Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 32, Num. 1 (2025) 136-142

RESEARCH ARTICLE

Deep Learning-Based Instance Segmentation for Enhanced Navigation of Agricultural Vehicles

Segmentação de Instâncias Baseada em Aprendizado Profundo para uma melhor Navegação em Veículos Agrícolas

Renato de Avila Lopes[1]*, Marcus Vinícius Leal de Carvalho[2], Edson Kitani[3], Francisco de Assis Zampirolli[1], Leopoldo Yoshioka[2], Luiz Antonio Celiberto Junior[1], Ugo Ibusuki[1]

Abstract: This paper presents the development of a computer vision application based on the YOLOv8 network, designed to assist the navigation of autonomous vehicles on rural roads, particularly those found in sugarcane fields. The application employs instance segmentation to differentiate between navigable and non-navigable areas and detect obstacles such as pedestrians, vehicles, and other potential hazards. This information is used to generate an occupancy map that helps the navigation planner identify the safest and most efficient routes. The system was trained on a dataset containing 1.018 images, and the results demonstrate that instance segmentation significantly enhances the precision and safety of autonomous navigation in complex rural environments. The proposed approach is compatible with the ROS2 framework, using its structure for data integration and enabling real-time decision making.
Keywords: Computer Vision — YOLOv8 — Instance Segmentation — Autonomous Vehicles — Rural Roads — Autonomous Navigation — Occupation Map — Obstacle Detection — Precision Agriculture — Sugarcane Crops

Resumo: Este artigo apresenta o desenvolvimento de uma aplicação de visão computacional baseada na rede YOLOv8, projetada para auxiliar a navegação de veículos autônomos em estradas rurais, particularmente aquelas encontradas em campos de cana-de-açúcar. A aplicação utiliza segmentação por instâncias para diferenciar entre áreas navegáveis e não navegáveis e detectar obstáculos como pedestres, veículos e outros perigos potenciais. Essas informações são utilizadas para gerar um mapa de ocupação que auxilia o planejador de navegação na identificação de rotas mais seguras e eficientes. O sistema foi treinado em um conjunto de dados contendo 1.018 imagens, e os resultados demonstram que a segmentação por instâncias melhora significativamente a precisão e a segurança da navegação autônoma em ambientes rurais complexos. A abordagem proposta é compatível com a estrutura *ROS2*, utilizando sua estrutura para a integração de dados e permitindo a tomada de decisões em tempo real.
Palavras-Chave: Visão Computacional — YOLOv8 — Segmentação por Instâncias — Veículos Autônomos — Estradas Rurais — Navegação Autônoma — Mapa de Ocupação — Detecção de Obstáculos — Agricultura de Precisão — Plantações de Cana-de-açúcar

[1] *Universidade Federal do ABC (UFABC), Brasil*
[2] *Universidade de São Paulo (USP), Brasil*
[3] *Faculdade de Tecnologia do Estado de São Paulo (FATEC), Santo André , Brasil*
***Corresponding author**: renato.avila@ufabc.edu.br
DOI: http://dx.doi.org/10.22456/2175-2745.143329 • **Received:** 16/10/2024 • **Accepted:** 12/12/2024

1. Introduction

Autonomous navigation in rural environments presents major challenges due to factors such as terrain variability and the presence of dynamic and static obstacles [1]. In particular, rural roads in sugarcane fields are often narrow, uneven, and surrounded by dense vegetation, requiring an accurate and adaptive navigation system.

As described by Borges et al. [1], several sensors can be employed in autonomous navigation combined with complex sensor fusion systems. Video cameras, LiDAR (Light Detection and Ranging) and radars are some of these sensors. Computer vision applied to video images, for example, stands out among these sensors due to its high level of detail in the data, low cost, and practicality [2].

To aid in this process, this article describes the development of a computer vision application based on the YOLOv8 [3], designed for instance segmentation, to enhance the safety and navigation of autonomous vehicles in these environments.

Following this introduction, the subsequent sections of this work are as follows: Section 2, which introduces the theoretical foundations; Section 3, which discusses related works; Section 4, which presents the method; Section 5, which presents results; and Section 6, which provides discussions. Finally, Section 7 presents conclusions and future work.

