Introspection of 2D Object Detection using Processed Neural Activation Patterns in Automated Driving Systems

[1]Hakan Yekta Yatbaz [1,2]Mehrdad Dianati [1]Konstantinos Koufos [1]Roger Woodman
[1]WMG, University of Warwick
[2]School of Electronics, Electrical Engineering and Computer Science (EEECS)
Queen's University of Belfast
(hakan.yatbaz, m.dianati, konstantinos.koufos, r.woodman)@warwick.ac.uk, m.dianati@qub.ac.uk

Abstract

While deep neural network (DNN) models have become extremely popular for object detection in automated driving systems (ADS), the dynamic and varied nature of the road traffic environment can still lead to model failures. To address this issue, researchers have recently explored introspection mechanisms, a.k.a, self-assessment, for monitoring the quality of perception in ADS. Subsequently, depending on the situation, these mechanisms can either hand over control to the human driver in SAE Level 3, or initiate a minimum risk maneuver in SAE Level 4 ADS. State-of-the-art introspection mechanisms for ADS train a neural network to learn the relationship between the raw neural activation patterns of the underlying DNN-based perception function per frame and the calculated mean average precision. In this paper, we show that the use of raw activation patterns may contain misleading information for introspecting 2D object detection in ADS. To this end, we investigate how to optimally pre-process these patterns for improving the error detection performance. We evaluate the developed mechanism with and without pre-processing of the raw neural activation patterns and compare its performance with a state-of-the-art algorithm highlighting that for the Berkeley Deep Drive (BDD) dataset, pre-processing reduced the ratio of missed errors by 14% and improved the overall detection performance by 3%.

1. Introduction

The mainstream architecture of automated driving systems (ADS) consists of separate subsystems for sensing, understanding, planning and control to facilitate their design, aiming to enhance passenger comfort, safety, and efficiency of future transportation. Perception, i.e., sensing and understanding the surrounding road traffic environment, is crucial since the rest of the ADS architectural pipeline heavily depends on it. Object detection, i.e., localising and classifying objects, is one of the key functions of the perception subsystem of ADS. Deep neural networks (DNN) are becoming widely accepted for their improved object detection performance compared to traditional computer vision techniques. Nevertheless, their reliability and stability in safety-critical situations such as those encountered in ADS remain to be a challenge for system developers [1, 17, 26]. Even the best perception systems are not bulletproof; perception errors cannot be eliminated despite the rapid advancements in DNNs. To this end, runtime monitoring is a crucial last line of defense against such malfunctions. This means that a DNN-based mechanism must be able to detect its errors, which is called *introspection* in this paper. Several introspection approaches have been proposed so far by considering different learning (input) representations and error identification methods in DNN-based systems. In this regard, one approach is confidence/uncertainty-based introspection, where the confidence or uncertainty of the output is used to detect errors [14]. Another approach is inconsistency-based introspection, which involves detecting inconsistencies among two or more diverse parallel systems, such as detection and track-

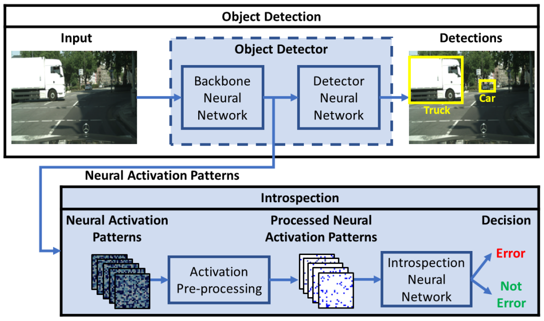
Object Detection
Input
Object Detector
Detections
Backbone Neural Network
Detector Neural Network
Truck
Car
Neural Activation Patterns
Introspection
Neural Activation Patterns
Processed Neural Activation Patterns
Decision
Activation Pre-processing
Introspection Neural Network
Error
Not Error

Figure 1. High-level summary of the proposed introspection mechanism.

ing in [21]. Past experience-based introspection is a promising approach for ADS operating in controlled environments, which involves encoding and storing of past experiences that the system can query for error detection [8]. Lastly, it is also possible to use estimation of an upper bound of a given performance metric, such as mean average precision (mAP) as an indicator for detecting events that the system performance degrades below a designated threshold [20].

