

UNIVERSITÀ POLITECNICA DELLE MARCHE
Repository ISTITUZIONALE

Edge AI for Monocular Cone Perception in Formula Student Driverless Racing

This is the peer reviewd version of the followng article:

Original
Edge AI for Monocular Cone Perception in Formula Student Driverless Racing / Fava, T., Berardini, D., Mancini, A., Galdelli, A., Falaschetti, L.. - ELETTRONICO. - (2026). (2026 IEEE International Conference on Omni Layer Intelligent Systems (COINS) Bologna, Italy 07-09 September 2026) [10.1109/COINS69995.2026.11714251].

Availability:
This version is available at: 11566/364752 since: 2026-10-15T16:26:36Z

Publisher:
IEEE

Published
DOI:10.1109/COINS69995.2026.11714251

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. The use of copyrighted works requires the consent of the rights' holder (author or publisher). Works made available under a Creative Commons license or a Publisher's custom-made license can be used according to the terms and conditions contained therein. See editor's website for further information and terms and conditions.
This item was downloaded from IRIS Università Politecnica delle Marche (https://iris.univpm.it). When citing, please refer to the published version.

Publisher copyright:

IEEE - Postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. To access the final edited and published work see 10.1109/COINS69995.2026.11714251

(Article begins on next page)

04 November 2026

Edge AI for Monocular Cone Perception in Formula Student Driverless Racing

Tommaso Fava, Daniele Berardini, Adriano Mancini, Alessandro Galdelli, Laura Falaschetti*
Department of Information Engineering, Università Politecnica delle Marche,
Ancona, Italy
*Corresponding author: l.falaschetti@staff.univpm.it

***Abstract*—Reliable perception is a fundamental requirement for autonomous racing systems, where real-time operation and limited onboard computational resources impose strict constraints on algorithm design and deployment. In the context of Formula Student Driverless (FSD), cone detection and localization are key perception tasks for track understanding and downstream planning. This paper presents an embedded monocular cone perception pipeline for autonomous racing, designed as a low-cost and low-complexity solution for real-world deployment. The proposed system combines a calibrated monocular camera, a YOLO-based object detector, and a geometric localization module based on the pinhole camera model to estimate cone positions in the vehicle reference frame. The pipeline is deployed on an NVIDIA Jetson Orin Nano platform and evaluated in terms of detection accuracy, localization error, runtime performance, and power consumption. Experimental results show that the best performing configuration, based on INT8-quantized YOLOv10s, achieves an mAP@50:95 of 48.53% and a localization RMSE of 0.760 m, while sustaining 29.90 FPS with an end-to-end power consumption of approximately 7.1 W. A lighter YOLOv10n configuration achieves 46.73% mAP@50:95, 0.840 m RMSE, and 31.80 FPS at 6.8 W, providing a favorable trade-off between accuracy and efficiency. These results indicate that monocular vision can provide an effective embedded perception baseline for FSD autonomous racing under tight cost, power, and complexity real-time constraints. The proposed pipeline is conceived as a perception building block for onboard artificial intelligence (AI) in autonomous racing systems.**

***Keywords*—Embedded perception, monocular vision, cone detection, Formula Student Driverless, autonomous racing, YOLO.**

I. Introduction

Autonomous racing represents a challenging testbed for real-time robotic perception and decision-making, as it requires accurate environment understanding under strict latency, power, and onboard computing constraints [1], [2]. Reliable perception is a core enabling component of onboard intelligence, as it supports track boundary estimation, landmark localization, and the environmental understanding required by downstream mapping, planning, and control modules. In Formula Student Driverless (FSD), one of the most critical perception tasks is cone detection and localization, since colored traffic cones define the track layout and represent the primary visual landmarks for navigation. Moreover, the known color and geometry of the cones make FSD a structured yet meaningful benchmark for the development and validation of embedded autonomous systems deployed in the real world [1], [3]–[5].

