IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 14, 2021, accepted January 17, 2021, date of publication January 27, 2021, date of current version February 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3055015

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

QUAZI MARUFUR RAHMAN, PETER CORKE, (Fellow, IEEE), AND FERAS DAYOUB
ARC Centre of Excellence for Robotic Vision, Queensland University of Technology, Brisbane, QLD 4000, Australia

Corresponding author: Quazi Marufur Rahman (quazi.rahman@qut.edu.au)

This work was supported in part by the Australian Research Council (ARC) Centre of Excellence for Robotic Vision under Grant CE140100016, and in part by the QUT Centre for Robotics.

ABSTRACT As deep learning continues to dominate all state-of-the-art computer vision tasks, it is increasingly becoming an essential building block for robotic perception. This raises important questions concerning the safety and reliability of learning-based perception systems. There is an established field that studies safety certification and convergence guarantees of complex software systems at design-time. However, the unknown future deployment environments of an autonomous system and the complexity of learning-based perception make the generalization of design-time verification to run-time problematic. In the face of this challenge, more attention is starting to focus on run-time monitoring of performance and reliability of perception systems with several trends emerging in the literature in the face of this challenge. This paper attempts to identify these trends and summarize the various approaches to the topic.

INDEX TERMS Machine learning, performance evaluation, reliability, robot learning.

I. INTRODUCTION

Deep Neural Networks (DNNs) show impressive results on many computer vision tasks such as image classification [1], object detection [2], depth estimation [3] and semantic segmentation [4]. This has led to their increased use for the perception pipeline of 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 DNNs suffer a drop in performance when tested on data that differs from their training and testing sets [9]–[11]. This fact is of particular importance for deep learning based robotic perception since a robot may experience a wide range of environmental conditions that were not represented in the training data. This can lead to unexpected perception failures which pose an unacceptable safety risk. Without the ability to assess the reliability of the deep learning based components of the robotic system at run-time, the whole system's safety must be questioned.

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 in the perception pipeline of a robot. 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 generalize 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 guarantees for deep learning models at 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 this scale issue, 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 guarantees during design-time extra challenging.

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

Consequently, the focus is shifting towards verification, validation and monitoring at run-time. Run-time monitoring checks a mobile robot's performance at its deployment phase, where ground-truth labels are not available. This monitoring is critically important 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 a human operator 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 organized as follows: Section II presents our categorization of the reviewed papers based on *how* they perform the run-time monitoring. In Section III, we revisit the same papers and re-categorize 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 IV.

II. RUN-TIME MONITORING

Although run-time monitoring of machine learning for robotic perceptions is an emerging research topic, we can identify several trends in the literature based on their approach of detecting or predicting run-time failures. The first trend includes techniques that utilize 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 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 own confidence in their output.

A. MONITORING BASED ON PAST EXPERIENCES

This section will discuss the literature that focuses on monitoring perception system performance based on previous experience. We can categorize this literature into two groups. The first group monitors perception performance using past examples of success and failures, and the second group uses the experience from the same workspace or context for performance monitoring.

1) 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 at 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 trained 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. *Alert* uses multiple generic hand-crafted image features to predict the accuracy of the vision system on that particular input image. They also introduced two new metrices – accuracy of vision system versus declaration rate, and risk-averse metric to evaluate the proposed performance prediction algorithm. Experimental results have shown *alert* is effective in predicting 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 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. Using 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 a mobile robot. They focused on the performance difference between the training and testing environment. They emphasized tracking the performance at run-time for the safety and reliability of object detection system during deployment. This work proposed a cascaded neural network to monitor performance by predicting the mean-average-precision metric over a sliding window of input images. In related work,

[20] used an object detector'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 a meta-classifier to discriminate between true-positive and false-positive, and performed meta-regression to predict the intersection-over-union (IoU) score without using any ground-truth labels during deployment. These approaches rely on the object detection output and hand-crafted features to evaluate object detection quality in run-time. Rabiee and Biswas [23] introduced a framework named introspective vision for obstacle avoidance (iVOA) consisting of a perception system and an 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 obstacles. The authors demonstrated the feasibility of the proposed introspection model for both indoor and outdoor dataset.

2) 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. One 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 was shown to be a viable alternative to hard-negative-mining without manually labeled data.