In recent years, performance-based introspection using raw neural activation patterns has gained interest due to its flexibility and ease of integration into other systems [20]. However, a recent study on out-of-distribution (OOD) detection has highlighted that raw neural activation patterns can be confusing when the system needs to identify images that do not belong to one of the known classes used during training [4]. This study also showed that simplification of these activation patterns may help identifying OOD samples without suffering from significant performance loss for the in-distribution samples. The OOD detection problem is a subset of error detection and machine learning safety focusing on errors due to unknown samples [16]. This motivates the investigation of the effect of pre-processed neural activation patterns for in-distribution error detection.

To this end, this paper explores the impact of pre-processing neural activation patterns for introspection, originally introduced in [4] using the term *activation shaping*, on the performance-based introspection for 2D object detection in ADS. For this purpose, we utilise the fully-convolutional one-stage object detection model (FCOS), and extract neural network activation patterns from the last layer of its backbone model. The errors are labelled using a mAP threshold equal to 0.5, and the (binary) labels are paired with the raw neural activation patterns. Subsequently, an introspector convolutional neural network (CNN) is trained on the generated pairs, hereafter referred to as the error dataset, where *pre-processing* is applied prior to the introspection model. A high-level summary of our mechanism is presented in Fig. 1. In summary, the contributions of this paper are:

- Adaptation of a pre-processing technique for OOD detection mechanism for classification to performance-based introspection for 2D object detection in ADS.

- Evaluation of the effectiveness of the adapted mechanism against the use of raw neural activation patterns, and a comparison of its performance versus state-of-the-art (SOTA) methods [20] for error detection in 2D object detection models in ADS.

- Investigation of the effectiveness of the adapted mechanism in terms of error detection capability on two well-known public driving datasets, KITTI [6] and Berkeley Deep Drive (BDD) [28].

The structure for the rest of this paper is as follows: In Section 2, we review the current literature on introspection of 2D object detection. Section 3 outlines the adapted mechanism and introspection framework. Experimental settings are presented and discussed in Section 4. Performance evaluation of the proposed method is displayed in Section 5. Finally, key takeaways of the papers are given in Section 6.

2. Related Work

While the output of the object detection task should be the location and class of objects, it is equally important to evaluate the model's confidence in doing so. This is of paramount importance for safety-critical applications such as accident-prone driving scenarios in ADS. The existing literature on the introspection of object detection is extensive and multi-disciplinary. In this section, we provide a comprehensive overview that can be categorised into confidence/uncertainty-based, performance-based, inconsistency-based and experience-based introspection.

Starting with the first category, the focus is on techniques that model uncertainty in the detection process to identify misdetections. One approach uses Bayesian inference and Monte Carlo dropout (MCD) to generate multiple detections and compute the covariance matrix per-anchor bounding box to represent uncertainty. Harakeh *et al.* in [9] proposed a Bayesian object detector, while Miller *et al.* in [14] extended the MCD approach to 2D object detection and used the uncertainty values to establish a find-and-reject mechanism for open-set errors. The method named as GMM-Det in [15] uses class-specific Gaussian mixture models (GMM) to calculate the uncertainty measure for deciding whether or not to reject a sample. Post-processing can also be leveraged to improve the uncertainty estimates of object detection models. In [22], the authors proposed the MetaDetect algorithm that uses the output of an object detector (regression and classification) to provide better uncertainty estimates. In [24], the authors designed a model to identify areas-of-concern, generating a heatmap to indicate regions where there is a high probability of missing objects. Finally, researchers have also focused on identifying OOD samples in object detection. In [5], the authors implemented a spatio-temporal unknown distillation (STUD) mechanism that extracts unknown objects from videos and regularises the model's decision boundary accordingly. Wilson *et al.* in [25] proposed a mechanism for OOD detection that leverages activation maps extracted from "OOD-sensitive layers" and aims to identify OOD samples.

