

Efficient Pothole Segmentation Using Quantization and Pruning for ADAS

Segmentação de Buracos em vias para ADAS Usando Poda De Modelos e Quantização

William D'Abruzzo Martins^{1*}, Fernando Santos Osório², Diego Renan Bruno¹

Abstract: Real-time semantic segmentation is critical for Advanced Driver-Assistance Systems (ADAS), but deploying deep learning models on resource-constrained hardware remains a challenge. This work presents an efficient pipeline for pothole segmentation using the lightweight PP-LiteSeg model, evaluating and comparing two key optimization techniques: post-training quantization and network pruning. Results demonstrate that 8-bit integer (INT8) quantization is highly effective, achieving a 3.54x speedup in GPU compute time with only a minimal, well-controlled loss in segmentation accuracy. While network pruning also reduces model size and latency, it comes at a more pronounced cost to performance. In conclusion, while quantization offers a superior trade-off between speed and accuracy, network pruning remains a valuable strategy for applications where memory compression is the primary objective.

Keywords: segmentation — ADAS — quantization — model pruning

Resumo: A segmentação semântica em tempo real é fundamental para Sistemas Avançados de Assistência ao Condutor (ADAS), mas a implantação de modelos de deep learning em hardware com recursos limitados continua sendo um desafio. Este trabalho apresenta um *pipeline* eficiente para a segmentação de buracos com o modelo leve PP-LiteSeg, avaliando as técnicas de otimização de quantização e poda de rede. Os resultados demonstram que a quantização para inteiros de 8 bits (INT8) é altamente eficaz, alcançando uma aceleração de 3.54x no tempo de computação da GPU com perda mínima de acurácia. Embora a poda de rede também reduza o tamanho e a latência do modelo, seu custo de desempenho é mais pronunciado. Conclui-se que a quantização oferece um *trade-off* superior entre velocidade e acurácia, enquanto a poda de rede se destaca em aplicações onde a compressão de memória é o objetivo principal.

Palavras-Chave: segmentação — ADAS — quantização — poda de modelos

¹ Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista (Unesp), Brazil

² Instituto de Ciências Matemáticas e de Computação., Universidade de São Paulo (USP), Brazil

*Corresponding author: diego.bruno@unesp.br

DOI: <http://dx.doi.org/10.22456/2175-2745.150751> • Received: 07/10/2025 • Accepted: 27/10/2025

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. Introduction

Road infrastructure quality is crucial for both economic development and public safety. Surface defects such as potholes not only shorten the service life of vehicles, affecting the automotive sector, but also create serious safety hazards by raising the probability of traffic accidents [1]. Potholes have severe global impacts, leading to thousands of complaints, injuries, and fatalities each year, while also imposing high repair costs on municipalities across countries such as the United States, the United Kingdom, India, New Zealand, and Brazil [2, 3].

Although investments in road infrastructure have increased, the pace of pothole repairs remains insufficient to keep up with road deterioration. As a result, the responsibility for preventing accidents often falls on drivers. However, human

factors continue to be among the leading causes of traffic accidents [4].

In this context, the automotive industry has increasingly invested in driver-assistance technologies, including systems capable of detecting road surface damage [1], mainly carried out using two approaches: 2D image analysis with cameras and 3D sensors based on laser technology [5]. Although 3D sensors provide high precision in capturing road surface topography, their implementation is costly due to the high price of such devices. In contrast, 2D image-based methods represent a more affordable alternative, offering accurate results with the potential for real-time processing.

Building on 2D image-based methods, traditional computer vision techniques often struggle to accurately detect and segment complex road surface defects such as potholes, hav-

ing their performance compromised under adverse weather conditions, such as heavy rain, fog, and low light, reducing their reliability in detecting road surface irregularities [6]. To overcome these limitations, Deep Learning has emerged as the state-of-the-art approach for image-based road damage detection because it is more resilient to variations in lighting, weather conditions, road textures and achieves better real-time performance [7].

