Hardware- and Situation-Aware Sensing for Robust Closed-Loop Control Systems

Sayandip De, Yingkai Huang, Sajid Mohamed, Dip Goswami, Henk Corporaal
Electronic Systems Group, Eindhoven University of Technology, The Netherlands
{sayandip.de, s.mohamed, d.goswami, h.corporaal}@tue.nl, {y.huang3}@student.tue.nl

***Abstract*—While vision is an attractive alternative to many sensors targeting closed-loop controllers, it comes with high time-varying workload and robustness issues when targeted to edge devices with limited energy, memory and computing resources. Replacing classical vision processing pipelines, e.g., lane detection using Sobel filter, with deep learning algorithms is a way to deal with the robustness issues while hardware-efficient implementation is crucial for their adaptation for safe closed-loop systems. However, while implemented on an embedded edge device, the performance of these algorithms highly depends on their mapping on the target hardware and situation encountered by the system. That is, first, the timing performance numbers (e.g., latency, throughput) depends on the algorithm schedule, i.e., what part of the AI workload runs where (e.g., GPU, CPU) and their invocation frequency (e.g., how frequently we run a classifier). Second, the perception performance (e.g., detection accuracy) is heavily influenced by the situation – e.g., snowy and sunny weather condition provides very different lane detection accuracy. These factors directly influence the closed-loop performance, for example, the lane-following accuracy in a lane-keep assist system (LKAS). We propose a hardware- and situation-aware design of AI perception where the idea is to define the situations by a set of relevant environmental factors (e.g., weather, road etc. in an LKAS). We design the learning algorithms and parameters, overall hardware mapping and its schedule taking the situation into account. We show the effectiveness of our approach considering a realistic LKAS case-study on heterogeneous NVIDIA AGX Xavier platform in a hardware-in-the-loop framework. Our approach provides robust LKAS designs with 32% better performance compared to traditional approaches.**

I. Introduction

A key step for successfully achieving a higher level of autonomy (e.g., level 5 autonomous vehicle) is effective usage of a combination of the camera (for being low-cost and well-developed) and artificial intelligence (for being ubiquitous). They form the basis for robust environmental sensing required for operational safety and functional correctness of next-generation closed-loop autonomous systems. Successful adaptation of these technologies in practice requires addressing two main challenges – hardware-efficiency for being used in resource-constraint edge devices (e.g., NVIDIA AGX Xavier [1]) and situation-robustness for safe and correct operations in a wide variety of situations (e.g., different lighting, road layout and marking) encountered in real-life.

For illustration, as shown in Fig. 1, we perform a trade-off analysis between lane detection accuracy over various situations and maximum achieved frames per second (FPS) on NVIDIA AGX Xavier for different lane detection algorithms. State-of-the-art (SOTA) convolutional neural network (CNN) based segmentation approaches such as VPGNet [2], LaneNet [3] etc. perform end-to-end learning with very high robustness to situational variations. However, these approaches provide low FPS when implemented on NVIDIA AGX Xavier, which makes them unsuitable to be used in closed-loop autonomous systems. On the other hand, traditional lane detection algorithms proposed in [4]–[7] detect lane segments based on diverse handcrafted cues like color-based features, structure tensor, bar filter and so on. Similarly, recent research [8], [9] has shown that approximate sensing can reduce high compute workloads at the cost of additional sensor noise which is tackled by robust control design. This class of algorithms achieve high FPS on edge devices, but they suffer from robustness issues due to situation variations raising safety concerns when used in closed-loop control systems.

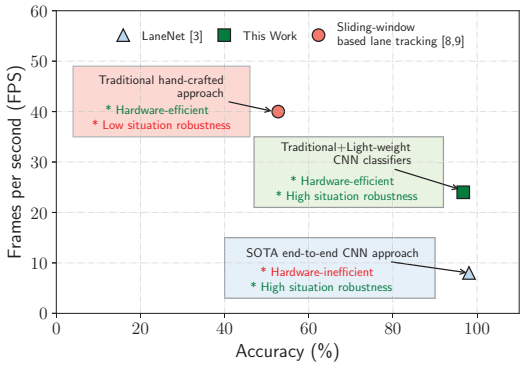


Fig. 1. Accuracy and FPS comparison between different lane detection techniques. Accuracies measured against the dataset introduced in Sec. III-C and FPS based on 512×256 resolution frames. NVIDIA AGX Xavier platform [1] constrained to power-budget of 30W is considered for deployment.