The second category of introspection methods for object detection focuses on utilising performance metrics for identifying error cases. For instance, object detection is commonly evaluated using the mAP for the detection performance. Hence, drops in the mAP or mAP regression either per-frame or per-window (a sequence of frames) can be utilised for error detection. Some researchers have also developed methods to introspect each detected object for identifying if it is missed or incorrectly detected. These methods can include monitoring systems such as those proposed in [20] and [19]. In [20], the authors used the output of the last layer of the backbone CNN to extract features and determine whether the estimated mAP is sufficient for the given image, while in [19], they proposed a cascaded neural network to extract features from multiple layers of the backbone network and multiple frames. Alternatively, in [18], the authors designed a monitoring system for false negatives in traffic sign detection by extracting activation maps from the object detector's backbone. Similarly, the authors in [29] proposed a method for introspection to identify false-negative samples in object detection, while [27] used a separate introspection model to extract common features for false-negative samples from the perception input during training.

Researchers have also proposed mechanisms that can monitor the inconsistency in different perception systems, such as object detectors and trackers, to identify errors. In [21], the authors defined an inconsistency-based error using both 2D object detection and tracking algorithms. Antonante *et al.* proposed a diagnostic graph in [3] where each vertex represents a processor, such as RADAR or camera, and edges represent consistency tests between the vertices. This formulation enables the system to identify errors in object detection and vehicle localisation with minimal overhead. The idea is extended in [2] where a graph neural network (GNN) detects inconsistencies in the diagnostic graph.

Lastly, creating a knowledge base with environment characteristics is another method for introspection where the object detector's ability is checked using a querying mechanism. In [8], a location-specific method for introspection grants autonomy to a robot only when its localisation is reliable. The method continuously records its past performance and offers probabilistic performance values for specific locations. The decision-making system uses the performance records to grant or deny autonomy in a limited environment. Similarly, in [7], the authors extended their location-based method with visual similarity-based experience in addition to performance records. In contrast, Hawke *et al.* in [10] presented a method for introspection using experience-based errors to retrain the network with false-negative samples extracted using scene filters. The developed experience-based classification mechanism turns out to improve the error detection performance.

3. Proposed Introspection Method

This section firstly gives a short overview of the activation pre-processing method proposed in [4]. This is followed by a discussion of how these activations are incorporated within our introspection framework. Next, we outline the key aspects of the main object detection system used in this paper and the proposed introspection models. Finally, we describe the process of training the introspection model, which is a crucial step for the system's effectiveness.

3.1. Pre-processing Method: Activation Shaping

In [4], the authors hypothesised that the raw neural activation patterns can be overly complicated for the task of OOD detection, and discussed various methods to simplify them. To test their hypothesis, they have proposed a two-stage approach consisting of the following:

1. Set equal to zero the activation elements whose values are less than the p-th percentile of the sample, i.e.,

$$\mathbf{x}' = \text{Shape}(\mathbf{x}) = \begin{cases} x_i, & \text{if } x_i \geq F^{-1}(p) \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathbf{x}$ is the activation pattern, $x_i, i = 1, \ldots, n$ is its i-th element, F^{-1} is the inverse (empirical) cumulative distribution function of the activation pattern, and $\mathbf{x}'$ is the shaped activation pattern.

2. Process the remaining activations using one of the following rules:

- Keep the remaining activations as it is, called activation shaping with pruning **(ASH-P)**
- Set all the values to a positive constant β calculated using the sum of all activations divided by the number of unpruned activations called **ASH by binarisation (ASH-B)**.

$$\mathbf{x}' = \begin{cases} \beta, & \text{if } x_i \geq F^{-1}(p) \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

  where $\beta = \frac{1}{|\{i : x'_i \neq 0\}|} \sum_i x_i$, and x'_i is an element of the shaped activation pattern.

- Scale up all the activations by the ratio calculated with the sum of the activations before and after pruning, called **ASH with scaling (ASH-S)**.

$$\mathbf{x}' = \begin{cases} \beta \, x_i, & \text{if } x_i \geq F^{-1}(p) \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $\beta = \exp\left(\frac{\sum_i x_i}{\sum_i x'_i}\right)$.

An example for all shaping techniques on a simplified 2D activation map is illustrated in Fig. 2. In the following section, we will compare the above-mentioned processing techniques to investigate their capability to identify errors depending on the performance metric of the object detector.

