INE410121-41000025DO/ME - Computer Vision 24 (2024) 1-18 Submitted 11/12; Revised 10/12; Published 11/12

Cristiano Langner, Kelmer Martins Cunha, Pedro Anderson Ferreira Castro

Real-Time Object Detection for Self-driving Cars

Cristiano Langner CRISTIANOLANGNER.EGE@GMAIL.COM
Postgraduate Program in Electrical Engineering
Federal University of Santa Catarina (UFSC)
Florianópolis, Brazil

Kelmer Martins Cunha KELMERMARTINSCUNHA@GMAIL.COM
Mycology Lab (MICOLAB - MIND.Funga)
Federal University of Santa Catarina (UFSC)
Florianópolis, Brazil

Pedro Anderson Ferreira Castro PEDRO.A.F.CASTRO@POSGRAD.UFSC.BR
Machine Learning and Applications Research Group (GAMA)
Federal University of Santa Catarina (UFSC)
Florianópolis, Brazil

Editor: Cristiano Langner, Kelmer Martins Cunha, Pedro Anderson Ferreira Castro

Abstract

Nowadays, there is a heavy dependence on land transportation from day-to-day population circulation to goods production, etc. With such a high traffic density, mainly in industrialized cities, there are also problems associated with this scenario, such as road accidents and traffic jams, or industry-related, costs associated with logistics or even accessibility aspects such as Persons with a Disability (PwD) integrations to vehicles. All those problems can benefit from the implementation of emerging technologies, such as autonomous cars (AC) and driver assistance systems (DAS), where both rely mostly on Computer Vision problems often using Deep-Learning-based solutions for implementations. This project aimed to develop a CNN-based, with the famous YOLO architecture, application for object detection in real-time considering the road and traffic contexts of Florianópolis, Santa Catarina, Brazil, using Transfer Learning and Fine Tuning techniques, addressing the issues related to sparse data from southern countries such as Brazil. Our results [1] show that, for the approach of using the custom dataset [2], we achieved a mAP50 and mAP50-95 of 0.714 and 0.475, respectively, for all classes on the test set, with a Recall above 0.6222 for all classes but the bicycle, where the model performed poorly due to the sparsity of samples of this class in our custom data [3].

Keywords: object detection, self-driving cars, deep learning, computer vision, YOLO

1 Introduction

The current flow in and between urban and suburban settings is dominated by paved roads in most industrialized cities, revealing a global dependence on transportation based

1. Demo video available at Google Drive
2. Custom Dataset available at RoboFlow Repository
3. Presentation slides at Google Presentation

on these conditions. Therefore, following urbanization and industrialization, transportation via roads has shaped a diversity of essential aspects related to modern civilization, from day-to-day population circulation to goods production and transportation. Regardless of being an efficient way of traveling, high road dependence creates critical health issues, especially in low-income countries, where related death rates can be three-fold higher when compared with high-income countries, as shown in Organization (2018). With that in mind, Brazil, despite having large land areas disconnected from main road networks and most of its network non-pavemented, it ranks among the countries with bigger road networks, with almost 2 million km of roads and more than 110 million registered vehicles, as CNT (2022) shows. Following global trends, associated with this large road network extension is a high accident and incident indices, with ca. 40,000 deaths each year caused by road traffic. This places Brazil fifth in the number of road-associated deaths globally, as the studies of Aquino et al. (2020) illustrate, an important health issue affecting the Brazilian population. Therefore, road safety is an urgent problem in the country, calling for solutions that considerably reduce mortality and injuries. Considering that human error is the major cause of road traffic accidents, as the work of dos Santos and de Souza (2022) exposes, the promotion and massive implementation of emerging technologies, such as autonomous cars (AC) and driver assistance systems (DAS), could represent an efficient solution for substantially ameliorate problems and avoid casualties, as Szénási et al. (2021) exhibits. The large development in the deep learning algorithms within the computer vision field, especially the advent of Convolutional Neural Networks (CNNs), has changed a lot the amount of information/decisions that DAS can make, increasing safety associated with these systems and leading to the popularization and commercialization of ACs as a result. At any level used to define ACs based on their autonomy and for any pipeline applied in this context, one fundamental task that permeates most of the implementations is real-time object detection, which is essential to the user and other traffic participants' safety. For this task, several CNNs architectures have been implemented, with most applications being broadly single or double-stage detectors, as the work of Grigorescu et al. (2020) shows, where a trade-off between performance and efficiency exists, with single-stage detectors displaying less accuracy, but with a better performance. While CNN applications to ACs and DAS are successful and widely adopted, with many datasets dedicated to training these models, the majority of these are collected from single cities in Northern/European countries, as referred in Grigorescu et al. (2020). These characteristics can hamper the direct use of these datasets in Southern countries with high road traffic casualties, such as Brazil, as the road scenarios and conditions in these countries can differ significantly from Northern/European scenes. Therefore, CNN implementation for object detection in ACs and DAS needs to be aligned with local contexts, guaranteeing that satisfactory performance is achieved, and resulting in increased road safety. This project aimed to develop a CNN-based application for object detection in real-time considering the road and traffic contexts of Florianópolis, Santa Catarina, Brazil, using Transfer Learning and Fine Tuning techniques[4]. For this, we took advantage of the You Only Look Once (YOLO) unified architecture, a CNN that approaches object detection as a regression problem, efficiently detecting and classifying objects based on bounding boxes and class probabilities, introduced by Redmon et al.

4. Code available at GitHub Repository

(2016), by fine-tuning it on both public and custom data for this application. Our results show that, for the approach of using a custom dataset, we achieved a mAP50 and mAP50-95 of 0.714 and 0.475, respectively, for all classes on the test set, with a Recall above 0.622 for all classes but the bicycle, where the model performed poorly due to the sparsity of samples of this class in our custom data[5].