Among deep learning techniques for visual perception, semantic segmentation has proven particularly effective for tasks such as pothole detection and drivable area estimation [8, 9]. Unlike object detection, which provides bounding boxes around targets, semantic segmentation assigns a class label to each pixel in the image, resulting in a fine-grained scene representation that enables precise localization and accurate delineation of potholes and other road surface irregularities. The fine-grained scene representation provided by semantic segmentation models is critical for advanced driver assistance systems (ADAS), especially in pothole detection, since the exact shape, size, and position of a pothole relative to the drivable surface directly influence vehicle control strategies.

Despite these advances, many existing studies still rely on datasets captured from overhead perspectives, which do not reflect the viewpoint of a vehicle-mounted camera. This mismatch between training data and real-world driving conditions often leads to poor generalization and degraded performance during in-vehicle inference, underscoring the importance of datasets that are properly aligned with ADAS applications and real driving scenarios [10]. In addition, few works address the challenges of deploying such systems in real vehicles, where real-time inference and computational efficiency are critical requirements.

To address these limitations, this work proposes a deep learning-based pothole detector using the PP-LiteSeg segmentation model [11], optimized for real-time applications. The model is trained on a dataset captured from a vehicle-mounted camera, and its performance is further enhanced through quantization and network pruning, ensuring efficient and accurate inference for practical ADAS deployment.

2. Related Work

Several studies in the literature explore the application of semantic segmentation for pothole detection. Fan *et al.* [12] proposed a disparity transformation to improve the discrimination between damaged and undamaged regions, combined with an adversarial domain adaptation strategy to convert road RGB images into disparity representations. However, their approach is primarily designed for road inspection rather than driver-assistance systems, relying on overhead imagery instead of the vehicle-mounted perspective required for ADAS applications. In their subsequent work, the authors further incorporated a Graph Attention Layer to optimize feature representations for semantic segmentation, achieving significant performance improvements [13]. Nevertheless, both approaches remain focused on road inspection using over-

head imagery rather than on driver-assistance systems with vehicle-mounted viewpoints.

Katsamenis *et al.* [9] employed the transformer-based architecture SegFormer for semantic segmentation of potholes. Similarly, methods such as ERCU-Net [14], MAFNet [15], and the framework proposed by Akagic *et al.* [16] have achieved promising results. However, these approaches remain primarily focused on road inspection using overhead imagery.

Masihullah *et al.* [17] proposed an attention-based coupled framework for joint road and pothole segmentation, built upon the DeepLabV3+ architecture. Their novel framework achieved 93.6% IoU for road segmentation and 73.8% IoU for pothole segmentation. However, the evaluation for pothole segmentation was conducted on a relatively small dataset of only 237 images, which limits the generalization of the results.

Several works have also focused on deploying pothole detection models on resource-constrained devices. Asad *et al.* [18] utilizes lightweight object detection models deployed on a Raspberry Pi 4B and further optimizes them by exporting to OpenVINO for improved performance on embedded hardware. Similarly, Musa *et al.* [19] employs knowledge distillation, a well-known technique for reducing model parameters, to deploy deep CNN models for pothole image classification on embedded systems. However, these works do not address image segmentation models.

3. Methods

As illustrated in Figure 1, the methodology of this work comprises three main stages. First, a dataset appropriate for ADAS applications is automatically annotated using the SAM 2 model, enabling accurate pixel-level labels for road surface and pothole regions. Next, the PP-LiteSeg architecture is trained on this dataset to perform semantic segmentation of potholes in real-time. Finally, model quantization and network pruning are applied and compared to reduce computational requirements and enable efficient inference, ensuring that the system can operate effectively on embedded hardware for practical ADAS deployment.

