Run-Time Performance Monitoring of Machine Learning for Robotic Perception: a Survey of Emerging Trends

Quazi Marufur Rahman, Peter Corke and Feras Dayoub *

Abstract

As deep learning continues to dominate all state-of-the-art tasks in computer vision, it is increasingly becoming the essential building blocks for robotic perception. As a result, the research questions concerning the safety and reliability of learning-based perception during run-time are gaining increased importance. Although there is an established field that studies safety certification and convergence guarantee of complex software systems during design-time, the uncertainty in the run-time conditions and the unknown deployment environments of autonomous systems and the complexity of learning-based perception systems makes the generalisation of the verification results from design-time to run-time problematic. In the face of such a challenge, more attention is starting to shift towards run-time monitoring of performance and reliability with a number of trends emerging in the literature – this paper attempt to identify these trends and provide a summary of the various approaches on the topic.

1 Introduction

Building on the impressive results on many computer vision tasks such as classification [1], object detection [2], depth estimation [3] and semantic segmentation [4]; Deep Neural Networks (DNNs) are increasingly becoming an essential part of the perception pipeline for robotic and autonomous systems such as driverless cars, service, agricultural and field robots [5–8]. However, a growing body of research is showing that state-of-the-art deep neural networks suffer a drop in performance when tested on data that differ from their training and testing sets [9–11]. This fact is of most importance for deep learning-based robotic perception as the robot may experience a wide range of environmental conditions that were not represented in the training data. This can lead to silent perception failures in unexpected ways, with unacceptable safety risks. Without the ability to assess the reliability of deep learning-based components in the robotics system during run-time, the whole system's safety comes under question.

The core of the problem is that deep learning models are currently developed using a large dataset split into training and test samples. As a result, the samples in the two sets are generated from the same distribution. In addition to that, most DNNs are trained with a closed-world assumption where all inputs are assumed to belong to one of a set of known categories. However, this is not always the case when using deep learning models as part of a robot's

*The authors are with the Australian Centre for Robotic Vision at Queensland University of Technology (QUT), Brisbane, QLD 4001, Australia. This research has been conducted by the Australian Research Council (ARC) Centre of Excellence for Robotic Vision (Grant CE140100016). The authors acknowledge continued support from the QUT Centre for Robotics. Contact: quazi.rahman@qut.edu.au

perception pipeline. The models' input data might come from unseen or different distributions than the training and testing sets due to environmental variables and novel contents that were not represented during design-time (i.e., the development and training stage). If this critical issue is not addressed, we can not generalise the model's performance on the test set to predict the performance during actual run-time (i.e., post-deployment on the robot) in a meaningful way.

Although there is an increasing interest in the area of safety certification and convergence guarantee of deep learning models during design-time (see [12] for a comprehensive overview), most of the current methods do not scale to large deep neural networks that are typical of what is used for robotic perception. In addition to the scale factor, there are the open-world deployment conditions that some mobile robots operate under (e.g. autonomous vehicles and field robots) that make the achievement of safety certification and convergence guarantee during design-time extra challenging. Consequently, more focus is being targeted towards verification, validation and monitoring during run-time. Run-time monitoring can be referred to as checking a mobile robot's performance during the deployment phase, where ground-truth labels are not available. This monitoring is of utmost importance for mobile robots' safety, and reliability as performance monitoring can work as a trigger to hand-over control to a less-capable-but-safe system or shift to a fail-safe mode. To this end, this paper identifies and discusses emerging research trends that address the run-time performance monitoring of the learning-based components in autonomous robotic systems.

The paper is organised as follows: Section 2 presents our categorisation of the reviewed papers based on *how* they perform the run-time monitoring. In Section 3, we revisit the same papers and re-categorise them based on *where* in the perception pipeline they perform the monitoring (i.e., whether at the input stage or in the target model itself or at its output stage or a combination of the above). Finally, we conclude with general observations in Section 4.

2 Run-Time Monitoring

Although run-time monitoring of machine learning for robotic perceptions is an emerging research topic, we can identify trends in the literature based on their approach of detecting or predicting run-time failures. The first trend includes techniques that utilise past examples of failures or predicting the quality of the output based on the similarity of context or the place of operation to previous experiences. The second trend includes the methods that detect inconsistencies in the perception output either over a stream of input data or input from different sensors or outputs from different models. The third trend is based on confidence learning and uncertainty estimation, where perception modules express their low confidence in their output, indicating low-quality performance or the detection of unknown inputs that were unseen during training.

2.1 Based on past experiences

2.1.1 Based on past examples of success and failures

Generally, performance monitoring using examples of failure depends on an auxiliary network to predict the base network's failure. The base network can be responsible for any specific task – image classification, segmentation or object detection. The auxiliary network is trained using both positive and negative samples where the base network performed its particular task with

expected accuracy. During the deployment phase, the auxiliary network operates along with the base network and predicts the base network's success or failure for performing the specific task.