Many autonomous racing platforms rely on multi-sensor perception stacks to improve robustness and accuracy [1], [2], [6]. However, such solutions often increase system cost, integration effort, and computational complexity, making them less suitable for early-stage prototypes and resource-constrained deployments [5], [6]. In this context, monocular vision is an attractive alternative due to its low cost, compactness, and rich semantic information, provided that real-time inference and geometric localization can be achieved reliably on embedded hardware [4], [7], [8].

This work builds upon the first author's master thesis [9] and presents an embedded monocular cone perception pipeline for FSD autonomous racing. The proposed system combines a calibrated monocular camera, a YOLO-based object detector, and a geometric localization module based on the pinhole camera model to estimate cone positions in the vehicle reference frame. The full pipeline is deployed on an NVIDIA Jetson Orin Nano platform, targeting a balanced trade-off between perception accuracy, runtime performance, power consumption, and deployment cost. Experimental results show that the proposed pipeline enables real-time embedded operation while maintaining useful perception accuracy for an early autonomous racing prototype. These findings indicate that low-cost monocular perception can provide a practical baseline for embedded autonomous racing systems operating under tight resource constraints.

The main contributions of this work are threefold. First, an embedded monocular perception pipeline for cone detection and localization in FSD is presented. Second, the deployment of the proposed pipeline on low-power Jetson hardware, specifically Jetson Orin Nano, is analyzed through an evaluation of detection accuracy, localization error, runtime performance, and power consumption. Third, the obtained results show that a low-cost camera-based solution can support real-time autonomous racing perception and serve as a scalable foundation for future system extensions. From this perspective, the proposed system is conceived not as an isolated vision module, but as an embedded artificial intelligence (AI) component for autonomous robotic racing, designed to operate under real-time and power constraints on onboard hardware.

II. Related Work

Formula Student Driverless (FSD) has been widely adopted as a benchmark scenario for the development of autonomous

racing systems. Several works describe complete driverless racecar architectures integrating perception, localization, planning, and control modules, often based on multi-sensor configurations designed to maximize robustness and performance [1]–[3]. While these systems demonstrate the effectiveness of tightly integrated pipelines, they also highlight the engineering complexity and computational cost associated with multi-sensor solutions.

Within this context, cone detection and localization have been extensively investigated, as they provide the basis for track reconstruction and vehicle navigation. Existing approaches include both vision-based methods and multi-sensor systems combining cameras with LiDAR or stereo setups. Vision-only methods typically rely on deep learning-based detectors coupled with geometric or learning-based depth estimation, offering compact and cost-effective solutions [4], [7], [10], [11]. In contrast, multi-sensor approaches improve robustness and localization accuracy at the expense of increased system complexity and computational requirements [6].

A growing body of work has focused on lightweight and resource-aware approaches for autonomous racing. In particular, embedded perception pipelines deployed on platforms such as NVIDIA Jetson have been proposed to enable real-time operation under strict power and hardware constraints [5], [8]. These approaches emphasize deployability and scalability, which are critical for early-stage prototypes and cost-constrained applications.

Deep learning-based object detectors from the YOLO family have become widely adopted in real-time perception tasks due to their favorable trade-off between detection accuracy and inference speed [12], [13]. These characteristics make them particularly suitable for embedded and resource-constrained environments.

In this landscape, the present work focuses on an embedded monocular perception pipeline for FSD autonomous racing. Compared with full multi-sensor stacks, the proposed approach prioritizes low cost, reduced complexity, and ease of deployment, while combining object detection and geometric localization within a unified embedded system.

III. System Design

A. System Overview

The proposed perception system is designed to provide real-time cone detection and localization for FSD autonomous racing using a minimal hardware setup. The overall pipeline takes as input RGB images acquired from a calibrated monocular camera and resizes them to a fixed resolution of 640×640 (corresponding to the input size of the detection network), extracted from a region of interest centered in the lower part of the image.