2 Related Works

Research in the field of autonomous driving (AD) has seen a relevant increase in recent years, with several works focused on the real-time object detection problem. Due to its simple approach and overall good performance, the YOLO system by Redmon et al. (2016) is massively used in the object detection context within AD research, as shown in Grigorescu et al. (2020). This network is mainly pre-trained on the open-source MS COCO dataset Lin et al. (2014), but generally applied with a fine-tuning approach based on niched public datasets, such as performed by Alahdal et al. (2024). This work evaluated the performance of three different YOLO versions (v5, v7, and v8) based on a virtual dataset simulated on VSim-AV, by Meftah and Braham (2021). The performance metrics included were precision, recall, and mean Average Precision (mAP) at an Intersection over the Union (IoU) threshold of 0.5, where models reached overall precisions from 44% to 94%. Interestingly, different YOLO versions achieved better performance depending on the considered metric, but the YOLOv8 version tested achieved the highest precision, reaching 91% for all classes considered together. Implementing virtual datasets is common in autonomous driving research as the work of Yuan et al. (2020) illustrates and has the advantage of acquiring a large diversity of scenarios with low associated costs, but with the potential to simplify complex scenarios encountered in real contexts. Contrasting implementations in the autonomous driving context take the burden of generating real-life data, as done by Sharma et al. (2024), where YOLOv8 was fine-tuned based on a new dataset of Canadian roads and vehicles. The authors used their custom dataset comprised of ca. 10,000 images and 11 classes and the public RoboFlow dataset B. et al. (2024) for training (using only the RoboFlow dataset or both combined). Models were evaluated based on recall, precision, the composite F1-score, and mAP at an IoU of 0.5. Unsurprisingly, the best-performing models were the ones that included both general and custom datasets, reaching accuracies from 14% to 91% depending on object class. These results are valuable to showcase that better overall performance is achieved when the local context is included in used datasets.

3 Methodology

In this section, the data used, the model's architecture and approaches used, such as the pre-processing techniques, and also the evaluation metrics are presented to achieve the objectives described earlier for this project.

5. Presentation video available at Youtube

3.1 Datasets

This project made use of 2 datasets to test how the application would work in each scenario. First, we fine-tuned the model on the public dataset KITTI by Geiger et al. (2012) and evaluated its performance using the model for real-time inference (camera stream inside the car). After this, we used our custom dataset, the CompVisionProjectUFSC (2024), for fine-tuning and again tested the model's performance in the real-time scenario.

3.1.1 KITTI Dataset

The KITTI Dataset (Karlsruhe Institute of Technology and Toyota Technological Institute) by Geiger et al. (2012) is one of the most popular datasets for use in mobile robotics and autonomous driving. It consists of 7481 training images and 7518 test images as well as the corresponding point clouds, comprising a total of 80.256 labeled objects. the images are in RGB color space and are stored as Portable Network Graphic (PNG) files, where the camera images are cropped to a size of 1382 x 512 pixels using libdc's format 7 mode. After rectification, the images get slightly smaller. The cameras are triggered at 10 frames per second by the laser scanner (when facing forward) with shutter time adjusted dynamically (maximum shutter time: 2 ms). The classes that the dataset contains are: car, pedestrian, van, cyclist, truck, misc, tram, and person sitting. Figure 1 illustrates some images taken from the dataset.

Figure 1: Sample images from the KITTI dataset, showing the diversity of their recordings. Source: Geiger et al Geiger et al. (2012).

3.1.2 Custom Dataset (CompVisionProjectUFSC)

For the CompVisionProjectUFSC (2024) custom dataset, we used a base of 9 videos recorded over the region of Florianópolis, in Santa Catarina (Brazil), from several conditions of illumination (rain, cloud, sunny, and night) to create diversity, while filming over the

region around the Federal University of Santa Catarina (UFSC) and also the main roads. Figure 2 shows examples of the data present in this dataset, while Figure 3 shows the distribution of the classes labeled. The videos were recorded using only a horizontal view, inside/out of a car, with a resolution of 720p (1280 x 720 pixels) at 30 Frames Per Second (FPS) using smartphones' cameras. The videos were then passed through a Python script that extracts random frames from it and stores them in a folder, where posteriorly, all frames were manually inspected to remove images that were not useful for posterior labeling. We used the B. et al. (2024) auto-labeling tool, further evaluating each label generated from the 1447 images in the dataset, also manually labeling if the tool doesn't. After this, the data was split into training, validation, and test sets, comprising 1110, 244, and 93 images, respectively, with 9570 annotations in total, where 800 images contain 2 to 7 objects annotated. The classes present in this dataset are bicycle, bus, car, motorcycle, person, sign (traffic signs), traffic light, and truck.

Figure 2: Sample images from the CompVisionProjectUFSC dataset, showing the diversity of its recordings. Source: CompVisionProjectUFSC (2024).

3.2 Pre-Processing

Before passing the images to the networks tested, a pre-processed step is taken. For the KITTI dataset, the images were split into training and testing sets, further using a script available at Kaggle to convert the annotations to the YOLO format, since it is originally in COCO format. Furthermore, in both datasets, the images were rescaled to 640x640 pixels, to match the input pattern of the YOLO models. Data augmentation, which is a technique applied to avoid overfitting on training data and also to increase the number of samples on sparse datasets, was applied to both the KITTI and the custom dataset, with the transformations and hyperparameters listed in Table 1, where the library Albumentations by Buslaev et al. (2020) is applied to easily create the transformations.