In the context of mobile robot Teach and Repeat, [25] proposed a *localization envelop* to capture the likely localization performance from the Teach phase to improve the performance during the Repeat phase. However, this approach is location-dependent and requires multiple Teach phases to learn the expected performance. To improve upon this work, Dequaire *et al*. [26] proposed an appearance-based approach to predict the localization 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 to 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 the Simultaneous Localization and Mapping (SLAM) paradigm, Rabiee and Biswas [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 randomly. Heavy traffic, complex intersections, adverse weather and illumination condition are conditions 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 whether the current driving conditions are safe or hazardous for an underlying speed and steering angle prediction network that uses images collected from a vehicle's front-view camera.

B. MONITORING BASED ON INCONSISTENCIES DURING INFERENCE

Another trend we can identify in the performance monitoring literature is the detection of inconsistency during the inference period. This inconsistency detection can be an indicator of performance degradation. The proposed approaches focus on using temporal and stereo vision, multiple sensor modalities, misalignment detection between the input and the output and abnormal neuron activation pattern. This section will provide a brief overview of these approaches.

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 neighboring LIDAR points, they recognized 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 models accuracy at run-time.

Yang and Patras [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 at run-time.

Gupta and Carlone [34] proposed an 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 at 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.

C. MONITORING BASED ON UNCERTAINTY ESTIMATION AND CONFIDENCE

In this section, we will provide a brief overview of the literature that focus to monitor the performance of a perception system using uncertainty estimation, prediction confidence and quality scores.

1) 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], and vehicle and object detection [46]–[48].

The work by Grimmett *et al*. [49] is one of the earlier attempts to use 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 assess 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 emphasize the repeatedly reported observation that Deep Ensembles [41] method for estimating uncertainty demonstrates 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) 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. This is frequently the case for DNNs deployed on mobile robots in open-world settings. An emerging research trend for failure prediction is learning a specialized confidence score that acts as a measure for the quality of the target model outputs or as an indicator of the difficulty of the input to flag potential low-quality predictions.

For estimating model confidence, Corbière *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 maximum class probability (MCP) directly is better at predicting the failure of convolutional neural networks for a classification and a semantic segmentation task.

An example of the use of quality score is the works by Rottman *et al*. [58]. They proposed a 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 or regressor to predict the IoU score with unknown ground-truth at run-time. Maag *et al*. [59] extend the work to account for temporal dependency between the input frames. As for input hardness prediction, Wang and Vasconcelos [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, the approach of Valindria *et al*. [61] 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. Instead of applying RCA to predict the DSC, Robinson *et al*. [62] proposed to use a convolutional neural network. Their approach provides real-time inference and better accuracy for predicting the DSC for image segmentation task.

3) OUT-OF-DISTRIBUTION DETECTION

Throughout the literature, out-of-distribution (OOD) detection is referred by multiple terms, for example anomaly, novelty or outlier 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. At 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 since the DNNs could make an over-confidently wrong predictions when operating on out-of-distribution data. Upon identifying out-of-distribution input, the mobile robot can enable a fail-safe mode or hand-over control to a 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 and Roy [66] proposed using 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 behavior which reduced the number of collisions and resulted in faster navigation time than the baseline approach. Cai and Koutsoukos [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 a variational autoencoder and a deep support vector machine. The experimental results show a decrease in the number of false-positive errors 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 concerned with designing robust model 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], where 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 a rejection option to Support Vector Machines, nearest neighbors 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 and El-Yaniv [78] proposed rejection option that can be jointly learned with the perception system. Reference [78] integrated a reject option with a deep neural network.

III. 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

TABLE 1. Re-categorization 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]			✓		✓

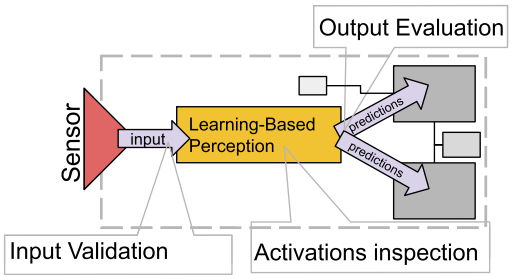

FIGURE 1. In a robotic system that uses a learning-based perception module, we can categorize the methods reviewed in this survey under methods that perform the monitoring by input validation or output evaluation or inner activations inspection or a combination of the above.