The system outputs the estimated (X, Y) coordinates of the detected cones in the vehicle reference frame, enabling the generation of a local bird's-eye view representation. The system is composed of four main stages: image acquisition, object detection, geometric localization, and result visualization. A schematic overview of the pipeline is shown in Fig. 1.

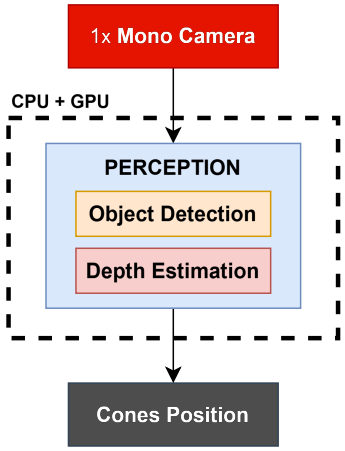


Fig. 1. Overview of the proposed embedded monocular cone perception pipeline, including image acquisition, YOLO-based detection, geometric localization, and bird's-eye view cone projection.

The design follows three main objectives: low cost, low complexity, and scalability. To this end, the hardware configuration is intentionally limited to a single camera and an embedded computing platform, avoiding the integration burden associated with more complex multi-sensor architectures. This choice is particularly suited to early-stage autonomous racing prototypes, where affordability and ease of deployment are key requirements.

B. Detection Module

Cone detection is performed using YOLO-based object detectors trained on the FSOCO dataset [14], [15], which contains annotated Formula Student Driverless cone instances under racing-like conditions. Among the evaluated models, YOLOv10n and YOLOv10s were selected for embedded deployment due to their favorable trade-off between detection accuracy and runtime performance.

Input images are acquired at a resolution of 1920×1200 pixels and preprocessed by extracting a 640×640 region of interest centered in the lower part of the image, matching the input size of the detection network. This ROI-based preprocessing focuses computation on the most informative region of the scene, where cones are more likely to appear. The detector outputs bounding boxes and class labels for the visible cones. To support efficient deployment on the target platform, trained PyTorch models are exported to ONNX and converted into TensorRT engines, with post-training quantization to INT8 precision. This optimization significantly reduces inference cost while preserving competitive detection performance, enabling real-time execution on embedded hardware.

C. Geometric Localization

Detected cones are localized through a monocular geometric estimation module based on the pinhole camera model. The method exploits prior knowledge of cone dimensions together

with the intrinsic parameters obtained from camera calibration. Starting from the cone height measured in the image, the depth is estimated by inverting the pinhole projection model. The resulting optical distance is then converted into ground distance by accounting for the mounting height of the camera, and the lateral coordinate is computed from the horizontal image position of the detection. In this way, each cone is represented by a 2D position in the vehicle reference frame. The camera intrinsic parameters were obtained through a standard calibration procedure, while the extrinsic parameters required by the geometric model (camera mounting height and orientation) were measured from the physical setup and used during localization.

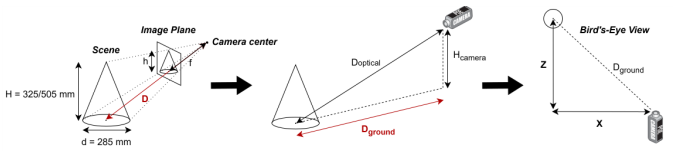


Fig. 2. Monocular geometric localization pipeline. The cone distance is estimated from its apparent height using the pinhole camera model, then converted to ground distance and projected into the vehicle reference frame.

As illustrated in Fig. 2, the localization process consists of three main steps: depth estimation from the apparent cone height, conversion to ground distance, and computation of the lateral position.

This approach was selected as a lightweight alternative to stereo or learning-based depth estimation methods. Although it relies on simplifying assumptions, most notably a flat ground plane and known object geometry, it offers low computational cost and straightforward integration with the detection module, enabling real-time execution on embedded hardware without requiring additional sensors. This formulation trades geometric simplicity for computational efficiency, making it particularly suitable for resource-constrained autonomous systems.

