



OPEN

A deep learning-based approach for axle counter in free-flow tolling systems

Bruno José Souza^{1,2✉}, Guinther Kovalski da Costa¹, Anderson Luis Szejka¹, Roberto Zanetti Freire³ & Gabriel Villarrubia Gonzalez⁴

Enhancements in the structural and operational aspects of transportation are important for achieving high-quality mobility. Toll plazas are commonly known as a potential bottleneck stretch, as they tend to interfere with the normality of the flow due to the charging points. Focusing on the automation of toll plazas, this research presents the development of an axle counter to compose a free-flow toll collection system. The axle counter is responsible for the interpretation of images through algorithms based on computer vision to determine the number of axles of vehicles crossing in front of a camera. The You Only Look Once (YOLO) model was employed in the first step to identify vehicle wheels. Considering that several versions of this model are available, to select the best model, YOLOv5, YOLOv6, YOLOv7, and YOLOv8 were compared. The YOLOv5m achieved the best result with precision and recall of 99.40% and 98.20%, respectively. A passage manager was developed thereafter to verify when a vehicle passes in front of the camera and store the corresponding frames. These frames are then used by the image reconstruction module which creates an image of the complete vehicle containing all axles. From the sequence of frames, the proposed method is able to identify when a vehicle was passing through the scene, count the number of axles, and automatically generate the appropriate charge to be applied to the vehicle.

Keywords Axle counter, Deep learning, Free-flow tolling systems, You Only Look Once

Guaranteeing efficient human mobility in cities brings economic benefits and improves the quality of life, both individually and collectively¹. The development in transportation considering both structural and operational aspects is important for achieving high-quality mobility. In light of these improvements, the Intelligent Transportation System (ITS) is a concept that has been refined in recent years². The ITS can be represented by a software and hardware system that uses information and communication technology in mobility to improve transportation efficiency from a strategic standpoint. Mobility improves due to its use in terms of security, punctuality, and real-time information³.

Advances in image processing considering embedded devices have resulted in significant improvements in image collection, storage, and sharing. This technological development is being exploited in a variety of applications, such as image and video capture where cameras are frequently used to perform analysis, identification, and tracking of objects in real-time⁴. This progress allows for the exploration of solutions in areas that are now getting attention, such as urban mobility and smart cities⁵.

Several examples show how technology applications are relevant components of innovation in urban mobility. There are currently applications in operation, such as traffic light control⁶, smart parking⁷, and traffic flow optimization⁸. Other opportunities for image and video processing are the current toll plaza systems. The possibility of the installation of free-flow tolling would allow transportation managers to perform more efficient operations. In addition to providing benefits from automatic tariff collection, this type of system eliminates the need to stop the vehicle at the toll, reduces both traffic conflicts and traffic jams, and improves the security of the tolling system.

Nowadays, the transport sector has been looking for ways to implement electronic toll collection systems in their most advanced configuration, such as the free-flow electronic toll systems⁹. Free-flow systems are an

¹Industrial and Systems Engineering Graduate Program, Pontifical Catholic University of Paraná-PUCPR, Curitiba, Paraná 80215-901, Brazil. ²Pumatronix Electronic Equipment Ltd., Bartolomeu L. de Gusmão 1970, Curitiba 81650-050, Brazil. ³Production and Systems Engineering - PPGEPS, Universidade Tecnológica Federal do Paraná-UTFPR, Curitiba, Paraná 80230-901, Brazil. ⁴Expert Systems and Applications Lab, Faculty of Science, University of Salamanca, Plaza de los Caídos s/n, 37008 Salamanca, Spain. ✉email: bruno.jose@pucpr.edu.br

evolution of electronic systems for automatic vehicle identification, in which gantries installed along a toll road recognize vehicles that cross them and charge them electronically and automatically. The implementation of this system brings productivity gains in toll collection. In addition, it is a viable alternative for the reduction of tariffs since operating costs are reduced. In commercial free-flow tolling systems usually more than one sensor is considered, assuming a multi-sensor approach¹⁰.

To ensure that the axle counting system works correctly, it is necessary that both the positioning of the objects in the scene and the camera are adequately set up. Given this need, an ideal scenario was defined for the application based on the project requirements of a Brazilian company that develops solutions for mobility on highways and urban environments, where this study was applied.

In this solution proposal, this scenario is composed of a portico that must cross the road 6 m over the ground. The number of cameras installed on the gantry should be equal to the number of lanes on the highway, as each camera is responsible for analyzing vehicles passing through one of the lanes. The cameras must be positioned on the gantry so that the captured image contains the vehicle axles. A 45-degree positioning is ideal for the highway's requirements. Figure 1a represents an application scenario on a dual carriageway, while Fig. 1b shows an example of the desired image.

Given this task, this paper proposes an axle counter based on only one sensor, focusing on improving the efficiency of the toll collection system. The proposed approach interprets images through deep learning algorithms to determine the number of axles in the vehicles present in the scene. The main contributions of this paper can be summarized as follows.

- A hybrid method that combines a deep learning structure for classification and a slice image reconstruction process is proposed. The proposed method reduces the necessary number of cameras to be used and automatically generates the charge in a continuous flow toll.
- A dataset, consisting of real and synthetic images, was created and made available in this paper. This dataset was used to train and evaluate the object detector model presented in this research and can be applied in other studies on the subject.
- To achieve the best possible results, the proposed method is based on the most suitable deep learning models, to ensure this outcome, several versions of You Only Look Once (YOLO) are evaluated.

The remainder of this paper is organized as follows: “[Related works](#)” describes previous work on computer vision, object detection, and new tolling systems. In “[Proposed method](#)”, the proposed method is detailed and the considered dataset is explained. The results of the computations are discussed in “[Results and discussions](#)”. Conclusions and also some suggestions for future work are given in “[Conclusions and future work](#)”.

Related works

Automatic toll collection plays an important role in improving highway efficiency and user satisfaction¹¹. With the increasing adoption of automatic toll systems worldwide, it is relevant to ensure that vehicle counting is accurate and reliable¹². Errors in counting can lead to a range of issues, such as incorrect charges, discrepancies in traffic data, and user dissatisfaction¹³. In the case of free-flow tolling, it is important that vehicles and their axles can be correctly detected for the proper functioning of the system.

A promising way to detect objects in images is through the use of computer vision techniques¹⁴. These techniques can be classical, such as the use of feature detection algorithms like the histogram of oriented gradients¹⁵, the viola-jones feature descriptor¹⁶, and texture descriptors¹⁷, which have been widely used for computer vision tasks and object detection applications.

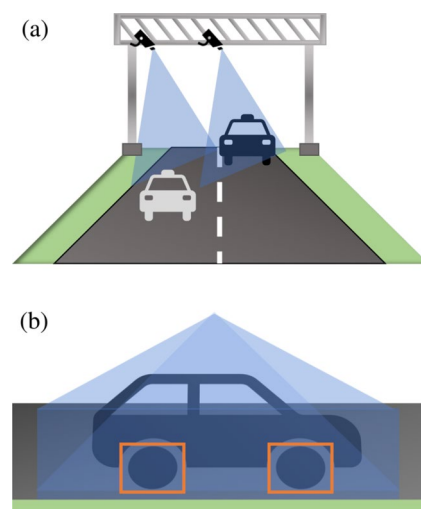


Figure 1. Proposed scenario for axle identification and counting.