2. Theoretical Foundation

The theoretical foundation of this article is based on fundamental concepts of computer vision, convolutional neural networks (CNNs), image segmentation, and autonomous navigation. To contextualize the development of the proposed application, it is essential to understand the evolution and application of these technologies.

2.1 Computer Vision

Computer vision is a branch of artificial intelligence that enables machines to interpret and understand visual content in a human-like manner [4]. Essential for autonomous navigation, it enables real-time perception and interaction with the environment. Image processing techniques such as edge detection, object segmentation, and pattern recognition are widely applied to extract information from images captured by vehicle sensors.

2.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of neural network architecture inspired by the human visual cortex structure [5]. CNNs are particularly effective for computer vision tasks, such as object recognition and segmentation, due to their ability to extract spatial hierarchies in images. The YOLOv8 model, which serves as the primary network architecture used in this work, is an evolution of traditional CNNs, optimized for real-time object detection and segmentation [3].

2.3 Instance Segmentation

Instance segmentation is an advanced technique that not only classifies the pixels of an image, but also distinguishes between individual instances of objects within the same class [6]. Unlike classical semantic segmentation [6], which groups all pixels of a class into a single category, instance segmentation identifies and separates individual objects. For example, in semantic segmentation, all cars in an image would receive the same label “car”. However, in instance segmentation, each car would have a distinct label “car$_i$”. This instance segmentation technique is particularly useful in autonomous navigation applications, where it is crucial to differentiate between multiple dynamic and static obstacles and safe navigation paths.

YOLO (You Only Look Once) is a family of neural networks designed for real-time object detection. The YOLOv8 network, one of the latest versions, introduces significant improvements in terms of accuracy and speed, making it ideal for real-time instance segmentation applications [3]. YOLOv8 can identify and delineate objects in an image using polygons, making it well-suited for this application.

2.4 Occupancy Map and Trajectory Planning

The occupancy grid map is a crucial data structure in robotics and autonomous vehicle navigation that represents the environment by dividing it into a grid of cells, each cell indicating whether it is occupied, free, or unknown [7]. This representation helps in identifying obstacles and free space, which is fundamental for trajectory planning. By processing segmented images to generate masks and then converting these masks into a cost map, the system creates a dynamic occupancy grid that reflects the current state of the environment.

Segmentation techniques that delineate obstacles with polygons, as opposed to simpler bounding boxes, not only provide a more precise classification of navigable paths but also reduce spatial requirements for map storage. This approach optimizes memory usage and enhances the efficiency of path planning calculations, as the defined navigable space is more accurately represented, allowing the navigation system to compute paths within a smaller, more relevant area of the map [8].

The cost map assigns cost to different regions based on the likelihood of encountering obstacles, allowing the navigation planner to calculate the safest and most efficient paths, avoid collisions, and optimize the vehicle’s trajectory through the environment [8].

2.5 ROS2

The ROS2 (Robot Operating System 2)[9] is an open-source framework widely used for the development of robotic systems [10]. It provides infrastructure for communication between components of an autonomous system, enabling integration of sensors, control algorithms, and user interfaces.

In the context of this study, ROS2 integrated the YOLOv8-based vision system into the vehicle, enabling real-time data exchange between the segmentation module and navigation planner for improved decision-making. Its distributed architecture ensures scalability and robustness in complex environments like rural roads between sugarcane fields.

3. Related works

Related approaches are found in the work of Huang *et al.* [11], which proposes an adaptation of Poly-YOLO [12] to enhance accuracy and efficiency in semantic segmentation for autonomous vehicles. This method generates polygons around objects to reduce computational load while maintaining high detection accuracy across various target sizes. Additionally, Ojha *et al.* [13] implement Mask R-CNN with transfer learning for vehicle detection via instance segmentation, producing bounding boxes and object masks.

These works differ from the one presented in this article as they do not provide the codes and datasets used, and primarily, they do not address the context of autonomous applications in vehicles on rural roads.

The DR-YOLO algorithm, optimized by Wu *et al.* [14], improves operations in mountainous tea plantations with its lightweight design and real-time performance for agricultural machinery. Future work aims to integrate DR-YOLO with the ROS system for intelligent operations in lychee orchards, enabling obstacle avoidance for harvesting robots. While similar to our work using the YOLOv8-seg model, its dataset and trained model were not reproducible.