3.1 Dataset collection

The training dataset was created by combining two publicly available datasets for pothole object detection¹. The first dataset [20] contains 3,777 vehicle-view images captured with a GoPro, of which 1,119 contain potholes. The second dataset [21] comprises 2,099 images captured under varying weather and lighting conditions, ensuring robustness in diverse real-world scenarios. To address class imbalance and increase data diversity, data augmentation techniques such as horizontal flipping were applied, and some negative samples were excluded, similarly used in the work of Martins *et al.* [10]. The final dataset consists of 5,602 training images (4,156 with

¹Dataset available under request.

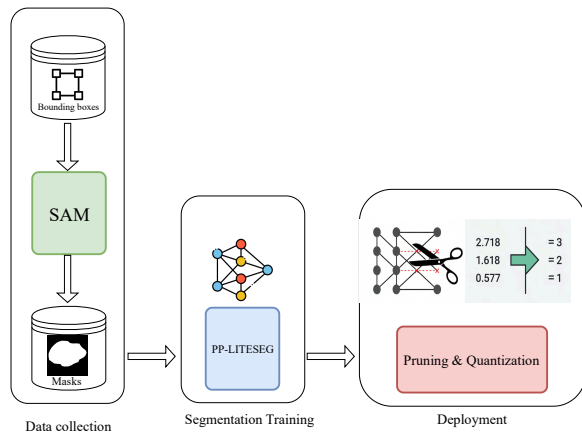


Figure 1. Pipeline of the proposed approach. First, the bounding box annotations are converted into segmentation masks using the Segment Anything Model 2 (SAM 2). With these masks, the PP-LiteSeg segmentation model is trained. Finally, pruning and quantization techniques are applied and evaluated to improve the model’s efficiency and enable deployment on resource-constrained devices.

potholes, 1,446 without) and 1,375 validation images (1,057 with potholes, 318 without). Bounding box annotations were converted into segmentation masks using the foundational model Segment Anything Model 2 (SAM 2) [22], followed by manual inspection to ensure annotation quality (Figure 2). SAM 2 has been widely adopted for automatic dataset labeling [23, 24, 25], particularly due to its strong zero-shot segmentation capabilities.

3.2 PP-LiteSeg Segmentation Model

PP-LiteSeg [11] is a segmentation model proposed by Baidu Inc. that adopts an encoder–decoder architecture and introduces two key components: the Flexible and Lightweight Decoder (FLD) and the Unified Attention Fusion Module (UAFM). In addition, the model incorporates an enhanced variant of the Pyramid Pooling Module originally proposed by Zhao *et al.* [26], termed the Simple Pyramid Pooling Module (SPPM). The overall architecture of PP-LiteSeg is illustrated in Figure 3.

The Flexible and Lightweight Decoder (FLD) introduces a strategy to balance computation between the encoder (feature extractor) and the decoder (segmentation predictor). Instead of keeping the same number of channels throughout the decoder, as commonly adopted in many segmentation models—which leads to unnecessary computational overhead and a decoder bottleneck—the FLD gradually reduces the number of feature channels from high- to low-level layers while simultaneously increasing spatial resolution. This design allows efficient upsampling and pixel-wise prediction while maintaining a favorable trade-off between accuracy and computational cost.

The Unified Attention Fusion (UAF) module is designed to effectively fuse low-level and high-level features, a pro-

cess that is critical for improving segmentation accuracy [27]. It achieves this by combining two complementary attention mechanisms. Spatial attention generates weights that highlight the relative importance of each pixel in the feature map, while channel attention assigns weights that capture the significance of each feature channel. By integrating these mechanisms, UAF enriches feature representations and enhances the model’s ability to capture both detailed spatial information and high-level semantic context.

3.3 Deployment and quantization

Although deep learning models achieve excellent performance in complex tasks such as object detection, segmentation, image classification, and natural language processing, their high computational cost poses significant challenges for embedded systems. Models with strong generalization often contain millions to billions of parameters, leading to increased memory, processing, and energy requirements, which are difficult to meet on resource-constrained, low-power devices.

