis which is performed during the safety lifecycle.

In this paper we propose a modular approach to engineering safe AI with concept modules as essential building blocks. Using the example of traffic sign classification, we perform an iterative development process with targeted optimization in each category of design choices: concept representation space, concept module aggregation, and concept module training. This exemplary development process illustrates the potential of using modular concept models for engineering safe AI systems that can be trusted and deployed in safety-critical domains with increased confidence, while maintaining efficiency and adaptability.

2. Related Work

Modular concept models mainly build upon two emerging fields of research: explaining ML models with high-level concepts (concept models) and more generally, the field of modular deep learning.

2.1. Concept Models

The use of high-level concepts to explain predictions in deep image recognition models encompasses two streams of research: the first one incorporates the concepts after training (post-hoc concept models) to understand which concepts are the most activated in the network, while the second one aims to directly train the model using these concepts (inherent concept models).

One approach to post-hoc concept-based explanations is testing with concept activation vectors (TCAV) [17], where concept activation vectors of user-defined concepts are calculated and their alignment with a trained CNN's latent space for the prediction of a specific class is measured. Other approaches also leverage the latent space of a trained network to identify mappings to one or a combination of predefined concepts [12, 17, 35, 36]. However, these explanations are not grounded in the internal mechanisms of the model, so they may lack reliability.

Other studies focus on constraining the model's latent space to accommodate a set of concepts, thereby making the model interpretable by design. This is achieved by integrating a concept bottleneck layer [18, 22, 23], a concept whitening layer [8], or the utilization of supplementary information such as image descriptions [34].

Instead of relying on datasets with concept annotations, a parallel stream of research consists of using concepts found in an unsupervised manner, both in post-hoc [11, 13] and in inherent concept models setups [7, 25]. A significant limitation of this approach is that the discovered concepts may not resonate with human understanding. Hence, we focus in this work on inherent concept models where the concepts of interest have been previously defined and annotated by a domain expert or user, thereby contributing to the development of engineered safe AI.

2.2. Modular Deep Learning

Modular Deep Learning (DL) is a more general approach than concept models and also applies to use cases beyond image recognition. Modular neural architectures incorporate modules that can be updated independently, similar to biological systems where specialized components perform distinct functions, leading to adaptability and resilience. They consist of the implementation of modules, a routing function for their selection, and an aggregation function for combining module outputs. This modularity enables local updates, facilitating adaptation to new tasks and improving sample efficiency [27]. The routing function can be fixed, when metadata such as expert knowledge about sub-tasks is available [29], or learned, when the modules are selected during training. In the aggregation step, the modules' outputs can be weighted and summed [24], or take the form of a function, where either a sequential [9, 26] or hierarchical (tree) structure dictates the aggregation order [2]. In the training phase, the modules are either jointly trained as in [20], are incrementally introduced during continual learning [30], or are incorporated post-pre-training as a means to fine-tune the model in transfer learning scenarios [28].

3. A Modular Approach to Engineering Safe AI

In this section we first give a brief overview of the ML safety lifecycle introduced by Burton et al. [5] and mentioned in ISO PAS 8800 [16]. We then describe our modular approach in more detail and explain why it may address some of the practical challenges that accompany the ML safety lifecycle.

3.1. ML Safety Lifecycle

Due to the inherent complexity of ML [6] and the resulting multi-layered uncertainty in the overall safety argumentation, only an iterative ML development process can help to identify and mitigate the safety assurance gaps in order to achieve the formulated safety goal(s). These safety assurance gaps can manifest themselves from uncertainties regarding data, model or environment and require suitable solutions, for instance concept models and/or modular ML approaches, to improve certain safety properties.

The presented ML safety lifecycle from Burton et al. [5] highlights the need of a continuous identification of (ML) insufficiencies and subsequent incremental improvement of the (ML) system to tackle the complexity and ultimately satisfy allocated safety requirements. The ML lifecycle therefore demands an iterative development, which requires frequent adaptation and/or optimization of the ML function towards safety goals. The main steps of the lifecycle include:

1. Data specification and collection for training and test
2. Selection of ML approach and design of architectural measures to minimize and mitigate insufficiencies
3. Evaluation of performance with respect to the derived safety requirements
4. Evaluation of the impact and causes of performance insufficiencies

The fundamental idea behind this approach is to eventually reach a state at which there is sufficient confidence in the achieved safety assurance argumentation by repeated cycles of evaluation and optimization.