4. Methods

The autonomous navigation method proposed in this work uses the specific version of the YOLOv8 architecture that performs instance segmentation. This technique, in contrast to traditional semantic segmentation, enables precise identification and delineation of individual objects using polygons. This more detailed representation of the environment allows for a deeper understanding of scenes and facilitates decision-making in autonomous navigation tasks. The practical implementation can be found in the GitHub repository [15], specifically in the Colab file `segmentation.ipynb`. See also the file `Method.png` for a general overview of the method.

To train the YOLOv8 network, a dataset was created with 1.018 images captured on rural roads across sugarcane fields. The Roboflow website [16] was used to generate the dataset required for training, and the generated image dataset is available in this GitHub repository [15].

The images used to build the dataset were extracted from videos obtained in Iracemápolis, in the interior of São Paulo, Brazil. The videos were recorded in sugarcane fields and urban areas of the city. For data collection, a FLIR BlackFly GigE monocular camera was installed in the cabin of a Mercedes Arocs truck, see Figure 1, recording its movement through urban roads and rural paths.

Figure 1. Highlighting the camera installed in the cabin of a Mercedes Arocs truck.

The selected camera model, BFLY-PGE-12A2C-CS, offers a combination of high-quality image resolution (1280×960) and reliable connectivity through the GigE interface.

The images were labeled with three main classes: navigable road, vegetation, and rural road, along with obstacle classes such as people, cars, trucks, and motorcycles. The network was trained to segment these classes and generate polygons for each region in the image. The training process is detailed in this GitHub repository [15], specifically in the Colab file `segmentation_train.ipynb`.

In this Colab notebook, after organizing the training directories and executing the required imports, the following command is used to initiate the training process:

```
!yolo task=segment mode=train model=yolov8n-seg.pt\
      data=/content/segmentacao/dataset/data.yaml \
      epochs=320 imgsz=640 plots=True
```

This command uses the YOLO tool to train image segmentation models. The main command, `!yolo`, is executed in Colab and invokes the YOLO tool via the terminal. The parameter `task=segment` defines the task of segmentation, which involves identifying and labeling regions of the image. The operation mode is defined by `mode=train`, specifying that the command is for training the model. The pre-trained model file is indicated by `model=yolov8n-seg.pt`, and the path to the data configuration file (`data.yaml`) is provided. The training was conducted for 320 epochs, as specified in `epochs=320`, with the size of the input images set to 640×640 pixels by `imgsz=640`.

The generation of plots during training is enabled by `plots=True`, allowing the visualization of metrics, which will be summarized in the next section. The training was concluded with 211 epochs and stopped as no further improvements were observed for 50 consecutive epochs. The complete process took 220 minutes using the Colab T4 GPU resource. This command uses the default behavior to split the dataset into 80% for training, 10% for validation, and 10% for testing.

The parameter `yolov8n-seg.pt` refers to a pre-trained deep learning model based on the YOLOv8 architecture, optimized for image segmentation tasks. The designation `yolov8n-seg.pt` denotes a more compact and efficient variant of the YOLOv8 architecture, making it ideal for applications with limited computational resources in embedded systems [17]. The suffix `-seg` indicates that the model was specifically trained for segmenting images, which assigns a class to each pixel and thereby delineates objects of interest[1].

The `.pt` extension indicates that the model is saved in the PyTorch format, a popular and versatile deep learning framework [18].

In practical applications, real-time images are captured using the OpenCV library [19], and the YOLOv8 model performs inference to identify and segment the objects present. The generated data is transmitted in real-time to the autonomous

[1]See other variants of the YOLOv8 variants optimized for image segmentation: docs.ultralytics.com/tasks/segment.

vehicle navigation system, which differentiates navigable areas, classified as “road”, from non-navigable areas, classified as “vegetation”, in addition to detecting obstacles. The segmentation process outputs the detected classes along with their corresponding polygons, which outline the area of each object, as shown in Figure 2.

