Computer Vision Model Applied to the Autonomous Husky Robot

Juan Caio Paronitti Galera*, Leila Cristina Carneiro Bergamasco* and Danilo Hernani Perico†
*Computer Science
Centro Universitário FEI, São Bernardo do Campo, São Paulo, Brazil
✉: juancaio.galera@gmail.com
†Robotics Engineering
Centro Universitário FEI, São Bernardo do Campo, São Paulo, Brazil

***Abstract*—The utilization of autonomous off-road vehicles has garnered increasing significance, particularly in unstructured terrains like agricultural fields and mines. Navigating such environments presents unique challenges due to the absence of conventional signage and the rugged nature of the terrain. This study focuses on the development and enhancement of computer vision model for improving navigation capabilities of the Husky Robot. YOLO model has been used to detect obstacles, particularly cones, facilitating informed decision-making regarding navigation routes. The results allow us to conclude that YOLO can be successfully used to detect cones in off-road environments.**

I. Introduction

The study of autonomous off-road vehicles is becoming increasingly important as these vehicles can be used in unstructured environments such as agricultural fields and mines for various activities.

The development of autonomous off-road vehicles poses several challenges, as the environments they operate in are not as controlled as urban streets and roads. Typically, these environments lack signage such as signs, lanes, and traffic lights, making navigation more complex.

The Husky Robot[1] (Fig. 1) is capable of performing tasks in environments that pose certain challenges for mobility. Given its ability to operate in these regions, a computer vision model has been developed to detect cones, aiming to indicate the routes where it can move.

This computer vision system enhances the Husky's navigation capabilities by identifying obstacles such as cones, allowing it to make informed decisions about its route and maneuver safely through challenging environments.

The Husky vehicle has been meticulously crafted for seamless navigation through agricultural landscapes. In this dynamic setting, our paramount focus lies in enabling autonomous navigation, ensuring adept traversal across rugged terrains such as sprawling farmlands and rustic rural areas. To achieve this, we have engineered a cutting-edge computer vision-based detection system, a technological marvel that furnishes the vehicle with a calibrated level of autonomy. This innovative system empowers the Husky to traverse even the most challenging regions with unparalleled ease and efficiency.

[1]https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

Fig. 1. Clearpath Husky UGV



Adhering steadfastly to the project's guiding principles, our foremost priority has been to enhance the vehicle's perceptual acuity and decision-making prowess within its operational environment. To this end, we have meticulously structured the following pivotal elements:

Vision Sensors: Equipped with an array of strategically positioned cameras, the Husky captures comprehensive visual data from myriad angles, providing a panoramic view of its surroundings.

Obstacle Detection: Employing sophisticated computer vision algorithms, our system adeptly identifies obstacles within the vehicle's path. Harnessing the potential of convolutional neural networks, it discerns and navigates through potential impediments with remarkable precision.

Decision-Making: Informed by the rich trove of data furnished by our computer vision system, the Husky seamlessly orchestrates real-time decision-making processes. Leveraging this invaluable insight, it adeptly charts its course through the terrain, reacting swiftly and decisively to any detected obstacles.

Through these concerted efforts, we endeavor to augment the Husky's navigational capabilities, charting new frontiers in autonomous vehicle technology. With an unwavering com-

mitment to innovation and excellence, we envision a future where the Husky autonomously navigates through the most challenging environments with unparalleled efficiency and reliability.

II. Related works

To develop this review, the Semantic Scholar and Google Scholar websites were utilized, where the following phrases were used as search keywords: ”cone detection,” ”computer vision in an outdoor environment,” ”computer vision,” and ”computer vision in off-road vehicles.” The searches returned several occurrences, and filters were applied to focus on traffic cone detection and developing Convolutional Neural Networks (CNNs) for this purpose.

In 2022, the author [1] and colleagues presented an article on the importance of algorithms responsible for cone detection, considering that these traffic signals do not have a fixed location. The program's development utilized a deep learning technique based on images of cones on roads to implement the algorithm, thereby creating an RGB dataset. A neural network model was proposed that classifies obstacles on the road using a learning model that simulates driving behavior. For the model's implementation, traffic cones were placed on the road to capture obstacle evasion scenes. Specifically, an RGB dataset was used to train the object detector. The model's structure uses Python and a graphical interface to store information about bounding boxes, with their dimensions represented by real numbers on a scale, indicating the relative location and size of the cone with the entire image.

