A Deep Learning Model for Heavy Vehicle Classification Based on Silhouette

Daniel José Cerqueira Brito
CETEC – PaveLab
UFRB
Cruz das Almas, Brazil
ORCID: 0009-0006-4018-255X

Acbal Rucas Andrade Achy
PPGEEC – UFRB
UFRB
Cruz das Almas, Brazil
ORCID: 0000-0002-4063-0774

Brenner dos Santos Araujo
CETEC – PaveLab
UFRB
Cruz das Almas, Brazil
ORCID: 0009-0001-9422-961X

Tiago Palma Pagano
PPGEEC – UFRB
UFRB
Cruz das Almas, Brazil
ORCID: 0000-0003-2457-9064

Weiner Gustavo Silva Costa
CETEC – PaveLab
UFRB
Cruz das Almas, Brasil
ORCID: 0000-0001-6151-8805

Mario Sergio Souza de Almeida
CETEC – PaveLab
UFRB
Cruz das Almas, Brazil
ORCID: 0000-0002-0222-4804

***Abstract*—This paper presents a deep learning model for silhouette-based multiclass heavy vehicle classification using computer vision and machine learning techniques. The objective of this work is to identify heavy vehicle categories from images by their number of axles, as specified by the DNIT guidelines. The model generates reliable and scalable data to support road planning, pavement design, and predictive maintenance of the road network. To achieve this, our proposed methodology combines a pre-trained YOLO model for automated image collection and a ResNet-50 model, fine-tuned via transfer learning. The model's practical application contributes to highway planning and maintenance by improving the accuracy of pavement load impact prediction. It achieved an Area Under the Precision-Recall Curve (AUC-PRC) of 97% and a validation loss of 0.15, even when faced with a limited and unbalanced dataset.**

***Index Terms*—Vehicle classification, machine learning, transfer learning, YOLO, ResNet-50, AUC-PRC, highways.**

I. Introduction

Vehicle classification is a fundamental tool for intelligent transportation systems (ITS), supporting traffic monitoring, infrastructure management, and public policy for road preservation. The rapid evolution of ITS has seen a shift from earlier methods to the prevalent use of video image analysis for managing complex traffic environments [1], [2]. Effective traffic management is a critical challenge, particularly in urban areas where high vehicle volume leads to economic losses, increased accidents, and environmental concerns [3]. In this context, accurate vehicle counting and detailed classification are paramount, providing essential data to optimize road pavement design and maintenance, as per DNIT guidelines [4], while also supporting public security and dynamic traffic management [1]–[3].

Traditionally, traffic is quantified via manual field counts during sample periods. From this data, annual and decennial traffic estimates are made based on projections that may diverge from local reality, compromising the accuracy of engineering projects. Consequently, automating this process with computer vision is a promising alternative. However, automated video analysis faces inherent difficulties, including shadows, vehicle occlusion, and adverse conditions like rain or fog [3]. These techniques are often confined to specific scenarios, frequently failing with complex backgrounds or camera instability, highlighting that robust and generalizable vehicle counting remains an open problem under real-world conditions [3].

These challenges underscore the need for more sophisticated approaches. The advancement of artificial intelligence (AI) and deep learning (DL) has made it feasible to develop robust models for real-time vehicle detection and classification from images. These advanced methods enable not only basic detection but also detailed vehicle characterization. For instance, fine-grained classification efforts aim to identify specific attributes (e.g., make and model), providing granular data for various ITS applications [1], [2]. Transfer learning with pre-trained Convolutional Neural Network (CNN) architectures has also proven effective for accurate models, even with limited datasets [5]–[7].

This work proposes an automated system for classifying heavy vehicles (trucks and buses) based on their silhouette. The system uses a pipeline combining YOLOv8 (version 8) detection [8] and multiclass classification with a ResNet-50 architecture [9], [10]. Our objective is to identify heavy vehicles by their number of axles, following DNIT guidelines [4], to generate reliable and scalable data for road planning and maintenance. While existing studies have explored various aspects of vehicle classification, a specific and robust solution for silhouette-based heavy vehicle classification by axle count, tailored to local regulatory guidelines, remains underexplored. Our research, therefore, offers a novel pipeline to address this gap.

This paper is organized as follows: Section II presents a review of the state of the art; Section III describes the proposed methodology; Section IV details and discusses the obtained results; and finally, Section V presents the conclusions and suggestions for future work.

II. State of the Art

Vehicle identification and counting using computer vision and artificial intelligence have been widely explored. The application of Convolutional Neural Networks (CNNs) with preprocessing and feature extraction techniques has shown particularly promising results in this field.