However, an optimal solution that satisfies all safety goals may not be guaranteed to be found within reasonable effort. Additionally, regular and comprehensive system updates can be very resource- and time-intensive [3]. It is therefore beneficial to apply suitable and efficient measures to identified insufficiencies, instead of retraining the ML model from scratch each iteration potentially optimizing towards conflicting goals. The design of modular concept models may help to address these challenges and facilitate the implementation of all steps within the ML safety lifecycle.

3.2. Modular Concept Models

Modular concept models break up a complex vision task into subtasks of recognizing relevant visual concepts (concept modules), the outputs of which are aggregated into a transparent model prediction for the primary task. Figure 1 illustrates an exemplary forward pass through a modular concept model for traffic sign classification. Based on an input image, the concept modules, which are trained independently, each predict one visual concept describing the shape or color of the traffic sign or symbols on it. The resulting concept predictions are then aggregated into a class prediction, using a classifier in this case.

When developing a modular concept model for a specific use case, there are three main areas of design possibilities: concept representation space, concept module aggregation, and concept module training. The goal in the design of the concept representation space is the selection of an optimal set of concepts relevant to the primary task. Concept module aggregation describes how the concept modules' outputs are aggregated into a task prediction. For classification, the aggregation may follow simple pre-defined rules without any training, when the mapping between concepts and classes is clearly defined. In case of more complex concept-class relationships, a classifier which was trained on data may be the better fit. Finally, the concept modules themselves can be designed and trained in various ways. The model architecture and training process can be optimized for each concept individually.

Modular concept models offer interpretability using the detected concepts and the concept module aggregation. The decision of the model can be explained through the presence or absence of certain visual concepts and their relevance for the final prediction. The many design possibilities for modular concept models broaden the solution space for optimizing towards safety goals. Additionally, the modularity allows for more targeted modifications of the model and therefore a more efficient search in this broader solution space.

These properties of modular concept models align well with the safety lifecycle presented in Section 3.1. The interpretability facilitates the evaluation of the impact and causes of performance insufficiencies (step 4 of the safety lifecycle). The many design possibilities increase the size of the toolbox for the selection of ML approach and design of architectural measures to minimize and mitigate insufficiencies (step 2). The design of the modular concept model can be optimized according to the requirements of the use case including e.g., assigning more importance to more safety-critical concepts. This can also be reflected in the evaluation of performance with respect to the derived safety requirements (step 3), in that additional more differentiated requirements can be defined for each concept module depending on their impact on the safety of the system. Finally,