The use of artificial intelligence-based models is increasing over time¹⁸, these applications can be focused on classification^{19–21}, time series forecasting^{22–24}, among others^{25–27}. Deep learning approaches are becoming popular²⁸, mainly because of their ability to deal with non-linearities^{29–31}. As presented by Moreno et al.³² the use of hybrid methods can be an outstanding approach in this regard.

Deep learning architectures like Convolutional Neural Networks (CNNs) have promising results for several applications³³, such as navigation³⁴, fault detection³⁵, security³⁶, image classification³⁷, and object detection³⁸. The CNNs have proven to be effective in learning high-level representations directly from input data, giving them greater generalization ability and superior performance in object detection tasks³⁹. Studies like^{40,41} prove that these applications are promising to be embedded.

Regarding embedded systems, Ref.⁴² presented a power-efficient optimizing framework for field-programmable gate array-based acceleration of the YOLO algorithm. Their framework incorporates efficient memory management techniques, quantization schemes, and parallel processing to enhance performance and reduce energy consumption. In the work of Ge et al.⁴³, the YOLOv3 is applied for monitoring full-bridge traffic load distribution. To determine load distribution, the method considers the vehicle's position, size, and weight, providing a more detailed understanding of the stress distribution on the bridge.

Also using YOLOv3, Rajput et al.⁴⁴ presented an approach for automatic vehicle identification and classification in a toll management system. This approach offers real-time, accurate, and efficient toll collection, thereby enhancing the overall effectiveness and user experience of toll facilities. The system's adaptability and potential for security enhancements make it a valuable asset for transportation infrastructure.

Deep learning models are becoming increasingly popular in mobility applications. In Ref.⁴⁵, the authors employ YOLOv3 in their vehicle detection and counting system on highways. Similarly, in Ref.⁴⁶, the authors propose an optimized version of YOLOv4 for vehicle detection and classification. Moreover, other applications, such as^{47,48}, demonstrate different uses of object detection, including license plate detection, road crack detection, and traffic light recognition respectively.

Zarei et al.⁴⁹ proposed the Fast-YOLO-Rec. This model combines the strengths of the YOLO-based architecture and recurrent-based prediction networks for vehicle detection in sequences of images. Based on YOLO, they localize vehicles in each frame, providing a strong initial detection foundation. In Ref.⁵⁰ the vehicle detection is based on a tiny version of YOLOv5 called T-YOLO. An innovation of T-YOLO is its ability to handle tiny vehicles, which are often challenging to detect due to their limited spatial presence.

To overcome the challenges of limited and imbalanced data, Dewi et al.⁵¹ employed Generative Adversarial Networks (GANs) to synthesize additional training samples, expanding the dataset and promoting model generalization. In their application, the YOLOv4 and multiple GANs are explored. The resulting model exhibits promising performance across a wide range of traffic sign types, sizes, and orientations, even in challenging environmental conditions.

To perform the axle counting and speed measurement, Miles et al.⁵² applied the YOLOv3. The model was implemented to track the vehicles and assign a wheel to a vehicle if the center point of the wheel bounding box fell within the vehicle bounding box. The outputs from the axle detection were then processed to produce axle counts for each vehicle, achieving an accuracy of 93% across all vehicles where all axles were visible.

Li et al.⁵³ also employed a tracking-based method for vehicle axle counting. In their work, the YOLOv5s is utilized for axle detection and classification into double and single-wheeled vehicles. The obtained results indicate that the axle detector achieved a mean Average Precision (mAP) of 95.2%. To address the issue of vehicles sometimes not fitting within a single frame, there are some alternative approaches to the tracking-based method, such as utilizing image stitching. However, performing real-time image stitching can be computationally costly, especially for online applications⁵⁴.

In Refs.^{55,56}, the authors focused on the identification of electrical insulators in images taken by unmanned aerial vehicles. They proposed modifications to the base structure of the model to obtain improvements in its performance. In these works, in addition to detecting the chain of electrical insulators, classification is also carried out in case of any visually identifiable defect, such as broken or flash-over insulators.

In Ref.⁵⁷ the application of a modified YOLOv5 model called T-YOLO for vehicle detection was proposed. The images in the dataset used were obtained from a camera with a top view of the parking lot scene, and just as in the case of insulator detection, the vehicles in question end up being relatively small objects in the image, which supports the application of YOLO models for objects which are not necessarily large. In Ref.⁵⁸ the application of a YOLO-v2 for vehicle detection in the Karlsruhe Institute of Technology and Toyota Technological Institute dataset was presented with promising results.

Proposed method

The proposed method aims to examine a sequence of frames to determine the vehicle's axle count. The case study scenario consists of a camera installed in a gantry, which will be responsible for analyzing a lane of the highway. In the scenario, the vehicles pass through the camera sequentially at different intervals of time.

The first challenge of the proposed task concerns identifying when a new vehicle enters the scene. To accomplish this task, a passage manager that uses optical flow techniques in conjunction with axle identification was developed to understand when a vehicle is passing through the image. When the system identifies that there is a vehicle on the scene, it analyzes whether a given frame corresponds to a new passage or a passage that is in progress.

The system must be consistent in identifying which passage each axis belongs to. The system also needs to be able to save the frames corresponding to a given passage and, at the end of the passage, use these frames to generate a panoramic image, where it is possible to visualize the entire vehicle. The slice reconstruction module

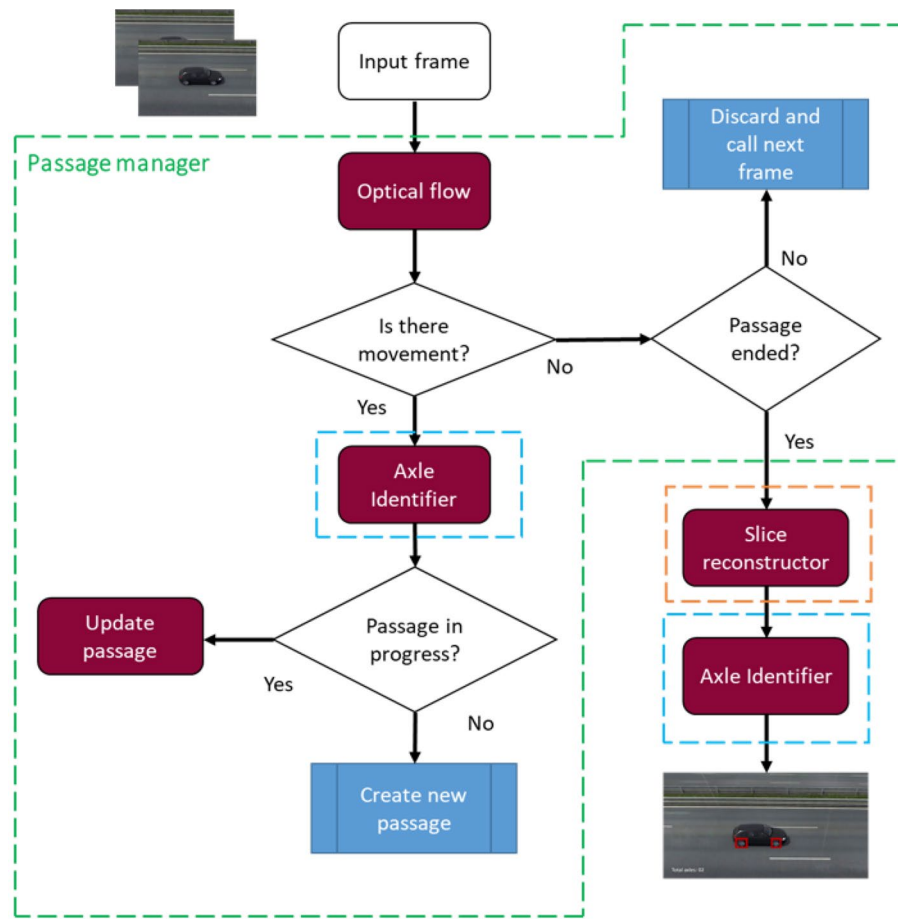


Figure 2. Flowchart of the proposed method.

performs this process. The last step aims to use the axle identifier in the image generated by the slice reconstructor to perform the vehicle axle count. Figure 2 presents a general flowchart of the pipeline of the proposed approach.