D. Embedded Deployment

The full perception pipeline is deployed on an NVIDIA Jetson Orin Nano Developer Kit, which serves as the target embedded computing platform of the system [16]. The software is implemented in C++ and relies on CUDA, cuDNN, and TensorRT to accelerate inference and support low-latency execution on the onboard GPU. The hardware setup includes a single IDS UI-3160CP-C-HQ industrial monocular RGB camera for image acquisition, integrated with the embedded computing platform. The camera is equipped with a global shutter and a native resolution of 1920 $\times$ 1200 pixels resolution, selected to provide adequate image quality while maintaining a compact and relatively low-cost configuration [17].

A detailed view of the implemented perception architecture is shown in Fig. 3, where the main software modules, configuration files, and CPU/GPU execution domains are explicitly identified. This modular organization supports system configurability and facilitates deployment on embedded hardware, while also simplifying future integration with higher-level

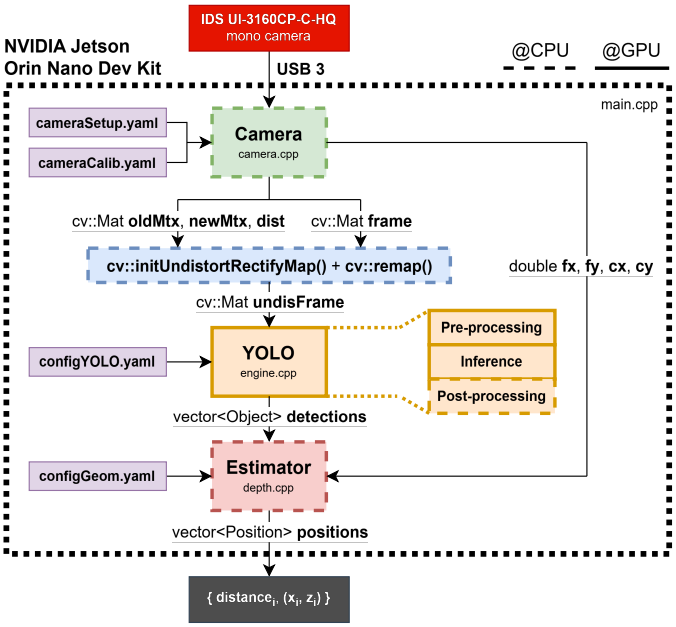


Fig. 3. Detailed diagram of the designed perception system. Modules are shown in colored boxes, labeled with their names and corresponding source code filenames. Solid outlines indicate GPU execution, while dashed outlines denote CPU execution. Violet boxes represent YAML configuration files.

autonomous racing modules. Besides returning cone detections and estimated positions, the application can generate a bird's-eye view representation of the perceived track landmarks. This output is intended to support future integration with mapping, planning, and control components within a broader autonomous racing software stack.

IV. Experimental Setup

A. Dataset and Model Selection

1) Dataset: The detection module was trained and evaluated on the FSOCO dataset [14], freely available from the official website [15], which contains annotated cone instances acquired in FSD scenarios and is therefore representative of the target application domain. FSOCO dataset is a collection of 11572 high-quality annotated images coming from 18 teams with a large variety of sites, lighting conditions and sensor setups. It includes five classes: yellow, blue, orange, large orange and other, non-compliant cones—from now on mentioned as the unknown class. More in detail, the dataset contains a total of 220816 cones. By class, there are 93230 (42.22%) yellow cones, 86097 (38.99%) blue cones, 25991 (11.77%) orange cones, 8217 (3.72%) large orange cones, and 7281 (3.29%) unknown cones. This results from the yellow and blue cones being present throughout the track, while the orange cones mark specific areas.

2) Split: As there is no predefined division into training, validation, and test sets, an 80/10/10 split is applied, which is a commonly used approach in machine learning, with a balanced distribution of cone instances across splits providing a sufficient amount of training data while preserving independent validation and test sets for model selection and final

evaluation. This leads to the cone distribution reported both in Table I and Fig. 4.