Contribution: We make the hardware-efficient traditional approaches [8], [9] robust to situations by introducing light-weight CNN classifiers and use this situational knowledge for runtime dynamic tuning of the required feature values. This translates to hardware-efficiency and situation-robustness, leading to fail-safe operation with a higher quality of control (QoC) in closed-loop systems.

Our proposed method[1] is demonstrated with a concrete case-study (Sec. II) and extensive experiments (Sec. IV).

II. Case-study

We consider an LKAS consisting of five main stages: camera sensor, image signal processing (ISP), perception (PR), control (T_c) and actuate (T_a). We introduce light-weight CNN classifiers, which dynamically tunes configuration knobs in the

[1]Our analysis framework is open sourced and can be accessed on github: **https://github.com/sayandipde/robust_dynamic_sesning**.

978-3-9819263-5-4/DATE21/©2021 EDAA

ISP, PR & T_c based on the situation and hardware setting. The camera and T_a are modelled and executed in the Webots physics simulation engine [10], while ISP, PR, CNN classifiers and T_c are executed in NVIDIA AGX Xavier using the IMACS framework [11]. Fig. 2 shows an overview of our setup adapted from [9], [11].

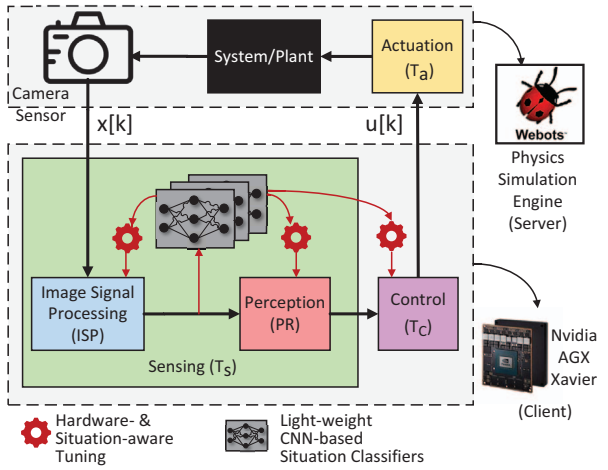
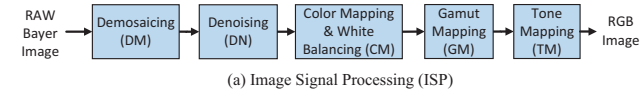


Fig. 2. Hardware-in-the-loop (HiL) setup for LKAS.

Below we provide an overview of LKAS from an algorithmic as well as hardware perspective.

Image Signal Processing (ISP): An ISP pipeline transforms a RAW image from Bayer domain to RGB domain. We consider five essential stages common to all ISP pipelines - demosaic, denoise, color map, gamut map and tone map, as defined in [8], [12] and shown in Fig. 3(a).

Perception (PR): During perception (PR), the lateral deviation (y_L) of the vehicle from the lane center is calculated. First, the region of interest (ROI) is selected based on the road situation. Then bird's eye view of the ROI is obtained using perspective transform, which is followed by binarization using dynamic thresholding. Finally, candidate lane pixels are determined using sliding windows from bottom to top of the image, and curve fitting is done using a second-order polynomial to calculate y_L at a look-ahead distance (LL). The different stages in PR are shown in Fig. 3(b).



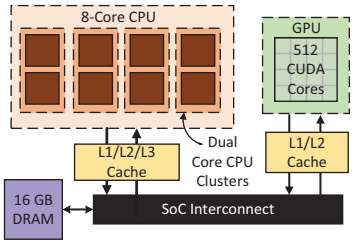
(a) Image Signal Processing (ISP)

(b) Perception (PR)

Fig. 3. Block-level overview of ISP and PR.

Discrete-time control (T_c): We consider the bicycle model [13] for the BMWX5 car model in Webots as introduced in [9]. The control takes y_L as an input and computes the actuating steering angle δ_f using the optimal linear quadratic regulator (LQR) [14]. A control design is annotated with a pair (h, τ) that models the sampling period h and the constant (worst-case) sensor-to-actuation delay τ associated with it for which we design the controller gains [15], [16]. In this work, the controller is designed considering LL = 5.5 m.