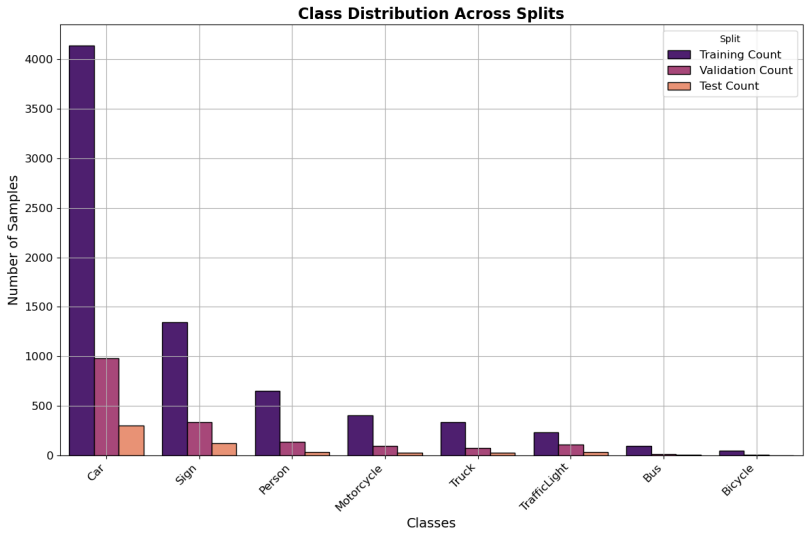

Figure 3: Classes distribution over the training, validation, and test sets from the CompVisionProjectUFSC dataset. Source: authors.

Transformation	**Parameters and Values**
Blur	Probability: $p = 0.01$ Blur limit: $(3, 7)$
MedianBlur	Probability: $p = 0.01$ Blur limit: $(3, 7)$
ToGray	Probability: $p = 0.01$ Output channels: 3 Method: Weighted Average
CLAHE	Probability: $p = 0.01$ Clip limit: $(1.0, 4.0)$ Tile grid size: $(8, 8)$

Table 1: Transformations done on both datasets using the Albumentations library by Buslaev et al. (2020). Source: authors.

3.3 Models

In this project, only 2 models from the same architecture, the YOLO, introduced by Redmon et al. (2016), were utilized. Here, tests were made with the two versions, the YOLOv3, implemented in the OpenCV library, and the YOLOv11, implemented on the Ultralytcs enterprise. The YOLO (that stands for "You Only Look Once") models are real-time object detection systems that identify and classify objects in a single pass of the image. In other words, the model only looks at the image once, and from this 'single pass' can identify objects in the image. The version 3 builds upon the original YOLO and

YOLOv2 architectures. It is designed to balance accuracy and speed, making it suitable for applications requiring real-time performance. Its key features are using Darknet-53 as its backbone, featuring residual connections for efficient feature extraction. It predicts bounding boxes at three scales for improved detection of objects of varying sizes. The architecture employs anchor boxes and outputs class probabilities, objectness scores, and refined bounding box coordinates, balancing speed, and accuracy for real-time applications, as the Figure 4 shows. Meanwhile, the latest YOLO version, YOLOv11, introduces a more efficient architecture with C3K2 blocks, SPFF (Spatial Pyramid Pooling Fast), and advanced attention mechanisms like C2PSA. YOLOv11 is designed to enhance small object detection and improve accuracy while maintaining the real-time inference speed that YOLO is known for, as explained in Rao (2024). Figure 5 illustrates the architecture, even though we used the medium version, which just differs by using a tinier backbone for feature extraction.

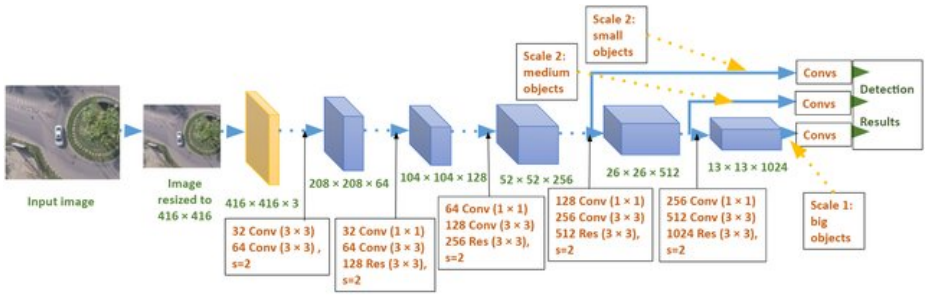

Figure 4: YOLOv3 architecture's schematic. Source: Ammar et al. (2021).

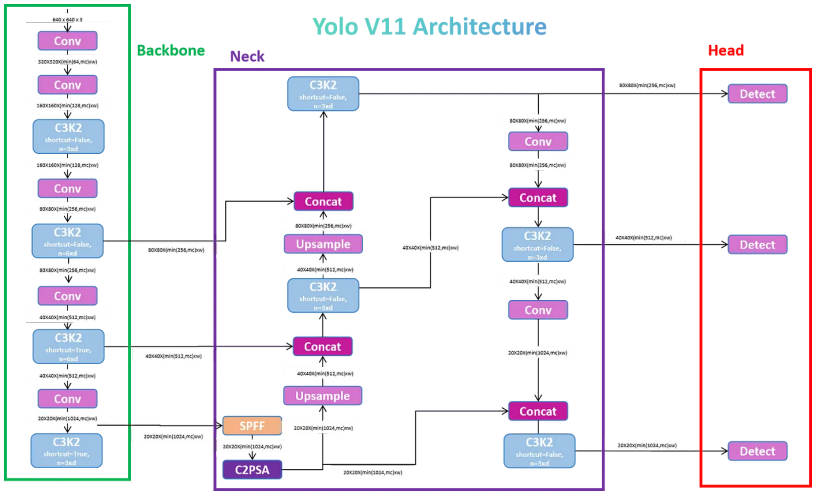

Figure 5: YOLOv11 architecture's schematic. Source: Rao (2024).