Souza Almeida et al. [11] proposed a methodology for continuous highway traffic monitoring, employing CNNs for vehicle identification. They highlighted the importance of silhouette-based counting for calibrating the DNIT's MeDiNa method, which is a central aspect of our work. However, a dedicated and automated system for multiclass heavy vehicle classification based strictly on DNIT axle count guidelines remains an area needing further development. For real-time object detection, the *YOLO (You Only Look Once)* algorithm [12], reviewed by Nazir et al. [13], represented a revolution by unifying localization and classification, and its fast approach was crucial for extracting vehicle silhouettes in this study.

Expanding on the use of YOLO, Kejriwal et al. [3] developed a detection system for urban traffic. While their automatic acquisition and detection methodology served as a reference, such systems often do not address the fine-grained, axle-based classification of heavy vehicles required for infrastructure planning. They also often fail to fully handle the specific challenges of varying highway conditions for silhouette analysis. To address these limitations, robust architectures are necessary.

He et al. [9] introduced *ResNet (Deep Residual Network)*, chosen for its balance between depth and efficiency. Its performance is often boosted by pre-training on large datasets like ImageNet [14], a technique similar to that used by Azizi et al. [15]. We used an ImageNet pre-trained ResNet-50, which is a widely adopted architecture [10], enabling high-performance transfer learning even with reduced datasets.

Other approaches for classification include the proposal by Shi and Liu [16], who used Fast R-CNN with preprocessing enhancements for scenarios with occlusion, and Hasanvand et al. [17], who emphasized feature engineering with a system based on morphological features and various classifiers.

While these studies demonstrate strong performance, they often do not focus specifically on the unique challenges of fine-grained heavy vehicle classification by axle configuration from silhouette images, which is critical for specialized applications like pavement design according to national guidelines.

These studies provide a solid foundation for our work. However, despite advancements in generic detection, a significant gap remains in developing an automated and robust system specifically for silhouette-based multiclass classification of heavy vehicles according to their axle count, as stipulated by DNIT guidelines. Our work directly addresses this need with a novel pipeline that leverages YOLO for data collection and ResNet-50 with transfer learning for precise classification. To the best of our knowledge, no works were found with an identical approach, indicating that this study offers highly relevant and original contributions to the field of computer vision applied to road infrastructure.

III. Methodology

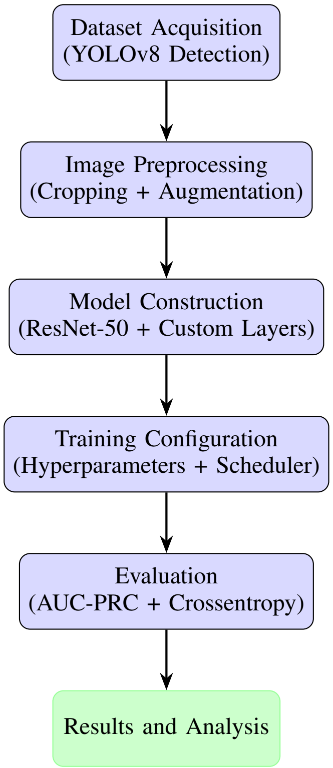


Figure 1: Flowchart representing the proposed methodology. Source: Own authorship.

The methodological workflow of this study, as illustrated in Figure 1, is structured into five sequential stages. It begins with dataset acquisition (Section III-A), where traffic images are collected and objects are detected using YOLOv8. In the image preprocessing phase (Section III-B), techniques such as cropping and data augmentation are applied to refine the input data. Next, the model construction stage (Section III-C) involves the assembly of a classification model based on ResNet-50 with additional custom layers. The fourth stage, training configuration (Section III-D), defines the learning process through the adjustment of hyperparameters and scheduling strategies. Finally, the evaluation step (Section III-E) measures the model's performance using metrics such as AUC-PRC and categorical crossentropy, guiding the analysis and interpretation of results.

A. Dataset Acquisition

Dataset acquisition was performed automatically through a detection pipeline based on the YOLOv8 model [8], pre-trained with COCO dataset weights. Detection was applied to real highway traffic videos from the BR-110, where a vehicle's silhouette was automatically cropped from the scene. This procedure ensured data capture was unsupervised, consistent, and highly scalable.

A total of 5,790 images were collected and divided into four classes of heavy commercial vehicles, categorized by their silhouette and number of axles, as per the official DNIT (2006) classification [4]. The choice of these four categories aimed to reduce the initial problem complexity, given the limited samples, while representing the main load profiles that impact pavement design.