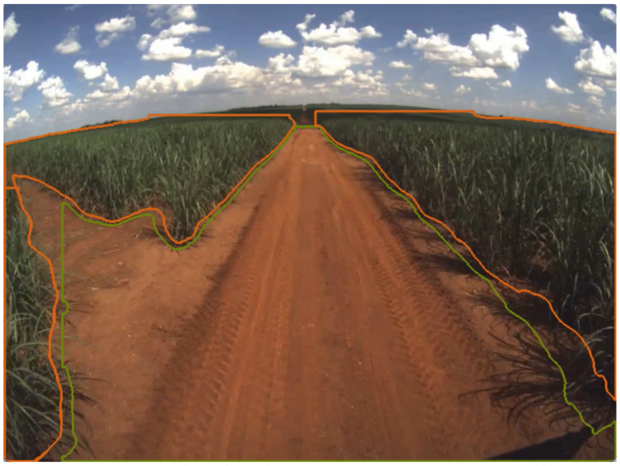

Figure 2. Image highlighting the labelled parts: navigable road (green polygon) and sugarcane plantation (orange polygon).

The contours shown in Figure 2 are scaled according to their dimensions and overlaid on the image frame, with specific colors assigned to each class. To reduce noise and concentrate on areas of interest, the application ignores the upper part of the image above the horizon line. The remaining portion, which contains relevant objects, is used for mask creation. A mask distinguishes navigable areas (roads) from non-navigable ones (vegetation, obstacles) by assigning a weight or cost to each region, guiding the planner to favor safe paths. This cost-based method helps avoid dangerous or high-cost areas, optimizing navigation for safety and efficiency.

An affine transformation aligns the mask with the forward path, creating a binary occupancy map. This map, where each pixel represents a cell, forms the basis for generating 3D occupancy maps and cost maps, critical for navigation planning. The goal is to send messages to the navigation planner to identify roads and obstacles, allowing it to verify if the current route aligns with the navigable area. Figure 3 presents the original segmented image and the mask created in the process.

The implementation of the mask can be found in the GitHub [15], in the file `segmentation_mask.ipynb`.

The final step in the processing is the creation of the occupancy map (OccupancyGrid), which represents the spatial distribution of the detected objects in the environment. Figure 4 shows a 3D occupancy map on the right, created using ROS2 and visualized with the ROS visualization tool, RVIZ [20].

After adjusting the mask, it is converted into an occupancy grid, where occupied cells (pixels in a polygon) are assigned the value 100 (depicted in pink), and free cells (pixels outside a polygon) receive the value 0 (depicted in gray). This grid

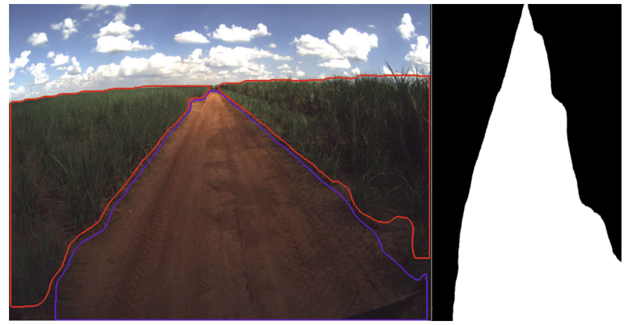

Figure 3. Mask image where the navigable region is shown in white.

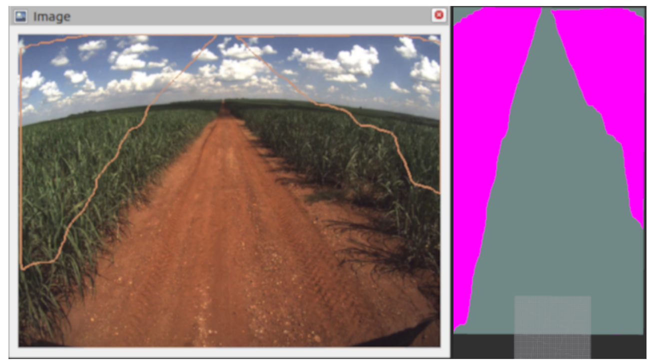

Figure 4. Occupancy map displayed on the right, generated within the ROS2 and visualized using the RVIZ. The map highlights navigable and non-navigable areas, aiding the autonomous vehicle’s navigation planning.

can be integrated into an `OccupancyGrid` message and published on a ROS2 topic, making it accessible to other components of the autonomous vehicle’s navigation system.