The idea of using past examples of failures for the training of a self-evaluation system that detects perception failures during run-time has roots before the breakthrough of deep learning. An early example is the work by Jammalamadak *et al.* [13] where *evaluator algorithms* were proposed to predict the accuracy of a human pose estimation algorithm. They introduced the idea of self-evaluation and framed that as a binary classification task that uses additional features extracted from the target model's output. The binary classifier is learnt as the evaluator using examples of failures on the target model's training set. During inference, a threshold on the evaluator's quality is used to determine the human pose estimator's successes and failures.

Following similar approach to [13], Zhang *et al.* [14] proposed *alert* – a generalized warning framework to detect the failure of any vision system – *basesys*. *Alert* uses multiple generic hand-crafted image features to predict the accuracy of *basesys* on that particular input image. They also introduced two new metrics – accuracy of *basesys* vs declaration rate and risk-averse metric to evaluate the proposed performance prediction algorithm. Experimental results have shown *alert* effectiveness to predict the failure of image segmentation, 3D layout estimation, image memorability and attributes-based scene and object recognition tasks. Daftry *et al.* [15] applied the *alert* framework to predict the perception failure of an autonomous navigation task. They trained the *alert* system to predict Micro-Air Vehicle (MAV) navigation failure from the input image and corresponding optical flow. Here, *alert* is designed using spatiotemporal convolutional neural network for feature extraction and Support Vector Machine to identify cases where MAV will fail to navigate safely. Following a similar framework, Saxena *et al.* [16] trained the vision system of an autonomous quadrotor to identify navigational failure and response accordingly to avoid the consequences.

Joining the trend of using a separate system to monitor and predict a target model's failure, Mohseni *et al.* [17] proposed an approach to train a student model to predict the target model's error for input instances based on a saliency map extracted from the input images. The failure predictor is trained on examples of steering angle prediction errors of the target model for frames from the training set. In [18], a secondary model is trained using the softmax probabilities outputs of a target model to predict if its predictions are correct or not, therefore estimating the true inference accuracy on new and unseen data. The accuracy monitoring model needs to be pre-trained using data relevant to the target domain.

Most recently, Rahman *et al.* [19] addressed the problem of run-time performance monitoring of object detection onboard mobile robot. They focused on the performance difference between training and testing environment. They emphasized to track the performance in run-time for the safety and reliability of object detection system during deployment. This work proposed a cascaded neural network to monitor the performance by predicting the mean average precision metric over a sliding window of input images. In related work, [20] used an object detection's internal features to predict if the mean average precision for a particular image will be higher or lower than a predefined threshold. Identifying the false-negative object has been used by [21] as a means of run-time performance monitoring of object detection. In this work, they exploited features from specific feature map locations to identify potential false-negative objects. In a similar context, Schubert *et al.* [22] proposed meta-classifier to discriminate between true-positive and false-positive, and performed meta-regression to predict the IoU score without using any ground-truth labels during the deployment. These approaches rely on the object detection output and hand-crafted features to evaluate object detection quality in run-time. Rabiee *et al.* [23] introduced a framework called an introspective vision for obstacle avoidance (iVOA)

consisting of a perception system and introspection module for the task of obstacle avoidance. The introspection module is trained to detect false-positive and false-negative patches of input images where the perception system fails to detect the obstacle. The authors demonstrated the feasibility of the proposed introspection model for both indoor and outdoor dataset.

2.1.2 Based on experiences in the same workspace or context

As mobile robots often operate in the same places or the same contexts over long periods encountering periodic and seasonal variations in the deployment conditions, performance monitoring and failure prediction methods can take advantage of this knowledge. Such example is the work by Hawke *et al.* [24] where they introduced an Experience-Based Classification (EBC) framework to improve mobile robot performance for pedestrian detection. They applied multiple scene filters to identify false-positive errors made by the pedestrian detector. They used those filtered out images to re-train the detector to achieve better performance on the same location during the next traversal. Through experimental evaluation, EBC has been shown as a viable alternative of hard-negative-mining without manually labelled data.

In the context of mobile robot Teach and Repeat, [25] proposed *localisation envelop* to capture the likely localisation performance from the Teach phase to improve the performance during the Repeat. However, this approach is location-dependent and requires multiple Teach phase to learn the expected performance. To improve upon this work, Dequaire *et al.* [26] proposed an appearance-based approach to predict the localisation envelop using a single Teach pass.

Using a probabilistic framework, Gurau *et al.* [27] predicted perception performance of a pedestrian detection system deployed on a mobile robot based on its previous visits of the same location. They estimated the detection performance for a particular place and granted or denied autonomy to the mobile robot based on the predicted performance. Most recently, in Simultaneous Localization and Mapping (SLAM) paradigm, Rabiee *et al.* [28] proposed the idea of introspective vision-based SLAM. A self-supervised approach for learning to predict sources of failure for visual SLAM and to estimate a context-aware noise model for image correspondences, moving objects, non-rigid objects and other causes of errors.