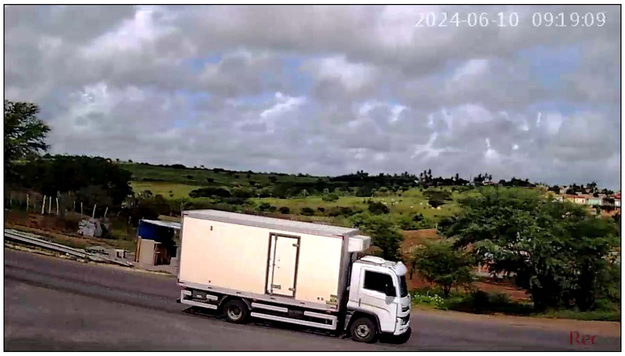


Figure 2: Example of a captured image for a 2C (2-axle) truck. Source: Own authorship.

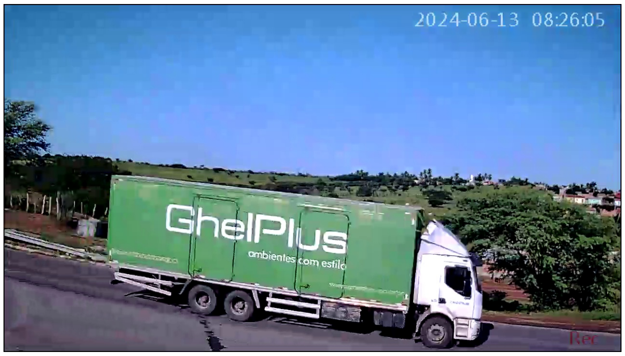


Figure 3: Example of a captured image for a 3C (3-axle) truck. Source: Own authorship.

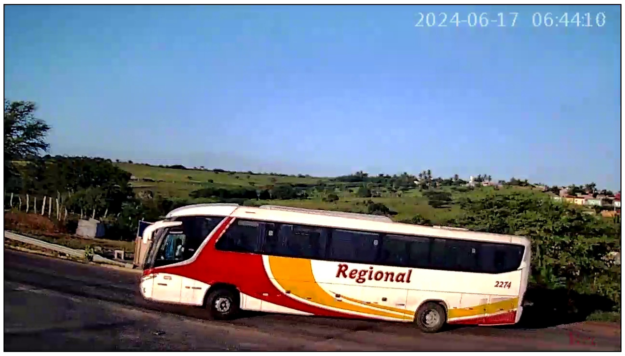


Figure 4: Example of a captured image for a 2CB (2-axle) bus. Source: Own authorship.

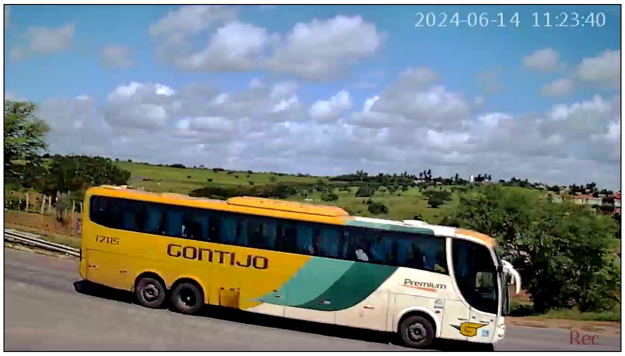
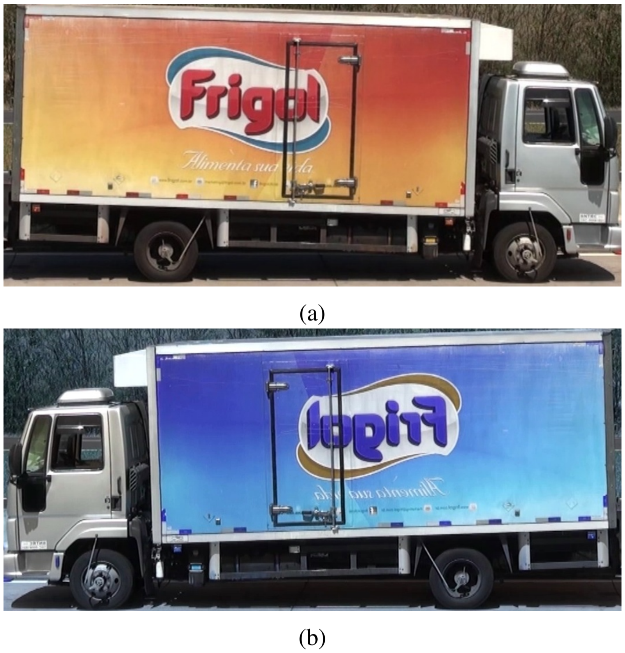


Figure 5: Example of a captured image for a 3CB (3-axle) bus. Source: Own authorship.

B. Image Preprocessing

Image preprocessing was performed in two main stages to optimize the input data and enhance model robustness. The first stage was automatic cropping of the YOLO-detected vehicle to remove irrelevant background (Figure 6a). The second involved data augmentation techniques [18] to increase dataset diversity, including resizing images to 224 × 224 pixels, horizontal flipping, brightness adjustments, and color channel shifts (Figure 6b).