This research highlights the need for robust algorithms to detect traffic cones, considering their importance in road safety, especially in dynamic and challenging environments such as roads and off-road settings. Advanced deep learning techniques like CNNs hold promise for addressing these challenges, providing a system capable of recognizing and responding to obstacles effectively. The use of diverse datasets and the adoption of suitable programming languages and graphical interfaces are essential for successfully developing and implementing these traffic cone detection systems.

In 2019, the author [2] and colleagues presented a vehicle developed at WPI, a level 3 autonomous model capable of executing all tasks necessary for driving, even though occasional intervention may still be required. Its development occurred in stages, beginning with tests of its remote control capability. Field tests provided various sensor data for developing decision-making software, even in the early stages. Despite significant progress, the responsible team observed that although the vehicle could simultaneously locate and map the unknown environment, it could not plan a path concurrently with speed control. The simultaneous construction and localization of maps played a significant role in the project and, in turn, proved to be one of the most challenging aspects to address. Moving to areas without previously acquired data sometimes complicates program execution. Precision movement necessitated a detailed map of the environment, and constructing an accurate map required the vehicle to move for object detection. However, the team mitigated the challenge of executing actions in parallel by manually mapping the environment, enabling the vehicle to plan a route using the pre-developed mapping.

In 2018, [3] presented a paper addressing image processing techniques for autonomous vehicles, which have their peculiarities in application. In the case of external environments, vehicle speed, and size are greater, leading to considerations of specific problems to be resolved: Basic lane detection, Sign detection and identification, Prediction of steering angle, and Proximity detection.

For the first issue, late detection was addressed to identify the lane in which the vehicle is located. Techniques involving smoothing followed by edge detection were employed for execution.

For the second issue, a code dataset was utilized, posing a relatively simple problem as an open-source code was used, requiring only minor adjustments.

The third issue tackled predicting the steering angle, employing a Convolutional Neural Network (CNN), a deep learning algorithm. However, problems arose during training, such as the emergence of sharp angles and an inability to learn adequately, necessitating the readjustment of the function, consequently increasing the amount of acquired data.

A camera mounted on the roof was utilized for the last issue, leveraging computer vision capabilities. Once the decision on which resources to use was made, a Support Vector Machine (SVM), a learning algorithm used to classify a set of points, was applied for vehicle classification.

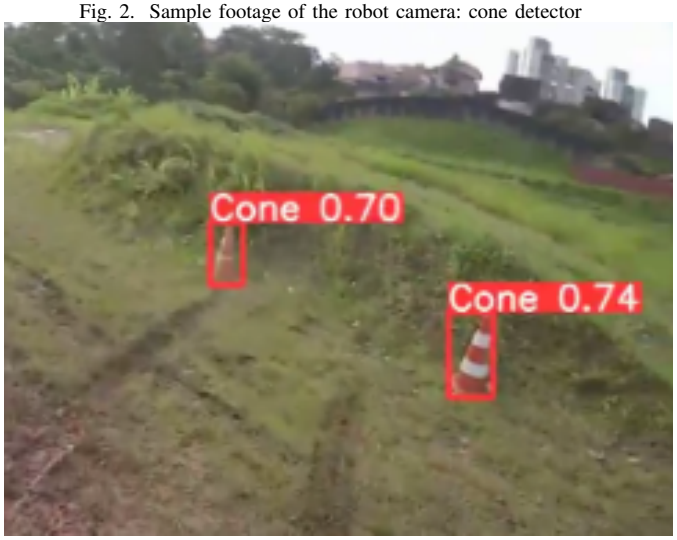
This research sheds light on the intricate challenges encountered in image processing for autonomous vehicles, highlighting the need for adaptive solutions and the integration of various techniques to ensure effective performance in diverse real-world scenarios.

In 2018, [4] authored a paper discussing Convolutional Neural Networks (CNNs), emphasizing their primary concept of filtering lines, curves, and edges to achieve visual recognition. Concerning image classification, it is common to use three-dimensional matrices with height, width, and depth, where convolutions perceive small squares analyzing the entire image to capture features. The output depth equals the number of applied filters, resulting in more detailed features with deeper layers. The filter contains weights updated with each new input, applied to a small input region known as the receptive field. When applied, these generate combinations that, upon finding compatibility, yield a result greater than zero.

The paper discussed the use of padding to prevent layers from quickly diminishing during learning, alongside functions enabling the network to learn various functionalities, necessitating selection based on computational needs. Finally, a pooling layer is applied to simplify information from the preceding layer, typically using max-pooling to store only the maximum value, thereby reducing the number of weights to be learned.