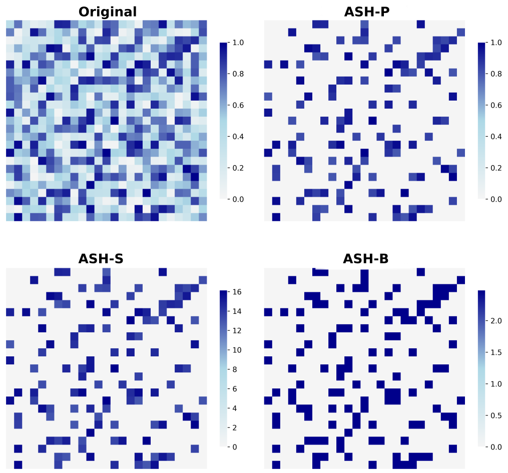

Figure 2. Summary of three pre-processing modes with 80 percentile presented in [4].

3.2. Introspection Framework

To assess the effectiveness of the adapted pre-processing method (activation shaping), we commence by dividing the driving dataset into three subsets: training, validation, and testing, in a proportion of $60 - 20 - 20$ %, respectively. Once the dataset is split, we follow the four stage introspection framework illustrated in Fig. 3. In the initial stage, we train an object detection model for driving-specific scenarios. This is necessary because most object detectors are pre-trained on generic datasets like COCO [12] or Pascal VOC [12]. In the second stage, we use the trained object detector to generate an error dataset for raw activation patterns. This dataset associates the raw neural activation patterns with binary labels generated by calculating the selected performance metric(s) and comparing them with predefined threshold(s). In this study, we adopt the mAP since it is widely used for object detection, and similar to [20], we set the decision threshold at 0.5 for a fair comparison. Subsequently, in the third stage, the introspection system is trained using the error dataset derived from the validation set, and in the last stage, we evaluate the performance of the introspection system on the error dataset generated from the test set. Note that in stages three and four, the raw neural activation patterns patterns are simplified according to the pre-processing modes described in Section 3.1.

3.3. Models & Training

The introspection framework consists of an object detector and an introspection neural network model. For object detection we have used a fully-convolutional one-stage (FCOS) [23] neural network, because one-stage detectors, due to their simplicity, are preferred in ADS applications over two-stage detectors, such as the Faster R-CNN. The selected detector is further trained on the KITTI and BDD datasets for domain-specific feature extraction.

For the introspection mechanism, we have utilised the ResNet18 [11] architecture for feature extractor from the shaped activations for error detection. Then, we coupled ResNet18 with a fully connected network (FCN) for final classification and applied hyper-parameter tuning to obtain optimal results. The summary of the parameters and their ranges, where applicable, are:

- Optimiser: Stochastic gradient descent (sgd).
- Early Stop Patience: 25 Epochs.
- Loss Function L : Focal loss,

$$L(\mathbf{q}) = -\sum_i \alpha_i (1 - q_i)^\gamma \log(q_i) \quad i = 0, 1, \tag{4}$$

  where $\log$ is the natural logarithm, $\mathbf{q}$ is the predicted probability vector for the classes not error (0) and error (1) with elements (q_0 and q_1 respectively), α_i is a scaling factor (class weights) that balances the contribution of the positive and negative examples for each class, and $\gamma \in \{0, 2, 5\}$ is a focusing parameter that downweights easy examples and emphasises hard ones.
- Number of Epochs: 600.
- Batch size: $16, 32, 64, 128$.
- Learning Rate: $0.01, 0.001, 0.005$.

4. Experimental Setup

This section provides the details of the experimental setup used for the evaluation of the proposed introspection mechanism. We first describe the selected datasets and the pre-processing steps undertaken to adapt them for our investigation. Then, the metrics used to quantify the efficacy of the proposed error detection mechanism are described. Finally, we present the best performing hyperparameter configurations for different pre-processing modes and datasets.

4.1. Datasets

There are various datasets available from companies and research institutes for training and developing object detectors for ADS. For this study, we have chosen two datasets that are popular in both introspection and ADS domains, i.e., KITTI and BDD.