5. Results

During the training of neural networks for object detection and segmentation tasks, such as the YOLOv8 network, various loss metrics and performance measures are calculated to evaluate and improve the model. In this section, we present the results of training and evaluation of the YOLOv8 model, specific for instance segmentation, applied to autonomous navigation in rural environments.

Performance metrics include Average Precision (AP), Mean Average Precision (mAP), Precision, Recall, and F1-score. These metrics are commonly used to evaluate the effectiveness of object detection and segmentation models [21, 22]. The mAP provides a measure of overall performance by averaging APs across all classes, delivering a comprehensive assessment of model effectiveness. These metrics are relevant in the context of autonomous navigation, where precision and recall are important for accurate object detection and path planning [23].

The “road” (AP = 85.1%), “vegetation” (AP = 91.3%), and “rural road” (AP = 98.8%) classes demonstrated high average precision (AP) values, indicating that the model was able to accurately identify and segment these classes consistently across tests. The obstacle classes, including “car”, “truck”, and "motorcycle", exhibited satisfactory performance, albeit with greater variability in AP values. This variability reflects the complexity of segmenting smaller-scale objects or those that appear less frequently in images. Notably, the "person" class achieved an AP of 99.5%, which stands out due to its high representation in the original YOLOv8 model [3].

Mean Average Precision (mAP) is the average of the AP values across all classes. The recorded mAP was high (88.2%), indicating that the model consistently performed well in segmenting the various classes. This outcome underscores the effectiveness of the training process, which successfully generalized across multiple categories of objects present in the scenes. The Average Precision (AP) was calculated for each class individually, reflecting the area under the Precision-Recall curve for different confidence thresholds, as shown in Figure 5.

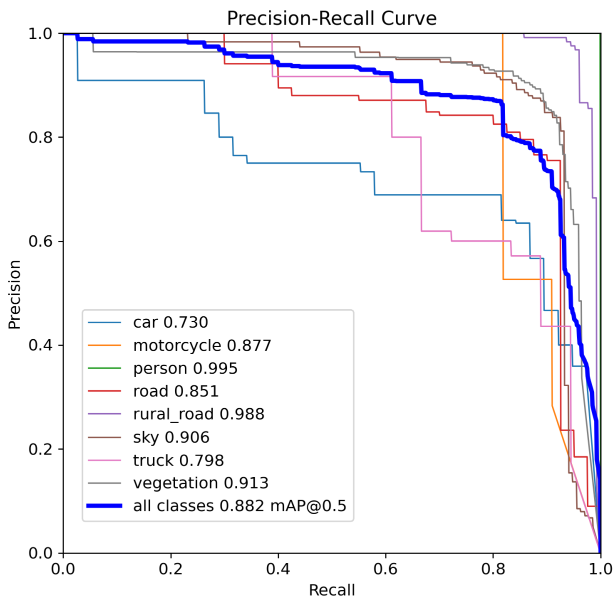

Figure 5. Precision-Recall curve for all classes.

The F1 Score, defined as the harmonic mean between Precision and Recall, was utilized to provide a balanced metric that accounts for both the ability to avoid false positives and minimize false negatives. Figure 6 illustrates a satisfactory balance between these two metrics, reaffirming the model's effectiveness in accurately segmenting the various object classes while maintaining a good balance between Precision and Recall.

The confusion matrix is essential for evaluating classification model performance by comparing predictions to actual classes. It shows the number of correct and incorrect classifications for each category, allowing for the calculation of key metrics like accuracy, precision, and recall. Accuracy indicates the overall correct classification rate, while accuracy and recall measure the model's ability to identify specific class instances. This analysis helps identify well-classified classes and those prone to errors, guiding tuning to improve model performance.