In the context of Autonomous Vehicles (AV), Hecker *et al.* [29] argued that failure in the onboard vision system is not uncommon, and this does not happen in random. Heavy traffic, complex intersection, adverse weather and illumination condition are places where the vision system will fail. They presented a method to learn to predict how challenging an environment is to a given vision-based model. Their proposed work predicts if the current driving condition is safe or hazardous for an underlying speed and steering angle prediction network from images collected from a vehicle's front-view camera.

2.2 Based on inconsistencies during inference

Another trend we can identify in the failure prediction and performance monitoring literature is detecting inconsistencies during inference to indicate failure or performance degradation. We found examples for detecting inconsistencies in the perception module's output from input of a stereoviews or stream, between different sensor modalities, by identifying misalignment between the input and the predictions or detecting abnormal neuron activation patterns.

Ramanagopal *et al.* [30] proposed using stereo and temporal inconsistency of a deployed object detection system to identify false negative instances. The stereo disparity is used to transfer detected object from one camera view to another for stereo inconsistency detection. A multi-

object tracker is used to construct tracklets using the detected objects, and any missing tracklets in subsequent frames work as a false negative hypothesis.

Building on the literature of multiprocessor diagnosability, Antonante *et al.* [31] developed the temporal diagnostic graphs, a framework to reason over the consistency of perception outputs over time and demonstrated the ability to detect perception failures in an autonomous driving simulator.

Zhou *et al.* [32] proposes an automatic validation pipeline incorporating an additional sensor (lidar) to examine the performance of a semantic segmentation model in run-time. Using the geometric properties of neighbouring lidar points, they recognized the road boundaries near the vehicle and automatically generated labels data for the road. By comparing the road segmentation model's predictions with the automatically generated labels, they measured the segmentation model accuracy during run-time.

Yang *et al.* [33] introduced the concept of mirrorability and mirror-error for object part localization and showed that mirror-error could be measured without any ground-truth data. They also showed a high correlation between the mirror-error and the corresponding ground-truth error. Because of this correlation, mirror-error can be used to indicate localization/alignment error in run-time.

Gupta *et al.* [34] proposed Adversarially-Trained Online Monitor (ATOM) to track the performance of neural networks that estimate 3D human shapes and poses from images. They address this problem by identifying the alignment inconsistency between the input image and the output mesh of a human shape and pose reconstruction network, GraphCMR [35]. ATOM generates a mesh correctness score and uses that to monitor the performance of GraphCMR prediction.

Henzinger *et al.* [36] proposed an abstraction-based framework to monitor a neural network by observing its hidden layers. This framework is a neural network architecture-independent, and the proposed abstraction represents all values encountered in the chosen layers during the training phase. During deployment, run-time monitoring is performed by comparing the current values in the layers with the abstraction. Another related work is proposed by Cheng *et al.* [37]. They stored the neuron activation pattern in an abstract form and used Hamming distance to compare the generated pattern during run-time to the abstract form. This comparison detects whether the run-time prediction made by the network is consistent with the prior training data.

2.3 Based on confidence and uncertainty estimation

2.3.1 Based on uncertainty estimation

Uncertainty estimation is an active area of deep learning research. It includes approaches as simple as softmax entropy [38] to more principled methods such Bayesian Neural Networks [39] and their approximation [40], and ensemble techniques [41] (for a comprehensive review of uncertainty estimation methods used in machine learning and deep learning see [42]). Due to the promising role of uncertainty estimation in increasing autonomous and robotic systems' safety by indicating low confidence in output predictions – and consequently detecting failures – many authors in the field of robotic perception investigated and compared variations of the main methods of estimating uncertainty from DNNs. Examples include uncertainty estimation for steering angle estimation [43], road segmentation [44], Visual Odometry [45], vehicle and object detection [46–48].

The work by Grimmett *et al.* [49] is one of the earlier attempts that uses uncertainty to monitor

learning-based robotic perception. They showed that, in the robotic context, traditional performance metrics are inadequate to train and evaluate classifiers used for mission-critical decision making. To overcome this shortcoming, they proposed the concept of introspection – the ability to associate appropriate assessment of confidence to mitigate overconfident classifications. Based on this idea, they analyzed the introspective capability expressed using uncertainty estimation of multiple image classifiers and suggested using model ensemble instead of using a single model to take critical decisions in safety-critical robotic applications.