3.3.1 Quantization

One of the techniques used for deploying those models on resource-constrained devices is the use of quantization [28], which consists of reducing the number of bits used to represent the weights and activations of network layers, typically decreasing from 32-bit floating-point to 16-bit floating-point or 8-bit integers. In addition to reducing memory consumption, quantization also improves inference time by using less expensive arithmetic operations. We quantized the PP-LiteSeg model from 32 to both 16-bit floating-point and 8-bit integers.

Quantization to 16-bit floating point is relatively straightforward, as it preserves the same data type while reducing memory requirements. In contrast, quantization to 8-bit integers (INT8) is more challenging, since values must be represented with only 256 discrete levels. This process requires mapping continuous values into a finite range $[\alpha, \beta]$, where inputs falling outside the range are clipped to the nearest boundary. The determination of α and β is performed through a calibration procedure [29]. In this work, we adopt the percentile calibration [30], which estimates the quantization range by running a representative subset of the dataset through the network and selecting the 99.99th percentile of the observed activations as the maximum range.

3.3.2 Network Pruning

Network pruning is a model compression technique that reduces the size and complexity of neural networks by systematically removing less significant weights or activation units [31]. Two main approaches are commonly employed (Figure 4): unstructured pruning, which removes individual weights below a given threshold [32], and structured pruning, which eliminates entire structures such as filters, channels, or layers [33]. Unstructured pruning provides higher flexibility and can achieve greater compression ratios, as any weight may be pruned. However, it is challenging to accelerate in practice because it requires specialized hardware to efficiently exploit

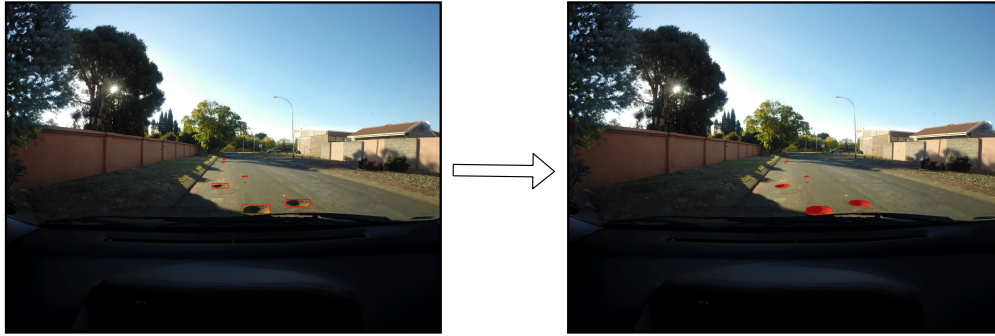


Figure 2. Results of automatic labeling with the Segment Anything Model 2 (SAM 2). The left image shows the original bounding box annotations, while the right image displays the masks generated by the model.

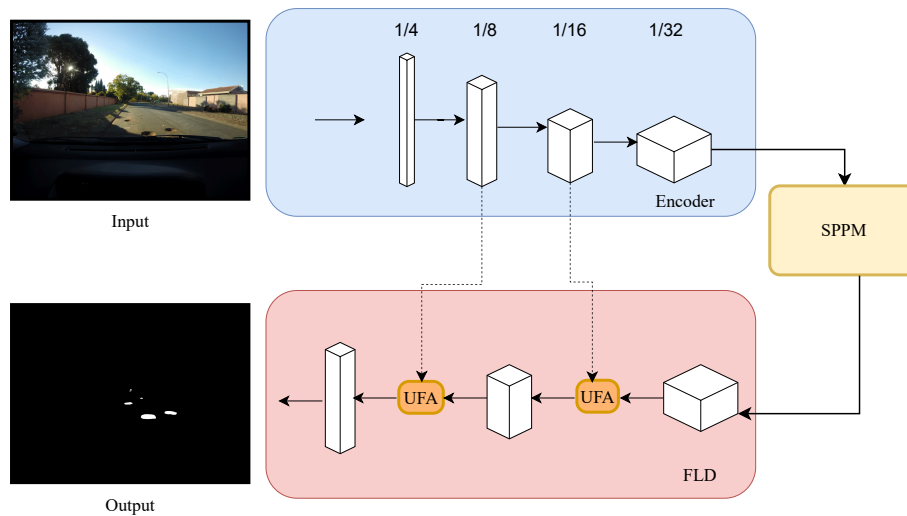
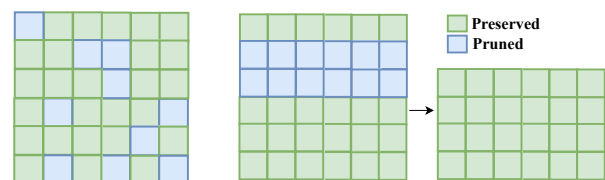