Axle identifier

To perform the axis identification task, an object detection YOLO model based on deep learning is applied to detect vehicle wheels and axle counting. The YOLO model is a CNN that divides images into grids, with each grid cell detecting objects within its boundaries⁵⁵. This model has a structure that needs only one shot to detect and classify the objects⁵⁶. The model falls under the category of single-stage detectors, as it only requires a single shot through the image to classify and detect the target object⁵⁹.

YOLO models have exhibited promising results for object detection in different areas, for example in Ref.⁶⁰ where the models were used in the health area, or in Refs.^{61,62} where it is used for safety and urban mobility respectively. Since its first version, YOLO has evolved and gained improvements in its architectures, leading to different versions⁶³.

The YOLOv5 is integrated with an algorithm known as AutoAnchor. This algorithm assesses and refines anchor boxes that do not suit the dataset and training parameters, such as image dimensions⁶⁴. In September 2022 the YOLOv6 was introduced with a new network structure composed of an efficient backbone employing Re-parameterization Very Deep Convolutional Networks (RepVGG) as its backbone⁶⁵. This novel backbone introduces enhanced parallelism compared to earlier YOLO backbones. Another key improvement was the loss for classification and a loss based on Scylla Intersection over Union (SIoU) or Generalized Intersection over Union (GIoU) for regression⁶⁶.

In the YOLOv7 an extended efficient layer aggregation network was proposed. This enables the models to learn more efficiently by managing the shortest longest gradient path⁶⁷. The YOLOv8 proposes a new solution for diverse visual tasks, spanning object detection, segmentation, pose estimation, tracking, and classification⁶⁸. Drawing inspiration from its predecessor YOLOv5, the YOLOv8 retains foundational elements while imbuing the CSP layer with new attributes, now called the C2f module⁶⁹.

For a complete evaluation, in this paper, the YOLOv5, YOLOv6, YOLOv7, and YOLOv8 versions are compared. Since there is a trade-off between computational effort and the desired result the medium variation of each model is considered. Considering that YOLOv7 doesn't have this variation the YOLOv7x is considered. The pseudocode to perform predictions using YOLO is presented in Algorithm 1.

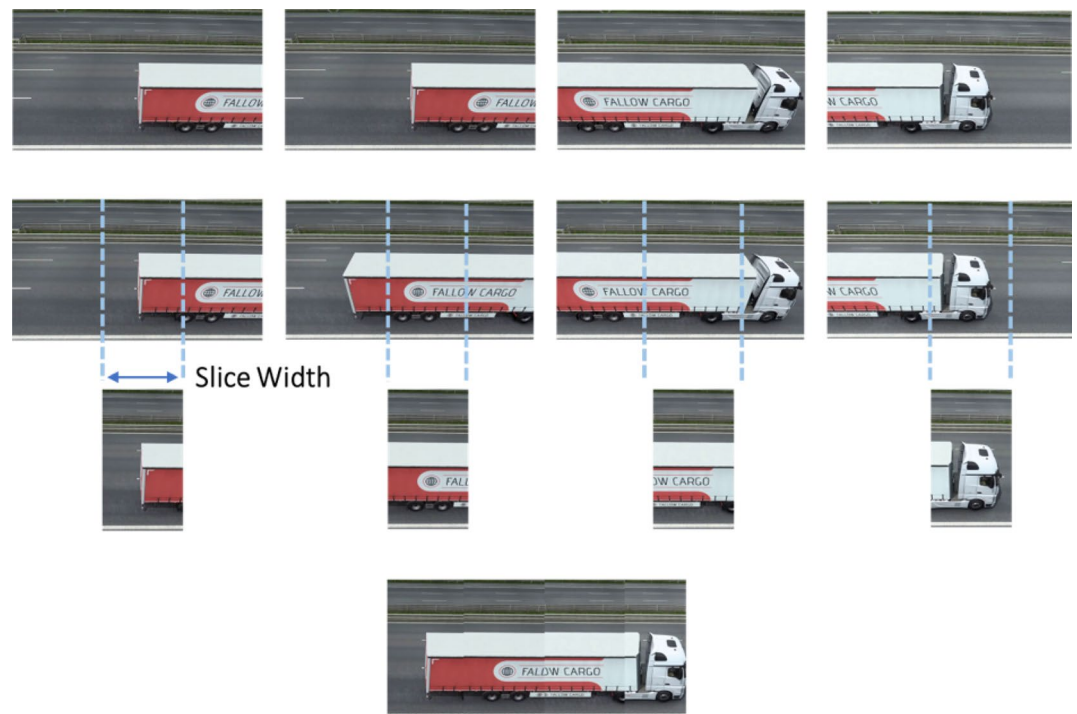


Figure 3. Slice reconstruction process.

-
- 1: Load pre-trained weights of the CNN model.
 - 2: Load input image.
 - 3: Preprocess image (resize, normalization, etc.).
 - 4: Pass the preprocessed image through the CNN.
 - 5: Obtain predictions, class probabilities, and confidence scores.
 - 6: Apply non-maximum suppression to filter out overlapping boxes.
 - 7: Select boxes with confidence scores above a certain threshold.
 - 8: Save the final bounding boxes with associated class labels.
-

Algorithm 1. You Only Look Once

Slice reconstructor

The image slice reconstructor module is responsible for creating an image containing all the vehicle axles present in a given passage. To perform its function, the slice reconstructor takes as input a sequence of frames.

Assuming a scenario in which a vehicle passes through a camera, with horizontal displacement in a single direction, a sequence of frames is obtained where it is possible to get visual information of the complete vehicle. To be able to select the parts necessary to rebuild the vehicle automatically, it was selected the slice from the center of each frame. Since it is considered that the sequence of frames has a complete vehicle passage, which indicates that every part of the vehicle will at some point be located in the center of the image.

Concerning the width of the selected slice, given a sequence of frames, it is possible to consider that the vehicle has a displacement in pixels at each frame. To select the information to rebuild the vehicle without redundancy of information, the ideal scenario is that the width of the slice should be equal to the displacement of the vehicle in pixels.

For the best operation of the algorithm, the theoretical scenario corresponds to a vehicle moving at a constant speed, otherwise, the reconstruction of the slices may show some deformation. However, as this cannot be controlled, the passage manager module performs a speed estimation to select the slice.