3.4 Evaluation Metrics

When evaluating a model's performance over different architectures or datasets, it is important to establish metrics to check the performance of each test on the proposed task. For this work, 5 metrics were adopted: precision, recall, F1-Score, mAP50, and mAP50-95, where the mAP50 and recall were our goal metrics.

3.4.1 Precision

It is important to specify how many of the predicted positive values are correct. Precision measures this aspect and is particularly useful when the number of False Positives is high. Precision is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

where TP is the number of True Positives and FP is the number of False Positives.

3.4.2 Recall

Also known as Sensitivity or True Positive Rate, is the ratio of correctly predicted positive observations to all observations in actual class. It answers the question "Of all the items that are positive, how many did we correctly identify as positive?". The Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

where FN is the number of False Negatives.

3.4.3 F1-Score

It is the weighted average of Precision and Recall. Therefore, it takes both false positives and false negatives into account. F1-Score is more useful than accuracy, especially if you have an uneven class distribution. The equation for the F1-Score is:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

3.4.4 mAP50

The mean Average Precision at a 50% Intersection over Union (IoU) threshold, commonly referred to as mAP50, is a metric used to evaluate the performance of object detection models. It calculates the mean of the average precision values across all classes for predictions that meet or exceed the 50% IoU threshold. The mAP50 metric is defined as:

$$\text{mAP50} = \frac{\sum_{c=1}^{C} \text{AP}_c}{C} \tag{4}$$

where C is the total number of classes, and AP_c represents the average precision for class c. The mAP50 metric emphasizes correct localization and is particularly useful for evaluating model precision at a fixed IoU threshold.

3.4.5 mAP50-95

The mean Average Precision over multiple IoU thresholds, commonly referred to as mAP50-95, provides a more comprehensive evaluation of an object detection model's performance. It is calculated as the mean of average precision values over IoU thresholds ranging from 50% to 95% in increments of 5%. The mAP50-95 metric is defined as:

$$\text{mAP50-95} = \frac{\sum_{t=1}^{T} \frac{\sum_{c=1}^{C} \text{AP}_{c,t}}{C}}{T} \tag{5}$$

where T is the total number of IoU thresholds, C is the total number of classes, and $\text{AP}_{c,t}$ represents the average precision for class c at threshold t. The mAP50-95 metric provides a balanced evaluation by considering both strict and relaxed IoU thresholds.

4 Experiments and Results

The evaluation process consisted of three main phases: testing a pre-trained model, fine-tuning with a public dataset focused on autonomous driving, and training with a custom dataset representing local conditions.

4.1 Model Training Setup

Each model was trained with specific configurations and setups as follows:

YOLOv3: Used directly without additional training, pre-trained on the COCO dataset.

YOLOv11s (Small): Fine-tuned on the KITTI dataset for 20 epochs.

YOLOv11m (Medium): Fine-tuned on the custom CompVisionProjectUFSC dataset for 200 epochs.

4.2 Model Performance Evaluation

The YOLOv3 model, pre-trained on the COCO dataset, was tested as a baseline. Despite its widespread use in autonomous driving research, it performed poorly when applied to Brazilian traffic. Misclassifications were frequent, particularly for bicycles and motorcycles, and the mean Average Precision (mAP) scores were low, highlighting the need for domain-specific adaptation.

The YOLOv11s model, fine-tuned on the KITTI dataset, showed improvements in mAP and recall scores over YOLOv3. While it performed better in recognizing urban traffic objects, challenges persisted, especially in distinguishing bicycles from motorcycles and accurately identifying buses under specific conditions.

Fine-tuning YOLOv11m on the custom dataset yielded good results, although it faced difficulties in specific scenarios, such as differentiating bicycles from motorcycles and identifying buses at non-typical angles.

Overall, these experiments highlight the importance of using datasets that represent local conditions to improve accuracy and safety in real-time object detection for ACC and DAS applications. The results also underscore the limitations of directly implementing popular public datasets in diverse regions, such as Brazilian cities, where traffic patterns, vehicle appearances, and environmental conditions differ significantly. This calls for greater

inclusion of globally diverse scenarios in widely adopted datasets to ensure more reliable performance in underrepresented contexts.

4.2.1 Performance Analysis of YOLOv11s on the KITTI Dataset

The normalized confusion matrix in the figure 6 displays classification performance across all classes in the KITTI dataset. Each row represents the actual class, and each column represents the predicted class, with values normalized to percentages (or fractions). Diagonal values represent correct predictions, while off-diagonal values indicate incorrect classifications, as referred in the work Duong et al. (2024).

The model performs well for certain classes like cars (92%) and trucks (98%), achieving high accuracy.

Pedestrian detection is weaker (71%), which might indicate challenges in recognizing smaller or less distinct features of pedestrians in urban environments.

Cyclists also present challenges (83% accuracy), which is consistent with the known difficulty in distinguishing between similar classes like cyclists, motorcycles, or pedestrians.