Figure 3. Overview of the PP-LiteSeg architecture, adapted from [11]. Features from the input image are first extracted by the encoder and aggregated in the Simple Pyramid Pooling Module (SPPM) to capture global context. Subsequently, the Flexible and Lightweight Decoder (FLD) combines detailed and semantic features, further enhanced by the Unified Attention Fusion (UAF).

sparse representations [34]. Structured pruning, on the other hand, removes well-defined computational structures, making it easier to achieve actual speedups on general-purpose hardware, albeit at the cost of reduced flexibility compared to unstructured pruning.

For PP-LiteSeg, we adopted the group pruning approach proposed by Fang *et al.* [35]. In this method, layers are first grouped according to their inter-dependencies by constructing a dependency graph. For each group, an importance score $I(w)$ is computed for each parameter using the L2-norm, and the group importance $I(g)$ is obtained by aggregating the importance scores of the parameters within the group. A sparse training phase is applied to sparsify parameters at the group level to be able to safely remove the group from the network. Finally, a relative importance score is calculated to identify and remove unimportant parameters.



(a) Unstructured (b) Structured pruning
Figure 4. Comparison between unstructured (left) and structured (right) pruning approaches.

4. Results and Discussion

4.1 Training

The PPLiteSeg model was trained using the STDC-1 backbone [36] within the SuperGradients [37] deep learning framework, initialized with pretrained weights from the Cityscapes dataset. Optimization was performed using the Adam optimizer with

an initial learning rate of $1e-5$, which was dynamically decayed using a cosine annealing scheduler down to $1e-6$. The model was trained by minimizing the Binary Cross Entropy Dice Loss with a batch size of 8, for up to 500 epochs, with early stopping based on the IoU metric for the pothole class. To enhance model robustness and generalization, we applied data augmentation techniques, including Random Erasing [38] and Affine transformations [39]. Table 1 summarizes the performance results for input resolutions of 640×640 and 1024×1024 . The higher resolution of 1024×1024 yields a significant performance improvement across all key segmentation metrics, most notably in Pothole IoU and F1-Score.

All experiments were carried out on a workstation with an NVIDIA RTX 3060 GPU (12 GB VRAM).

Table 1. Performance metrics of PPLiteSeg on input resolutions 640×640 and 1024×1024 .

Metric	640×640	1024×1024
IoU Metrics		
Pothole IoU	0.6255	0.6672
Background IoU	0.9993	0.9994
Mean IoU	0.8124	0.8333
Dice Metrics		
Pothole Dice	0.7696	0.8004
Background Dice	0.9996	0.9997
Mean Dice	0.8846	0.9000
Other Metrics		
Pixel Accuracy	0.9984	0.9984
Recall	0.7279	0.7599
Precision	0.8140	0.8437
F1 Score	0.7685	0.7996

4.2 Quantization

After training, quantization was applied to the 1024×1024 PP-LiteSeg model in order to improve inference efficiency. Specifically, INT8 quantization was performed using the *percentile calibration* method [30]. The trained model was first exported to the ONNX format and subsequently optimized with NVIDIA TensorRT, enabling low-latency inference on NVIDIA GPU-powered systems.