The KITTI dataset comprises over 14,000 images that are annotated with a camera and a Velodyne LiDAR mounted on a vehicle driving through urban areas in Karlsruhe, Germany. It includes a training set of 7,481 annotated images with various object classes, and a test set of 7,518 images. However, only three classes - car, pedestrian, and cyclist - are typically used for benchmarking, and the labels for the test set are not publicly accessible to ensure

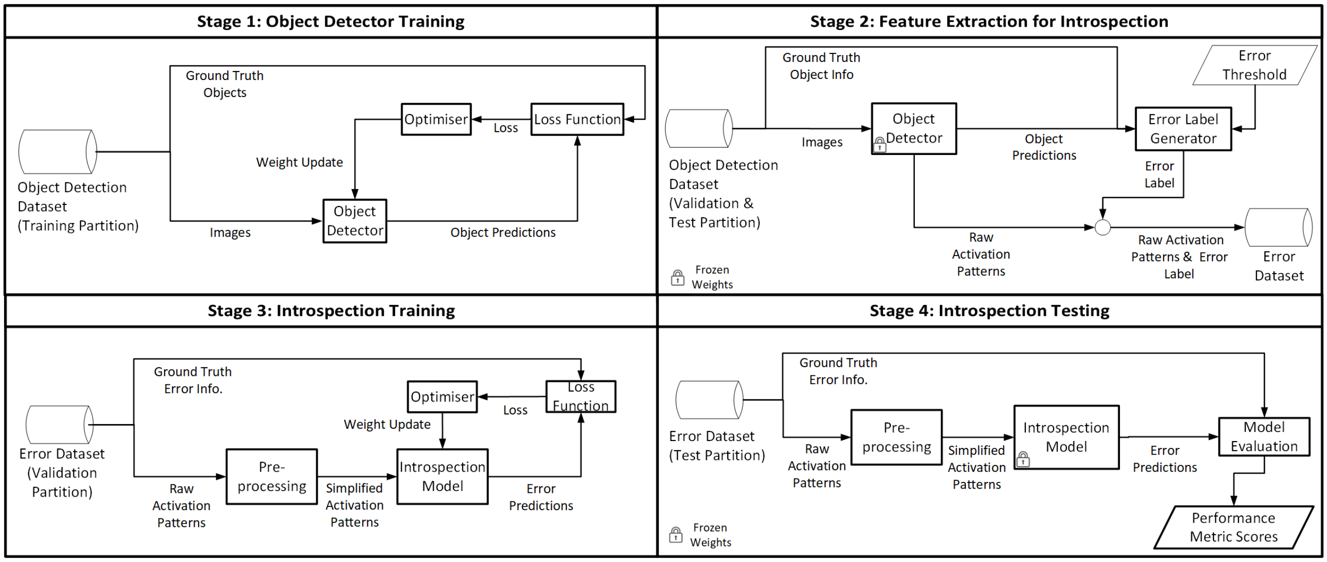

Figure 3. **Four-stage introspection framework:** (1) Train a driving-specific object detection model, moving away from generic datasets like COCO and Pascal VOC. (2) Generate an error dataset linking neural activations to binary labels using mAP at a 0.5 threshold. (3) Train the introspection system with error dataset from the validation set. (4) Test the system's performance using error dataset from the test set , simplifying neural activations.

fair benchmarking. In contrast, the BDD dataset comprises 100,000 annotated images taken from videos recorded in different parts of the United States, featuring ten object classes, including cars, pedestrians, bicycles, and motorcycles, but not cyclists. The dataset is divided into three sets - train, test, and validation - with 70%, 20%, and 10% proportions, respectively. The test set does not have publicly available labels.

To ensure compatibility between the two datasets, we merge the object classes into two categories: "vehicle" and "people." All types of vehicles are relabelled as "vehicle," and classes for people who are walking or sitting are merged into the "people" class. This allows for a more direct comparison between the datasets in our experiments.

4.2. Metrics

Since the proposed introspection model performs a classification task, we have utilised two well-known classification metrics, namely area under receiver operating characteristic curve (AUROC) and false negative rate (FNR), to evaluate its performance. The main reason we have selected the AUROC metric is to highlight the model's ability to sufficiently identify not only the errors but also the safe samples, as well as to be able to compare our proposed mechanism with SOTA. Furthermore, we have selected the FNR metric for highlighting the performance in error cases since they are vital for other traffic actors' safety.