(a)

(b)

Figure 6: Examples of the preprocessing pipeline: (a) A vehicle after automatic cropping and (b) The same image after augmentation (horizontal flip and channel shifts range). Source: Own authorship.

C. Model Architecture

The model was built based on the ResNet-50 architecture [9], a 50-layer deep residual network that uses shortcuts (skip connections) to mitigate the vanishing gradient problem in very deep networks. ResNet-50 is a deep, efficient, and widely adopted architecture for large-scale image classification tasks, with approximately 25.6 million parameters.

The original architecture begins with a 7 × 7 convolutional layer and a *MaxPooling* operation. Its core structure is composed of 4 main stages containing multiple *bottleneck* blocks, totaling 48 convolutional layers. Between these layers, *Batch Normalization* [19] [20] operations and *ReLU* [21] [22] activation functions are applied to provide training stability and network non-linearity. The network concludes with a *Global Average Pooling* [23] [24] layer before the final output.

In this implementation, ResNet-50 with ImageNet pre-trained weights was used. To adapt the model for heavy vehicle classification, custom dense and regularization layers were added, as described in **Table I**. During the fine-tuning [6] [25] process, approximately 80% of the ResNet-50 layers were frozen, allowing the training to focus on the latter layers to specialize the model for the four vehicle classes. The dataset was split into 90% for training and 10% for validation, and the final output layer uses a softmax [26] [27] activation function.

D. Training Parameters

During model training, the following hyperparameters were used: a batch size of 16 and a maximum of 100 epochs, with training automatically interrupted by callbacks if necessary.

The Adam optimizer [28] [29] was used to update the network weights during training. This optimizer combines the advantages of AdaGrad and RMSProp, adjusting parameters based on moving averages of the gradient and its variance, following the update rule in Equation 1.

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{1}$$

Where θ_t represents the updated weights at epoch t, η is the learning rate, $\hat{m}_t$ is the biased first moment estimate (average of gradients), $\hat{v}_t$ is the biased second raw moment estimate (average of squared gradients), and ϵ is a small constant for numerical stability (e.g., 10^{-8}).

An initial learning rate [30] [31] of 1×10^{-5} was used, which helps ensure stable learning and avoids large jumps that could hinder convergence. The Early Stopping [32] [33] technique, with a patience of $p = 60$, was also employed to prevent overfitting and reduce computational time by terminating training if the validation loss did not improve.

A Learning Rate Scheduler with *cosine annealing* decay, as defined by Equation 2, was employed to dynamically adjust the learning rate during training.

$$\eta_t = \eta_0 \cdot \frac{1 + \cos\left(\frac{\pi \cdot t}{2 \cdot T}\right)}{2} \tag{2}$$

Where:

- η_0 is the initial learning rate (defined as 1×10^{-5} in this work);
- t is the current epoch index;
- T is the total number of training epochs (defined as 100).

This decay approach promotes larger weight updates at the beginning of training, when the model is far from the optimum, and smaller updates as training progresses, facilitating convergence to a good local minimum.

E. Evaluation Metrics

Model performance evaluation was based on two main metrics: the Categorical Crossentropy [34] [35] loss function and the *AUC-PRC (Area Under the Precision-Recall Curve)* [36] [37]. Both were chosen for their suitability for multiclass classification tasks with unbalanced data, a selection consistent with practices in similar deep learning studies addressing imbalanced datasets.

The Categorical Crossentropy loss function (Equation 3) measures the divergence between the model's predicted distribution and the actual distribution of classes. It was used to quantify the model's error, with the objective of minimization during training. A lower loss value indicates a better model performance on the classification task.

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \cdot \log(p_i) \tag{3}$$

Where:

- C is the total number of classes;
- y_i is the true value (0 or 1) for class i;
- p_i is the predicted probability for class i.

The Area Under the Precision-Recall Curve (AUC-PRC) was adopted as the main performance metric, especially considering class imbalance. AUC-PRC is appropriate for scenarios with unbalanced classes as it considers the relationship between *Precision* (Equation 4) and *Recall* (Equation 5), which are used to construct the Precision-Recall Curve (PRC).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

Where:

- TP = True Positives;
- FP = False Positives;
- FN = False Negatives.

The AUC-PRC value, defined by Equation 6, indicates excellent performance when it is close to 1.

$$\text{AUC-PRC} = \int_0^1 \text{Precision}(r)\, dr \tag{6}$$

Although Accuracy (Equation 7) is a common metric, it can be misleading with unbalanced classes. For this reason, AUC-PRC was chosen as the primary evaluation metric.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Where:

Table I: Custom architecture added on top of ResNet-50. Source: Own authorship.