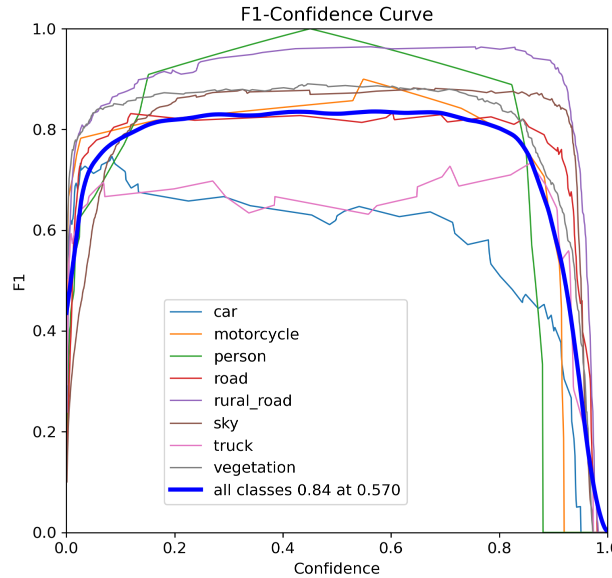

Figure 6. Precision-Recall curve for all classes.

Figure 7 presents the confusion matrix related to model validation, providing information about classification performance in different classes.

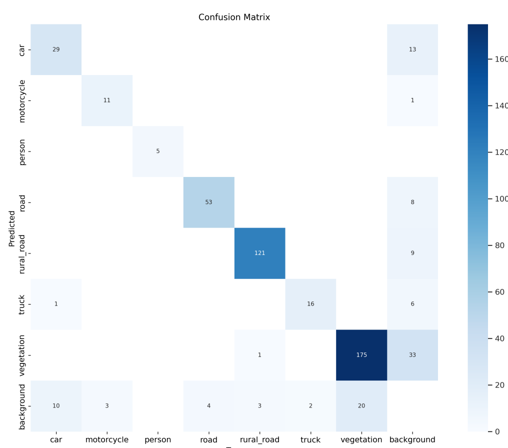

Figure 7. Confusion matrix related to the model validation.

Analysis of the confusion matrix shows that the class "person" was correctly identified in all instances, with no misclassifications. This highlights the effectiveness of the model in detecting this category. The classes "rural road" and "vegetation", with a high number of instances, also demonstrated high accuracy, with few misclassifications. This demonstrates the robustness of the model in accurately distinguishing these classes, even in complex scenarios where visual differences may be subtle.

These results confirm the model's ability to accurately classify objects, reinforcing its suitability for autonomous navigation tasks in complex rural settings. Figures 8 and 9 illustrate the convergence of metrics during training. Figure 8 presents the training and validation loss curves, indicating the convergence behaviour of the model, while Figure 9 shows the curves for Precision, Recall, and Mean Average Precision (mAP) over the training epochs.

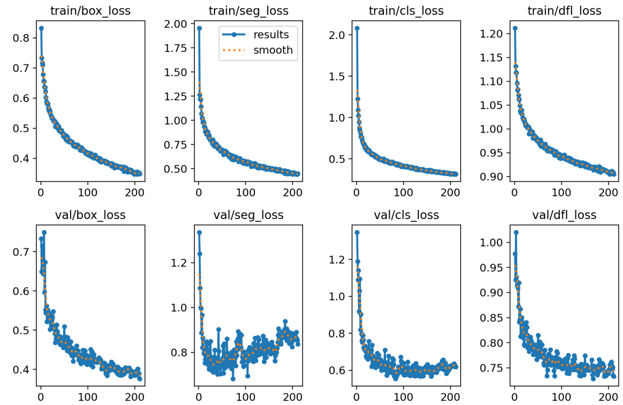

Figure 8. Metrics over epochs during training and validation.

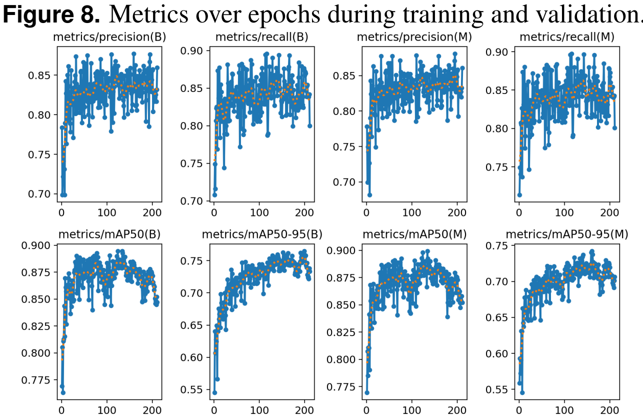

Figure 9. Precision, Recall, and mean Average Precision (mAP).