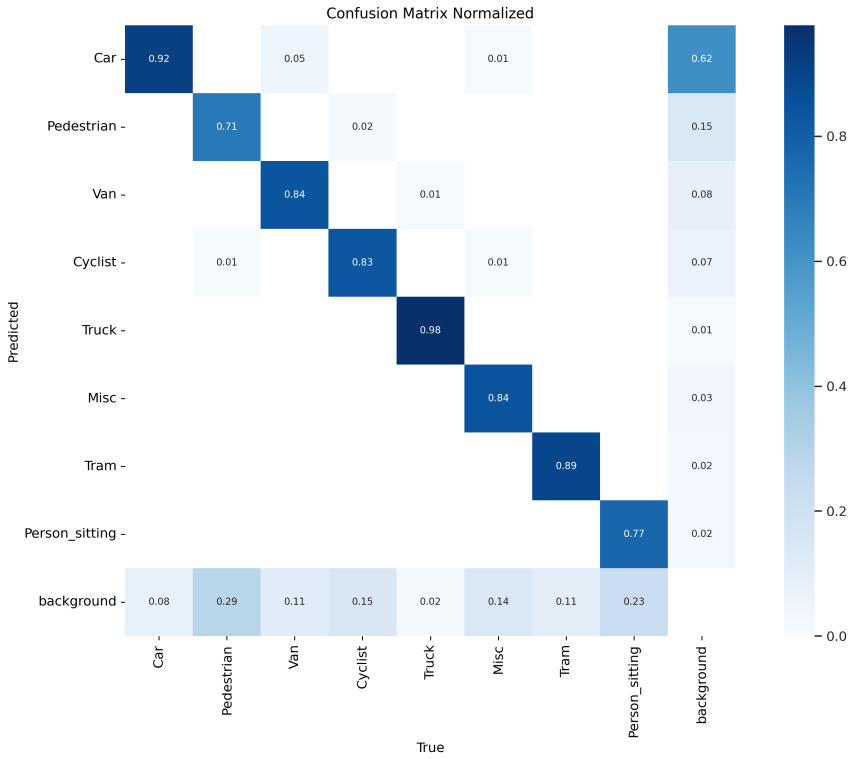

Figure 6: Normalized Confusion Matrix KITTI Dataset. Source: authors.

The precision-recall curve measures in the figure 7 trade-off between precision (how many of the predicted positives are correct) and recall (how many of the actual positives were correctly identified) for each class. The area under the curve (AUC) gives an idea

of the overall class performance, while the mAP score summarizes the model's detection capabilities across all classes, as discussed in the work Duong et al. (2024).

Truck (0.994 AUC) and Car (0.957 AUC): These classes have the highest precision-recall values, indicating strong and reliable detection.

Cyclists (0.855 AUC): This class shows reasonable performance but still has room for improvement, particularly in edge cases (e.g., cyclists in crowded or complex scenes).

Pedestrians (0.786 AUC): Relatively weaker performance, aligning with the confusion matrix results, suggesting difficulty in recognizing smaller or partially occluded objects.

The mAP50 for all classes is 0.885, which is a good result, but with variability across different object categories.

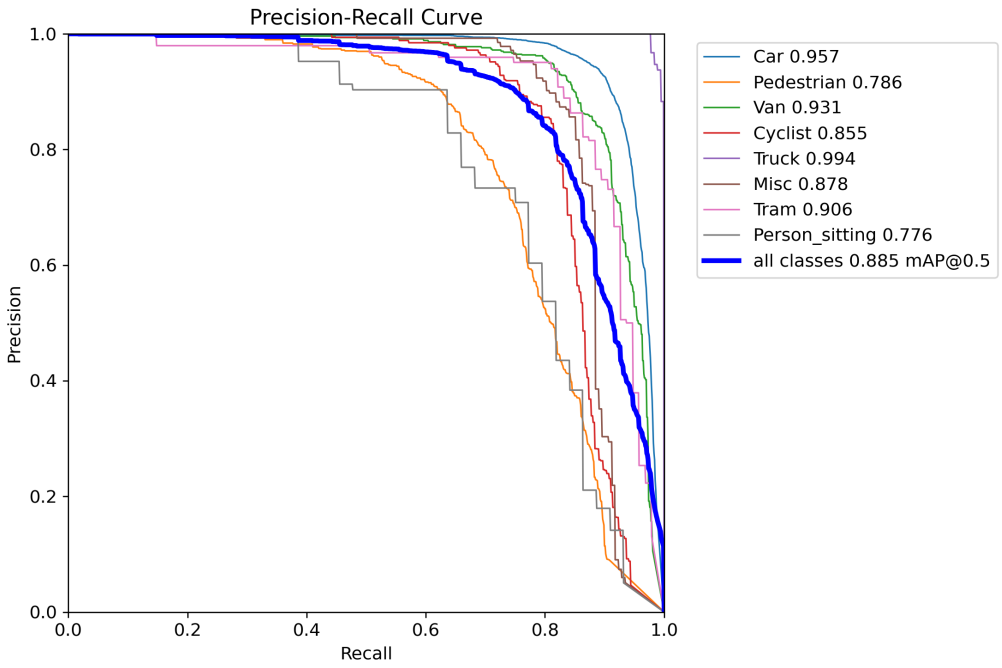

Figure 7: Precision Recall Curve KITTI Dataset. Source: authors.

The Recall steadily increases, surpassing 0.80, which demonstrates the model's improving ability to detect a higher proportion of relevant objects across epochs.

The mAP50 metric reaches approximately 0.90, indicating strong detection performance with sufficient overlap between predicted and ground truth boxes.

The mAP50-95 stabilizes at around 0.65, which suggests good performance even under more stringent overlap conditions.

In conclusion, the YOLOv11s model trained on the KITTI dataset exhibits consistent progress in terms of recall and high detection performance, as evidenced by the high mAP50 and mAP50-95 values. These metrics highlight the model's ability to detect objects accurately, even in complex urban environments, demonstrating its effectiveness in real-world applications.