During the passage of the vehicle, the average distance of displacement of the object between the frames is calculated. This average distance indicates how many pixels an object moved at each passing frame and corresponds to the slice width. After defining the slice width, the central slice of each frame is concatenated horizontally, creating a complete image of the vehicle. Figure 3 presents how this approach is applied to building the final image.

Dataset

The dataset created for this project is based on mixed image data, containing real and synthetic images. The dataset was generated and annotated considering its use in different projects, because of this, different classes of vehicles were considered in the annotation stage. Only the wheel category labels will be considered in this

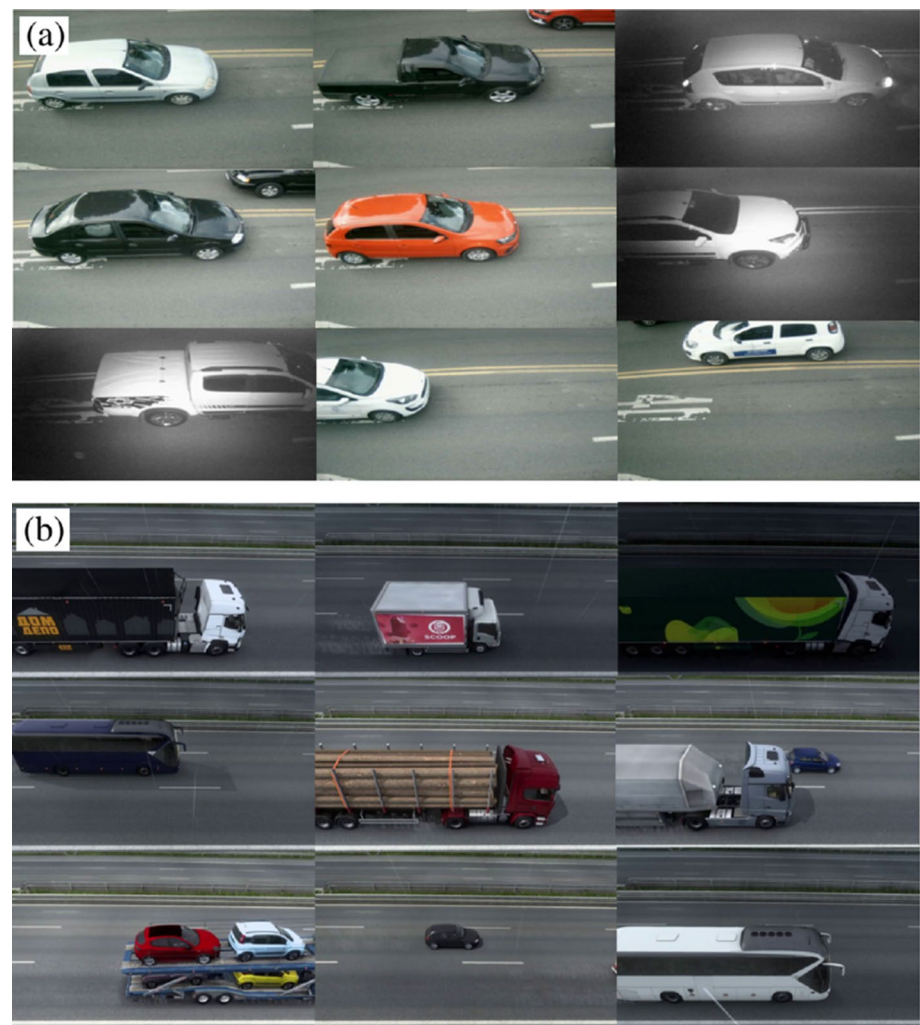


Figure 4. Representation of the mixed dataset: (a) real samples, (b) synthetic samples.

research. To have the database for training the object detection model, the company under evaluation installed a camera on a highway to simulate the real scenario of the application. With this, the acquisition of images of vehicles that passed in front of the camera was carried out.

Since detecting axles for large vehicles is part of the scope of this project, the images of bus and truck wheels must be available in the dataset. Due to the lower number of heavy vehicles on the highway, few samples were obtained. To increase the representativeness of the axles of these vehicles in the dataset, the selected alternative was to use synthetic images.

To create the synthetic data Euro Truck Simulator 2⁷⁰ was used. This simulator was used because it focuses on heavy vehicles such as trucks and buses. Another factor that led to the choice of this simulator was the possibility of camera adjustment. In this simulator is possible to modify the camera's view, making the captured image at an angle similar to the image of the real application. The use of Euro Truck Simulator 2 was considering the results presented in Ref.⁷¹, where the authors use this simulator for a vehicle detection application and show that the use of synthetic data can have a positive impact on the generalization of the detection model. Figure 4a,b show real and synthetic samples from the dataset. The division of the dataset is presented in Table 1.

Images	Training set	Validation set	Test set
Real	968	342	637
Synthetic	3061	685	1163
Total	4029	1027	1800

Table 1. Data distribution of the dataset.

Class	Training objects	Valid. objects	Test objects
Bus	300	99	154
Car	1386	419	876
Motorcycle	61	12	35
Truck	2751	570	1163
Total wheel	9755	2417	4693

Table 2. Distribution of classes in the dataset.

Focusing on a project feasibility study, it is necessary to analyze the influence caused by the difference between the camera angle of the real installation and the camera angle of the simulator, a fact that also generates a difference in the final size of the vehicles. Given this, only the axles will be considered in the detection task, and the proportion between the vehicles and their respective axles is visually close to the real proportions.

The differences in the proportion of vehicles allow the solution to be analyzed for different camera positions. The dataset is composed of the sum of the real and synthetic data, its distribution is presented in Table 2. All images (original and synthetic) were resized to 640×640 pixels to be standardized for the object detection task. The original images are not publicly available due to being from a private company. The dataset can be requested to the corresponding author on reasonable request. Examples of synthetic data can be found at: https://github.com/Bru-Souza/axles_dataset.

Experiment setup

In the evaluation presented in this paper, an Intel i7-13700K with a graphics processing unit RTX4090 (24GB), and 32GB of random-access memory was considered to perform the experiments. The proposed approach was written in Python language. In this paper, the precision, recall, and mAP metrics are considered.

Precision is defined by the number of positive detections divided by the total number of detections⁷². It is a measure of how often the model predicts correctly, and it indicates how much it's possible to rely on the model's positive predictions, given by:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

where TP are the true positives and FP are the false positives⁷³. The recall is used to evaluate if the model is having missing detections and is given by:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

where FN are the false negatives⁷⁴.

The Intersection over Union (IoU) provides information on the similarity between the region that the algorithm found and the real region of the object present in the image, being defined by the area of the intersection divided by the union of the area of the object and the area detected⁷⁵.

A true positive detection is defined by $IoU > T$, where T is a predefined threshold. In this research, the algorithms use $T = 0.5$. The mAP is calculated as the weighted mean of precisions at each threshold (given by the IoU), and the weight is the increase in recall from the prior threshold. The equation to calculate the mAP is according to:

$$mAP = \frac{1}{\Lambda} \sum_{i=1}^{\Lambda} AP_i \quad (3)$$

where Λ is the number of classes.