The consistent performance across classes, evidenced by the high AP, mAP, precision, recall, and F1-score metrics, along with the analysis of the confusion matrix, supports the effectiveness of both the training process and the architecture of the YOLOv8 model for instance segmentation in sugarcane field environments.

6. Discussion

The use of instance segmentation, as opposed to semantic segmentation, proved advantageous for autonomous navigation on rural roads. Representing objects through polygons enables more precise and adaptive detection of various elements, thereby enhancing vehicle safety during navigation. Furthermore, the detection of obstacles such as pedestrians and other vehicles improves the system's ability to avoid collisions and make real-time decisions.

This approach is efficient because the area of interest for navigation, located directly in front of the vehicle, is updated in real-time, resulting in minimal information loss compared to a map generated from a higher vantage point. This real-time updating allows the vehicle to dynamically adjust its navigation in response to changes in road direction and the presence of obstacles.

Furthermore, the use of instance segmentation with polygons, rather than bounding boxes, provides a more precise delineation of navigable spaces in complex rural environments. This level of detail is particularly beneficial in scenarios where paths are irregular and closely surrounded by vegetation. This precision improves the system ability to make safer and more adaptive navigation decisions, offering clear advantages over simpler segmentation methods, especially in challenging rural terrains. Figure 10 illustrates the reduction of the navigable area in the presence of an obstacle (a vehicle).

Figure 10. Segmentation with the presence of an obstacle (a vehicle) in the navigable area.

Threats to Validity

Our study has some notable threats to validity. The dataset, consisting of 1,018 images, may be insufficient to fully capture the complexity and variability of rural environments. Diverse conditions, such as variations in lighting, weather, and obstacle types, are not comprehensively represented, which may limit the generalizability of the proposed approach.

The classes included as obstacles only illustrate the possibility of including any other necessary class, and do not cover all possible scenarios.

Video analysis alone may not provide sufficient accuracy to estimate the distance of objects ahead of the vehicle. The integration of additional sensors, such as LiDAR and radar, is necessary to enhance the system's reliability and precision, particularly in scenarios where precise depth estimation is critical for safe navigation.

7. Conclusion

This article discusses a computer vision application for autonomous vehicle navigation on rural roads, utilizing the YOLOv8 network for instance segmentation. The results show that the system effectively identifies navigable and non-navigable areas and detects obstacles, ensuring safe navigation. Additionally, it generates an occupancy map, useful for creating a cost map for navigation. The application is fully compatible with the ROS2 framework, leveraging its resources and infrastructure.

Future research could integrate this system with additional sensors like LiDAR and radar, along with navigation techniques such as Visual SLAM, to improve autonomous navigation in complex rural environments. Adapting the model to diverse conditions, including dense vegetation and adverse weather, is another crucial avenue to explore.

Expanding the dataset is essential for enhancing model generalization. As suggested by Haimer et al. [22], adding images of potholes and diverse objects like pedestrians, animals, cars, and motorcycles will help create a more robust model for various scenarios.

Acknowledgements

The authors would like to thank the entire (Fundep Rota 2030/Linha V, N. 27192.02.01/2021.02.00) project team for their support and collaboration.

Public Data

The data from this study are freely available for reuse, provided the original research is properly cited to ensure author recognition and academic integrity. All materials produced in this article are available in the GitHub repository [15].

Author contributions

Renato de Avila Lopes: Conceptualization, Investigation, Methodology, Resources, Validation, Visualization, Writing - original draft, Writing - review & editing; Marcus Vinicius Leal de Carvalho: Investigation, Methodology, Validation, Visualization, Writing - original draft, Writing - review & editing; Edson Kitani, Francisco de Assis Zampirolli, Leopoldo Yoshioka, Luiz Antonio Celiberto Junior, and Ugo Ibusuki: Investigation, Validation, Writing - original draft, writing - review & editing.

References

[1] BORGES, P. V. et al. A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges. *Field Rob.*, v. 2, n. 1, p. 1567–1627, 2022.