A detailed analysis of the inference performance is presented in Table 2. For this analysis, latency is defined as the sum of the host-to-device data transfer, GPU compute time, and device-to-host data transfer. The throughput is calculated by dividing the total number of queries by the host walltime, measured from the moment the first query is enqueued until the last query is completed. Additionally, the model's segmentation performance across different precisions is summarized in Table 3.

Table 3. Performance of PP-LiteSeg at 1024×1024 resolution across different precisions. All values are expressed as percentages. Values in parentheses indicate the absolute percentage-point difference relative to the FP32 baseline.

Metric	FP32 (%)	FP16 (%)	INT8 (%)
IoU Metrics			
Pothole IoU	66.72	66.72 (0.00)	64.63 (-2.09)
Background IoU	99.94	99.94 (0.00)	99.93 (-0.01)
Mean IoU	83.33	83.33 (0.00)	82.28 (-1.05)
Dice Metrics			
Pothole Dice	80.04	80.00 (-0.04)	78.51 (-1.53)
Background Dice	99.97	99.97 (0.00)	99.96 (-0.01)
Mean Dice	90.00	90.00 (0.00)	89.24 (-0.76)
Other Metrics			
Pixel Accuracy	99.84	99.84 (0.00)	99.83 (-0.01)
Recall	75.99	75.97 (-0.02)	76.75 (+0.76)
Precision	84.37	84.37 (0.00)	80.41 (-3.96)
F1 Score	79.96	79.94 (-0.02)	78.53 (-1.43)

Table 2. Inference performance comparison across different precisions with a 1024×1024 input. Speedup values in parentheses are relative to the FP32 baseline.

Metric	FP32	FP16	INT8
Latency (ms)	6.72	3.42 (1.96x)	2.54 (2.65x)
GPU Compute Time (ms)	5.84	2.53 (2.31x)	1.65 (3.54x)
Throughput (QPS)	170	393 (2.31x)	603 (3.55x)

The summarized results demonstrate a minimal performance difference between the FP32 and FP16 models, confirming that FP16 offers a significant 2.31x speedup in both throughput and GPU compute time with a negligible impact on accuracy. While INT8 quantization introduces a slight, well-controlled performance loss, its impact is mitigated by the percentile-based calibration method, which effectively handles activation outliers by ensuring an adequate clipping range based on the dataset distribution. In return for this minor accuracy trade-off, the INT8 model delivers the highest performance, achieving a peak speedup of 3.55x in GPU compute time. This makes INT8 a compelling option for latency-critical applications where maximum efficiency is paramount.

4.3 Pruning

We also evaluated the impact of different pruning schedules on the trained model to verify performance improvements. We employed and compared two distinct strategies, summarized in the Figure 5:

Table 4. Performance and inference speedup of the Direct Pruning method at different model sizes.

Remaining Channels	Pothole IoU (%)	GPU Compute Time (ms)
100%	66.72	5.84
90%	61.40	5.31 (1.10x)
80%	56.03	4.59 (1.27x)
70%	43.08	3.93 (1.49x)
60%	35.57	3.49 (1.67x)
50%	30.32	2.75 (2.12x)
40%	18.11	2.46 (2.37x)
30%	0.00	1.91 (3.06x)

Iterative Pruning: Following the gradual pruning methodology [32], this strategy involves removing a fixed fraction of the network’s weights over several steps. After each pruning step, the model was fine-tuned for 20 epochs to allow it to recover from the removal of parameters.

Direct Pruning: In this more direct approach, the target percentage of weights was removed all at once. The resulting smaller network was then fine-tuned for 20 epochs to recover its performance.