This comprehensive overview provides insights into the fundamental principles of CNNs, elucidating their operations

Fig. 2. Sample footage of the robot camera: cone detector



in image processing and underscoring their role in visual recognition tasks. Additionally, it underscores the importance of design choices, such as padding and pooling, in optimizing network performance and efficiency.

In 2016, the author [5] and collaborators published a paper addressing real-time understanding of off-road environments, which poses a challenging task. To address this, an existing CNN architecture trained for urban scenarios was adopted to develop its capability to classify off-road scenes. Training time for the network needed to be significantly reduced compared to training from scratch. In cases where a specialized dataset is partially labeled, satisfactory results can be achieved. The chosen CNN architecture already contained labeled images of urban scenarios. However, new images captured by a camera mounted on a vehicle were labeled for off-road data. The vehicle was driven through various environments to access various information, with the data divided into 90% for training and the remaining 10% for testing.

As a result, 8 class labels were assigned to the off-road dataset, along with 3 already present labels. The process of labeling pixels began with the available data. However, the team decided not to fully label the images due to the time-consuming nature of the task and the difficulty delineating classes in certain scenes. The program's completion involved training seven versions, each initialized with a specific number of iterations.

This study highlights the adaptation of existing CNN architectures for off-road scene classification, demonstrating the importance of efficient training processes and the challenges involved in data labeling for complex environments. It underscores the iterative nature of model development and the need for practical solutions to address real-world constraints in machine learning applications.

III. Robot vision

To develop computer vision, we trained neural networks using an extensive dataset containing various images of cones in off-road environments [2]. Subsequently, we explored and experimented with various neural network architectures.

To achieve this objective, we focused on the YOLO model for object detection and developed the YOLOv8 and YOLOv4 models, known for delivering promising results regarding MAp. These models were selected after carefully analyzing their capabilities and performance across different scenarios. We aimed to ensure the computer vision model was robust enough to handle the complexities and challenges encountered in off-road environments.

Finally, we used the metric Mean Average Precision (MAp) to evaluate different detection approaches, shown in Equation 1, where $AveP$ is the average precision score for each query q. The term Q represents the amount of total queries performed.

$$MAp = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{1}$$

These development and selection steps were crucial to ensure that our computer vision system could accurately detect cones and provide valuable information for autonomous vehicle navigation in challenging environments. Figure 2 depicts how the robot's camera detects cones.

A. YOLOv4

In developing the YOLOv4 model, we extensively utilized the Darknet neural network framework. To attain a Mean Average Precision (MAP) that guarantees robust performance, the model underwent rigorous training for 2000 epochs. This prolonged training duration was imperative to yield a model with a MAP of 88% (Figure 3).

B. YOLOv8

After the YOLOv4 model development and its application in real-world scenarios involving recorded footage, the decision was made to transition to a more advanced model to enhance both the precision and speed of responses. Consequently, the YOLOv8 model was developed, leveraging a foundation comprising various models for further development. The flexibility to choose from multiple models within the YOLOv8 framework allowed developers to tailor the selection based

[2]https://universe.roboflow.com/search?q=cone%20dataset

Fig. 3. Map YOLOv4

```
 detections_count = 175, unique_truth_count = 327
class_id = 0, name = Cone, ap = 88.06%           (TP = 87, FP = 8)

 for conf_thresh = 0.25, precision = 0.92, recall = 0.27, F1-score = 0.41
 for conf_thresh = 0.25, TP = 87, FP = 8, FN = 240, average IoU = 74.13 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.880571, or 88.06 %
Total Detection Time: 8 Seconds
```

Fig. 4. Data from the YOLOv8 models

Model	Input size (pixels)	mAP 50-95	params (M)	FLOPS (B)
YOLOv8n	640	37.3	3.2	8.7
YOLOv8s	640	44.9	11.2	28.6
YOLOv8m	640	50.2	25.9	78.9
YOLOv8l	640	52.9	43.7	165.2
YOLOv8x	640	53.9	68.2	257.8

Source:
https://docs.ultralytics.com/models/yolov8/supported-modes

on specific requirements and objectives. This transition to YOLOv8 marked a strategic move towards optimizing the performance and efficacy of the computer vision system, ensuring that it could effectively address the evolving challenges and demands of real-world applications. Figure 4 shows features of YOLOv8 model.

C. YOLOv8s

Initially, the YOLOv8s model was developed, which exhibited a low mAP attribute of 50-95. However, it demonstrated good response speed for the task at hand. Considering that the final model would be integrated into a vehicle, response speed was deemed crucial.