4.2.2 Evaluation of YOLOv11m Performance on a Custom Dataset

The confusion matrix in the figure 8 shows high accuracy in detecting Signs (85%), Traffic Lights (77%), and Cars (78%).

Significant confusion between Bicycles (33%) and other classes such as Motorcycles, highlighting the model's difficulty in distinguishing visually similar objects.

Bus (60%) is often misclassified, possibly as cars or trucks, due to variations in angles or insufficient representation in the training dataset.

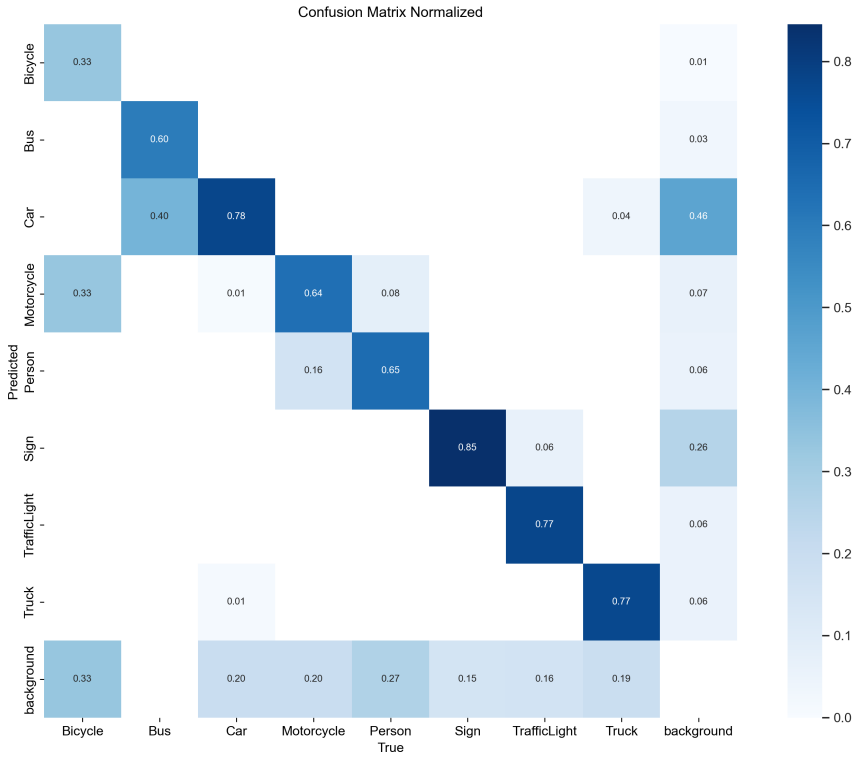

Figure 8: Normalized Confusion Matrix Custom Dataset. Source: authors.

Figure 9 evaluates the trade-off between precision and recall for each class. The mAP is an aggregated measure of the model's performance across all object categories.

Best performing classes:

Traffic Light (0.843 AUC): High reliability in detecting traffic lights.

Car (0.804 AUC): Demonstrates strong performance for vehicles.

Motorcycle (0.777 AUC): Shows reasonable consistency in recognizing motorcycles.

Challenging classes:

Bicycle (0.389 AUC): The poorest performance due to significant misclassifications.

Bus (0.717 AUC): Moderate detection performance, but still below the reliability threshold for real-world deployment.

The model achieves a mAP50 of 0.714, indicating decent overall performance but with room for improvement in handling challenging classes like bicycles and buses.

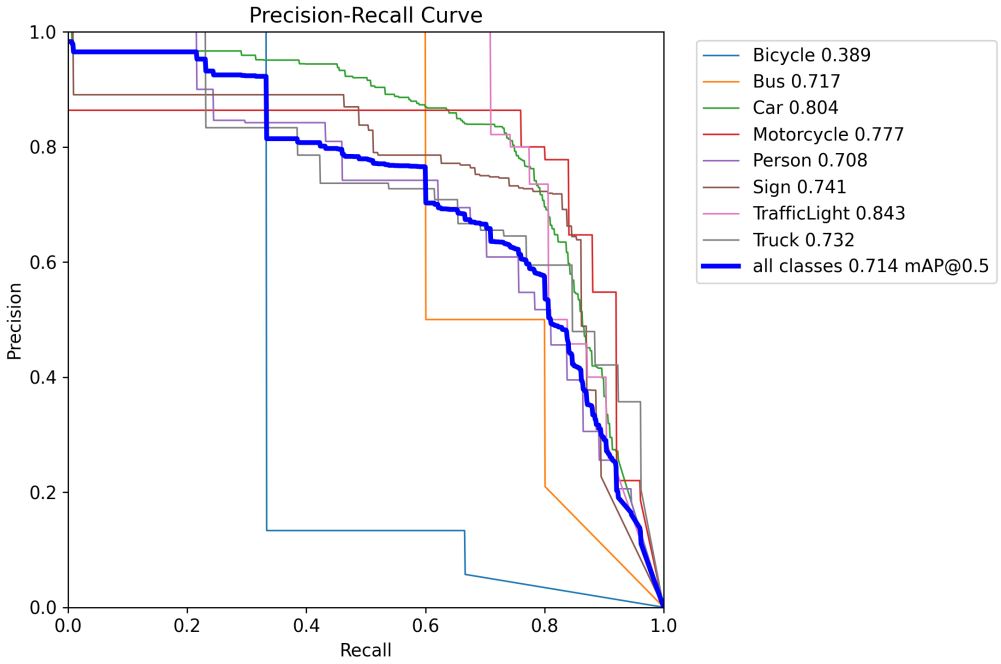

Figure 9: Precision-Recall Curve Custom Dataset. Source: authors.