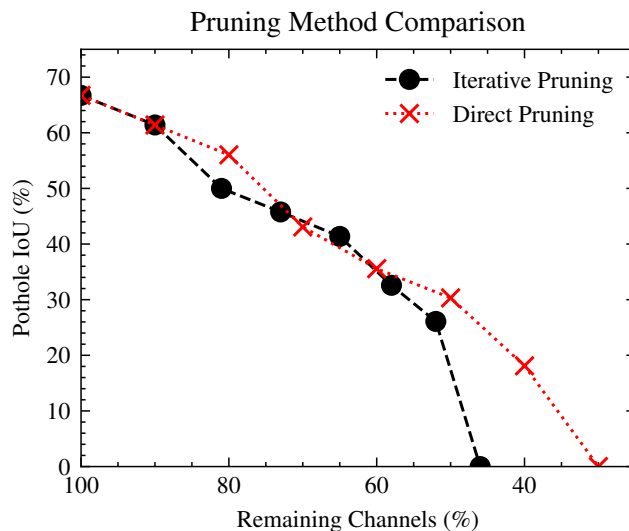


Figure 5. Performance comparison between iterative and one-shot pruning. The graph shows the Pothole IoU as a function of the percentage of remaining model parameters. Direct pruning achieve higher results in similar pruning rates.

The graph shows that pruning directly from the target architecture yields better results than the iterative prune-and-finetune approach. As noted by Liu *et al.* [40], learning the important weights of a larger model through iterative pruning does not benefit the pruned model in structured pruning. In fact, it can perform equivalently or even worse than using random weights, as the prior pruning configuration may serve as a poor starting point for further pruning. Further from direct pruning results are presented in Table 4.

5. Conclusion

This work presents a pipeline for real-time semantic segmentation using the PP-LiteSeg model and evaluates two deployment techniques: Model Pruning and Quantization. The results indicate that quantization is highly effective, with the INT8 model achieving a significant 3.54x speedup in GPU compute time while incurring only a minimal loss in segmentation accuracy. Network pruning, on the other hand, reduces memory constraints and improves inference time but at a more pronounced cost to performance compared to quantization. Although quantization outperformed network pruning in its balance between speed and accuracy, pruning remains a valuable strategy for applications where memory compression is the primary objective, such as on highly resource-constrained hardware.

Future work may involve testing the optimized models on real-world embedded hardware to confirm their practical speed and efficiency. Additionally, techniques such as knowledge distillation could be explored to help the compressed models recover accuracy lost during pruning or quantization.

Acknowledgements

This study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number #2025/03224-4.

Author Contributions

Martins, WD and Bruno, DR conceived of the presented idea; Martins, WD contributed to the implementation of the research; All authors contributed to the analysis of the results and to the writing of the manuscript.

References

- [1] SAFYARI, Y.; MAHDIANPARI, M.; SHIRI, H. A review of vision-based pothole detection methods using computer vision and machine learning. *Sensors*, MDPI AG, v. 24, n. 17, p. 5652, ago. 2024. ISSN 1424-8220. Disponível em: <http://dx.doi.org/10.3390/s24175652>.