Considered Platform & Task Mapping: We consider NVIDIA AGX Xavier [1] for implementing LKAS. It is an edge device with a maximum power budget of 30W, making it suitable for use in modern electric vehicles (EVs). As shown in Fig.4(a), it has an 8-core NVIDIA Carmel ARMv8.2 CPU and an integrated 512-core NVIDIA Volta GPU, along with 16GB of LPDDR4x off-chip DRAM memory. Fig. 4(a) only shows the IPs used in this work.



(a) Simplified Block diagram of NVIDIA AGX Xavier

	CPU	GPU	Runtime
ISP	• Input data loading	• DM • DN • CM • GM • TM	21.5 ms
PR	• LT • LD	• ROI • PT • IT	3.0 ms
T_c	• Entire control		2.5 us

(b) LKAS task mapping & profiled runtimes

Fig. 4. Block-level platform details and LKAS task mappings. See Fig. 3 for details on the acronyms.

Fig. 4(b) shows the CPU-GPU task mapping for LKAS. Profiled runtimes for ISP and PR make it evident that sliding window based lane detection technique is suitable for real-time operation on NVIDIA AGX Xavier, achieving up to 40FPS (see Fig. 1). However, due to lack of situation-awareness, this technique has poor lane detection accuracy (52%, see Fig. 1), making it unsuitable for safety-critical LKAS. So, we introduce light-weight CNN classifiers for improving the robustness of this technique, as explained in Sec. III.

III. Hardware- and Situation-aware Optimization

Closed-loop control systems operate in a range of real-life situations which is crucial to be sensed to maintain safe and correct closed-loop functionality. As discussed, existing SOTA approaches are either performance optimized (in terms of temporal behavior and qualtify of control) while lacking robustness [9] or are highly robust lacking performance [3], as shown in Fig. 1. Proposed work finds a balance between robustness and closed-loop performance, which is essential for adaptation of these technologies in practice.

For improving the QoC of closed-loop systems, prior research [8], [9] has focused on reducing the delay τ by approximating the ISP. Additionally, traditional sliding window based lane detection techniques are used for PR, which are efficient for edge implementation. However, such an approach lacks robustness due to high sensitivity to the operating situation. So, in this work, we tackle this robustness problem by introducing CNN classifiers, which identifies the situation and dynamically re-configures the ISP and PR based on hardware- and situation-aware pre-characterization. A detailed step-wise overview of our proposed method is shown in Fig. 5. Below we outline the key steps involved in our methodology.