The precision shows rapid improvement during the first 50 epochs, stabilizing between 0.75 and 0.80, indicating the model's effectiveness in minimizing false positives. The recall curve also increases consistently, reaching values near 0.75, showcasing the model's strong ability to detect relevant objects.

The mAP50 metric, which evaluates average precision, reaches close to 0.714, indicating excellent performance in detecting objects with significant overlap. Meanwhile, the mAP50-95, which measures mAP across a range of overlap thresholds, converges around 0.475. Although this is slightly lower than the KITTI benchmark of 0.65, it still demonstrates solid performance under stricter overlap conditions.

Overall, the model shows competitive precision and recall, with stable loss and metric curves. The slightly lower mAP50-95 might suggest that our custom dataset exhibits higher variability or fewer samples for certain classes. Nevertheless, the model is well-fitted to our dataset, showcasing good generalization and high accuracy. Next steps could include increasing the sample diversity for under performing classes, analyzing false positives and negatives for targeted improvements, or applying data augmentation techniques to further enhance generalization.

4.3 Discussion of Results

An analysis of the model's performance in different traffic scenarios reveals its strong detection capabilities even in unfavorable conditions. As illustrated in Figure 10, which shows four frames captured during the usage process, the model demonstrated accurate

detections on high-speed highways and urban streets. In highway scenarios, the model reliably identified objects such as cars, motorcycles, and buses despite higher speeds and varying lighting. In urban environments, it excelled in detecting pedestrians, traffic signs, and other critical objects, demonstrating its adaptability to diverse contexts.

Notably, even for classes that previously presented challenges, such as buses and motorcycles, the model achieved satisfactory detections with reasonable confidence levels, as long as the angle and lighting were favorable for these two classes in question. For example, motorcycles were accurately identified in congested urban conditions, while buses were successfully detected in high-speed environments.

The use of night footage in the dataset provided an improvement not only in nighttime environments, but also in places such as indoor parking lots or tunnels that have less lighting.

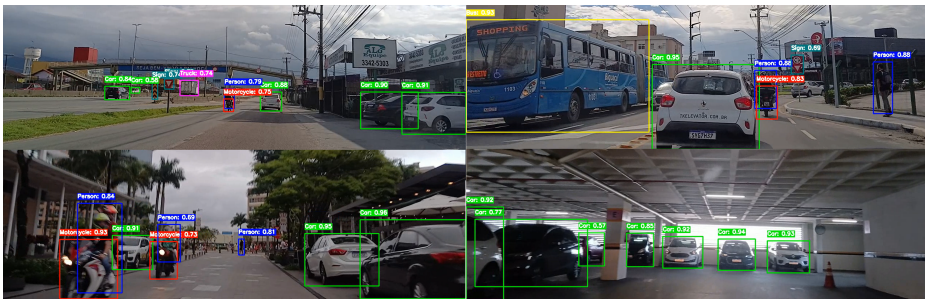

Figure 10: Good Object Detection. Source: authors.

Despite the overall improvement, certain challenges persisted:

Bicycle/Motorcycle Confusion: The model sometimes mistakes bicycles for motorcycles, especially in cases where the objects appear similar in shape or pose.

Bus Identification: Buses were occasionally misclassified as trucks, particularly when viewed from non-typical angles or under suboptimal lighting conditions.

Figure 11 illustrates four frames captured during the object detection tests with the model. The first two frames, on the left, show a situation where a bus was inconsistently identified, alternating between the classes bus and truck. In both cases, the model showed more confidence in the incorrect classification, assigning a probability of 0.77 to the truck class and only 0.62 to the bus class, indicating that the model was more certain in its error than in its correct identification. The two frames on the right show two examples of bicycle identification: one correct and one incorrect. The analysis of the results revealed that bicycles in motion, meaning in use, are more likely to be confused with motorcycles, while stationary bicycles are more easily identified correctly.

Additionally, in the same image where the bicycle was misidentified as a motorcycle, a person was erroneously detected as riding a motorcycle, even though they were simply walking. This rare mistake occurred due to the person's hand position and the shape of their legs, which resembled the posture typically associated with riding a motorcycle. These examples highlight the challenges the model faces in differentiating certain objects, especially under specific angles, movement conditions, and unusual unusual human motion conditions and poses rarely seen in the dataset.

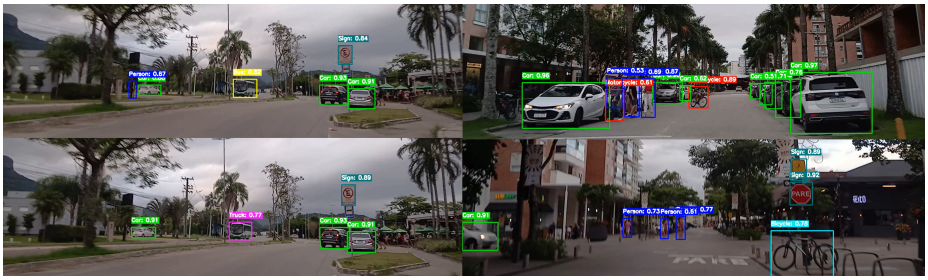

Figure 11: Object Detection Confusion: Bus vs. Truck and Bicycle Identification. Source: authors.

In summary, while the model demonstrated considerable improvement after fine-tuning on the custom dataset, challenges remain in achieving perfect classification accuracy for all object types. The confusion between bicycles and motorcycles, along with issues in bus identification, emphasizes the difficulty of generalizing models trained in different geographical and environmental contexts. However, the results from fine-tuning on the CompVision-ProjectUFSC dataset show promise, with significant gains in mAP and recall, making the model a more reliable tool for real-world applications in Brazil. Future work should focus on further expanding the custom dataset, particularly to address underrepresented classes, and fine-tuning the model for additional robustness under varying road conditions, lighting, and object poses. These efforts will help mitigate the identified issues and improve the model's performance for deployment in real-world autonomous driving systems.