[2] YASUDA, Y. D.; MARTINS, L. E. G.; CAPPABIANCO, F. A. Autonomous visual navigation for mobile robots: A systematic literature review. *ACM Comp. Surveys (CSUR)*, ACM New York, NY, USA, v. 53, n. 1, p. 1–34, 2020.

[3] JOCHER, G.; CHAURASIA, A.; QIU, J. *Ultralytics YOLOv8*. 2023. Disponível em: ⟨https://github.com/ultralytics/ultralytics⟩.

[4] SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Nature, 2022.

[5] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. ⟨http://www.deeplearningbook.org⟩.

[6] HAFIZ, A. M.; BHAT, G. M. A survey on instance segmentation: state of the art. *Int. Jour. of Mult. Inf. Retr.*, Springer, v. 9, n. 3, p. 171–189, 2020.

[7] RACINSKIS, P.; ARENTS, J.; GREITANS, M. Constructing maps for autonomous robotics: An introductory conceptual overview. *Electronics*, MDPI, v. 12, n. 13, p. 2925, 2023.

[8] CRISTÓFORIS, P. D. et al. Hybrid vision-based navigation for mobile robots in mixed indoor/outdoor environments. *P. R. Letters*, Elsevier, v. 53, p. 118–128, 2015.

[9] FOUNDATION, O. S. R. *Robot Operating System 2 (ROS2)*. 2024. ⟨https://docs.ros.org/en/foxy/⟩. Acces.: 2024-08-20.

[10] MACENSKI, S. et al. From the desks of ros maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2. *Robotics and Autonomous Systems*, Elsevier, v. 168, p. 104493, 2023.

[11] HUANG, Z. Semantic road segmentation based on adapted poly-yolo. In: *J. of Phy.: Conf. Series*. [S.l.: s.n.], 2023. v. 2580, n. 1, p. 012015.

[12] HURTIK, P. et al. Poly-yolo: higher speed, more precise detection and instance segmentation for yolov3. *Neural Comp. and Appl.*, Springer, v. 34, n. 10, p. 8275–8290, 2022.

[13] OJHA, A.; SAHU, S. P.; DEWANGAN, D. K. Vehicle detection through instance segmentation using mask r-cnn for intelligent vehicle system. In: IEEE. *Int. conf. on Intel. Comp. and Control Systems (ICICCS)*. [S.l.], 2021. p. 954–959.

[14] WU, W. et al. Instance segmentation of tea garden roads based on an improved yolov8n-seg model. *Agricul.*, MDPI AG, v. 14, n. 7, p. 1163, 2024.

[15] LOPES, R. de A. *Segmentation Article*. 2024. ⟨https://github.com/natoavilalopes/segmentationArticle⟩. Acces.: 2024-10-09.

[16] ROBOFLOW. *Roboflow*. 2024. ⟨https://roboflow.com⟩. Acces.: 2024-10-09.

[17] AVILA, R. et al. Comparisons of neural networks using computer vision for agricultural automation. In: *2023 15th IEEE International Conference on Industry Applications (INDUSCON)*. [S.l.: s.n.], 2023. p. 466–470.

[18] KETKAR, N. et al. Introduction to pytorch. *Deep learning with python: learn best practices of deep learning models with PyTorch*, Springer, p. 27–91, 2021.

[19] TEAM, O. *OpenCV: Open Source Computer Vision Library*. 2023. ⟨https://opencv.org⟩. Acces.: 2024-08-20.

[20] DEVELOPERS, R. *RViz*. 2024. ROS Visualization Tool. ⟨http://wiki.ros.org/rviz⟩.

[21] YUE, X. et al. Improved yolov8-seg network for instance segmentation of healthy and diseased tomato plants in the growth stage. *Agriculture*, MDPI, v. 13, n. 8, p. 1643, 2023.

[22] HAIMER, Z. et al. Pothole detection: A performance comparison between yolov7 and yolov8. In: *2023 9th International Conference on Optimization and Applications (ICOA)*. [S.l.: s.n.], 2023. p. 1–7.

[23] PADILLA, R.; NETTO, S. L.; SILVA, E. A. D. A survey on performance metrics for object-detection algorithms. In: IEEE. *2020 Int. Conf. on Systems, Signals and Image Processing (IWSSIP)*. [S.l.], 2020. p. 237–242.