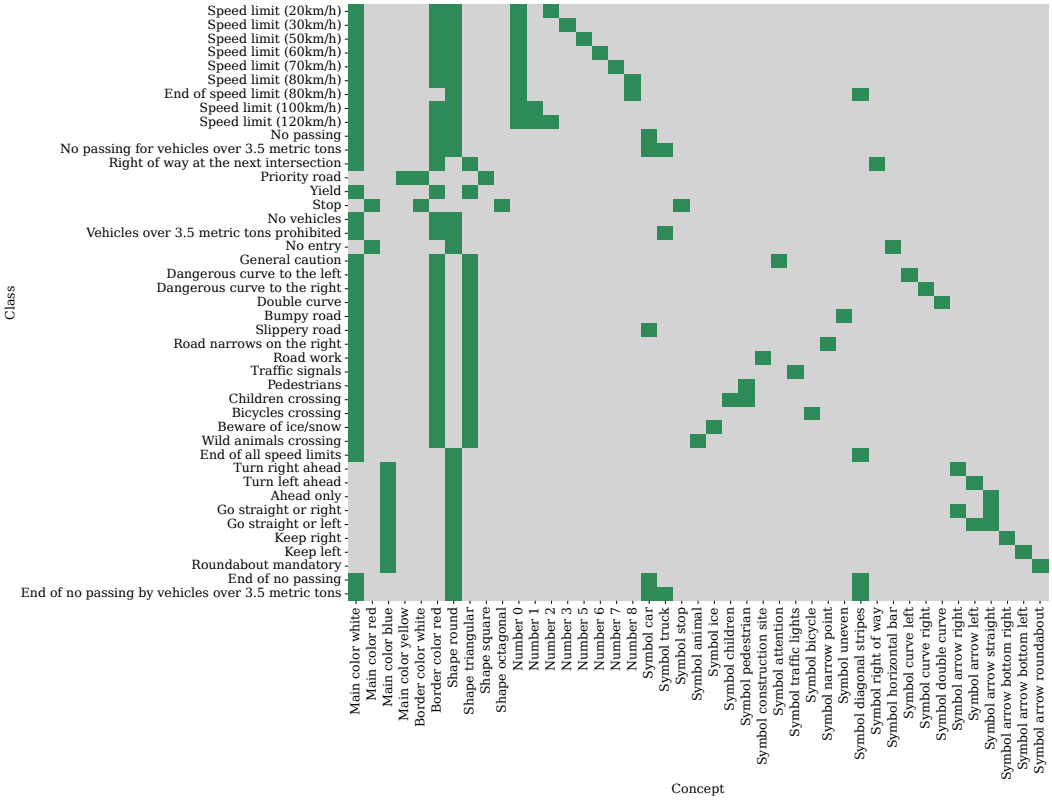


Figure 2. Mapping between concepts and classes in GTSRB. Green denotes that the concept must be present for an image to be classified as the respective class, gray that the concept should not be present.

the targeted modifications of the modular concept model may decrease the effort for the frequent model updates intended by the iterative development process. Updates can be restricted to only one concept module and may include measures like using additional data sources for this specific concept module (step 1).

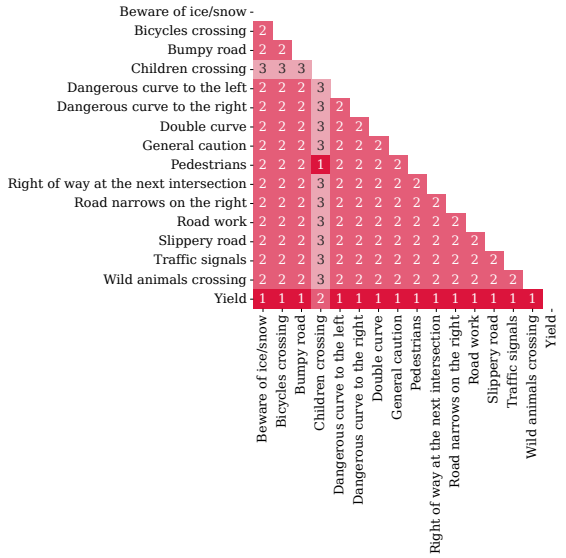
4. Iterative Development of Modular Concept Models for Traffic Sign Classification

In the following we iterate through each step in building a modular concept model and present potential measures to take during an iterative development process for safety-critical systems. The steps are the design of the concept representation space, the concept module aggregation, and the concept module training. We perform our experiments mainly on the German Traffic Sign Recognition Benchmark (GTSRB) dataset [31], while for some concept modules we also consider three additional traffic sign datasets: the Arabic Traffic Sign dataset (ArTS) [21], the Belgian Traffic Sign dataset (BelgiumTS) [33], and a dataset of African traffic signs (AfTS) [1] extracted from the Mapillary traffic sign dataset [10] and the DFG traffic sign dataset [32]. We use ResNet-18 models [15] for training the concept modules and nearest centroid, decision tree, and random forest classifiers for concept module aggregation. While we see potential for improvement of this setup, e.g., in optimizing model architecture for each module, we choose this simple setup to allow a broader analysis covering all steps.

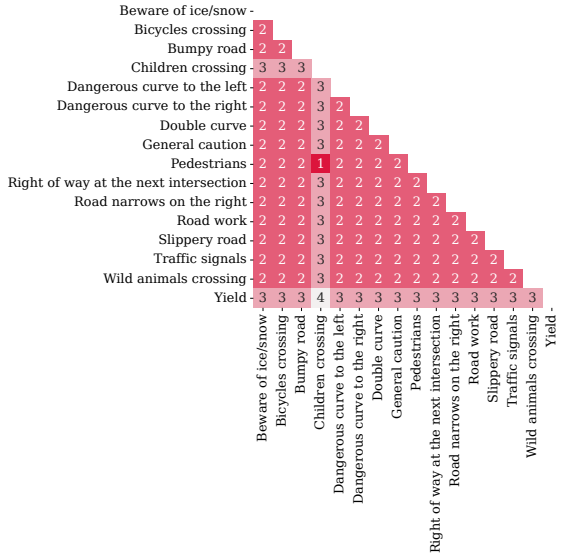
4.1. Concept Representation Space

The iterative development process of a modular concept model begins with the selection of concepts relevant to the classification task, i.e., designing the concept representation space. For the GTSRB dataset [31] we extend the concept definitions of [19] to 43 visual concepts (colors, shapes, numbers, and symbols) and annotate them.