TABLE I
NUMBER OF IMAGES AND CONE DISTRIBUTION IN THE FSOCO DATASET SPLIT INTO TRAIN (80%), VALIDATION (10%) AND TEST (10%) SETS.

	Imgs	% Yellow	% Blue	% Orange	% L. Orange	% Unk
Train	9256	42.36	39.08	11.81	3.72	3.18
Val	1157	41.69	38.68	12.46	3.63	3.54
Test	1158	41.58	38.57	12.02	3.85	3.98

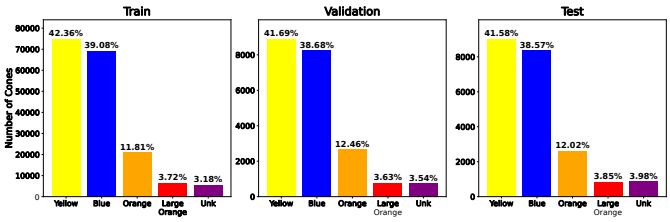


Fig. 4. Cone distribution in the FSOCO dataset split into train (80%), validation (10%) and test (10%) sets. Each columns color represents the correponding class, where 'L. orange' means yellow large orange cones and 'Unk' stands for unknown cones.

3) Model Selection: Several YOLO-based architectures were considered during development, including models from the YOLOv5, YOLOv8, YOLOv10, and YOLOv11 families. Preliminary experiments showed that YOLOv10 models provided the best trade-off between detection accuracy, latency, and power consumption on the target embedded platform. For this reason, the final evaluation focuses on YOLOv10n and YOLOv10s configurations. Additional comparisons with other YOLO variants are available in the extended thesis version of this work [9].

4) Training and Deployment: The detection models were trained offline using the Ultralytics framework [18], [19], based on PyTorch. Training was performed on a separate workstation, while the embedded platform is used exclusively for inference. For deployment on the embedded platform, the trained models were exported to ONNX and converted into TensorRT engines [20]. The inference pipeline was implemented in C++ using TensorRT 8.6, CUDA Runtime 12.2.1, and OpenCV 4.10.0 to ensure real-time performance on the Jetson device. It consists of four main steps: object creation, pre-processing, inference, and post-processing.

In addition to standard reduced-precision execution, INT8 post-training quantization was applied using a calibration set of 1000 images, reducing inference latency and computational cost while preserving competitive detection performance [21].

B. Hardware Platform and Runtime Conditions

All experiments were conducted on an NVIDIA Jetson Orin Nano Developer Kit with 8 GB of LPDDR5 memory, used as the target embedded computing platform. Image acquisition was performed with an IDS UI-3160CP-C-HQ industrial monocular RGB camera equipped with a global shutter and a native resolution of 1920×1200 pixels. Two interchangeable lenses with focal lengths of 6 mm and 8 mm were evaluated to assess the impact of camera optics on monocular cone localization. The 6 mm lens provides a wider field of view, while the 8 mm lens increases the apparent object size but narrows the field of view. Both configurations were calibrated and tested under the same setup. Although detection accuracy was evaluated on the public FSOCO dataset, the proposed hardware platform was used to validate the geometric localization module and to benchmark the embedded perception pipeline under the target deployment conditions.

Prior to deployment, the camera was calibrated to estimate intrinsic parameters and lens distortion coefficients, while the extrinsic parameters required by the localization model (camera mounting height and orientation) were measured from the physical setup. This configuration was selected to provide adequate image quality for cone perception while maintaining a compact and relatively low-cost embedded setup.

Runtime benchmarks were executed under the Jetson 7 W and 15 W power modes, corresponding to low-power and performance-oriented configurations of the platform, as defined by the NVIDIA Jetson power management profiles [22], [23], with Jetson Clocks enabled to ensure stable performance measurements.