- **AUROC:** Provides an indicator of how well a classifier distinguishes between the positive and negative classes. For introspection systems, the convention is to associate the positive class with an error in the input.
- **FNR:** Indicates the ratio of cases where the introspection system could not detect the error. In other words, it shows the probability of missing an error in the system, which can be read as:

$$\text{FNR} = \frac{\text{False Negatives}}{\text{False Negatives} + \text{True Positives}}.$$

Furthermore, it is essential to highlight that there is a class imbalance, i.e., for each error sample in KITTI, there are ten 'not error' samples, and a low number of error samples for the experimentation on the KITTI dataset. This is due to data partitioning prior to the four-stage framework, and the overall number of samples available in the KITTI dataset, which is low. Hence, the AUROC metric can provide high values if the model is able to correctly identify non-erroneous samples that constitute the majority class. To better understand the model performance in such cases, we have also examined the FNR value to ensure that the model can sufficiently identify both erroneous and safe cases. On the contrary, due to the diversity and higher number of samples in the BDD dataset, there is no significant imbalance in either the training or testing set for introspection.

4.3. Hyperparameter Tuning

To optimise the introspection performance, we have extensively evaluated the performance for various combinations of hyperparameters, i.e., batch size, learning rate and

Table 1. Best performing hyperparameters for each configuration.

Dataset	Method	Percentile	Batch Size	γ	Learning Rate
BDD	S	90	16	0	0.005
		85	16	5	0.001
		80	16	5	0.001
		75	16	0	0.005
		70	16	0	0.005
	P	90	16	5	0.010
		85	16	0	0.005
		80	16	0	0.001
		75	16	0	0.005
		70	16	0	0.001
KITTI	S	90	16	0	0.005
		85	32	0	0.001
		80	64	5	0.010
		75	64	5	0.005
		70	64	5	0.001
	P	90	64	5	0.010
		85	128	5	0.010
		80	128	5	0.005
		75	128	5	0.010
		70	128	5	0.005

focusing parameter γ. Their best values for each dataset, pre-processing mode and removal percentiles ranging from 70% to 90% can be retrieved from Table 1. The above-mentioned imbalance can also be seen in the parameter setups for best performing models, especially considering the γ value. Specifically, for the KITTI dataset, better results are obtained with higher γ values in which the dominance of the majority samples are reduced.

5. Performance Evaluation

This section presents the evaluation of the developed introspection system for object detection using the experimental setup presented in Section 4. It is worth noting that the introspection mechanism trained using the ASH-B pre-processing mode did not yield promising outcomes, as it consistently predicted most samples to belong to a single class. Consequently, we have excluded the performance evaluation results of this mode.

5.1. Comparison between Different Pre-processing Modes

The performance evaluation for both datasets is presented in Table 2 for the pre-processing modes: only pruning (P), and pruning and scaling (S), where it is demonstrated that only pruning yields the best overall performance. This is in contrast with the outcome of the numerical results obtained in [4] where it is shown that the mode S outperforms P. This difference may be attributed to the use of pre-preprocessed activation patterns. In [4], these patterns are used to calculate an energy score [13], while our introspection method utilises them for feature extraction and learning patterns for error detection. Hence, keeping the original scales of the activation values, i.e., only pruning, provides better results for introspection.

For the BDD dataset, pruning alone produces consistent results, although the FNR still varies between 0.11 and 0.35. On the contrary, it is apparent that the FNR significantly fluctuates when scaling is applied after pruning (S). Overall, S achieves an AUROC of 0.76-0.80, indicating good performance. Nonetheless, considering also the FNR metric, we observe that high AUROC values are accompanied by higher FNR values such as 33%. This behaviour indicates that the model tends to provide better performance for the not error cases as compared to the erroneous cases.

In the KITTI dataset, we observe a similar performance pattern as in the BDD dataset for the S mode, where results tend to lean towards one of the classes, e.g., on the one hand, a model with performance AUROC/FNR equal to 0.8088/0.7144 correctly identifies mostly non-erroneous cases, while on the other hand, a model with performance AUROC/FNR equal to 0.4584/0.0102 correctly detects mainly the error cases. Additionally, due to the limited number of samples in comparison with the BDD dataset, the performance inconsistencies between different percentiles are more pronounced.

5.2. Comparison between Raw Neural Activation Patterns and SOTA

In this section, we evaluate the model's performance without pre-processing and compare it with the SOTA, i.e., the results presented in [20], which follows the same experimental procedure as we do in this paper. This comparison is required to highlight the efficacy of the proposed mechanism over other baselines.