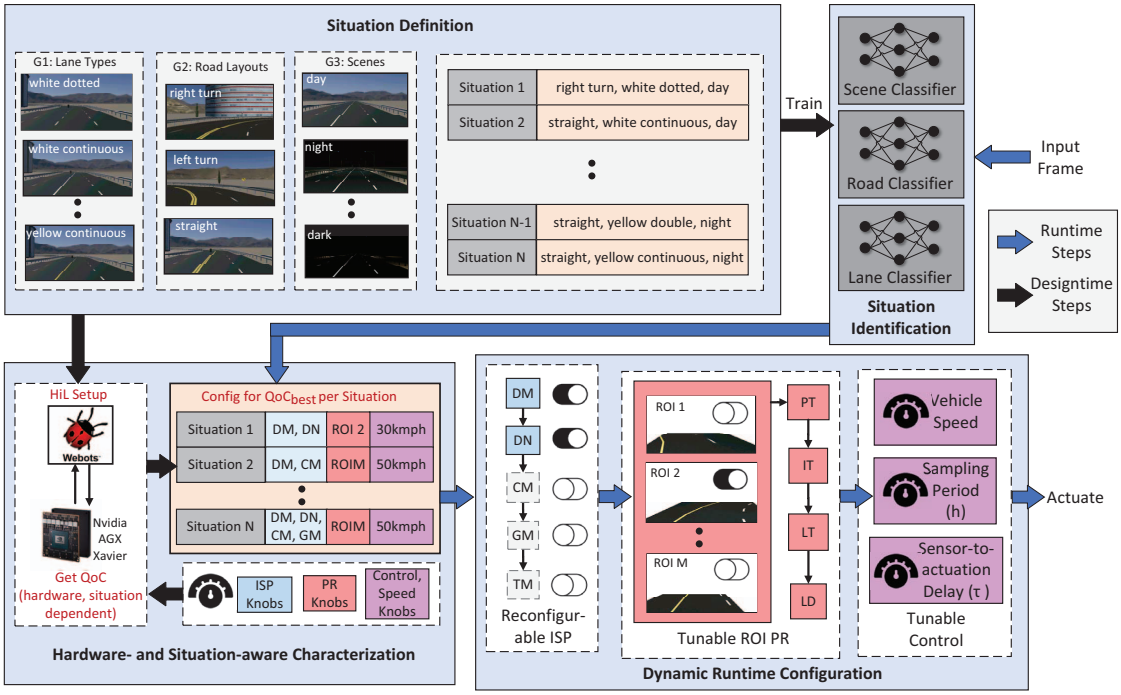


Fig. 5. Step-wise overview of our proposed hardware- and situation-aware method.

TABLE I
CONSIDERED FEATURES PER SITUATION.

Features	Detailed list
type of lane	color: white, yellow
	form: dotted, continuous, double continuous
layout of road	left turn, right turn, straight
type of scene/weather	day, night, dark, dawn, dusk

TABLE II
CONSIDERED CONFIGURABLE KNOBS IN THE SYSTEM.

Knobs	Detailed list	Runtime[2]
ISP Knobs	S0 : (DM, DN, CM, GM, TM)	21.5 ms
	S1 : (DM, CM, GM, TM)	18.9 ms
	S2 : (DM, DN, GM, TM)	20.9 ms
	S3 : (DM, DN, CM, TM)	3.3 ms
	S4 : (DM, DN, CM, GM)	3.2 ms
	S5 : (DM, DN)	3.1 ms
	S6 : (DM, CM)	3.2 ms
	S7 : (DM, GM)	3.1 ms
	S8 : (DM, TM)	3.2 ms
PR Knobs[3]	ROI 1 : (60, 0) (300, 0) (160, 65) (280, 65)	3.0 ms
	ROI 2 : (208, 0) (469, 0) (308, 72) (439, 72)	
	ROI 3 : (188, 0) (469, 0) (298, 72) (429, 72)	
	ROI 4 : (69, 0) (333, 0) (117, 72) (221, 72)	
	ROI 5 : (49, 0) (312, 0) (109, 72) (222, 72)	
Control Knobs	Vehicle speed (v) : 30 kmph, 50 kmph, Period (h), Sensor-to-actuation delay (τ)	2.5 us

DM, DN, CM, GM, TM - see Fig. 3 (a)
[3] profiled on NVIDIA AGX Xavier.
[4] ROI pixels are reported for 512x256 resolution frames.

A. Situation Definition

Situations are combinations of environmental factors that potentially influence the closed-loop performance. In this work, we define situations at design time by considering three main features that have the most impact on QoC: (1) type of lanes, (2) layout of roads and (3) type of scene/weather. Table I provides a detailed list of the considered features.

B. Hardware- and Situation-aware Characterization

The goal of hardware- and situation-aware characterization is to identify the set of LKAS parameters which perform best (QoC$_{best}$) under a specific situation at design time. For this, first, we determine the system parameters that are sensitive to the operating situation using Monte-Carlo simulations of the entire system [9] using our HiL setup . We observe that the approximations in the ISP, ROI selection in PR and vehicular speed selection in the controller (T_c) are the system parameters that heavily influence the closed-loop QoC. We will refer to these parameters as configurable knobs for the rest of the paper. Table II lists these considered configurable knobs.

After identifying system configurable knobs, design time characterization is done for each situation. For a particular situation, first, hardware-specific profiling is done for different sets of knob tunings. QoC is then obtained using closed-loop HiL simulations. The knob tuning values giving the best QoC per situation are then recorded. These recordings are later used for dynamic reconfiguration of knobs at runtime. Table III[4] provides a list of the pre-characterized situation-specific knob tunings for the best QoC. It is observed that not many ISP stages can be skipped for left turns with dotted lanes as it is

[4]We select a subset of the situations considered in Table I which are most frequently encountered. A similar approach can be followed for evaluating the rest of the situations.

TABLE III
PRE-CHARACTERIZED SITUATION-SPECIFIC KNOB TUNINGS FOR QoC_{best}.