Each latency measurement consisted of 2000 benchmark runs preceded by 200 warm-up runs on 1920×1200 input images. This procedure was adopted to reduce measurement variance and obtain representative throughput estimates under realistic deployment conditions. During inference, input images acquired from the monocular RGB camera were preprocessed by extracting a 640×640 region of interest before being fed to the detection network. The overall pipeline includes object detection, geometric localization, and projection into the vehicle reference frame, resulting in end-to-end cone position estimation.

C. Evaluation Metrics

The performance of the proposed system is evaluated in terms of detection accuracy, localization precision, runtime performance, and power consumption. Detection performance is evaluated using the standard mAP@50:95 metric, as defined in the COCO benchmark [24], which averages precision across multiple intersection-over-union (IoU) thresholds and provides a more stringent evaluation than single-threshold metrics, while localization accuracy was assessed through the root mean square error (RMSE) of the estimated cone positions in the vehicle reference frame.

Embedded execution performance was quantified through end-to-end frame rate, total power consumption, and energy efficiency expressed in frames per second per watt. Power consumption was measured through the VDD_IN value reported by jetson-stats [25].

This evaluation setup was intended to capture the main design trade-offs of embedded perception for autonomous racing, namely the balance between detection performance, geometric localization quality, computational throughput, and power efficiency.

V. Experimental Results

All reported results correspond to end-to-end execution of the full perception pipeline, including detection and geometric localization. Since the focus of this work is the deployment of embedded perception systems, the experimental evaluation reports only the final INT8 configurations.

A. Quantitative Performance

Table II summarizes the performance of the final embedded configurations evaluated on FSOCO [14], [15] in terms of detection accuracy, localization error, throughput, and power consumption. Among the evaluated solutions, the INT8-quantized YOLOv10s configuration achieved the best overall perception performance, reaching 48.53% mAP@50:95 and a localization RMSE of 0.760 m, while maintaining real-time operation at 29.90 FPS with an end-to-end power consumption of approximately 7.1 W. The YOLOv10n configuration achieves slightly lower accuracy, with 46.73% mAP@50:95 and 0.840 m RMSE, while providing the highest throughput at 31.80 FPS and a lower power consumption of approximately 6.8 W.

TABLE II
Performance comparison of the final embedded configurations in terms of detection accuracy, localization error, throughput, and power consumption.

Model	Precision	mAP@50:95 (%)	RMSE (m)	FPS	Power (W)	FPS/W
YOLOv10n	INT8	46.73	0.84	**31.80**	**6.8**	**4.68**
YOLOv10s	INT8	**48.53**	**0.76**	29.90	7.1	4.21

These results show that both configurations satisfy the real-time requirements of an embedded autonomous racing perception module, while exposing a clear trade-off between accuracy and efficiency. In particular, YOLOv10s improves detection accuracy and localization precision, reaching the best overall performance, at the cost of a slight reduction in throughput and a marginal increase in power consumption. Conversely, YOLOv10n represents a lighter alternative, providing higher frame rates and better energy efficiency. This trade-off is particularly relevant in embedded autonomous systems, where performance must be balanced against strict resource constraints. Consequently, YOLOv10n is preferable for resource-constrained deployments, whereas YOLOv10s offers the best overall perception performance. The observed localization error is consistent with the simplifying assumptions of the geometric model, including the flat-ground approximation and the reliance on bounding box height for depth estimation.

To further analyze the effect of camera optics on the proposed geometric localization module, Table III reports the maximum error and RMSE obtained with the two calibrated lens configurations. The results confirm that the system achieves sub-meter localization accuracy, with the best performance obtained using YOLOv10s and a 6 mm lens, reaching an RMSE of 0.760 m. The 6 mm configuration consistently achieved lower localization errors than the 8 mm lens for both evaluated models, and was therefore selected as the reference configuration for the final embedded system. However, the observed differences are relatively small, and further investigation is required to fully characterize the influence of lens parameters on localization performance.

TABLE III
Localization error for cone position estimation using YOLOv10n and YOLOv10s (INT8) under different camera lens configurations.