Before we start training the concept modules, we can first analyze and optimize the current concept space and its relation to the classes with minimal effort. All concepts and their relationship to the traffic sign classes of GTSRB are shown in Figure 2. We see that most classes share many concepts and only differ in one or two concept annotations. The distance between classes in terms of concept annotations might give an indication on how likely these classes will be confused when one or two relevant concept modules provide a wrong prediction. Figure 3a gives an overview of the distances between all triangular traffic sign classes in



(a) Original concept representation space with one triangular concept.



(b) Concept representation space with two distinct triangular concepts: pointing upwards and downwards.

Figure 3. Distances between triangular traffic sign classes measured as the distance between their concept representations. The introduction of two triangular shape concepts (3a→3b) increases the distance between the *Yield* sign and all other triangular signs by 2.

GTSRB, calculated as the Hamming distance between their concept representations. We choose to depict this subset of classes because it contains the most class pairs with a distance of only 1. In particular, the *Yield* sign (downwards pointing triangular white sign with a red border) has a concept distance of 1 to most of the other triangular signs. This means, if the symbol on a triangular sign is not detected correctly, the model will wrongly classify this sign as *Yield*. This type of misclassification might be hazardous in an automated driving scenario and should be prevented.

One strategy for error mitigation is to focus on directly improving the concept modules that differentiate the *Yield* from the warning signs. For these traffic signs, a modification in the concept space might already alleviate the issue, since the signs differ in another visual concept currently not captured by the concept space: All signs are triangular, but the triangle of the *Yield* sign points downwards. We therefore split the triangular shape concept into two concepts describing the downward and upward pointing triangles respectively. This change results in an increase in concept distance between the *Yield* sign and all other triangular signs from 1 to 3 (see Figure 3b). Although there is a distance of 1 remaining between the *Pedestrian* sign and the *Children crossing* sign, a potential confusion will likely not be safety relevant. However, the final judgment on the severity depends on the following downstream task.

We further evaluate the potential effects of this modification in the concept space on the class predictions. For GTSRB the concept-class relations are clearly defined but the concept modules may provide wrong predictions or concepts might be occluded in the image. Consequently, when evaluating the effects of a change in the concept space, we should focus on how robust the class predictions are to errors in the concept space. To that end, we apply different concept aggregation methods to the ground-truth annotations and test them with varying levels of error in the concept input. An error of 5% would correspond to randomly setting 5% of the positive concept annotations (concept is present) to negative (concept is not present), and vice versa. The methods we include are nearest centroid, decision tree, and random forest classifiers. Nearest centroid classification simply assigns the class whose centroid is the closest to the concept input. The centroids are defined by the concept-class mapping (see Figure 2). We additionally train decision tree and random forest classifiers. For each of those, we apply two variants: either trained with the ground-truth concept annotations or trained with an introduced error of 5% in the input. The introduced error in the training input data is expected to decrease overfitting and increase the robustness to errors in the concept space during testing.

Figure 4 shows the per-class accuracy of different concept aggregation methods for varying levels of error in the concept input of a hold-out validation set of GTSRB. The first row is trained and evaluated on the baseline concept space with one triangular concept, the second row on the extended concept space with two distinct triangular concepts (pointing upwards and downwards). For the baseline concept space, we observe for all methods that there is a noticeably steeper decline in accuracy with increasing input error for the *Yield* sign compared to other classes. This is particu-