Sit. No.	Situation Details	ISP knobs	PR knobs	T_c knobs [v, h, τ]
1	straight, white continuous, day	S3	ROI 1	[50, 25, 23.1]
2	straight, white dotted, day	S7	ROI 1	[50, 25, 22.4]
3	straight, yellow continuous, day	S4	ROI 1	[50, 25, 22.5]
4	straight, yellow double, day	S6	ROI 1	[50, 25, 22.5]
5	straight, white continuous, night	S6	ROI 1	[50, 25, 22.5]
6	straight, yellow continuous, night	S8	ROI 1	[50, 25, 23.0]
7	straight, white continuous, dark	S8	ROI 1	[50, 25, 23.0]
8	right, white continuous, day	S6	ROI 2	[30, 25, 22.5]
9	right, yellow continuous, day	S3	ROI 2	[30, 25, 23.1]
10	right, yellow double, day	S3	ROI 2	[30, 25, 23.1]
11	right, white continuous, night	S8	ROI 2	[30, 25, 23.0]
12	right, yellow continuous, night	S3	ROI 2	[30, 25, 23.1]
13	right, white dotted, day	S3	ROI 3	[30, 25, 23.1]
14	right, white dotted, night	S8	ROI 3	[30, 25, 23.0]
15	left, white continuous, day	S3	ROI 4	[30, 25, 23.1]
16	left, yellow continuous, day	S8	ROI 4	[30, 25, 23.0]
17	left, yellow double, day	S8	ROI 4	[30, 25, 23.0]
18	left, white continuous, night	S3	ROI 4	[30, 25, 23.1]
19	left, yellow continuous, night	S8	ROI 4	[30, 25, 23.0]
20	left, white dotted, day	S2	ROI 5	[30, 45, 40.7]
21	left, white dotted, night	S2	ROI 5	[30, 45, 40.7]

difficult to track them in the presence of approximation errors.

C. Situation Identification

For selecting the best tuned knobs, we need to identify the situations under which LKAS is operating at runtime. For this, three different light-weight CNN classifiers (scene, road & lane) are considered based on the Resnet-18 [17] architecture, as shown in Fig. 5. Sec. IV explains in details our design decision of choosing three different classifiers. Table IV gives a brief overview of the three classifiers considered in this work.

D. Dynamic Runtime Reconfiguration

For dynamic runtime reconfiguration, the input frames are analyzed first, and the operating situation is determined using the three classifiers introduced in Sec. III-C. Post situation identification, best hardware- and situation-specific knob tunings are selected based on the pre-characterization performed in Sec. III-B. It is noted that the situation identification is performed on the RGB images obtained from the ISP (see Fig. 2). So, the PR and control knobs are configured in the same cycles. However, the ISP knobs are configured in the next cycle. We argue that configuring the ISP with one cycle delay does not destabilize the closed-loop system (evident from results in Sec. IV) as real-life situations do not change per frame. For instance, while operating at 40 FPS @ 50 kmph, the vehicle progresses only 35 cms per frame, which is well below the look-ahead distance (LL = 5.5 m) considered for designing the controller. Controller stability while switching between different situations i due to varying τ_i and h_i is guaranteed by the existence of a common quadratic Lyapunov function (CQLF) as explained in [15], [16].

TABLE IV
CONSIDERED CLASSIFIERS FOR SITUATION IDENTIFICATION.

Classifiers	Details
Road Classifier	*dataset*: 5866 images (train: 5353, val: 513) *output classes*: straight, left turn, right turn *classification accuracy*: 99.92% *Profiled runtime on NVIDIA AGX Xavier*: 5.5 ms
Lane Classifier	*dataset*: 4781 images (train: 3939, val: 842) *output classes*: white continuous, white dotted, yellow continuous, yellow double *classification accuracy*: 99.97% *Profiled runtime on NVIDIA AGX Xavier*: 5.5 ms
Scene Classifier	*dataset*: 4703 images (train: 3892, val: 811) *output classes*: day, night, dark, dawn, dusk *classification accuracy*: 99.90% *Profiled runtime on NVIDIA AGX Xavier*: 5.5 ms

IV. EXPERIMENTAL RESULTS

A. Experimental Settings

We evaluate our hardware- and situation-aware method using the HiL setup introduced in Sec. II. For our evaluation, we consider a camera frame rate of 200 FPS in Webots. The actuation dynamics are modelled based on [18]. Lane width of 3.25 m is considered, as per standard road safety guidelines. It is noted that for our experiments, the left lane changes per situation, but the right lane is always set to white dotted, which is more common in the real world. The Webots simulation step is set to 5 ms[5], while the vehicle speed is set to either 50 km/hr or 30 km/hr depending on the operating situation, for all of our evaluations.