Model	Lens	Max Localization Error (m)	RMSE (m)
YOLOv10n	6 mm	0.987	0.840
	8 mm	0.988	0.870
YOLOv10s	6 mm	**0.931**	**0.760**
	8 mm	1.059	0.823

B. Embedded Deployment Analysis

The results reported in Table II confirm that TensorRT optimization and INT8 post-training quantization enable efficient execution of YOLO-based detectors on the Jetson Orin Nano platform. Both evaluated configurations satisfy real-time constraints while operating within a limited power envelope, demonstrating the suitability of the proposed pipeline for real-time autonomous racing applications.

From a system-level perspective, this outcome is significant because it demonstrates that a single-camera perception pipeline can provide a practical baseline for autonomous racing without requiring more complex and integration-heavy sensing stacks. Although the geometric localization stage relies on simplified assumptions, including known cone geometry and a flat-ground approximation, the resulting pipeline achieves a favorable compromise between deployability, cost, runtime performance, and perception quality. From a computational perspective, the object detection stage dominates the overall latency of the pipeline, indicating that further optimizations should primarily target neural network inference. Overall, these observations confirm that the proposed system is well-suited for real-time deployment on resource-constrained embedded platforms.

C. Discussion

The obtained results indicate that monocular embedded perception is a viable solution for early-stage FSD prototypes and for autonomous racing systems operating under tight hardware and power constraints. The proposed approach does not aim to replace richer multi-sensor stacks in terms of absolute robustness or localization precision; rather, it provides a lightweight and scalable perception baseline that can be deployed on low-cost hardware and integrated into a broader autonomous racing architecture.

At the same time, the current system still presents limitations related to depth estimation accuracy, sensitivity to de-

tection noise, and dependence on scene assumptions inherent to monocular geometric localization. Although the evaluation is conducted on the FSOCO dataset, further validation on real-world racing environments would be beneficial to assess robustness under varying lighting conditions and track layouts. These aspects suggest several directions for future work, including temporal filtering, sensor fusion, and tighter integration with higher-level mapping and planning modules.

Overall, the results demonstrate that the proposed monocular perception pipeline achieves a practical balance between accuracy, efficiency, and deployability, making it suitable for real-time autonomous racing applications on embedded platforms.

VI. Conclusion

This paper presented an embedded monocular cone perception pipeline for Formula Student Driverless autonomous racing. The proposed system combines YOLO-based cone detection and geometric localization within a compact and low-cost architecture deployed on an NVIDIA Jetson Orin Nano platform. Experimental evaluation showed that the pipeline achieves real-time embedded operation while maintaining useful detection and localization performance under strict power and computational constraints. The obtained results indicate that monocular vision can provide a practical baseline for autonomous racing perception in resource-constrained scenarios. In particular, the evaluated configurations highlighted a clear trade-off between perception accuracy and deployment efficiency, with YOLOv10s offering the best overall accuracy and YOLOv10n providing the highest throughput and energy efficiency.

The proposed pipeline is conceived as a perception building block for onboard AI in autonomous racing systems, enabling future integration with higher-level decision-making, mapping, and control modules.

Future work will focus on improving localization robustness through temporal filtering and sensor fusion, as well as on further system-level integration and validation in real-world racing conditions.

References

[1] J. Kabzan, M. de la Iglesia Valls, V. Reijgwart, H. F. C. Hendrikx, C. Ehmke, M. Prajapat, A. Bühler, N. B. Gosala, M. Gupta, R. Sivanesan, A. Dhall, E. Chisari, N. Karnchanachari, S. Brits, M. Dangel, I. Sa, R. Dubé, A. Gawel, M. Pfeiffer, A. Liniger, J. Lygeros, and R. Siegwart, "AMZ driverless: The full autonomous racing system," *CoRR*, vol. abs/1905.05150, 2019.

[2] S. Nekkah, J. Janus, M. Boxheimer, L. Ohnemus, S. Hirsch, B. Schmidt, Y. Liu, D. Borbély, F. Keck, K. Bachmann, and L. Bleszynski, "The autonomous racing software stack of the KIT19d," *CoRR*, vol. abs/2010.02828, 2020.