In the context of end-to-end controllers for self-driving cars, Michelmore *et al.* [50] explored the effectiveness of multiple measures of uncertainty and showed that mutual information, a measure of epistemic uncertainty [51], is a promising indicator of forthcoming crashes of the car. The evaluation was done using self-driving car simulator. In the context of vehicle detection, Feng *et al.* [52] proposed a probabilistic Lidar vehicle detection network that captures model epistemic uncertainty by Monte Carlo Dropout [53] and aleatoric uncertainty [54] by adding an auxiliary output layer to the vehicle detection network.

Tian *et al.* [55] showed that different uncertainty measures correlate differently to different types of sensory data degradation, and proposed a method to combine multiple types of uncertainties in an adaptive fusion scheme for unseen degradation with application to RGB-D semantic segmentation.

Henne *et al.* [56] compared several methods for estimating uncertainty for image classification task against safety-related requirements and metrics designed to measure the model's performance in safety-critical domains. Their findings emphasise the repeatedly reported observation that Deep Ensembles [41] method for estimating uncertainty demonstrate strong performance. They also found that learned confidence methods, the subject of the next section, produce consistently low confidence scores and can reject false predictions while producing higher confidence scores for correct predictions.

2.3.2 Based on confidence and quality scores

As shown in [10], deep neural networks often produce erroneous predictions with high confidence (low predictive uncertainty) when tested with data that differ from their training and test set, which is typical for DNNs deployed on mobile robots in open-world settings. An emerging research trend for failure prediction is learning a specialised confidence score that acts as a measure for the quality of the target model outputs or as a hardness indicator of the input to flag potential low-quality predictions.

For estimating model confidence, Corbiere *et al.* [57] defined a new confidence criterion called the True Class Probability (TCP) and proposed a network, ConfidNet, to learn the target confidence criterion. They provided theoretical guarantees and empirical evidence that predicting TCP instead of using max class probability (MCP) directly is better at predicting the failure of convolutional neural networks for classification and semantic segmentation.

An example of the use of quality score is the works by Rottman *et al.* [58]. They proposed meta-classifier to monitor the performance of a semantic segmentation model. The proposed approach uses pixel-wise uncertainty estimation and hand-crafted features corresponding to the target model segmentation's geometry to train a meta classifier/regressor to predict the intersection over union (IoU) with unknown ground-truth during run-time. Maag *et al.* [59] extend the work to account for temporal dependency between the input frames. As for input hardness prediction, Wang *et al.* [60] proposed an adversarially trained hardness predictor for a convolutional neural network classifier. The hardness-predictor is an auxiliary network that predicts a score for each input to the classifier denoting how hard it will be on the classifier.

Based on this score, the classifier can either accept to classify the image or reject it altogether.

Although not directly applied to a robotic application, Valindria *et al.* [61] approach to semantic segmentation quality monitoring can be extended to robotic perception. They introduced the concept of Reverse Classification Accuracy (RCA) to evaluate a deployed segmentation model's performance without using any ground-truth labels. RCA is a reverse classifier trained using the predictions of the target segmentation model as pseudo-ground-truth. Dice similarity coefficient (DSC) – aka F1-score – between RCA's outputs and the target model's predictions is used as a quality score. Robinson *et al.* [62] proposed to use a convolutional neural network instead of applying RCA to predict the DSC. Their approach provides real-time inference and better accuracy for predicting the DSC for image segmentation task.

2.3.3 Based on Out-of-distribution detection

Throughout the literature, out-of-distribution (OOD) detection is referred to and interchangeably used as an anomaly, novelty or outliers detection [63, 64]. Nevertheless, these approaches' common objective is to identify testing samples that do not belong to the training set's data distribution. Concretely, let us assume we have trained a perception system to perform some specific task – image classification, segmentation or object detection, using a dataset sampled from the distribution D_{in}. Any dataset that is not a member of D_{in} will be referred to as out-of-distribution. During run-time, we want to detect when the input comes from a distribution very different from D_{in}. (See [11] for a recent review of the different methods that tackle OOD detection and [65] for an empirical evaluation of several of these methods). In the context of robots that operate in open-world settings, this knowledge is essential as the DNNs tends to make over-confident but wrong predictions when operating on out-of-distribution data. Upon identifying out-of-distribution input, the mobile robot can enable fail-safe mode or hand-over the human operator's to ensure safety and reliability. This section discusses examples that use ideas related to OOD detection to monitor mobile robot performance.

In the context of a mobile robot's safe visual navigation, Richter *et al.* [66] proposed to use an autoencoder along with a collision-avoidance system. The autoencoder decides whether an input image is similar enough to the training data to be confident about the collision avoidance system's prediction. In the case of low confidence, the mobile robot reverts to safe recovery behaviour which reduced the number of collisions and resulted in faster navigation time than the baseline approach. Cai *et al.* [67] demonstrated the application of out-of-distribution control input detection in the context of a self-driving car. Their approach is based on conformal prediction [68] and anomaly detection. The nonconformity score is computed using variational autoencoder and deep support vector machine. The experimental results show a decrease in the number of false-positives and a faster execution time during inference.