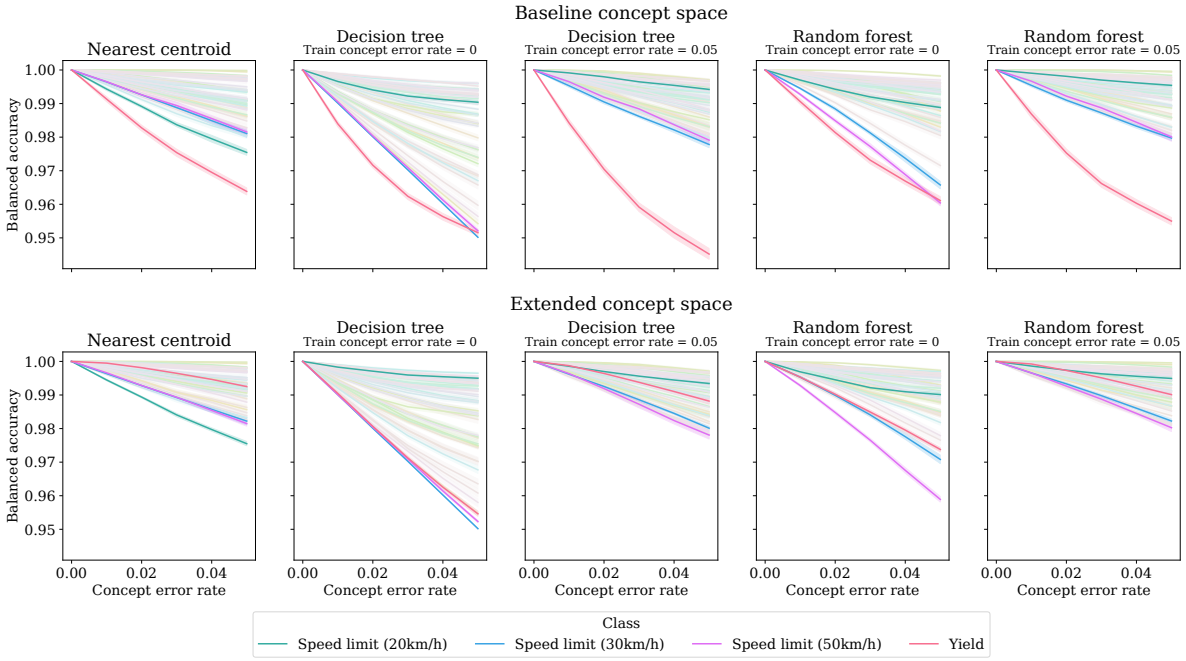


Figure 4. Balanced per-class accuracy on the GTSRB test set with different error rates in the concept input (*concept error rate*). The *train concept error rate* denotes the error rate in the concept input during the training of the respective classifier (except for nearest centroid which does not require training). For better readability, only relevant classes are listed in the legend.

larly visible, when the introduced error in the training input for decision tree and random forest classifiers lead to less overfitting for most classes, while the performance of *Yield* barely changes or even worsens. Notably, by introducing a concept for downward pointing triangles and therefore increasing the distance between Yield and other classes, the robustness of *Yield* to input concept error increases substantially. We can observe this in the second row of Figure 4, where Yield shows a smaller decrease in accuracy with increasing input error, while other class accuracies are barely affected by the concept space modification.

The design of the concept space, i.e., the selection of relevant concepts, already has a considerable impact on the resulting modular concept model. Using only the concept and class annotations and without training concept modules, we can already perform an analysis of the concept space and apply targeted measures for improving the model. The *Yield* traffic sign example shows how even a small modification in the concept space may lead to more robust class predictions.

4.2. Concept Module Aggregation

Another crucial step in the development process of modular concept models is the choice of concept module aggregation. Given the selected concepts and trained concept modules, we want to find a suitable method for concept aggregation aiming for high task accuracy. We select the concept space with two triangular concepts, as described in Section 4.1, and train a concept module for each concept. The trained concept modules achieve a mean balanced concept accuracy of 97.551% ranging from 83.333% for *Symbol arrow bottom left* to 99.996% for *Symbol stop*. The output of these concept modules is then used as the input for the concept module aggregation methods. The type of concept representation can either be binary, i.e., 0 for not present or 1 for present, or continuous, i.e., a confidence value between 0 and 1.

Method	Concept representation type Binary	Continuous
Nearest centroid	96.01	**97.66**
Decision tree	96.27 ± 0.01	96.10 ± 0.11
Random forest	96.53 ± 0.23	97.04 ± 0.09

Table 1. Class accuracy on the GTSRB test set (in %). Mean and standard deviation over 10 training runs with different seeds.