Order	Layer	Description and Parameters
1	`Dense`	Fully connected layer with 120 units and `tanh` activation
2	`Dropout`	Regularization layer with a rate of 30% to prevent overfitting
3	`Dense`	Fully connected layer with 100 units and `relu` activation
4	`BatchNormalization`	Normalization of the previous layer's data to stabilize training
5	`Dropout`	Regularization layer with a rate of 30%
6	`Dense`	Fully connected layer with 50 units and `relu` activation
7	`BatchNormalization`	Applied to normalize the previous layer's values
8	`Dropout`	Regularization layer with a rate of 10%
9	`Dense (Output)`	Output layer with 4 units (one for each class) and `softmax` activation

- TP = True Positives.
- TN = True Negatives.
- FP = False Positives.
- FN = False Negatives.

To assess the final performance, the loss function and the AUC-PRC metric were explicitly calculated for the training and validation datasets. The numerical results are presented and discussed in Section IV.

IV. Results

This section presents the results of the model training and evaluation, assessed using a comprehensive suite of metrics. The discussion begins with an overview of the proposed model's performance, followed by a detailed analysis of class-specific metrics and a critical comparison against two distinct baseline models to validate our approach.

A. Proposed Model Performance

During training, a rapid convergence of the loss function (Figure 7b) and the AUC-PRC metric (Figure 7a) was observed. The model achieved high success in multiclass classification, with key validation metrics of **AUC-PRC 0.9707** and **Loss 0.1551**. These results, illustrated in Figure 7, demonstrate a strong ability to generalize, with a low loss value on the training set combined with a controlled loss on the validation set. Regularization techniques such as data augmentation, dropout, and early stopping were vital in counteracting overfitting.

For a more granular evaluation, a confusion matrix and a classification report were used. The confusion matrix (Figure 8a) offers a visual representation of the classification results, showing consistently high diagonal elements and confirming a strong overall performance. This is further supported by the classification report (Table II), which provides key insights into the model's per-class behavior.

The performance in the truck categories is exceptionally high, with F1-scores of 0.95 and 0.98. The minority class, '3-Axle Bus (3CB)', also shows a robust F1-score of 0.93, indicating that the regularization and fine-tuning strategies effectively mitigated the negative impact of class imbalance. The model proved efficient in identifying visual patterns associated with the number of axles, a fundamental element of classification according to DNIT's methodology.

B. Comparison with Baseline Models

To validate the effectiveness of the proposed fine-tuning approach, a comparison was conducted against two distinct baseline models: a ResNet-50 with 80% of its layers frozen and a Support Vector Machine (SVM) on handcrafted HOG features. Overall metrics are in Table III, while detailed F1-scores are in Table IV.

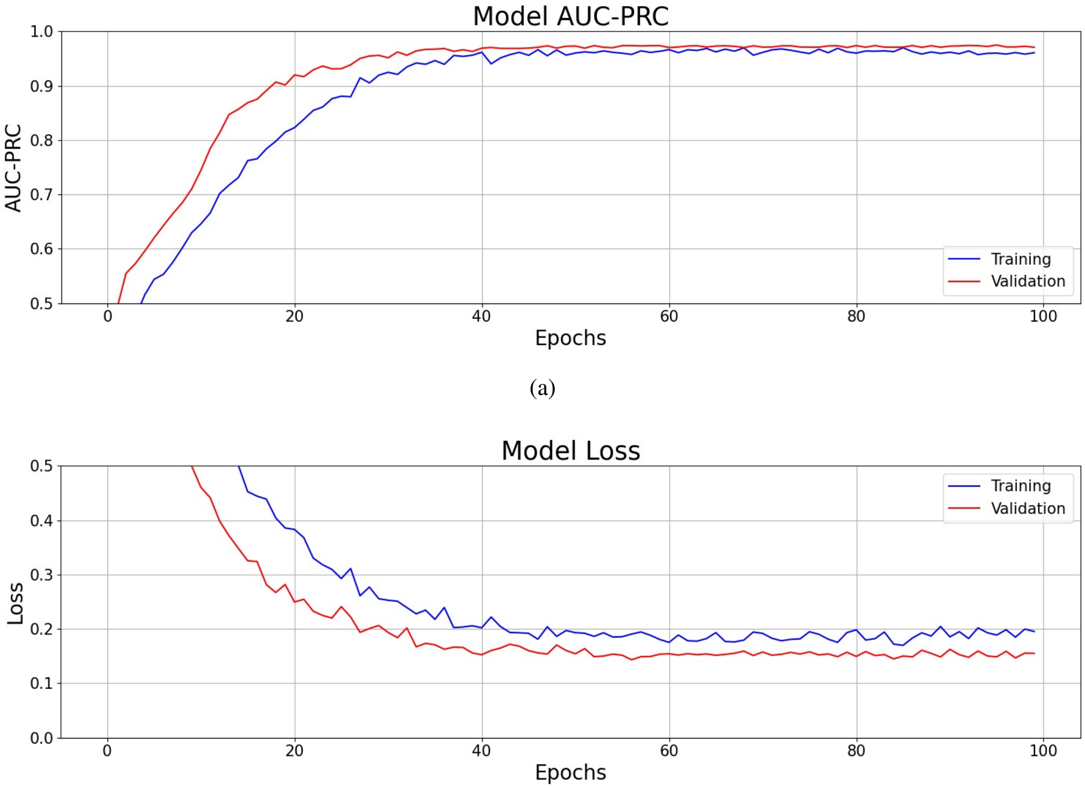
The comparison clearly demonstrates the superiority of the fine-tuned ResNet-50 model. The proposed model, with its F1-score of 0.95 and an AUC-PRC of 0.9707, significantly outperforms the ResNet-50 baseline with a large portion of its layers frozen (F1-score: 0.77, AUC-PRC: 0.7142). This highlights the critical role of allowing the pre-trained weights to adapt to the specific characteristics of the vehicle silhouette dataset.