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. References [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.

IV. 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 modeling 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 *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, in Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., Long Beach, CA, USA, vol. 97, Jun. 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," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[3] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, "Monocular depth estimation based on deep learning: An overview," *Sci. China Technol. Sci.*, vol. 63, pp. 1–16, Jun. 2020.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 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. Agricult.*, vol. 148, pp. 107–120, May 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 Robot. Autom. Lett.*, vol. 2, no. 2, pp. 765–772, Apr. 2017.

[7] N. Sunderhauf, 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," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 5729–5736.

[8] F. Codevilla, M. Muller, A. Lopez, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 1–9.

[9] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?" in *Proc. Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., Long Beach, CA, USA, vol. 97, 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 *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13991–14002.

[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," 2018, *arXiv:1809.04729*. [Online]. Available: https://arxiv.org/abs/1809.04729

[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," *Comput. Sci. Rev.*, vol. 37, Aug. 2020, Art. no. 100270.

[13] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. Jawahar, "Has my algorithm succeeded? An evaluator for human pose estimators," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 114–128.

[14] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh, "Predicting failures of vision systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1743–1750.

[16] D. M. Saxena, V. Kurtz, and M. Hebert, "Learning robust failure response for autonomous vision based flight," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2017, pp. 5824–5829.

[17] S. Mohseni, A. Jagadeesh, and Z. Wang, "Predicting model failure using saliency maps in autonomous driving systems," 2019, *arXiv:1905.07679*. [Online]. Available: http://arxiv.org/abs/1905.07679

[18] Z. Shao, J. Yang, and S. Ren, "Increasing trustworthiness of deep neural networks via accuracy monitoring," in *Proc. Workshop Artif. Intell. Saf.*, 2020, pp. 1–8.

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

[20] Q. M. Rahman, N. Sünderhauf, and F. Dayoub, "Per-frame mAP prediction for continuous performance monitoring of object detection during deployment," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV) Workshops*, Jan. 2021, pp. 152–160.

[21] Q. M. Rahman, N. Sunderhauf, and F. Dayoub, "Did you miss the sign? A false negative alarm system for traffic sign detectors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 3748–3753.

[22] M. Schubert, K. Kahl, and M. Rottmann, "MetaDetect: Uncertainty quantification and prediction quality estimates for object detection," 2020, *arXiv:2010.01695*. [Online]. Available: http://arxiv.org/abs/2010.01695

[23] S. Rabiee and J. Biswas, "IVOA: Introspective vision for obstacle avoidance," 2019, *arXiv:1903.01028*. [Online]. Available: http://arxiv.org/abs/1903.01028

[24] J. Hawke, C. Gurau, C. Tong, and I. Posner, "Wrong today, right tomorrow: Experience-based classification for robot perception," in *Proc. FSR*, 2015, pp. 173–186.

[25] W. Churchill, C. H. Tong, C. Gurau, I. Posner, and P. Newman, "Know your limits: Embedding localiser performance models in teach and repeat maps," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 4238–4244.

[26] J. Dequaire, C. H. Tong, W. Churchill, and I. Posner, "Off the beaten track: Predicting localisation performance in visual teach and repeat," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 795–800.

[27] C. Gurău, D. Rao, C. H. Tong, and I. Posner, "Learn from experience: Probabilistic prediction of perception performance to avoid failure," *Int. J. Robot. Res.*, vol. 37, no. 9, pp. 981–995, Aug. 2018.

[28] S. Rabiee and J. Biswas, "IV-SLAM: Introspective vision for simultaneous localization and mapping," 2020, *arXiv:2008.02760*. [Online]. Available: http://arxiv.org/abs/2008.02760

[29] S. Hecker, D. Dai, and L. Van Gool, "Failure prediction for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1792–1799.

[30] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to learn: Autonomously identifying perception failures for self-driving cars," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3860–3867, Oct. 2018.

[31] P. Antonante, D. I. Spivak, and L. Carlone, "Monitoring and diagnosability of perception systems," 2020, *arXiv:2005.11816*. [Online]. Available: http://arxiv.org/abs/2005.11816

[32] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1951–1963, May 2020.

[33] H. Yang and I. Patras, "Mirror, mirror on the wall, tell me, is the error small?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4685–4693.