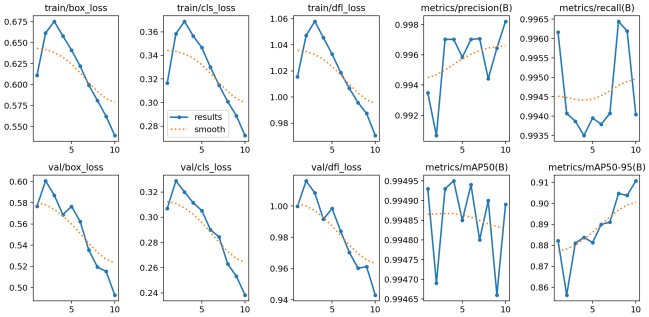
The test results of the developed model are presented in Figure 5. The training progress is shown in Figure 6. The results obtained from the trained model are displayed, with particular attention given to the metrics/mAP50 and metrics/mAP50-95, revealing the precision achieved by the model in its last training epoch. The model achieved 91 precision, indicating it was operating near its detection and accuracy capacity.

However, the final model exhibited extremely low precision, rarely returning valid detection. When applied to the test video used for the YOLOv4 model, it did not produce any valid detection. As a result, it was discarded as the final model for project implementation.

Fig. 5. Results - YOLOv8s



Fig. 6. Training progress for YOLOv8s



D. YOLOv8m

After the previous model failed in practical tests, training commenced on a superior model to observe the results at the end of training. The new model underwent the same number of training epochs and used the same dataset as the base to allow for the comparison of results with the increased parameters of the more powerful model.

The test results of the developed model are presented in Figure 7. The training progress is shown in Figure 8. The results obtained from the trained model are displayed. It can be observed that the model achieved 91 precision in mAP 50-95, indicating that it reached the limit of the model's capacity.

At the end of training, the model presented reasonable results, with a detection rate below that of the YOLOv4 model. During the final practical test, the model efficiently detected objects with an mAP of 60. However, it failed to detect some objects that the older model detected.

E. YOLOv8L

After the failure of the previous model in practical tests with Husky, training a more advanced model was initiated to

Fig. 7. Results - YOLOv8m



Fig. 8. Training progress for YOLOv8m

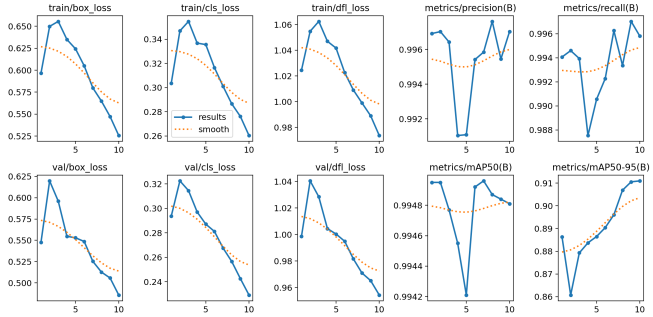
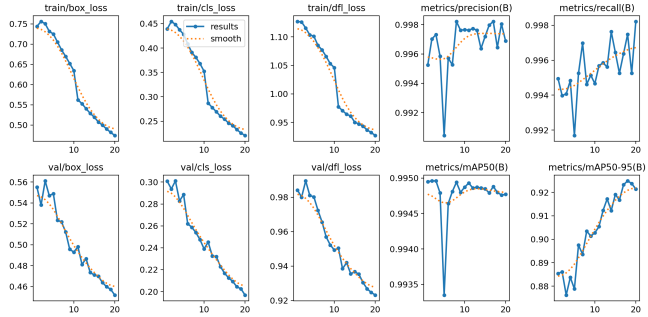


Fig. 10. Training progress for YOLOv8l



observe the results at the end of the process. This new model underwent the same number of training epochs and used the same dataset as a base, allowing the observed results to be related to the increase in parameters that the more powerful model presented.

The test results of the developed model are presented in Figure 9. The training progress is shown in Figure 10. The results extracted from the trained model are displayed, demonstrating a precision of 91 in mAP 50-95, indicating that it reached the limit of the model's capacity.

At the end of the training, the model presented reasonable results, with a detection rate below that of the YOLOv4 model.

F. YOLOv8x

After the YOLOv8m model delivered satisfactory results, more powerful models were developed to assess if they could match the precision of the v4 model. The YOLOv8L models were developed, but they did not yield superior results to the previous model. This lack of improvement did not justify their replacement, especially considering the increased parameters would result in slower detection responses.