B. Quality of control

We evaluate the closed-loop QoC of LKAS using Mean Absolute Error (MAE) which is the mean of the cumulative sum of absolute errors. A lower MAE implies better performance.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y[k]| \tag{1}$$

where n is the no. of samples and $y[k]$ is the lateral deviation, y_L, at the k^{th} sample and ideally y_L should be *zero*.

C. Static Situation-specific Results and Analysis

To motivate the need for hardware- and situation-aware sensing for robust system design, we evaluate four different cases. We start our evaluation with a static LKAS design as the baseline with no classifiers in case 1. In case 2, we add only the road classifier. In case 3, we also add the lane classifier along with the road classifier. Finally, in case 4, we consider all the three classifiers together. Table V lists the knob settings for all these four cases. In this section, each situation is evaluated separately in a static manner. Dynamic switching from one situation to another is considered in Sec. IV-D.

Fig. 6 shows a comparative analysis of LKAS robustness and QoC for the above-mentioned four cases over different situations. All values are normalized to case 3. It is observed that considering no classifiers (case 1) lead to system failure for situations with left and right turns, thus, resulting in low LKAS

[5]h and τ are ceiled to the nearest factor of the simulation step for proper capture of frames and actuation of steering angles in Webots.

Fig. 6. Analysis of LKAS robustness and performance for different static situations. Four different cases are considered by incremental addition of classifiers. Lower MAE implies better LKAS performance. All values are normalized to case 3. Situations 1-21 are defined in Table. III.

TABLE V
CONSIDERED CASES FOR ANALYZING ROBUSTNESS AND QOC OF LKAS.

Case No.	Details	ISP knobs	PR knobs	T_c knobs [v, h, τ]
1	No classifiers Considered	S0	ROI 1	[50, 25, 24.6]
2	Only Road Classifier Considered	S0	VS	[VS, 35, 30.1]
3	Road+Lane Classifier Considered	S0	VS	[VS, 40, 35.6]
4	All Three Classifier Considered	VS	VS	[VS, VS, VS]

VS: Varied as per situation (see Table III for details).

robustness. This is due to a fixed region of interest (ROI 1). This motivates the need for a road classifier for situation-specific change of the ROI. Considering only the road classifier (case 2) improves LKAS robustness as it works for situations with right and left road layout, with the exception of situations with right/left turns in combination with dotted lanes. Fine-grained switching of ROIs (ROI 3 & 5) is needed for proper LKAS functionality in case of right/left turns with dotted lanes. So, we consider the lane classifier along with the road classifier (case 3) to identify situations with dotted lanes, and perform fine-grained ROI switching. This further improves LKAS robustness resulting in proper LKAS functionality for all situations under consideration, as shown in Fig. 6.

Improved LKAS robustness, however, comes at the cost of degraded performance. It is evident from Fig. 6 that case 1 performs the best for situations with straight road layout. This is due to lower τ and lower h. Each classifier adds runtime penalty of 5.5 ms[6], thus, resulting in a higher τ and h. This impacts LKAS QoC, resulting in case 1 performing better than case 2, which in turn performs better than case 3. Only exceptions are situations 15 & 16, where case 3 performs slightly better than case 2 due to additional sensor noise encountered in left turns. This can be solved by modeling the sensor noise in a linear-quadratic gaussian (LQG) controller, which is an interesting future research direction. Recent literature has shown that LKAS performance can be improved by approximating the ISP [8], [9], which reduces the τ and thus allows faster sampling and improved QoC. However, the loss in image quality due to approximations may degrade the QoC. Balancing this interaction is essential in determining if we gain or lose in final QoC. ISP approximations are taken into account in case 4. For hardware-and situation-specific ISP tuning of knobs as per Table III, a scene classifier is introduced along with the road and lane classifiers. Case 4 performs better than case 3 across all situations with the only exception of situation 15. In case of situation 15, the extra approximation error degrades the QoC much more than the QoC improvements due to faster sampling, thus, resulting in a worse QoC for case 4 compared to case 3.