Interestingly, the traditional SVM + HOG pipeline (F1-score: 0.91, Accuracy: 0.91) proved to be highly competitive with the fine-tuned deep learning model, outperforming the frozen ResNet-50 baseline on several metrics. The detailed analysis of the confusion matrix (Figure 8b) reveals that while all models showed high diagonal values, the fine-tuned ResNet-50 demonstrates a unique ability to learn a hierarchical and abstract representation of features, resulting in a superior overall performance and a more balanced F1-score across all classes.

In conclusion, the results unequivocally validate the effectiveness of the proposed fine-tuning approach. While traditional methods like SVM with HOG features can achieve competitive results, the fine-tuned deep learning model demonstrates a clear edge in overall performance and robustness, highlighting the power of transfer learning and specialized feature adaptation for this challenging task.

V. Conclusion

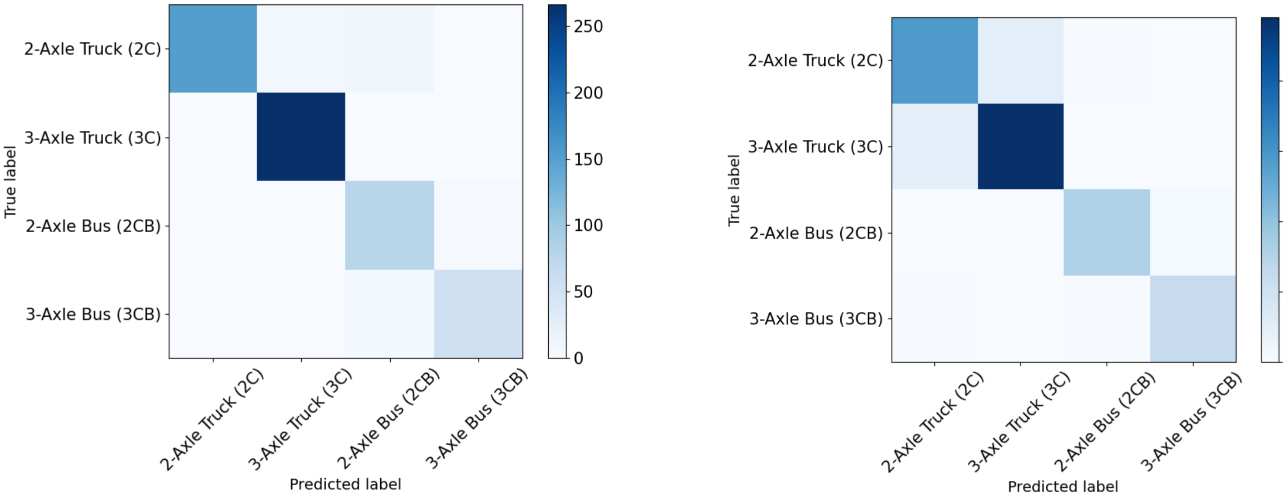
The objective of this work is to identify heavy vehicle categories from images according to their number of axles, following DNIT guidelines [4], thereby generating more reliable and scalable data to support road planning, pavement design, and predictive maintenance of the road network. To achieve this, an automated system was developed employing a deep learning model that integrates automatic vehicle detection with YOLOv8 for efficient image capture, followed by multiclass



(a)

(b)

Figure 7: Performance graphs of the proposed model throughout training: (a) AUC-PRC performance and (b) Error (Loss) during training. Source: Own authorship.



(a)

(b)

Figure 8: Confusion matrices performance on the validation set: (a) Proposed fine-tuned ResNet-50 model and (b) SVM + HOG baseline model. Source: Own authorship.