Fig. 9. Results - YOLOv8l model



Moving on to the YOLOv8x model, it proved to be the most powerful and updated model available, containing more parameters for detection. Following its training, the detections presented were superior to those of the YOLOv8M model, although they did not surpass the YOLOv4 model.

The test results of the developed model are presented in Figure 11. The training progress is shown in Figure 12. The results from the trained model highlighted that it underwent 100 epochs of training to observe the increase in the model's learning capacity. Ultimately, it achieved a 92 mAP 50 precision, which did not warrant further training despite a significant decrease in training loss data.

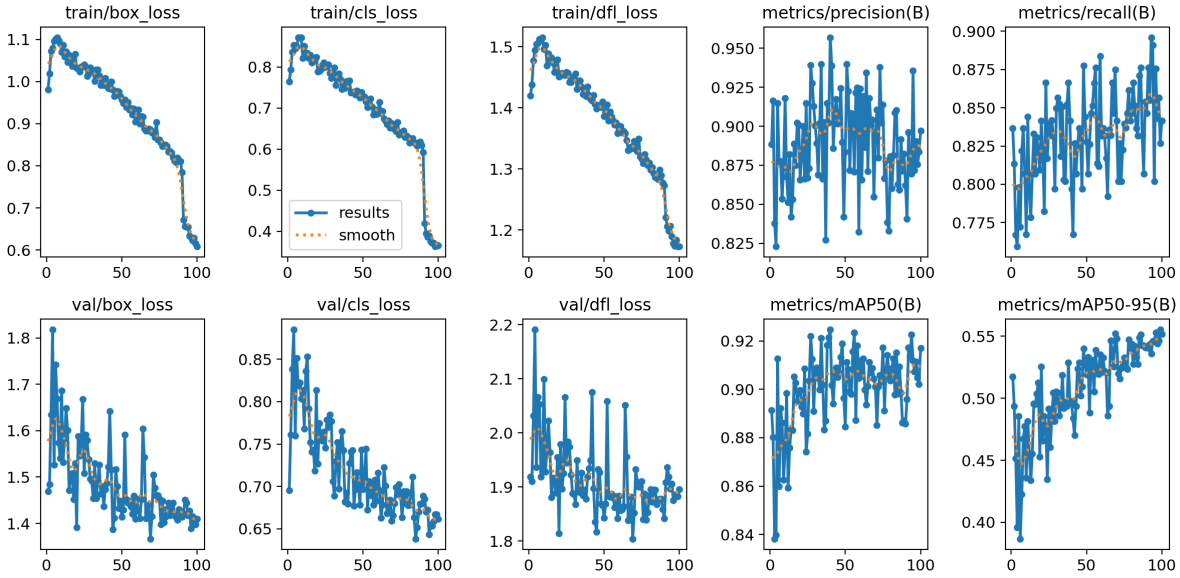
IV. Conclusion

This work presented the development of a computer vision system based on the YOLO model for detecting cones in off-road environments. Our goal is to use this system in

Fig. 11. Results - YOLOv8x



Fig. 12. Training progress for YOLOv8x



the Husky robot to improve its navigation capabilities in challenging off-road scenarios. According to the results, it is possible to conclude that the YOLO model can be successfully used to perform this task, where the YOLOv8x model shows the best performance. In conclusion, although this work has demonstrated the successful development of a computer vision system based on the YOLO model for cone detection in off-road environments, there are several areas for future development:

- Parameter Optimization: As YOLO offers a variety of parameters that can significantly affect model performance, thorough investigation could lead to enhanced precision in cone detection.
- Integration Details: Providing insights into how the vision system will operate and communicate with the robot's hardware will improve understanding and facilitate seamless integration.

ACKNOWLEDGMENT

We acknowledge Centro Universitário FEI for its continuous support and financial assistance.

REFERENCES

[1] K. I, " A novel dataset for real-time traffic cones detection using deep learning.," 2022.
[2] E. G., "A platform for autonomizing off-road vehicles," 2023.
[3] R. Gandikota, "Image Processing Techniques for Indoor and Outdoor Self Driving cars," 2018.
[4] G. Alves, "Entendendo Redes Convolucionais (CNNs)," 2018.
[5] H. C., "From On-Road to Off: Transfer Learning within a Deep Convolutional Neural Network for Segmentation and Classification of Off-Road Scenes," 2016.