Table 1 shows the test accuracy on traffic sign classification for three basic concept aggregation methods: Nearest centroid, decision tree, and random forest classifiers. The classifiers are trained and tested either on binary or continuous concept representations. Only nearest centroid classifiers are not trained since the centroids are defined by the concept-class mapping (see Figure 2). As a result, there is no variance in the accuracy for nearest centroid classi-

Concept module	Dataset			
	GTSRB	AfTS	ArTS	BelgiumTS
Symbol attention	✓	✓	✗	✓
Symbol arrow bottom left	✓	✓	✓	✗
Symbol curve left	✓	✓	✓	✓
Symbol arrow roundabout	✓	✓	✓	✓

Table 2. Available datasets for each concept module.

Data	Concept module			
	Symbol attention	Symbol arrow bottom left	Symbol curve left	Symbol arrow roundabout
GTSRB	91.05 ± 1.67	84.45 ± 3.15	93.73 ± 10.43	95.67 ± 2.76
GTSRB + others	$\mathbf{95.52 \pm 1.17}$	$\mathbf{89.17 \pm 5.82}$	$\mathbf{96.75 \pm 7.06}$	$\mathbf{97.58 \pm 1.22}$

Table 3. Balanced concept accuracy on the GTSRB test set (in %). Mean and standard deviation over 10 training runs with different seeds.

fiers. The highest test accuracy of 97.66% is achieved with a nearest centroid classifier using the continuous concept representation as input.

For a rather simple use case like traffic sign classification, where the concept-class mapping is clearly defined, it may not be surprising that assigning classes solely based on distance works well already. With more complex concept-class relationships, the optimal choice of concept module aggregation may vary depending on the dataset or task and a wider selection of methods may be required. Yet the complexity of the method for concept module aggregation should not be higher than necessary in order to preserve an adequate level of interpretability. A low level of complexity would also allow to easily incorporate additional information, like uncertainty metrics. This would enable a tailored solution to the risks associated with the ML system.

4.3. Concept Module Training

In addition to designing the concept space and selecting a suitable concept module aggregation method, it is also possible to optimize the concept modules themselves. While most concept modules we trained for the concept space defined in Section 4.1 perform well, there are some modules with larger room for improvement. Due to the modularity of our approach, we can focus efforts on refining only these low-performing concept modules.

One approach for improving performance is to use more data for training. For our use case we consider three additional traffic sign datasets besides GTSRB: African TS [1, 10, 32], Arabic TS (ArTS) [21], and Belgium TS [33]. These datasets have overlapping classes but also classes specific to each dataset. For our modular approach this is not an issue as we can include any dataset where the respective concept is present and annotated. When training a standard end-to-end image classifier, a combined coherent dataset containing only the classes of the target dataset would be needed, which might require a tedious manual merging of the datasets. At the same time, a standard classification model would have to be retrained each time, affecting the classification of other classes, too. Adding data from other sources would be similarly challenging if we used only one model to predict all concepts instead of one independently trained module per concept. Any additional dataset would have to contain annotations for all the defined concepts, not only the one that needs improvement.

In our example, we focus on improving the performance of four different concept modules (*Symbol Attention*, *Symbol Arrow Bottom Left*, *Symbol Curve Left*, *Symbol Arrow Roundabout*) by including more data from other sources. Out of all concept modules with a balanced test accuracy $< 96\%$ the selected concepts are the ones which are present in at least one of the other traffic sign datasets and would likely benefit from this additional data. Not all four concepts are present in all traffic sign datasets. This is not an issue for modular concept models since we can tailor the datasets to each concept module. Table 2 lists the available datasets for each concept module. We train concept modules for each concept on GTSRB and on all datasets available for the respective concept. Table 3 shows the balanced class accuracy of these modules on the GTSRB test set. We can observe that including more datasets in the training leads to a higher mean balanced test accuracy overall for all four concept modules. The spread in accuracy across different random initializations is quite high for some concept modules, which might be due to a smaller number of positive test samples (samples where the concept is present) in GTSRB (*Symbol Attention*: 390, *Symbol Arrow Bottom Left*: 90, *Symbol Curve Left*: 60, *Symbol Arrow Roundabout*: 90). For *Symbol Attention*, the increase in mean balanced test accuracy exceeds one standard de-

viation. For the other concepts we can still infer a trend towards higher performance through adding more datasets. We also evaluate the best performing concept aggregation method, nearest centroid. We see that this improvement of just 4 out of 44 concept modules already leads to a small increase in class accuracy from 97.66% (see Table 1) to 97.83%.