Nitsch *et al.* [69] proposed an uncertainty based OOD detection technique that uses auxiliary training along with post-hoc statistics without requiring any external out-of-distribution dataset. The proposed approach takes advantage of Generative Adversarial Network (GAN) to enforce the object classifier to assign low confidence on OOD data and uses cosine similarity to identify OOD samples. Whereas Che *et al.* [70] proposed Deep Verifier Network to detect OOD and adversarial input to a deep neural network using conditional variational autoencoder.

Recently, Jafarzadeh *et al.* [71] formalized the open-world recognition reliability problem and proposed multiple automatic reliability assessment policies using only the reported probability distribution of a classifier. The proposed open-world reliability assessment works for both closed-set and open-set settings and shows significant improvement over a baseline algorithm.

A related topic to the approaches in this category is abstention or rejection learning, which is the

study of designing models that can reject an input assuming the possibility of making a wrong decision. Abstention learning can be used as an implicit approach for run-time performance monitoring. In abstention learning, each error and rejection incur a predefined cost, and the goal of abstention learning is to keep this error-reject cost at an optimal level. The error-reject tradeoff was first introduced by Chow [72, 73], here the author formalized the optimal rejection rule and derived the relation between the error and rejection probabilities. Following this work, [74], [75] and [76] introduced rejection option in Support Vector Machine, nearest neighbour and boosting algorithms respectively. The rejection module in these approaches is trained separately from the targeted perception approach. Later Cortes *et al.* [77] and Geifman *et al.* [78] proposed reject option that can jointly be learned with the perception system. [78] integrated a reject option with a deep neural network.

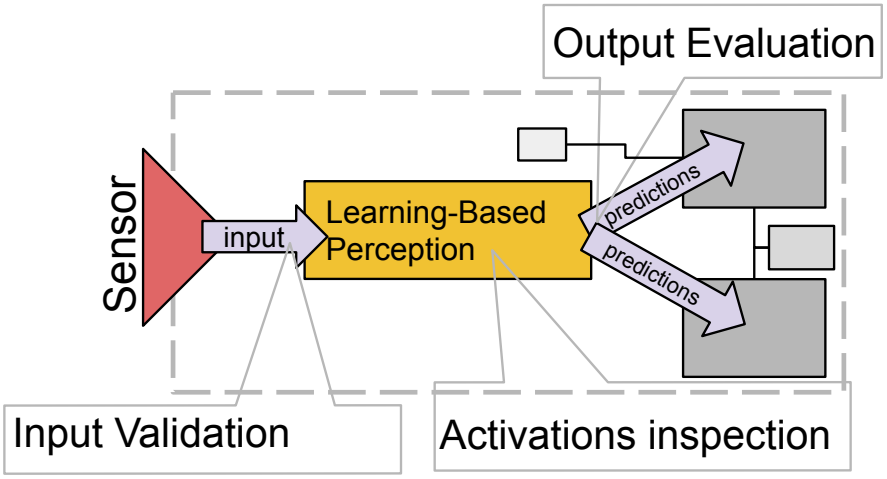

Figure 1: In a robotic system that uses a learning-based perception module, we can categorise the methods reviewed in this survey under methods that perform the monitoring by input validation or output evaluation or inner activations inspection or a combination of the above.

3 Run-time monitoring mapped to the robotic perception pipeline

Another way to categorize the papers reviewed in this survey is by where they perform the monitoring in the robotic perception pipeline (Figure 1). We can categorize the literature above according to whether they perform the monitoring by input validation or output evaluation or inner activations inspection or a combination of them.

Input validation means the performance monitoring system directly uses the same input as the perception system to predict failures and/or monitor the performance. As an example, [14] and [15] predict the success or failure of the perception system using a classifier that uses the same input as the perception system. In this case, performance monitoring is separated from the perception system. Output evaluation refers to the cases where performance monitoring is done by evaluating the perception system's output to predict its quality and express low or high confidence in it. [30] and [58] are examples of this paradigm. The activation inspection related research utilizes the perception system DNN's internal layer activations to monitor its performance and detect failures. [36] and [57] are examples of this category.

Moreover, we can categorize the performance monitoring literature into *explicit* and *implicit*

monitoring. In *explicit* monitoring, performance monitoring utilizes examples of success and failure from design-time before deployment. As an example, in [18] and [17], the performance monitor is trained using the perception system input and their corresponding accuracy and steering error, respectively. On the other hand, *implicit* monitoring does not require training using examples of success or failure. For example, [30] uses the stereo and temporal inconsistency to identify false-negatives and [66] identifies inputs dissimilar to the training set as a potential cause of navigation failure. Table 1 lists all the papers and their corresponding categorization.