Table II: Classification Report for the Validation Set. Source: Own authorship.

Class	**Precision**	**Recall**	**F1-score**	**Support**
2-Axle Truck (2C)	1.00	0.90	0.95	168
3-Axle Truck (3C)	0.97	1.00	0.98	267
2-Axle Bus (2CB)	0.83	0.96	0.89	80
3-Axle Bus (3CB)	0.95	0.90	0.93	62
Macro Avg	**0.94**	**0.94**	**0.94**	**577**
Weighted Avg	**0.96**	**0.95**	**0.95**	**577**

Table III: Overall performance metrics of the models on the validation set. Source: Own authorship.

Model	**Validation AUC-PRC**	**Validation Loss**	**Accuracy**	**F1-score (W. Avg)**
Proposed Model (ResNet-50 Fine-tuned)	0.9707	0.1551	0.95	0.95
Baseline 1 (ResNet-50 Fixed)	0.7142	0.6037	0.76	0.77
Baseline 2 (SVM + HOG)	N/A	N/A	0.91	0.91

Table IV: Per-class F1-scores for the models on the validation set. Source: Own authorship.

Model	**F1-score (2C)**	**F1-score (3C)**	**F1-score (2CB)**	**F1-score (3CB)**
Proposed Model (ResNet-50 Fine-tuned)	0.95	0.98	0.89	0.93
Baseline 1 (ResNet-50 Fixed)	0.74	0.92	0.57	0.51
Baseline 2 (SVM + HOG)	0.86	0.91	0.97	0.96

classification using the ResNet-50 architecture with transfer learning.

Despite the constraints imposed by a limited and unbalanced dataset, the developed model demonstrated good performance, achieving an AUC-PRC of 97% on the validation set and a loss function of 0.15. These results, complemented by a detailed classification report and confusion matrix, confirm the network's ability to learn the morphological differences between vehicle categories, even in scenarios with sampling constraints. The preprocessing stage, particularly the automated cropping of detected vehicles and data augmentation techniques, was essential for enhancing the learning process and mitigating the risk of overfitting.

The technical robustness of the proposed model is coupled with its practical relevance. Applying the methodology in real-world scenarios can represent a significant advancement in the automated and reliable acquisition of highway traffic data. Such information is crucial for the proper design of pavements, definition of public transport policies, logistical planning, and highway preservation, optimizing investments and increasing the safety of the national road network.

As future enhancements, we recommend dataset expansion to include new vehicle classes and increase the representation of currently under-sampled categories, such as 3-axle buses (3CB). In addition, it is important to validate the model in videos captured in different regions and under various climatic conditions to test its robustness. A final prospective development is the real-time implementation of the system, using low-cost embedded devices, such as the *Raspberry Pi* or AI-specific hardware like the *Jetson Nano*. For such deployments, evaluating the model's inference speed and latency, along with its computational resource requirements (number of parameters, FLOPs, memory footprint), becomes crucial. This analysis will ensure the system's viability for continuous, high-throughput processing in resource-constrained environments, allowing for direct field application and enabling continuous and automated monitoring of heavy vehicles with greater efficiency and scalability. Such advances can consolidate the proposal as a practical and scalable solution for the automatic classification of heavy vehicles on Brazilian highways.

Acknowledgements

The authors would like to thank the technicians and scholarship holders of the Paving Laboratory of the Federal University of Recôncavo da Bahia (PaveLab-UFRB) and the technical and financial support of the Institute of Transport Research (IPR) of DNIT through the research project linked to the Decentralized Execution Term (TED) No. 507/2023, in partnership with the UFRB. We also thank the Federal University of Recôncavo da Bahia (UFRB) and the PIBIC/UFRB program for granting the scientific initiation scholarship, which made this research possible.

References

[1] S. Yu, Y. Wu, W. Li, Z. Song, and W. Zeng, "A model for fine-grained vehicle classification based on deep learning," *Neurocomputing*, vol. 257, pp. 97–103, 2017.

[2] S. Tas, O. Sari, Y. Dalveren, S. Pazar, A. Kara, and M. Derawi, "Deep learning-based vehicle classification for low quality images," *Sensors*, vol. 22, no. 13, p. 4740, 2022.

[3] R. Kejriwal, R. H. J., and A. Arora, "Artificial intelligence enabled vehicle detection and counting using deep learning," in *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI)*, 2022, pp. 1–6.

[4] Departamento Nacional de Infraestrutura de Transportes (DNIT), "Manual de estudos de tráfego," https://www.gov.br/dnit/pt-br/assuntos/planejamento-e-pesquisa/ipr/coletanea-de-manuais/vigentes/723_manual_estudos_trafego.pdf, 2006, instituto de Pesquisas Rodoviárias (IPR).