Naturally the training of the concept modules has a high impact on the modular concept model's performance. The modularity of the approach allows for a targeted optimization of selected concept modules and the use of datasets tailored to each concept. In our traffic sign classification example, we see that performance increases by including datasets from other sources in the training of the concept modules.

5. Conclusion

The iterative development process required in the safety assurance of AI systems may entail challenging engineering with potentially conflicting safety goals and frequent adaptations. In this paper, we propose modular concept models for engineering safe AI. Modular concept models comprise multiple independently trained concept modules, whose outputs are aggregated into a prediction for the primary task. Such a modular approach allows for a tailored solution to the data and task at hand, as well as a variety of targeted measures to optimize towards a safe AI system.

To demonstrate the potential of modular concept models, we guide through an exemplary iterative development process of a traffic sign classifier, covering each part of the modular concept model: concept representation space, concept module aggregation, and concept module training. Our experiments on GTSRB [31] show how even small changes that do not require any retraining of large models and an individual tailoring of datasets can lead to an improvement of the model's performance.

The design of the concept representation space can already have a substantial impact on the modular concept model. For classification, the challenge is in selecting the relevant concept modules which separate the classes well and therefore reduce class confusion due to concept prediction errors. A metric that can aid in optimizing the selection of concept modules is the distance between classes measured in distance between their concept representations. In the traffic sign example, we observe a particularly small distance between the *Yield* sign and other triangular signs. To increase this distance, we apply a small change in the design of the concept representation space, that is replacing the concept for triangular shape with two concepts, one for a triangular shape pointing upwards and one for downwards. We show how this easy modification reduces the misclassification of the *Yield* sign, as it increases the distance to other triangular signs in terms of concept representation.

The selection of the method for aggregating the outputs of the concept modules also plays a crucial role in the development of modular concept models. Given a set of trained concept modules for traffic signs, we compare three different aggregation methods on binary and continuous concept representations. We find that, for our example, the nearest centroid classifier using continuous concept representations performs best and achieves a class accuracy of 97.66%.

Finally, we focus on improving the concept modules themselves. We select low-performing concept modules and enrich their training data with data from other traffic sign datasets (African TS [1, 10, 32], Arabic TS (ArTS) [21], and Belgium TS [33]). This works particularly well for modular concept models because it neither requires matching classes, as would be the case for standard classification models, nor concept annotations for all concepts, as would be the case for a single model predicting all concepts. For modular concept models, the only requirement for additional datasets is the presence and annotation of the concept of the module that needs improvement. For the traffic sign example, we demonstrate an improvement in concept accuracy of four concept modules by including more data in the training. We also show that this effort concentrated on only 4 out of 44 concept modules already results in a small improvement of class accuracy from 97.66% to 97.83%.

This paper illustrates the potential of modular concept models for engineering safe AI using the example of traffic sign classification. The simplicity of the use case, in terms of concepts as well as the mapping between concepts and classes, aids in demonstrating the proposed modular approach. However, future work should focus on exploring other use cases with a higher degree of complexity as well. Additionally, research on each part of the modular concept model can be extended. Concept module aggregation methods could incorporate additional information, like concept prediction uncertainty. The training data can be further tailored to each concept module, including the use of synthetic data. Additionally, the optimization of the concept modules themselves can be enhanced, e.g., by using different model architectures for each concept module and by potentially including non-AI methods as well. This optimization may also focus on other aspects than performance, e.g., domain generalization in order to enable the reuse of concept modules. These future research directions may help with evaluating and tapping the full potential of modular concept models and advancing towards engineered safe AI.

Acknowledgments

This work was funded by the Bavarian Ministry for Economic Affairs, Regional Development and Energy as part of a project to support the thematic development of the Institute for Cognitive Systems.