Table 1: Re-categorisation of the papers based on where in the robotic perception pipeline they perform the monitoring and whether they required training during design-time or not.

Paper	Input Validation	Activation Inspection	Output Evaluation	Explicit	Implicit
Jammalamadaka *et al.* [13]	✓			✓	
Zhang *et al.* [14]	✓			✓	
Daftry *et al.* [15]	✓			✓	
Saxena *et al.* [16]	✓			✓	
Mohseni *et al.* [17]		✓		✓	
Shao *et al.* [18]			✓	✓	
Rahman *et al.* [19, 20]		✓		✓	
Rahman *et al.* [21]		✓	✓		✓
Schubert *et al.* [22]			✓	✓	
Rabiee *et al.* [23]	✓			✓	
Valindria *et al.* [61]			✓		✓
Robinson *et al.* [62]			✓	✓	
Cortes *et al.* [77]	✓				✓
Geifman *et al.* [78]	✓				✓
Hawke *et al.* [24]			✓		✓
Churchill *et al.* [25]	✓				✓
Gurau *et al.* [27]	✓			✓	
Rabiee *et al.* [28]	✓		✓	✓	
Hecker *et al.* [29]	✓			✓	
Ramanagopal *et al.* [30]			✓		✓
Yang *et al.* [33]			✓		✓
Gupta *et al.* [34]	✓		✓	✓	
Henzinger *et al.* [36]		✓			✓
Antonante *et al.* [31]			✓		✓
Grimmett *et al.* [49]			✓		✓
Feng *et al.* [52]			✓		✓
Corbiere *et al.* [57]		✓			✓
Rottman *et al.* [58]			✓	✓	
Wang *et al.* [60]	✓				✓
Richter *et al.* [66]	✓				✓
Cai *et al.* [67]	✓				✓
Tian *et al.* [55]			✓		✓
Nitsch *et al.* [69]			✓		✓
Che *et al.* [70]			✓		✓
Jafarzadeh *et al.* [71]			✓		✓

4 Conclusion

Run-time monitoring of learning-based perception systems – dominated by deep neural networks – is crucial for robotic applications due to the difficulty in applying design-time formal verification and safety guarantees for such systems, mainly due to their complexity and the complexity of modelling their deployment environments. In this survey, we identified an emerging research direction that focuses on run-time verification and monitoring. The approaches we reviewed tackle the problem in various ways. Some depend on past experiences and examples of success and failures to train a monitoring system that verifies some input/output/neural activations properties for the target model. Other approaches detected run-time inconsistencies in the input/output/internal activations as a mean to predict failures. The last group of methods use uncertainty estimation, learned confidence, and detect out-of-distribution input to predict the low-quality output from the target model. We also mapped these approaches based on where they perform the monitoring in the perception system pipeline and whether they require training during design-time. Due to the importance of this line of research for many safety-critical systems that use learning-based components such as deep neural networks with millions of parameters, a more principled approach to run-time monitoring is needed – one that considers not only the target perception module by itself but also the whole robotic system and the interaction between its various modules overtime.

References

[1] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html

[2] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, pp. 261–318, 2019.

[3] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, "Monocular depth estimation based on deep learning: An overview," *Science China Technological Sciences*, pp. 1–16, 2020.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2018.

[5] D. Hall, F. Dayoub, T. Perez, and C. McCool, "A rapidly deployable classification system using visual data for the application of precision weed management," *Comput. Electron. Agric.*, vol. 148, pp. 107–120, 2018.

[6] I. Sa, C. Lehnert, A. English, C. McCool, F. Dayoub, B. Upcroft, and T. Perez, "Peduncle detection of sweet pepper for autonomous crop harvesting—combined color and 3-d information," *IEEE Robotics and Automation Letters*, vol. 2, pp. 765–772, 2017.

[7] N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford, "Place categorization and semantic mapping on a mobile

robot," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5729–5736, 2016.

[8] F. Codevilla, M. Müller, A. Dosovitskiy, A. López, and V. Koltun, "End-to-end driving via conditional imitation learning," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9, 2018.

[9] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?" ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 5389–5400. [Online]. Available: http://proceedings.mlr.press/v97/recht19a.html

[10] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 991–14 002.

[11] A. Shafaei, M. Schmidt, and J. Little, "Does your model know the digit 6 is not a cat? a less biased evaluation of "outlier" detectors," *ArXiv*, vol. abs/1809.04729, 2018.

[12] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020.

[13] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. Jawahar, "Has my algorithm succeeded? an evaluator for human pose estimators," in *ECCV*, 2012.

[14] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh, "Predicting failures of vision systems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3566–3573.

[15] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert, "Introspective perception: Learning to predict failures in vision systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1743–1750.

[16] D. M. Saxena, V. Kurtz, and M. Hebert, "Learning robust failure response for autonomous vision based flight," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5824–5829.