In [20], the authors proposed extracting statistical features from activation maps using mean, maximum and standard deviation functions, and used these values to train an artificial neural network for error detection. It should be noted that they have utilised a two-stage object detector, but they have used the same pre-trained backbone model utilised in this paper for extracting the raw neural activation patterns.

Table 3 provides a comparison between the best performing pre-processing mode based on AUROC metric, their no-shaping equivalents and the state-of-the-art model presented in [20]. The results show that for the BDD dataset, shaping significantly reduced the FNR by 14%, while also marginally increasing the overall performance by 3% com-

Table 2. Comparison of the different pre-processing modes (S and P).

Dataset	Type	Percentile	AUROC	FNR
BDD	S	90	0.7994	0.3302
		85	0.8057	0.2996
		80	0.7612	0.0180
		75	0.8021	0.0952
		70	0.7971	0.1114
	P	90	0.8009	0.2611
		85	0.8068	0.3521
		80	0.7972	0.2374
		75	**0.8103**	**0.1069**
		70	0.7999	0.2306
KITTI	S	90	0.8088	0.7143
		85	0.6759	0.2143
		80	0.4584	0.0102
		75	0.5362	0.2449
		70	0.6102	0.1327
	P	**90**	**0.8409**	**0.4898**
		85	0.8346	0.4082
		80	0.8238	0.4796
		75	0.8235	0.4592
		70	0.8330	0.4388

pared to raw activation patterns. Similarly, shaping provides better results than the SOTA method in both metrics. In the KITTI dataset, the improvement in the overall performance due to shaping comes at the cost of increased FNR. Despite the competitive results, pre-processing did not perform better than [20]. Additionally, it is important to emphasise that there are better results in terms of FNR despite the lower AUROC for the KITTI dataset, which may provide more competitive results, see Table 2. As previously mentioned, the discrepancy in results across different datasets may be due to the reduced sample size resulting from partitioning. However, it is noteworthy that our findings closely align with those of [20] for the KITTI dataset.

Table 3. Comparison of pre-processed, raw neural activation patterns and state-of-the-art.

Dataset	Method	Percentile	AUROC	FNR
BDD	with ASH	75	**0.8103**	**0.1069**
	w/o ASH	-	0.7793	0.2439
	[20]	-	0.7950	0.4760
KITTI	with ASH	90	0.8409	0.4898
	w/o ASH	-	0.8065	0.3674
	[20]	-	**0.8460**	**0.3400**

6. Summary & Conclusions

In this study, we explored the impact of simplified the neural activation patterns to enhance the error detection capability of 2D object detection in automated driving systems (ADS). This concept has been previously examined within the context of out-of-distribution detection for classification problems. In the present work, we utilised a one-stage object detection model for extracting neural network activation patterns from its backbone model's final layer. An error dataset was generated using the mean average precision (mAP) with a decision threshold at 0.5 to pair the binary labels with raw neural activation patterns. A separate neural network was trained and tested on this error dataset, employing pre-processing before inference. The modified approach was assessed using KITTI and BDD datasets.

Our findings indicate that by pre-processing neural activation patterns via pruning and scaling, the introspection model's capacity to detect erroneous patterns in 2D object detection is enhanced. Furthermore, our findings underscore the importance of large and diverse datasets while also revealing encouraging outcomes for smaller datasets. As this research constitutes an initial exploration of pre-processing within the domain of introspection for 2D object detection, further, more extensive studies are necessary. These should examine the effects of pre-processing on larger data samples, employ various state-of-the-art models, explore alternative pre-processing techniques, and account for domain-shift. Additionally, investigating other pre-processing techniques, neural network architectures and wider hyper-parameter spaces is essential to gain a broader understanding of the effect of simplified neural activation patterns on error detection in ADS.

Acknowledgement

This research has been conducted as part of the EVENTS project, which is funded by the European Union, under grant agreement No 101069614. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them. This research has been also supported by the Centre for Doctoral Training (CDT) to Advance the Deployment of Future Mobility Technologies at the University of Warwick.

References

[1] Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles.

[2] Pasquale Antonante, Heath Nilsen, and Luca Carlone. Monitoring of perception systems: Deterministic, probabilistic, and learning-based fault detection and identification. *arXiv preprint arXiv:2205.10906*, 2022.