5 Conclusion

Experiments were conducted in this project based on the YOLOv3 and YOLOv11 architectures, aiming to develop models for application in AC and DAS real-time object detection within the Florianópolis region (Santa Catarina, Brazil) traffic context. Efforts were made to construct a custom dataset representing local scenarios and traffic-related objects in several climate/illumination conditions, comprising 1447 images with 9570 annotations representing 8 classes with diverse scene conditions captured during daytime and night. The first set of experiments was designed to evaluate the pre-trained YOLOv3 system in the Florianópolis local context, which performed poorly, indicating that popular pre-trained systems used in AD research are not reliable for direct implementation in underrepresented regions, such as Brazilian cities. Accordingly, the second set of experiments focused on training and evaluating the YOLOv11 architecture fine-tuned with an AD-focused dataset, resulting in better but still unreliable performance for safe real implementations. The final experiment with models fine-tuned on the custom dataset generated in this work performed substantially better than the previous results, with a mAP50 and mAP50-95 of 0.714 and 0.475, respectively. Also, the recall was above 0.6 for all classes, except for the bicycle class. As expected, for some challenging classes at certain conditions, such as bicycles and motorcycles in similar angles, the implementation struggles to assign classes correctly some-

times. Overall, the results obtained here indicate that custom datasets representing local conditions are critical for accurate, and thus safety-enhancing, real-time object detection in AC and DAS contexts. Additionally, the results showed that the direct implementation of popular public datasets does not result in a good performance for Brazilian cities' traffic scenes, highlighting the sub-representation of globally diverse images in widely adopted datasets, and calling for caution when implementing these in Southern countries' settings.

Acknowledgments and Disclosure of Funding

Our team thanks Prof. Dr. rer.nat. Aldo von Wangenheim for providing hardware to train networks more efficiently, as well as the knowledge passed on in the course, along with the teaching assistant interns. We also thank RoboFlow (B. et al. (2024)) and Ultralytcs for the tools that facilitated the implementation of this project.

References

Nusaybah M Alahdal, Felwa Abukhodair, Leila Haj Meftah, and Asma Cherif. Real-time object detection in autonomous vehicles with yolo. *Procedia Computer Science*, 246: 2792–2801, 2024.

Adel Ammar, Anis Koubaa, Mohanned Ahmed, Abdulrahman Saad, and Bilel Benjdira. Vehicle detection from aerial images using deep learning: A comparative study. *Electronics*, 10:820, 03 2021. doi: 10.3390/electronics10070820.

Érika Carvalho de Aquino, José Leopoldo Ferreira Antunes, and Otaliba Libânio de Morais. Mortality by road traffic injuries in brazil (2000–2016): capital cities versus non-capital cities. *Revista de saude publica*, 54:122, 2020.

Dwyer B., Nelson J., and Hansen T. et. al. Research — roboflow.com. `https://roboflow.com/research#cite`, 2024. [Accessed 04-12-2024].

Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. ISSN 2078-2489. doi: 10.3390/info11020125. URL `https://www.mdpi.com/2078-2489/11/2/125`.

CNT. *Anuário CNT do transporte*. National Transport Confederation, 2022.

CompVisionProjectUFSC. Object detection dataset of florianópolis streets for self-driving cars applications. `https://universe.roboflow.com/compvisionprojectufsc/obj_detection_autonomousdriving`, nov 2024. URL `https://universe.roboflow.com/compvisionprojectufsc/obj_detection_autonomousdriving`. visited on 2024-12-03.

Damião Flávio dos Santos and Yuri Machado de Souza. Binary logistic regression model applied to data on accidents occurred on federal highways in brazil. *Research, Society and Development*, 11(15):e120111536833–e120111536833, 2022.

Viet Hung Duong, Duc Quyen Nguyen, Thien Van Luong, Huan Vu, and Tien Cuong Nguyen. Robust data augmentation and ensemble method for object detection in fisheye camera images. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7017–7026. IEEE, 2024.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

Leila Haj Meftah and Rafik Braham. Vsim-av: A virtual simulation platform for autonomous vehicles. In *International Conference on Intelligent Systems Design and Applications*, pages 379–388. Springer, 2021.

World Health Organization. Global status report on road safety 2018 — who.int. `https://www.who.int/publications/i/item/9789241565684`, 2018. [Accessed 03-12-2024].

S Nikhileswara Rao. YOLOv11 Explained: Next-Level Object Detection with Enhanced Speed and Accuracy — nikhil-rao-20. `https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376` 2024. [Accessed 04-12-2024].

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. URL `https://arxiv.org/abs/1506.02640`.

Teena Sharma, Abdellah Chehri, Issouf Fofana, Shubham Jadhav, Siddhartha Khare, Benoit Debaque, Nicolas Duclos, and Deeksha Arya. Deep learning-based object detection and classification for autonomous vehicles in different weather scenarios of quebec, canada. *IEEE Access*, 2024.

Sándor Szénási, Gábor Kertész, Imre Felde, and László Nádai. Statistical accident analysis supporting the control of autonomous vehicles. *Journal of Computational Methods in Sciences and Engineering*, 21(1):85–97, 2021.

Wei Yuan, Ming Yang, Chunxiang Wang, and Bing Wang. Vrdriving: A virtual-to-real autonomous driving framework based on adversarial learning. *IEEE Transactions on Cognitive and Developmental Systems*, 13(4):912–921, 2020.