[6]as ResNet-18 architecture is considered for all three classifiers.

D. Analysis of Dynamic Switching between Situations

For studying the dynamic switching between situations, a world model is developed in Webots consisting of different situations commonly encountered in real world, as shown in Fig. 7. The entire track consists of nine different sectors, where each sector corresponds to a particular situation. The vehicle starts from sector 1 (see Fig. 7), and as it progresses along the track, LKAS switches from one situation to another. The considered track covers dynamic road layout changes, lane type & color changes as well as scene changes (transition from night with street lights to dark with no street lights in sector 8 to 9). This track is considered as a concrete case study for evaluating the robustness and QoC of LKAS in the presence of proposed hardware- and situation-aware sensing.

Fig. 8 shows that the vehicle crashes while moving from sector 1 to 2 when no classifiers are considered (case 1), due to fixed ROI in PR. Inclusion of road classifier (case 2) improves LKAS robustness, but the vehicle crashes while moving from sector 5 to 6. Road classifier allows coarse-grained ROI changes based on road layout. But for better robustness, fine-grained ROI tunings are needed based on lane features (explained in Sec. IV-C). Thus, considering both road and lane classifiers (case 3) results in a robust LKAS design and the vehicle progresses through all the different sectors without crashing. Improved robustness comes with a performance penalty. This is evident from Fig. 8. On average, case 3 performs 55%, 22% worse compared to case 1 and 2 respectively[7]. To regain the performance with sacrificing robustness, ISP approximations are considered (case 4). For tuning ISP knobs, inputs from all three (road, lane & scene) classifiers are considered. On average, case 4 improves the QoC of LKAS by 30% compared to case 3 (robust baseline).

E. Tuning Invocation Frequency of Classifiers

In Sec. IV-D, the three classifiers (road, lane & scene) are invoked every frame in case 4. An option to improve LKAS QoC further is to vary the invocation frequency of the classifiers. This is motivated by the fact that certain situation-specific features do not change per frame, e.g., a scene transition from

[7]only considering sectors with no LKAS failure

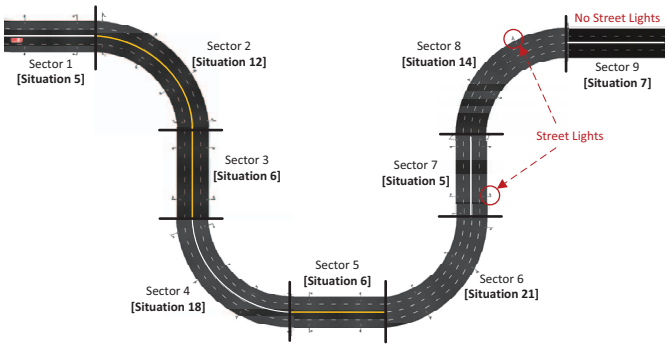


Fig. 7. Considered case study for studying dynamic switching of situations.

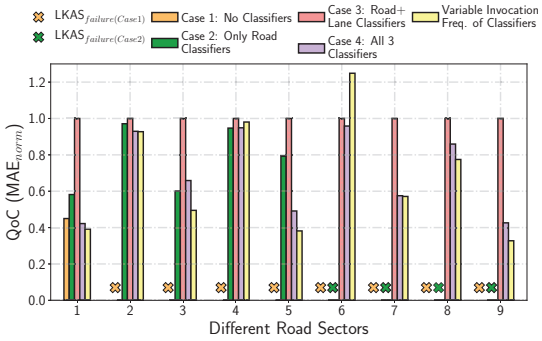


Fig. 8. Study of LKAS robustness and performance while switching between different road sectors/situations. All values are normalized to case 3.

day to night, lane color and type. In this work, we devise a simple scheme of invoking one classifier every frame. We observe that LKAS robustness is more sensitive to the road classifier compared to the lane and the scene classifier. So, we invoke the road classifier every frame for a duration of 300 ms[8]. After every 300 ms, we invoke the lane classifier in place of the road classifier. In the next frame, only the scene classifier is invoked. Again, for the next 300 ms, the road classifier is invoked every frame, and the cycle is repeated.