[34] A. Gupta and L. Carlone, "Online monitoring for neural network based monocular pedestrian pose estimation," 2020, *arXiv:2005.05451*. [Online]. Available: http://arxiv.org/abs/2005.05451

[35] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4501–4510.

[36] T. A. Henzinger, A. Lukina, and C. Schilling, "Outside the box: Abstraction-based monitoring of neural networks," 2019, *arXiv:1911.09032*. [Online]. Available: http://arxiv.org/abs/1911.09032

[37] C.-H. Cheng, G. Nuhrenberg, and H. Yasuoka, "Runtime monitoring neuron activation patterns," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 300–303.

[38] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–12.

[39] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

[40] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1050–1059.

[41] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. NIPS*, 2017, pp. 1–15.

[42] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," 2020, *arXiv:2011.06225*. [Online]. Available: http://arxiv.org/abs/2011.06225

[43] C. Hubschneider, R. Hutmacher, and J. M. Zollner, "Calibrating uncertainty models for steering angle estimation," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1511–1518.

[44] B. Phan, S. Khan, R. Salay, and K. Czarnecki, "Bayesian uncertainty quantification with synthetic data," in *Proc. SAFECOMP Workshops*, 2019, pp. 378–390.

[45] G. Costante and M. Mancini, "Uncertainty estimation for data-driven visual odometry," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1738–1757, Dec. 2020.

[46] D. Feng, A. Harakeh, S. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," 2020, *arXiv:2011.10671*. [Online]. Available: http://arxiv.org/abs/2011.10671

[47] D. Miller, F. Dayoub, M. Milford, and N. Sunderhauf, "Evaluating merging strategies for sampling-based uncertainty techniques in object detection," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 2348–2354.

[48] A. Harakeh, M. Smart, and S. L. Waslander, "BayesOD: A Bayesian approach for uncertainty estimation in deep object detectors," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2020, pp. 87–93.

[49] H. Grimmett, R. Triebel, R. Paul, and I. Posner, "Introspective classification for robot perception," *Int. J. Robot. Res.*, vol. 35, no. 7, pp. 743–762, Jun. 2016.

[50] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," 2018, *arXiv:1811.06817*. [Online]. Available: http://arxiv.org/abs/1811.06817

[51] Y. Gal, *Uncertainty in Deep Learning*, vol. 1, no. 3. Cambridge, U.K.: Univ. Cambridge, 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," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3266–3273.

[53] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3581–3590.

[54] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 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 *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 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 *Proc. SafeAI@AAAI*, 2020, pp. 83–90.

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

[58] M. Rottmann, P. Colling, T. P. Hack, R. Chan, F. Huger, P. Schlicht, and H. Gottschalk, "Prediction error meta classification in semantic segmentation: Detection via aggregated dispersion measures of softmax probabilities," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–9.

[59] K. Maag, M. Rottmann, and H. Gottschalk, "Time-dynamic estimates of the reliability of deep semantic segmentation networks," in *Proc. IEEE 32nd Int. Conf. Tools With Artif. Intell. (ICTAI)*, Nov. 2020, pp. 502–509.

[60] P. Wang and N. Vasconcelos, "Towards realistic predictors," in *Proc. ECCV*, 2018, pp. 36–51.

[61] V. V. Valindria, I. Lavdas, W. Bai, K. Kamnitsas, E. O. Aboagye, A. G. Rockall, D. Rueckert, and B. Glocker, "Reverse classification accuracy: Predicting segmentation performance in the absence of ground truth," *IEEE Trans. Med. Imag.*, vol. 36, no. 8, pp. 1597–1606, Aug. 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 *Proc. MICCAI*, 2018, pp. 578–585.

[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," 2020, *arXiv:2009.11732*. [Online]. Available: http://arxiv.org/abs/2009.11732

[64] M. Masana, I. Ruiz, J. Serrat, J. van de Weijer, and A. M. Lopez, "Metric learning for novelty and anomaly detection," 2018, *arXiv:1808.05492*. [Online]. Available: http://arxiv.org/abs/1808.05492

[65] S. Rabanser, S. Günnemann, and Z. C. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," 2018, *arXiv:1810.11953*. [Online]. Available: http://arxiv.org/abs/1810.11953

[66] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics: Science and Systems XIII*. Cambridge, MA, USA: Massachusetts Institute of Technology, Jul. 2017.