[3] Pasquale Antonante, David I Spivak, and Luca Carlone. Monitoring and diagnosability of perception systems. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 168–175. IEEE, 2021.

[4] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. 2022.

[5] Xuefeng Du, Xin Wang, Gabriel Gozum, and Yixuan Li. Unknown-aware object detection: Learning what you don't know from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13678–13688, 2022.

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[7] Corina Gurău, Dushyant Rao, Chi Hay Tong, and Ingmar Posner. Learn from experience: Probabilistic prediction of perception performance to avoid failure. *The International Journal of Robotics Research*, 37(9):981–995, 2018.

[8] Corina Gurău, Chi Hay Tong, and Ingmar Posner. Fit for purpose? predicting perception performance based on past experience. In *International Symposium on Experimental Robotics*, pages 454–464. Springer, 2016.

[9] Ali Harakeh, Michael Smart, and Steven L Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2020.

[10] Jeffrey Hawke, Corina Gurău, Chi Hay Tong, and Ingmar Posner. Wrong today, right tomorrow: Experience-based classification for robot perception. In *Field and Service Robotics*, pages 173–186. Springer, 2016.

[11] 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.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings*, pages 740–755. Springer, 2014.

[13] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

[14] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018.

[15] Dimity Miller, Niko Sünderhauf, Michael Milford, and Feras Dayoub. Uncertainty for identifying open-set errors in visual object detection. *IEEE Robotics and Automation Letters*, 7(1):215–222, 2021.

[16] Sina Mohseni, Haotao Wang, Chaowei Xiao, Zhiding Yu, Zhangyang Wang, and Jay Yadawa. Taxonomy of machine learning safety: A survey and primer. *ACM Comput. Surv.*, 55(8), dec 2022.

[17] Quazi Marufur Rahman, Peter Corke, and Feras Dayoub. Run-time monitoring of machine learning for robotic perception: A survey of emerging trends. *IEEE Access*, 9:20067–20075, 2021.

[18] Quazi Marufur Rahman, Niko Sünderhauf, and Feras Dayoub. Did you miss the sign? a false negative alarm system for traffic sign detectors. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3748–3753. IEEE, 2019.

[19] Quazi Marufur Rahman, Niko Sünderhauf, and Feras Dayoub. Online monitoring of object detection performance during deployment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4839–4845. IEEE, 2021.

[20] Quazi Marufur Rahman, Niko Sünderhauf, and Feras Dayoub. Per-frame map prediction for continuous performance monitoring of object detection during deployment. In *IEEE Winter Conference on Applications of Computer Vision Workshops*, pages 152–160, 2021.

[21] Manikandasriram Srinivasan Ramanagopal, Cyrus Anderson, Ram Vasudevan, and Matthew Johnson-Roberson. Failing to learn: Autonomously identifying perception failures for self-driving cars. *IEEE Robotics and Automation Letters*, 3(4):3860–3867, 2018.

[22] Marius Schubert, Karsten Kahl, and Matthias Rottmann. Metadetect: Uncertainty quantification and prediction quality estimates for object detection. In *International Joint Conference on Neural Networks*, pages 1–10. IEEE, 2021.

[23] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[24] Yongxin Wang and Duminda Wijesekera. Pixel invisibility: Detect object unseen in color domain. In *VEHITS*, pages 201–210, 2021.

[25] Samuel Wilson, Tobias Fischer, Feras Dayoub, Dimity Miller, and Niko Sünderhauf. Safe: Sensitivity-aware features for out-of-distribution object detection, 2022.

[26] Dannier Xiao, William Gonçalves Geiger, Hakan Yekta Yatbaz, Mehrdad Dianati, and Roger Woodman. Detecting hazardous events: A framework for automated vehicle safety systems. In *IEEE 25th International Conference on Intelligent Transportation Systems*, pages 641–646, 2022.

[27] Qinghua Yang, Hui Chen, Zhe Chen, and Junzhe Su. Introspective false negative prediction for black-box object detectors in autonomous driving. *Sensors*, 21(8), 2021.

[28] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[29] Xuechen Zhang, Samet Oymak, and Jiasi Chen. Post-hoc models for performance estimation of machine learning inference. *arXiv preprint arXiv:2110.02459*, 2021.