Fig. 8 shows that our invocation scheme improves LKAS QoC further compared to case 4, for all sectors except sector 4 and 6. It is observed that situations with left turns are difficult to track because of the dotted right lane being away from the camera frame. This introduces sensor noise and thus, degrades QoC of LKAS. Fine-grained ROI tuning for dotted lanes with situational knowledge from lane classifier helps to compensate for this. Proposed variable scheme invokes the lane classifier once every 300 ms, which doesnot allow fine-grained ROI tuning, resulting in degraded QoC for sectors 4 & 6. Sector 6 shows a higher QoC degradation compared to sector 4 due to both left and right lanes being dotted. On average, variable invocation frequency for classifiers improves LKAS QoC by 32%, 3% compared to case 3 and 4, respectively.

V. Conclusion and Future Work

In the context of applying AI in closed-loop systems, we tackled the problem of balancing between situation robustness and performance in terms of temporal behavior and QoC. This is an important design challenge to be addressed for adaptation of these technologies in many domains, including autonomous driving. The essence of our approach is to perform situation identification and optimization at design time and runtime reconfiguration based on the encountered situation. As opposed to the state-of-the-art end-to-end CNN-based approaches, we propose to use multiple light-weight CNN classifiers since it offers flexibility in terms of situation-specific implementation and invocations. We reported a simple invocation scheme. A more complete invocation scheme can be developed in future.

While our results are application-specific, our methodology can be adopted for other domains by re-defining the situations and application-specific optimizations. Choice of light-weight classifiers needs to be reconsidered based on situation definitions. Once the situations & classifiers are defined, the rest of the methodology can be seamlessly adopted.

[8]Evaluation window of 300 ms is considered for the invocation frequency study. This is because our LQR controller calculates y_L at a look-ahead distance of 5.5 m and the top vehicle speed is 50 kmph. So, the current control decision is valid for a position the vehicle will reach after $\sim$ 400 ms. So, to not make the system unstable, we consider a window of 300 ms.

VI. Acknowledgment

This work has received partial funding from the NWO Perspectief program ZERO and the ECSEL Joint Undertaking grant H2020-ECSEL-2017-2-783162 (FitOptiVis).

References

[1] "NVIDIA AGX Xavier, Scalable AI Platform for Autonomous Driving." [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier
[2] S. Lee et al., "VPGNet: Vanishing point guided network for lane and road marking detection and recognition," in *ICCV*, 2017.
[3] D. Neven et al., "Towards end-to-end lane detection: an instance segmentation approach," in *IEEE IV*, 2018.
[4] J. C. McCall et al., "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," in *IEEE T-ITS*, 2006.
[5] Kuo-Yu Chiu et al., "Lane detection using color-based segmentation," in *IEEE Proceedings. Intelligent Vehicles Symposium (IV)*, 2005.
[6] H. Loose et al., "Kalman particle filter for lane recognition on rural roads," in *IEEE Intelligent Vehicles Symposium (IV)*, 2009.
[7] Z. Teng et al., "Real-time lane detection by using multiple cues," in *ICCAS*, 2010.
[8] S. De et al., "Approximation trade offs in an image-based control system," in *DATE*, 2020.
[9] S. De et al., "Approximation-aware design of an image-based control system," *IEEE Access*, 2020.
[10] O. Michel, "Cyberbotics ltd. webots™: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, 2004.
[11] S. Mohamed et al., "IMACS: a framework for performance evaluation of image approximation in a closed-loop system," in *MECO*, 2019.
[12] M. Buckler et al., "Reconfiguring the imaging pipeline for computer vision," in *ICCV*, 2017.
[13] J. Kosecka et al., "Vision-based lateral control of vehicles," in *Proceedings of Conference on Intelligent Transportation Systems*, 1997.
[14] G. F. Franklin et al., *Digital control of dynamic systems*, 1998.
[15] S. Mohamed et al., "Designing image-based control systems considering workload variations," in *CDC*, 2019.
[16] S. Mohamed et al.,"A scenario-and platform-aware design flow for image-based control systems," in *MICPRO*, 2020.
[17] K. He et al., "Deep residual learning for image recognition," in *CVPR*, 2016.
[18] R. Frank, "Steering in the Right Direction," in *Electronic Design*, 2016.