[67] F. Cai and X. Koutsoukos, "Real-time out-of-distribution detection in learning-enabled cyber-physical systems," in *Proc. ACM/IEEE 11th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2020, pp. 174–183.

[68] G. Rhafer and V. Vovk, "A tutorial on conformal prediction," *J. Mach. Learn. Res.*, vol. 9, pp. 371–421, Mar. 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," 2020, *arXiv:2011.01413*. [Online]. Available: http://arxiv.org/abs/2011.01413

[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," 2019, *arXiv:1911.07421*. [Online]. Available: http://arxiv.org/abs/1911.07421

[71] M. Jafarzadeh, T. Ahmad, A. R. Dhamija, C. Li, S. Cruz, and T. E. Boult, "Automatic open-world reliability assessment," 2020, *arXiv:2011.05506*. [Online]. Available: http://arxiv.org/abs/2011.05506

[72] C. K. Chow, "An optimum character recognition system using decision functions," *IRE Trans. Electron. Comput.*, vol. 6, no. 4, pp. 247–254, Dec. 1957.

[73] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 1, pp. 41–46, Jan. 1970.

[74] G. Fumera and F. Roli, "Support vector machines with embedded reject option," in *Proc. Int. Workshop Support Vector Mach.* Berlin, Germany: Springer, 2002, pp. 68–82.

[75] M. E. Hellman, "The nearest neighbor classification rule with a reject option," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-6, no. 3, pp. 179–185, Jul. 1970.

[76] C. Cortes, G. DeSalvo, and M. Mohri, "Boosting with abstention," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1660–1668.

[77] C. Cortes, G. DeSalvo, and M. Mohri, "Learning with rejection," in *Proc. Int. Conf. Algorithmic Learn. Theory*. Cham, Switzerland: Springer, 2016, pp. 67–82.

[78] Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4878–4887.

QUAZI MARUFUR RAHMAN received the bachelor's and master's degrees in computer science and engineering from the University of Dhaka, Bangladesh. He is currently pursuing the Ph.D. degree with the Centre for Robotics, Queensland University of Technology, and the Australian Centre for Robotic Vision, Australia. His research interest includes monitoring the run-time performance of the vision-based mobile robotic system using deep learning techniques.

PETER CORKE (Fellow, IEEE) received the bachelor's and master's degrees in electrical engineering and the Ph.D. degree from the University of Melbourne. He is currently a Distinguished Professor in robotic vision with the Queensland University of Technology, the Director of the ARC Centre of Excellence for Robotic Vision and the QUT Centre for Robotics (QCR). His research interests include with enabling robots to see, and the application of robots to mining, agriculture, and environmental monitoring. He created widely used open-source software for teaching and research, wrote the best-selling textbook *Robotics, Vision, and Control*, created several MOOCs and the Robot Academy, and has won national and international recognition for teaching, including 2017 Australian University Teacher of the Year. He is a Fellow of the Australian Academy of Technology and Engineering and the Australian Academy of Science, the Former Editor-in-Chief of the *IEEE Robotics and Automation Magazine*, the Founding Editor of the *Journal of Field Robotics*, the Founding Multi-Media Editor and an Executive Editorial Board Member of *The International Journal of Robotics Research*, a member of the Editorial Advisory Board of the *Springer Tracts in Advanced Robotics* series. He was a recipient of the Qantas/Rolls-Royce and Australian Engineering Excellence awards; and has held visiting positions at Oxford, the University of Illinois, Carnegie Mellon University, and the University of Pennsylvania.

FERAS DAYOUB received the Ph.D. degree in robotic vision from the University of Lincoln, U.K., in 2012. He is currently a Senior Lecturer with the School of Electrical Engineering and Robotics, Queensland University of Technology (QUT), Brisbane, QLD, Australia. He is a Chief Investigator with the QUT Centre for Robotics and the Chief Investigator with the ARC Centre of Excellence for Robotic Vision. His research interests include the deployment of machine learning and computer vision on mobile robots in challenging environments. He worked with various types of real-world robotic applications, such as Agricultural Robotics, Autonomous Underwater Vehicles (AUV), Unmanned Aerial Vehicles (UAV), and mobile service robots.