[3] h. TIAN, J. NI, and J. HU, "Autonomous driving system design for formula student driverless racecar," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–6, 2018.

[4] L. Qie, J. Gong, H. Zhou, S. Wang, S. Zhou, and N. B. Chetan, "Cone detection and location for formula student driverless race," in *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, pp. 440–444, 2020.

[5] T. Friedrich, M. Müller, A. Bauske, S. Härtl, J. Herrmann, D. Förster, T. Tietze, and S. Sartor, "Winning through simplicity: Autonomous car design for formula student," 2024.

[6] A. Paleczny, R. Górecki, K. Baradziej, and P. Kohut, "Cone detection system with camera and LiDAR in formula student driverless car," in *2024 15th Annual Undergraduate Research Conference on Applied Computing (URC)*, pp. 1–6, 2024.

[7] M. Svecovs and F. Hörnschemeyer, "Real time object localization based on computer vision: Cone detection for perception module of a racing car for formula student driverless," M.S. thesis, Chalmers University of Technology, 2020.

[8] J. Albaráñez Martínez, L. Llopis-Ibor, S. Hernández-García, S. Pineda de Luelmo, and D. Hernández-Ferrándiz, "A case of study on traffic cone detection for autonomous racing on a jetson platform," in *Pattern Recognition and Image Analysis: 10th Iberian Conference, IbPRIA 2022, Aveiro, Portugal, May 4–6, 2022, Proceedings*, (Berlin, Heidelberg), p. 629–641, Springer-Verlag, 2022.

[9] T. Fava, "Design of a visual perception system on an embedded platform for formula student driverless racing vehicles." https://tesi.univpm.it/handle/20.500.12075/20903. Accessed: 2026-04-26.

[10] H. Gong, Y. Feng, T. Chen, Z. Li, and Y. Li, "Fast and accurate: The perception system of a formula student driverless car," in *2022 6th International Conference on Robotics, Control and Automation (ICRCA)*, pp. 45–49, 2022.

[11] A. Dhall, D. Dai, and L. V. Gool, "Real-time 3D traffic cone detection for autonomous driving," *CoRR*, vol. abs/1902.02394, 2019.

[12] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[13] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-time end-to-end object detection," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, 2024.

[14] D. Dodel, M. Schötz, and N. Vödisch, "FSOCO: the formula student objects in context dataset," *CoRR*, vol. abs/2012.07139, 2020.

[15] "The FSOCO dataset." https://fsoco.github.io/fsoco-dataset/. Accessed: 2026-04-26.

[16] "NVIDIA - the ultimate platform for physical ai and robotics." https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/. Accessed: 2026-04-26.

[17] "IDS imaging." https://en.ids-imaging.com/store/ui-3160cp-rev-2-1.html. Accessed: 2026-04-26.

[18] "Ultralytics." https://www.ultralytics.com/. Accessed: 2026-04-26.

[19] "Ultralytics github." https://github.com/ultralytics/ultralytics. Accessed: 2026-04-26.

[20] "NVIDIA TensorRT." https://developer.nvidia.com/tensorrt. Accessed: 2026-04-26.

[21] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *CoRR*, vol. abs/1506.02626, 2015.

[22] NVIDIA, "NVIDIA Jetson Orin Nano developer kit user guide." https://developer.nvidia.com/embedded/learn/jetson-orin-nano-devkit-user-guide/index.html. Accessed: 2026-04-26.

[23] NVIDIA, "NVIDIA Jetson platform power and performance." https://docs.nvidia.com/jetson/archives/r36.4.3/DeveloperGuide/SD/PlatformPowerAndPerformance.html. Accessed: 2026-04-26.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 740–755, Springer International Publishing, 2014.

[25] "NVIDIA jetson-stats." https://github.com/rbonghi/jetson_stats. Accessed: 2026-04-26.