The mAP incorporates the trade-off between precision and recall. This property makes mAP a suitable metric for most detection applications. Performing the plot of the precision-recall curve is also a way to obtain the mAP. The Average Precision (AP) is defined as the area under the curve, and the mAP is defined as the average of AP of each class⁷⁶. To compare the computational effort of the models the Floating Point Operations (FLOPs) are considered. The considered hyperparameters and main definition setup to compute the experiments are presented in Table 3.

Results and discussions

In this section, the results will be presented and discussed, the experiments are divided into two parts. The first part aims to evaluate and compare the object detection models, while the second part presents the results regarding axle counting, which is the main focus of this paper.

Object detection analysis

The performance evaluation of the models is conducted considering a test set of images for this purpose. Table 4 presents a comparison of the results obtained after training the models. To evaluate the compared model, it was

Definition	Specification
Language	Python 3
Framework	PyTorch
Main libraries	cv2, numpy, ultralytics, torch
Image size	640×640
Max number epochs	100
Learning rate	0.01
Momentum	0.937
Warmup bias	0.1
Weight decay	0.0005
Optimizer	Stochastic gradient descent

Table 3. Hyperparameters and model setup.

Model	Precision	Recall	mAP @0.50	mAP @0.50:0.95	FLOPs
YOLOv5m	0.994	0.982	0.994	0.800	49.0
YOLOv6m	0.990	0.832	0.989	0.792	82.2
YOLOv7x	0.979	0.923	0.975	0.665	189.9
YOLOv8m	0.992	0.980	0.994	0.820	78.9

Table 4. Comparative model results. Best performance values are in bold.

Author	Application	Model	Precision	Recall	mAP @0.50	mAP @0.50:0.95
Stefenon et al. ⁵⁵	Insulator	YOLOv5u	0.981	0.975	0.983	0.905
Souza et al. ⁵⁶	Insulator	YOLOv5x	0.983	0.990	0.993	0.957
Han et al. ⁵⁸	Vehicle location	O-YOLO-v2	0.940	0.940	0.940	–
Padilla Carrasco et al. ⁵⁷	Vehicle location	T-YOLO	0.963	0.996	0.998	0.997
Li et al. ⁵³	Vehicle wheels	YOLOv3-SPP	0.718	–	0.955	–
Miles et al. ⁵²	Vehicle wheels	YOLOv3	0.927	–	0.927	–
Our	Vehicle wheels	YOLOv5m	0.994	0.982	0.994	0.800

Table 5. Other researchers’ results.

considered transfer learning and a fine-tuning process. The transfer learning is based on the pre-trained weights of the COCO dataset⁷⁷.

All evaluated models were able to achieve values above 0.97 for precision. The best result was obtained by YOLOv5m with a precision of 0.994, while the lowest precision was 0.979 achieved by YOLOv7x. The YOLOv5m achieved also the best performance regarding the recall which was 0.982. The YOLOv6m had the lowest recall in this evaluation.

The YOLOv5m and YOLOv8m models achieved the best mAP result (0.994), and in this measure, the YOLOv7x performed the worst (0.975). Taking into account a wider range of confidence thresholds, ranging from 0.50 to 0.95, the YOLOv8m model exhibited the best performance, reaching a value of 0.820, followed by YOLOv5m with 0.800 and YOLOv6m with 0.792.

The YOLOv5m model showed better performance in most of the metrics. Additionally, it has fewer parameters and fewer FLOPs when compared to the other models considered. Based on these results, the YOLOv5m was selected for axle identification. YOLO has been applied for several tasks by other researchers, some results comparing to our application are presented in Table 5.

When comparing the proposed method with other works that also use versions of the YOLO to detect vehicle axes, the camera angle ends up making it difficult to identify all axes since this isometric image ends up causing the occlusion of some axes, compromising the potential of the application. The lateral view of the vehicle is desired because, in addition to facilitating the reconstruction process of the complete vehicle, it generates the possibility of using information about the axle position, such as height and distance between axes, so that in future applications it can differentiate which axles are image belong to the same vehicle, and whether these axles are lowered, raised, or double-wheeled⁵².



Figure 5. Results from the reconstruction process.

Axle counting analysis

The axle counting analysis concerns the evaluation of the system that receives the sequence of frames and divides these frames into individual vehicle passes, the part that receives the separate frames and generates an image of the complete vehicle, and also the evaluation regarding the number of axles found on each vehicle.

Passage manager

The passage manager is the module that, through the optical flow algorithm, can identify and separate vehicle passages in a sequence of frames. The test was performed on real and synthetic images, and for this purpose, a video containing vehicle passages collected in the Euro Truck Simulator 2⁷⁰ simulator was created.

The video that was developed has a total of 30 vehicle passages, of which the system was able to correctly separate 26 passages, obtaining a total of 86.67% accuracy. Concerning the real data, 60 passages of different types of vehicles, such as cars, trucks, and buses were collected, and the system was able to correctly separate 46 passages, obtaining a total of 76.67% accuracy.

Reconstruction analysis

Based on the result of the passage manager module, a set of frames is obtained for a specific vehicle passage. These vehicle passages are then sent to the module responsible for slice reconstruction. Figure 5 presents examples where the algorithm can reconstruct the image without deformations that impair the visual analysis of the vehicle (using images from the Euro Truck Simulator 2⁷⁰).

Figure 6 presents the results of the slice reconstructor application in a real-world scenario. In this case, for the reconstruction algorithm to function properly, a preprocessing step is necessary on the frames to remove possible camera distortions.

In these examples, the synthetic images had more sharpness than the real images. This happened because it was a cloudy day when the data were recorded, based on this example it is possible to observe that real data have additionally the influence of weather conditions, that in some cases may impair axle identification.

Axle counter analysis

The results of applying the trained model to the reconstructed images can be observed in Figs. 7 and 8, where examples of application on synthetic and real data are shown, respectively. The object detector was able to successfully identify the axles of the vehicles, meeting the needs of this project.

Limitations: After the slice reconstruction process, although the height remains the same as the original, the width is defined by the slice width multiplied by the number of frames in the passage. If the image has larger dimensions than the images used to train the model, resizing it for inference can lead to errors in detection results.

Conclusions and future work

This study has addressed the challenges of implementing an axle counter system based on computer vision. A deep learning model was applied to identify axles, and after that, an algorithm for vehicle management and axle counting was implemented.

The computer vision-based systems have some limitations compared to the system proposed in this paper due to several outliers that may occur in the traffic scenario. The use of deep learning for axle counting proved



Figure 6. Results from the reconstruction in real images.



Figure 7. Axle counter inference.

capable of identifying vehicle axles. The YOLOv5m was selected after comparing this model to the YOLOv6m, YOLOv7x, and YOLOv8m models, where the YOLOv5 family model achieved a precision of 0.994 and a recall of 0.982 during the training process. Furthermore, the values of $mAP@0.50$ and $mAP@0.50:0.95$ were 0.994 and 0.800, respectively.

The passage manager module and the slice reconstruction module were developed, enabling the determination of vehicle passages in the scenario and creating an image containing the complete vehicle for wheel counting. Regarding the outcomes of the passage manager obtained during the experiments, the proposed approach had an accuracy rate of 86.67% for synthetic data, correctly identifying 26 out of 30 passage samples, and 76.67% for real-world images, accurately classifying 46 out of 60 passages. These results underscore the effectiveness of the implemented system in reliably managing and processing axle passages, both in synthetic and real-world scenarios.