[17] S. Mohseni, A. Jagadeesh, and Z. Wang, "Predicting model failure using saliency maps in autonomous driving systems," *arXiv preprint arXiv:1905.07679*, 2019.

[18] Z. Shao, J. Yang, and S. Ren, "Increasing trustworthiness of deep neural networks via accuracy monitoring," *Workshop on Artificial Intelligence Safety 2020*, 2020.

[19] Q. M. Rahman, N. Sünderhauf, and F. Dayoub, "Online monitoring of object detection performance post-deployment," *arXiv preprint arXiv:2011.07750*, 2020.

[20] ——, "Performance monitoring of object detection during deployment," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, 2021.

[21] ——, "Did you miss the sign? a false negative alarm system for traffic sign detectors," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3748–3753, 2019.

[22] M. Schubert, K. Kahl, and M. Rottmann, "Metadetect: Uncertainty quantification and prediction quality estimates for object detection," *ArXiv*, vol. abs/2010.01695, 2020.

[23] S. Rabiee and J. Biswas, "Ivoa: Introspective vision for obstacle avoidance," *arXiv preprint arXiv:1903.01028*, 2019.

[24] J. Hawke, C. Gurau, C. Tong, and I. Posner, "Wrong today, right tomorrow: Experience-based classification for robot perception," in *FSR*, 2015.

[25] W. Churchill, C. Tong, C. Gurau, I. Posner, and P. Newman, "Know your limits: Embedding localiser performance models in teach and repeat maps," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4238–4244, 2015.

[26] J. Dequaire, C. Tong, W. Churchill, and I. Posner, "Off the beaten track: Predicting localisation performance in visual teach and repeat," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 795–800, 2016.

[27] C. Gurau, D. Rao, C. Tong, and I. Posner, "Learn from experience: Probabilistic prediction of perception performance to avoid failure," *The International Journal of Robotics Research*, vol. 37, pp. 981 – 995, 2018.

[28] S. Rabiee and J. Biswas, "Iv-slam: Introspective vision for simultaneous localization and mapping," *ArXiv*, vol. abs/2008.02760, 2020.

[29] S. Hecker, D. Dai, and L. Gool, "Failure prediction for autonomous driving," *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1792–1799, 2018.

[30] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to learn: Autonomously identifying perception failures for self-driving cars," *IEEE Robotics and Automation Letters*, vol. 3, pp. 3860–3867, 2018.

[31] P. Antonante, D. I. Spivak, and L. Carlone, "Monitoring and diagnosability of perception systems," *ArXiv*, vol. abs/2011.07010, 2020.

[32] W. Zhou, J. Berrio, S. Worrall, and E. M. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 1951–1963, 2020.

[33] H. Yang and I. Patras, "Mirror, mirror on the wall, tell me, is the error small?" *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4685–4693, 2015.

[34] A. Gupta and L. Carlone, "Online monitoring for neural network based monocular pedestrian pose estimation," *ArXiv*, vol. abs/2005.05451, 2020.

[35] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4501–4510.

[36] T. Henzinger, A. Lukina, and C. Schilling, "Outside the box: Abstraction-based monitoring of neural networks," *ArXiv*, vol. abs/1911.09032, 2020.

[37] C.-H. Cheng, G. Nührenberg, and H. Yasuoka, "Runtime monitoring neuron activation patterns," *2019 Design, Automation and Test in Europe Conference*, pp. 300–303, 2019.

[38] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=Hkg4TI9xl

[39] D. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Computation*, vol. 4, pp. 448–472, 1992.

[40] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *ArXiv*, vol. abs/1506.02142, 2016.

[41] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NIPS*, 2017.

[42] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, A. Khosravi, U. R. Acharya, V. Makarenkov *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *arXiv preprint arXiv:2011.06225*, 2020.

[43] C. Hubschneider, R. Hutmacher, and J. M. Zöllner, "Calibrating uncertainty models for steering angle estimation," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1511–1518, 2019.

[44] B. Phan, S. Khan, R. Salay, and K. Czarnecki, "Bayesian uncertainty quantification with synthetic data," in *SAFECOMP Workshops*, 2019.

[45] G. Costante and M. Mancini, "Uncertainty estimation for data-driven visual odometry," *IEEE Transactions on Robotics*, vol. 36, pp. 1738–1757, 2020.

[46] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *ArXiv*, vol. abs/2011.10671, 2020.

[47] D. Miller, F. Dayoub, M. Milford, and N. Sünderhauf, "Evaluating merging strategies for sampling-based uncertainty techniques in object detection," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2348–2354.

[48] A. Harakeh, M. Smart, and S. L. Waslander, "Bayesod: A bayesian approach for uncertainty estimation in deep object detectors," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 87–93, 2020.