References

[1] Nouman Ahsan. Traffic Signs Dataset (Mapillary and DFG). https://www.kaggle.com/datasets/nomihsa965/traffic-signs-dataset-mapillary-and-dfg, 2022. 4, 7, 8

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural Module Networks, 2017. arXiv:1511.02799 [cs]. 2

[3] Rob Ashmore, Radu Calinescu, and Colin Paterson. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. *arXiv:1905.04223 [cs, stat]*, 2019. 1, 3

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. 1

[5] Simon Burton and Benjamin Herd. Addressing uncertainty in the safety assurance of machine-learning. *Frontiers in Computer Science*, Hypothesis and theory article(5), 2023. 1, 3

[6] Simon Burton, John McDermid, Philip Garnett, and Rob Weaver. Safety, complexity and automated driving – holistic perspectives on safety assurance. *IEEE Computer*, page 11, 2021. 3

[7] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems*, 2019. 2

[8] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. 2

[9] Asa Cooper Stickland, Alexandre Berard, and Vassilina Nikoulina. Multilingual Domain Adaptation for NMT: Decoupling Language and Domain Information with Adapters. In *Proceedings of the Sixth Conference on Machine Translation*, pages 578–598, Online, 2021. Association for Computational Linguistics. 2

[10] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, Gerhard Neuhold, and Yubin Kuang. The Mapillary Traffic Sign Dataset for Detection and Classification on a Global Scale. In *Computer Vision – ECCV 2020*, pages 68–84, Cham, 2020. Springer International Publishing. 4, 7, 8

[11] Zhengqing Fang, Kun Kuang, Yuxiao Lin, Fei Wu, and Yu-Feng Yao. Concept-based Explanation for Fine-grained Images and Its Application in Infectious Keratitis Classification. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 700–708, 2020. 2

[12] Ruth Fong and Andrea Vedaldi. Net2Vec: Quantifying and Explaining How Concepts Are Encoded by Filters in Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8730–8738, 2018. 2

[13] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards Automatic Concept-based Explanations. In *Advances in Neural Information Processing Systems*, 2019. 2

[14] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. XAI—Explainable artificial intelligence. *Science Robotics*, 4(37): eaay7120, 2019. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[16] International Organization for Standardization. ISO/PAS 8800. Technical report, in work. 1, 3

[17] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pages 2668–2677, 2018. 2

[18] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept Bottleneck Models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5338–5348, 2020. 2

[19] Jan Kronenberger and Anselm Haselhoff. Dependency Decomposition and a Reject Option for Explainable Models. *arXiv:2012.06523*, 2020. 4

[20] Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond Distillation: Task-level Mixture-of-Experts for Efficient Inference, 2021. arXiv:2110.03742 [cs]. 2

[21] Ghazanfar Latif, Jaafar Alghazo, Danyah A. Alghmgham, and Loay Alzubaidi. ArTS: Arabic Traffic Sign Dataset, 2020. 4, 7, 8

[22] Chi Li, M. Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D. Hager, and Manmohan Chandraker. Deep Supervision with Intermediate Concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1828–1843, 2019. 2

[23] Max Losch, Mario Fritz, and Bernt Schiele. Interpretability Beyond Classification Output: Semantic Bottleneck Networks. *arXiv:1907.10882*, 2019. 2

[24] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1930–1939, London United Kingdom, 2018. ACM. 2

[25] Meike Nauta, Ron van Bree, and Christin Seifert. Neural Prototype Trees for Interpretable Fine-Grained Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021. 2

[26] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer, 2020. arXiv:2005.00052 [cs]. 2

[27] Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. Modular Deep Learning, 2023. 2

[28] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters, 2017. arXiv:1705.08045 [cs, stat]. 2

[29] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks, 2017. arXiv:1706.05098 [cs, stat]. 2

[30] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks, 2022. arXiv:1606.04671 [cs]. 2

[31] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pages 1453–1460, 2011. 1, 4, 8

[32] Domen Tabernik and Danijel Skočaj. Deep Learning for Large-Scale Traffic-Sign Detection and Recognition. *IEEE Transactions on Intelligent Transportation Systems*, 21(4): 1427–1440, 2020. 4, 7, 8

[33] Radu Timofte, Karel Zimmermann, and Luc Van Gool. Multi-view traffic sign detection, recognition, and 3D localisation. *Machine Vision and Applications*, 25(3):633–647, 2014. 4, 7, 8

[34] Sandareka Wickramanayake, Wynne Hsu, and Mong Li Lee. Comprehensible Convolutional Neural Networks via Guided Concept Learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. 2

[35] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable Basis Decomposition for Visual Explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 2

[36] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting Deep Visual Representations via Network Dissection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2131–2145, 2019. 2