To enhance the performance of the axle counter, in future works, an alternative would be to include images of the reconstructed vehicle in the training dataset. It would be promising to incorporate images of vehicles in different backgrounds to obtain a more generalizable model. A crucial aspect to improve the robustness of the application is to develop a strategy for handling cases in which one vehicle is carrying another, such as tow and stork trucks.



Figure 8. Axle counter inference in real images.

Future work can also explore other situations, such as traffic congestion, where there is minimal or no displacement of vehicles between two consecutive frames, or scenarios involving a vehicle and a motorcycle passing simultaneously in a single lane. The tariff collection for trucks currently has differences for axles that are lowered, raised, and double-wheeled, factors that can be approached in future works.

Data availability

The datasets generated and/or analysed during the current study are not publicly available due to being from a private company but are available from the corresponding author on reasonable request.

Received: 25 October 2023; Accepted: 5 February 2024

Published online: 10 February 2024

References

- Barbosa, H. *et al.* Human mobility: Models and applications. *Phys. Rep.* **734**, 1–74. <https://doi.org/10.1016/j.physrep.2018.01.001> (2018).
- Damaj, I. *et al.* Intelligent transportation systems: A survey on modern hardware devices for the era of machine learning. *J. King Saud Univ. Comput. Inform. Sci.* **34**(8), 5921–5942. <https://doi.org/10.1016/j.jksuci.2021.07.020> (2022).
- Jabbar, R. *et al.* Blockchain technology for intelligent transportation systems: A systematic literature review. *IEEE Access* **10**, 20995–21031. <https://doi.org/10.1109/ACCESS.2022.3149958> (2022).
- Ribaric, S., Ariyaeinia, A. & Pavesic, N. De-identification for privacy protection in multimedia content: A survey. *Signal Process. Image Commun.* **47**, 131–151. <https://doi.org/10.1016/j.image.2016.05.020> (2016).
- Kim, H. *et al.* A systematic review of the smart energy conservation system: From smart homes to sustainable smart cities. *Renew. Sustain. Energy Rev.* **140**, 110755. <https://doi.org/10.1016/j.rser.2021.110755> (2021).
- Chibber, A., Anand, R., & Singh, J., *et al.* Smart traffic light controller using edge detection in digital signal processing. in *Wireless Communication with Artificial Intelligence*, 251–272. (CRC Press, 2022). <https://doi.org/10.1201/9781003230526>.
- Lee, C. P., Leng, F. T. J., Habeeb, R. A. A., Amanullah, M. A. & Rehman, M. H. Edge computing-enabled secure and energy-efficient smart parking: A review. *Microprocess. Microsyst.* <https://doi.org/10.1016/j.micpro.2022.104612> (2022).
- Friedanelli, L. *et al.* Traffic flow detection using camera images and machine learning methods in its for noise map and action plan optimization. *Sensors* **22**(5), 1929. <https://doi.org/10.3390/s22051929> (2022).
- Bari, C., Gupta, U., Chandra, S., Antoniou, C., & Dhamaniya, A. Examining effect of electronic toll collection (etc) system on queue delay using microsimulation approach at toll plaza—A case study of Ghoti Toll Plaza, India. in *International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, vol. 7, 1–6 (2021). <https://doi.org/10.1109/MT-ITS49943.2021.9529325>.
- Singhal, N. & Prasad, L. Sensor based vehicle detection and classification—A systematic review. *Int. J. Eng. Syst. Model. Simulat.* **13**(1), 38–60. <https://doi.org/10.1504/IJESMS.2022.122731> (2022).
- Wong, Z. J., Goh, V. T., Yap, T. T. V., & Ng, H. Vehicle classification using convolutional neural network for electronic toll collection. in *Computational Science and Technology: 6th ICCST 2019, Kota Kinabalu, Malaysia*, 29–30 August 2019, 169–177. (Springer, 2020). https://doi.org/10.1007/978-981-15-0058-9_17.
- Ahmed, S., Tan, T. M., Mondol, A. M., Alam, Z., Nawal, N., & Uddin, J. Automated toll collection system based on RFID sensor. in *International Carnahan Conference on Security Technology (ICCST)*, 1–3. (IEEE, 2019). <https://doi.org/10.1109/CCST.2019.8888429>.
- Hari Charan, E. V. V., Pal, I., Sinha, A., Baro, R. K. R., & Nath, V. Electronic toll collection system using barcode technology. in Nath, V., Mandal, J.K. (eds.) *Nanoelectronics, Circuits and Communication Systems*, 549–556. (Springer, 2019). https://doi.org/10.1007/978-981-13-0776-8_51.
- Sun, Z., Wang, P., Wang, J., Peng, X. & Jin, Y. Exploiting deeply supervised inception networks for automatically detecting traffic congestion on freeway in China using ultra-low frame rate videos. *IEEE Access* **8**, 21226–21235. <https://doi.org/10.1109/ACCESS.2020.2968597> (2020).