[49] H. Grimmett, R. Triebel, R. Paul, and I. Posner, "Introspective classification for robot perception," *The International Journal of Robotics Research*, vol. 35, pp. 743 – 762, 2016.

[50] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," *ArXiv*, vol. abs/1811.06817, 2018.

[51] Y. Gal, "Uncertainty in deep learning," *University of Cambridge*, vol. 1, no. 3, 2016.

[52] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3266–3273, 2018.

[53] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Advances in neural information processing systems*, 2017, pp. 3581–3590.

[54] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[55] J. Tian, W. Cheung, N. Glaser, Y.-C. Liu, and Z. Kira, "Uno: Uncertainty-aware noisy-or multimodal fusion for unanticipated input degradation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5716–5723.

[56] M. Henne, A. Schwaiger, K. Roscher, and G. Weiss, "Benchmarking uncertainty estimation methods for deep learning with safety-related metrics," in *SafeAI@AAAI*, 2020.

[57] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," *Advances in Neural Information Processing Systems*, vol. 32, pp. 2902–2913, 2019.

[58] M. Rottmann, P. Colling, T.-P. Hack, F. Hüger, P. Schlicht, and H. Gottschalk, "Prediction error meta classification in semantic segmentation: Detection via aggregated dispersion measures of softmax probabilities," *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2020.

[59] K. Maag, M. Rottmann, and H. Gottschalk, "Time-dynamic estimates of the reliability of deep semantic segmentation networks," *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 502–509, 2020.

[60] P. Wang and N. Vasconcelos, "Towards realistic predictors," in *ECCV*, 2018.

[61] V. Valindria, I. Lavdas, W. Bai, K. Kamnitsas, E. Aboagye, A. Rockall, D. Rueckert, and B. Glocker, "Reverse classification accuracy: Predicting segmentation performance in the absence of ground truth," *IEEE Transactions on Medical Imaging*, vol. 36, pp. 1597–1606, 2017.

[62] R. Robinson, O. Oktay, W. Bai, V. Valindria, M. M. Sanghvi, N. Aung, J. Paiva, F. Zemrak, K. Fung, E. Lukaschuk, A. Lee, V. Carapella, Y. Kim, B. Kainz, S. Piechnik, S. Neubauer, S. Petersen, C. Page, D. Rueckert, and B. Glocker, "Real-time prediction of segmentation quality," in *MICCAI*, 2018.

[63] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *arXiv preprint arXiv:2009.11732*, 2020.

[64] M. Masana, I. Ruiz, J. Serrat, J. van de Weijer, and A. M. Lopez, "Metric learning for novelty and anomaly detection," *arXiv preprint arXiv:1808.05492*, 2018.

[65] S. Rabanser, S. Günnemann, and Z. C. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," *ArXiv*, vol. abs/1810.11953, 2019.

[66] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," 2017.

[67] F. Cai and X. Koutsoukos, "Real-time out-of-distribution detection in learning-enabled cyber-physical systems," in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2020, pp. 174–183.

[68] G. Shafer and V. Vovk, "A tutorial on conformal prediction," *Journal of Machine Learning Research*, vol. 9, no. Mar, pp. 371–421, 2008.

[69] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena, "Out-of-distribution detection for automotive perception," *arXiv preprint arXiv:2011.01413*, 2020.

[70] T. Che, X. Liu, S. Li, Y. Ge, R. Zhang, C. Xiong, and Y. Bengio, "Deep verifier networks: Verification of deep discriminative models with deep generative models," *arXiv preprint arXiv:1911.07421*, 2019.

[71] M. Jafarzadeh, T. Ahmad, A. R. Dhamija, C. Li, S. Cruz, and T. E. Boult, "Automatic open-world reliability assessment," *arXiv preprint arXiv:2011.05506*, 2020.

[72] C.-K. Chow, "An optimum character recognition system using decision functions," *IRE Transactions on Electronic Computers*, no. 4, pp. 247–254, 1957.

[73] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on information theory*, vol. 16, no. 1, pp. 41–46, 1970.

[74] G. Fumera and F. Roli, "Support vector machines with embedded reject option," in *International Workshop on Support Vector Machines*. Springer, 2002, pp. 68–82.

[75] M. E. Hellman, "The nearest neighbor classification rule with a reject option," *IEEE Transactions on Systems Science and Cybernetics*, vol. 6, no. 3, pp. 179–185, 1970.

[76] C. Cortes, G. DeSalvo, and M. Mohri, "Boosting with abstention," *Advances in Neural Information Processing Systems*, vol. 29, pp. 1660–1668, 2016.

[77] ——, "Learning with rejection," in *International Conference on Algorithmic Learning Theory*. Springer, 2016, pp. 67–82.

[78] Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," in *Advances in neural information processing systems*, 2017, pp. 4878–4887.