[5] M. Hussain, J. J. Bird, and D. R. Faria, "A study on cnn transfer learning for image classification," in *Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK*. Springer, 2019, pp. 191–202.

[6] E. Cetinic, T. Lipic, and S. Grgic, "Fine-tuning convolutional neural networks for fine art classification," *Expert Systems with Applications*, vol. 114, pp. 107–118, 2018.

[7] J. Gupta, S. Pathak, and G. Kumar, "Deep learning (cnn) and transfer learning: a review," in *Journal of Physics: Conference Series*, vol. 2273, no. 1. IOP Publishing, 2022, p. 012029.

[8] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, "A review on yolov8 and its advancements," in *International Conference on Data Intelligence and Cognitive Informatics*. Springer, 2024, pp. 529–545.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[10] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, vol. 1. IEEE, 2021, pp. 96–99.

[11] M. S. de S. Almeida, A. R. A. Achy, W. G. S. Costa, V. R. Santana, N. W. Filho, C. A. Abramides, C. A. T. Carmo, and G. L. O. Marques, "Metodologia para determinação do tráfego rodoviário utilizando a técnica de processamento de imagens," in *XIX Congresso de Pesquisa e Ensino em Transportes – ANPET*, 2023.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] A. Nazir and M. A. Wani, "You only look once-object detection models: a review," in *2023 10th International conference on computing for sustainable global development (INDIACom)*. IEEE, 2023, pp. 1088–1095.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[15] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic data from diffusion models improves imagenet classification," *arXiv preprint arXiv:2304.08466*, 2023.

[16] Z. Shi and M. Liu, "Moving vehicle detection and recognition technology based on artificial intelligence," *International Journal of Circuits, Systems and Signal Processing*, vol. 16, pp. 399–404, 2022.

[17] M. Hasanvand, M. Nooshyar, E. Moharamkhani, and A. Selyari, "Machine learning methodology for identifying vehicles using image processing," *Artificial Intelligence and Applications*, vol. 1, no. 3, pp. 154–162, 2023.

[18] R. Takahashi, T. Matsubara, and K. Uehara, "Data augmentation using random image cropping and patching for deep cnns," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2917–2931, 2019.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[20] R. Balestriero and R. G. Baraniuk, "Batch normalization explained," *arXiv preprint arXiv:2209.14778*, 2022.

[21] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with relu activation function and linear spline-type methods," *Neural Networks*, vol. 110, pp. 232–242, 2019.

[22] A. A. Waoo and B. K. Soni, "Performance analysis of sigmoid and relu activation functions in deep neural network," in *Intelligent Systems: Proceedings of SCIS 2021*. Springer, 2021, pp. 39–52.

[23] A. Al-Sabaawi, H. M. Ibrahim, Z. M. Arkah, M. Al-Amidie, and L. Alzubaidi, "Amended convolutional neural network with global average pooling for image classification," in *International conference on intelligent systems design and applications*. Springer, 2020, pp. 171–180.

[24] F. Bieder, R. Sandkühler, and P. C. Cattin, "Comparison of methods generalizing max-and average-pooling," *arXiv preprint arXiv:2103.01746*, 2021.

[25] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature machine intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

[26] B. K. Iwana, R. Kuroki, and S. Uchida, "Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 4176–4185.

[27] S. Mehra, G. Raut, R. D. Purkayastha, S. K. Vishvakarma, and A. Biasizzo, "An empirical evaluation of enhanced performance softmax function in deep learning," *IEEE Access*, vol. 11, pp. 34 912–34 924, 2023.

[28] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] R. O. Ogundokun, R. Maskeliunas, S. Misra, and R. Damaševičius, "Improved cnn based on batch normalization and adam optimizer," in *International Conference on Computational Science and Its Applications*. Springer, 2022, pp. 593–604.

[30] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2018.

[31] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "The effect of adaptive learning rate on the accuracy of neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, 2021.

[32] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 2002, pp. 55–69.

[33] M. K. Anam, S. Defit, H. Haviluddin, L. Efrizoni, and M. B. Firdaus, "Early stopping on cnn-lstm development to improve classification performance," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1175–1188, 2024.

[34] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[35] P. Li, X. He, X. Cheng, M. Qiao, D. Song, M. Chen, T. Zhou, J. Li, X. Guo, S. Hu *et al.*, "An improved categorical cross entropy for remote sensing image classification based on noisy labels," *Expert Systems with Applications*, vol. 205, p. 117296, 2022.

[36] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, "The area under the precision-recall curve as a performance metric for rare binary events," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 565–577, 2019.

[37] J. Miao and W. Zhu, "Precision–recall curve (prc) classification trees," *Evolutionary intelligence*, vol. 15, no. 3, pp. 1545–1569, 2022.