15. Zhou, W., Gao, S., Zhang, L. & Lou, X. Histogram of oriented gradients feature extraction from raw Bayer pattern images. *IEEE Trans. Circ. Syst. II Express Briefs* **67**(5), 946–950. <https://doi.org/10.1109/TCSII.2020.2980557> (2020).
16. Puspita, Z. G., Novamizanti, L., Rachmawati, E., & Nasari, M. Fuzzy local binary pattern and weber local descriptor for facial emotion classification. in *International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, 6–11 (2021). <https://doi.org/10.1109/ICICyTA53712.2021.9689087>.
17. Fernandes, A. M. D. R. *et al.* Detection and classification of cracks and potholes in road images using texture descriptors. *J. Intell. Fuzzy Syst.* **44**(6), 10255–10274. <https://doi.org/10.3233/JIFS-223218> (2023).
18. Stefenon, S. F. *et al.* Fault detection in insulators based on ultrasonic signal processing using a hybrid deep learning technique. *IET Sci. Meas. Technol.* **14**(10), 953–961. <https://doi.org/10.1049/iet-smt.2020.0083> (2020).
19. Sopelsa Neto, N. F. *et al.* A study of multilayer perceptron networks applied to classification of ceramic insulators using ultrasound. *Appl. Sci.* **11**(4), 1592. <https://doi.org/10.3390/app11041592> (2021).
20. Stefenon, S. F., Silva, M. C., Bertol, D. W., Meyer, L. H. & Nied, A. Fault diagnosis of insulators from ultrasound detection using neural networks. *J. Intell. Fuzzy Syst.* **37**(5), 6655–6664. <https://doi.org/10.3233/JIFS-190013> (2019).
21. Corso, M. P. *et al.* Classification of contaminated insulators using k-nearest neighbors based on computer vision. *Computers* **10**(9), 112. <https://doi.org/10.3390/computers10090112> (2021).
22. Stefenon, S. F. *et al.* Photovoltaic power forecasting using wavelet neuro-fuzzy for active solar trackers. *J. Intell. Fuzzy Syst.* **40**(1), 1083–1096. <https://doi.org/10.3233/JIFS-201279> (2021).
23. Yamasaki, M. *et al.* Optimized hybrid ensemble learning approaches applied to very short-term load forecasting. *Int. J. Electr. Power Energy Syst.* **155**, 109579. <https://doi.org/10.1016/j.ijepes.2023.109579> (2024).
24. Stefenon, S. F. *et al.* Group method of data handling using Christiano-Fitzgerald random walk filter for insulator fault prediction. *Sensors* **23**(13), 6118. <https://doi.org/10.3390/s23136118> (2023).
25. Stefenon, S. F. *et al.* Electric field evaluation using the finite element method and proxy models for the design of stator slots in a permanent magnet synchronous motor. *Electronics* **9**(11), 1975. <https://doi.org/10.3390/electronics9111975> (2020).
26. Starke, L. *et al.* Interference recommendation for the pump sizing process in progressive cavity pumps using graph neural networks. *Sci. Rep.* **13**(1), 16884. <https://doi.org/10.1038/s41598-023-43972-4> (2023).
27. Westarb, G. *et al.* Complex graph neural networks for medication interaction verification. *J. Intell. Fuzzy Syst.* **44**(6), 10383–10395. <https://doi.org/10.3233/JIFS-223656> (2023).
28. Wang, X., Gao, H., Jia, Z. & Li, Z. BL-YOLOv8: An improved road defect detection model based on YOLOv8. *Sensors* **23**(20), 8361. <https://doi.org/10.3390/s23208361> (2023).
29. Branco, N. W., Cavalca, M. S. M., Stefenon, S. F. & Leithardt, V. R. Q. Wavelet LSTM for fault forecasting in electrical power grids. *Sensors* **22**(21), 8323. <https://doi.org/10.3390/s22218323> (2022).
30. Moreno, S. R., Mariani, V. C. & dos Santos Coelho, L. Hybrid multi-stage decomposition with parametric model applied to wind speed forecasting in Brazilian northeast. *Renew. Energy* **164**, 1508–1526 (2021).
31. Ribeiro, M. H. D. M., Stefenon, S. F., de Lima, J. D., Nied, A., Mariani, V. C., & Coelho, L. D. S. Electricity price forecasting based on self-adaptive decomposition and heterogeneous ensemble learning. *Energies* **13**(19), 5190. <https://doi.org/10.3390/en13195190> (2020).
32. Moreno, S. R., Seman, L. O., Stefenon, S. F., dos Santos Coelho, L. & Mariani, V. C. Enhancing wind speed forecasting through synergy of machine learning, singular spectral analysis, and variational mode decomposition. *Energy* **292**, 130493. <https://doi.org/10.1016/j.energy.2024.130493> (2024).
33. Stefenon, S. F. *et al.* Classification of insulators using neural network based on computer vision. *IET Generat. Transmission Distribution* **16**(6), 1096–1107. <https://doi.org/10.1049/gtd.2.12353> (2021).
34. dos Santos, G. H. *et al.* Static attitude determination using convolutional neural networks. *Sensors* **21**(19), 6419. <https://doi.org/10.3390/s21196419> (2021).
35. Borré, A. *et al.* Machine fault detection using a hybrid CNN-LSTM attention-based model. *Sensors* **23**(9), 4512. <https://doi.org/10.3390/s23094512> (2023).
36. Vieira, J. C. *et al.* Low-cost CNN for automatic violence recognition on embedded system. *IEEE Access* **10**, 25190–25202. <https://doi.org/10.1109/ACCESS.2022.3155123> (2022).
37. Stefenon, S. F., Yow, K.-C., Nied, A. & Meyer, L. H. Classification of distribution power grid structures using inception v3 deep neural network. *Electr. Eng.* **104**, 4557–4569. <https://doi.org/10.1007/s00202-022-01641-1> (2022).
38. Dai, Y., Liu, W., Wang, H., Xie, W. & Long, K. Yolo-former: Marrying yolo and transformer for foreign object detection. *IEEE Trans. Instrument. Meas.* **71**, 1–14. <https://doi.org/10.1109/TIM.2022.3219468> (2022).
39. Gu, J. *et al.* Recent advances in convolutional neural networks. *Pattern Recognit.* **77**, 354–377. <https://doi.org/10.1016/j.patcog.2017.10.013> (2018).
40. Chen, S., & Lin, W. Embedded system real-time vehicle detection based on improved YOLO network. in *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 3, 1400–1403 (2019). <https://doi.org/10.1109/IMCEC46724.2019.8984055>.
41. Glasenapp, L. A., Hoppe, A. F., Wisintainer, M. A., Sartori, A. & Stefenon, S. F. OCR applied for identification of vehicles with irregular documentation using IoT. *Electronics* **12**(5), 1083. <https://doi.org/10.3390/electronics12051083> (2023).
42. Bao, C., Xie, T., Feng, W., Chang, L. & Yu, C. A power-efficient optimizing framework FPGA accelerator based on winograd for YOLO. *IEEE Access* **8**, 94307–94317. <https://doi.org/10.1109/ACCESS.2020.2995330> (2020).
43. Ge, L., Dan, D. & Li, H. An accurate and robust monitoring method of full-bridge traffic load distribution based on YOLO-v3 machine vision. *Struct. Control Health Monit.* **27**(12), 2636. <https://doi.org/10.1002/stc.2636> (2020).
44. Rajput, S. K. *et al.* Automatic vehicle identification and classification model using the YOLOv3 algorithm for a toll management system. *Sustainability* **14**(15), 9163. <https://doi.org/10.3390/su14159163> (2022).
45. Song, H., Liang, H., Li, H., Dai, Z. & Yun, X. Vision-based vehicle detection and counting system using deep learning in highway scenes. *Eur. Transp. Res. Rev.* **11**, 51. <https://doi.org/10.1186/s12544-019-0390-4> (2019).
46. Zhao, J. *et al.* Improved vision-based vehicle detection and classification by optimized YOLOv4. *IEEE Access* **10**, 8590–8603. <https://doi.org/10.1109/ACCESS.2022.3143365> (2022).
47. Fan, R., Bocus, M. J., Zhu, Y., Jiao, J., Wang, L., Ma, F., Cheng, S., & Liu, M. Road crack detection using deep convolutional neural network and adaptive thresholding. in *IEEE Intelligent Vehicles Symposium (IV)*, Paris, France, 474–479 (2019). <https://doi.org/10.1109/IVS.2019.8814000>.
48. Kulkarni, R., Dhavalikar, S., & Bangar, S. Traffic light detection and recognition for self driving cars using deep learning. in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1–4. (IEEE, 2018). <https://doi.org/10.1109/ICCUBEA.2018.8697819>.
49. Zarei, N., Moallem, P. & Shams, M. Fast-yolo-rec: Incorporating yolo-base detection and recurrent-base prediction networks for fast vehicle detection in consecutive images. *IEEE Access* **10**, 120592–120605. <https://doi.org/10.1109/ACCESS.2022.3221942> (2022).
50. Carrasco, D. P., Rashwan, H. A., García, M. Á. & Puig, D. T-yolo: Tiny vehicle detection based on yolo and multi-scale convolutional neural networks. *IEEE Access* **11**, 22430–22440. <https://doi.org/10.1109/ACCESS.2021.3137638> (2023).
51. Dewi, C., Chen, R.-C., Liu, Y.-T., Jiang, X. & Hartomo, K. D. Yolo v4 for advanced traffic sign recognition with synthetic training data generated by various gan. *IEEE Access* **9**, 97228–97242. <https://doi.org/10.1109/ACCESS.2021.3094201> (2021).

52. Miles, V., Gurr, F. & Giani, S. Camera-based system for the automatic detection of vehicle axle count and speed using convolutional neural networks. *Int. J. Intell. Transport. Syst. Res.* **20**(3), 778–792. <https://doi.org/10.1007/s13177-022-00325-1> (2022).
53. Li, L., Wu, J., Luo, H., Wu, R., & Li, Z. (2021). A video axle counting and type recognition method based on improved YOLOv5s. in *Data Mining and Big Data: 6th International Conference, DMBD 2021*, Guangzhou, China, October 20–22, 2021, Proceedings, Part 1 6, 158–168. https://doi.org/10.1007/978-981-16-7476-1_15.
54. Wei, L., Zhong, Z., Lang, C. & Yi, Z. A survey on image and video stitching. *Virtual Reality Intell. Hardw.* **1**(1), 55–83. <https://doi.org/10.3724/SPJ.2096-5796.2018.0008> (2019).
55. Stefenon, S. F., Singh, G., Souza, B. J., Freire, R. Z. & Yow, K.-C. Optimized hybrid YOLOu-Quasi-ProtoPNet for insulators classification. *IET Generation Transmission Distribution* **17**(15), 3501–3511. <https://doi.org/10.1049/gtd2.12886> (2023).
56. Souza, B. J., Stefenon, S. F., Singh, G. & Freire, R. Z. Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV. *Int. J. Electr. Power Energy Syst.* **148**, 108982. <https://doi.org/10.1016/j.ijepes.2023.108982> (2023).
57. Padilla Carrasco, D., Rashwan, H. A., Garcia, M. A. & Puig, D. T-YOLO: Tiny vehicle detection based on YOLO and multi-scale convolutional neural networks. *IEEE Access* **11**, 22430–22440. <https://doi.org/10.1109/ACCESS.2021.3137638> (2023).
58. Han, X., Chang, J. & Wang, K. Real-time object detection based on yolo-v2 for tiny vehicle object. *Procedia Comput. Sci.* **183**, 61–72. <https://doi.org/10.1016/j.procs.2021.02.031> (2021) (Proceedings of the 10th International Conference of Information and Communication Technology).
59. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. You only look once: Unified, real-time object detection. in *IEEE Conference on Computer Vision and Pattern Recognition*, 779–788 (2016). <https://doi.org/10.1109/CVPR.2016.91>.
60. Ali, F., Khan, S., Abbas, A. W., Shah, B., Hussain, T., Song, D., Ei-Sappagh, S., & Singh, J. A two-tier framework based on googlenet and YOLOv3 models for tumor detection in MRI. *Comput. Mater. Continua.* (2022). <https://doi.org/10.32604/cmc.2022.024103>.
61. Mantau, A. J., Widayat, I. W., Leu, J.-S. & Köppen, M. A human-detection method based on YOLOv5 and transfer learning using thermal image data from UAV perspective for surveillance system. *Drones* **6**(10), 290. <https://doi.org/10.3390/drones6100290> (2022).
62. Murthy, J. S. *et al.* Objectdetect: A real-time object detection framework for advanced driver assistant systems using YOLOv5. *Wireless Commun. Mobile Comput.* **2022**, 9444360. <https://doi.org/10.1155/2022/9444360> (2022).
63. Singh, G., Stefenon, S. F. & Yow, K.-C. Interpretable visual transmission lines inspections using pseudo-prototypical part network. *Machine Vision Appl.* **34**(3), 41. <https://doi.org/10.1007/s00138-023-01390-6> (2023).
64. Zheng, Y., Zhan, Y., Huang, X. & Ji, G. YOLOv5s FMG: An improved small target detection algorithm based on YOLOv5 in low visibility. *IEEE Access* **11**, 75782–75793. <https://doi.org/10.1109/ACCESS.2023.3297218> (2023).
65. Huang, L., Huang, W., Gong, H., Yu, C. & You, Z. PEFNet: Position enhancement faster network for object detection in roadside perception system. *IEEE Access* **11**, 73007–73023. <https://doi.org/10.1109/ACCESS.2023.3292881> (2023).
66. Zhang, L. *et al.* Improved object detection method utilizing yolov7-tiny for unmanned aerial vehicle photographic imagery. *Algorithms* **16**(11), 520. <https://doi.org/10.3390/a16110520> (2023).
67. Liu, S., Wang, Y., Yu, Q., Liu, H. & Peng, Z. CEAM-YOLOv7: Improved YOLOv7 based on channel expansion and attention mechanism for driver distraction behavior detection. *IEEE Access* **10**, 129116–129124. <https://doi.org/10.1109/ACCESS.2022.3228331> (2022).
68. Terven, J., Córdova-Esparza, D.-M. & Romero-González, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learn. Knowl. Extraction* **5**(4), 1680–1716. <https://doi.org/10.3390/make5040083> (2023).
69. Karna, N. B. A., Putra, M. A. P., Rachmawati, S. M., Abisado, M. & Sampedro, G. A. Toward accurate fused deposition modeling 3d printer fault detection using improved YOLOv8 with hyperparameter optimization. *IEEE Access* **11**, 74251–74262. <https://doi.org/10.1109/ACCESS.2023.3293056> (2023).
70. SCS Software: Euro Truck Simulator . <https://eurotrucksimulator2.com/>. Online; accessed 25 January 2024.
71. Cecchetti, V. B., Souza, B. J., & Freire, R. Z. Framework for automated synthetic image generation for vehicle detection. in *2023 International Conference on Control, Robotics Engineering and Technology*, vol. 1 (2024).
72. Surek, G. A. S., Seman, L. O., Stefenon, S. F., Mariani, V. C. & Coelho, L. S. Video-based human activity recognition using deep learning approaches. *Sensors* **23**(14), 6384. <https://doi.org/10.3390/s23146384> (2023).
73. Qian, S. *et al.* Detecting taxi trajectory anomaly based on spatio-temporal relations. *IEEE Trans. Intell. Transport. Syst.* **23**(7), 6883–6894. <https://doi.org/10.1109/TITS.2021.3063199> (2022).
74. Corso, M. P. *et al.* Evaluation of visible contamination on power grid insulators using convolutional neural networks. *Electr. Eng.* **105**, 3881–3894. <https://doi.org/10.1007/s00202-023-01915-2> (2023).
75. Rahman, M. A. & Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. *Int. Symp. Visual Comput.* **10072**, 234–244. https://doi.org/10.1007/978-3-319-50835-1_22 (2016).
76. Yao, J., Li, Y., Yang, B. & Wang, C. Learning global image representation with generalized-mean pooling and smoothed average precision for large-scale cbir. *IET Image Process.* **17**(9), 2748–2763. <https://doi.org/10.1049/ipr2.12825> (2023).
77. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. Microsoft coco: Common objects in context. in *Computer Vision—ECCV 2014: European Conference*, vol. 13, 740–755 (2014). https://doi.org/10.1007/978-3-319-10602-1_48.

Author contributions

B.J.S. wrote the original draft and performed the computation; A.L.S., R.Z.F., and G.V.G. contributed to the review, editing, and supervision.

Funding

This work was supported by caracterización, análisis e intervención en la prevención de riesgos laborales en entornos de trabajo tradicionales mediante la aplicación de tecnologías disruptivas—Consejería de Empleo e Industria—J125.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to B.J.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024