- [2] DHIMAN, A.; KLETTE, R. Pothole detection using computer vision and learning. *IEEE Transactions on Intelligent Transportation Systems*, v. 21, n. 8, p. 3536–3550, 2020.
- [3] Confederação Nacional do Transporte (CNT). *Radar CNT do Transporte – Pontos Críticos 2023*. Brazil, 2023. [CNT Transport Radar – Critical Points 2023]. Available at: <https://cnt.org.br/analises-transporte>.
- [4] MCKENNA, F. P. The human factor in driving accidents an overview of approaches and problems. *Ergonomics*, Informa UK Limited, v. 25, n. 10, p. 867–877, out. 1982. ISSN 1366-5847. Disponível em: <http://dx.doi.org/10.1080/00140138208925047>.
- [5] LYDON, D. et al. A review of vision-based methods for pothole detection and road profile analysis. In: *Civil Engineering Research in Ireland 2020*. Cork, Ireland: Cork Institute of Technology, 2020.
- [6] MA, N. et al. Computer vision for road imaging and pothole detection: a state-of-the-art review of systems and algorithms. *Transportation Safety and Environment*, v. 4, n. 4, p. tdac026, 11 2022. ISSN 2631-4428. Disponível em: <https://doi.org/10.1093/tse/tdac026>.
- [7] CHENG, L.; ZHANG, X.; SHEN, J. Road surface condition classification using deep learning. *Journal of Visual Communication and Image Representation*, Elsevier BV, v. 64, p. 102638, out. 2019. ISSN 1047-3203. Disponível em: <http://dx.doi.org/10.1016/j.jvcir.2019.102638>.
- [8] MA, N. et al. Computer vision for road imaging and pothole detection: a state-of-the-art review of systems and algorithms. *Transportation Safety and Environment*, Oxford University Press (OUP), v. 4, n. 4, nov. 2022. ISSN 2631-4428. Disponível em: <http://dx.doi.org/10.1093/tse/tdac026>.
- [9] YANG, J. et al. Semantic segmentation for autonomous driving. In: FAN, R.; GUO, S.; BOCUS, M. J. (Ed.). *Autonomous Driving Perception: Fundamentals and Applications*. Singapore: Springer Nature Singapore, 2023. p. 101–137. ISBN 978-981-99-4287-9. Disponível em: https://doi.org/10.1007/978-981-99-4287-9_4.
- [10] MARTINS, W. D.; OSORIO, F. S.; BRUNO, D. R. Real time deep learning based pothole detection on low cost embedded devices for adas. In: *2025 Brazilian Conference on Robotics (CROS)*. Belo Horizonte, Brazil: IEEE, 2025. v. 1, p. 1–6.
- [11] PENG, J. et al. *PP-LiteSeg: A Superior Real-Time Semantic Segmentation Model*. 2022. Disponível em: <https://arxiv.org/abs/2204.02681>.
- [12] FAN, R. et al. We learn better road pothole detection: From attention aggregation to adversarial domain adaptation. In: *Computer Vision – ECCV 2020 Workshops*. Springer International Publishing, 2020. p. 285–300. ISBN 9783030668235. Disponível em: http://dx.doi.org/10.1007/978-3-030-66823-5_17.
- [13] FAN, R. et al. Graph attention layer evolves semantic segmentation for road pothole detection: A benchmark and algorithms. *IEEE Transactions on Image Processing*, v. 30, p. 8144–8154, 2021.
- [14] TRIPATHI, R.; INDU, S.; KUMAR, R. Ercu-net: segmentation of road potholes using enhanced residual convolutional block based on u-net for adas. *Signal, Image and Video Processing*, Springer Science and Business Media LLC, v. 18, n. S1, p. 385–394, abr. 2024. ISSN 1863-1711. Disponível em: <http://dx.doi.org/10.1007/s11760-024-03161-w>.
- [15] FENG, Z. et al. Mafnet: Segmentation of road potholes with multimodal attention fusion network for autonomous vehicles. *IEEE Transactions on Instrumentation and Measurement*, v. 71, p. 1–12, 2022.
- [16] AKAGIC, A.; BUZA, E.; OMANOVIC, S. Pothole detection: An efficient vision based method using rgb color space image segmentation. In: *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Opatija, Croatia: IEEE, 2017. p. 1104–1109.
- [17] MASIHULLAH, S. et al. Attention based coupled framework for road and pothole segmentation. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. Milan, Italy: IEEE, 2021. p. 5812–5819.
- [18] ASAD, M. H. et al. Pothole detection using deep learning: A real-time and ai-on-the-edge perspective. *Advances in Civil Engineering*, Wiley, v. 2022, n. 1, jan. 2022. ISSN 1687-8094. Disponível em: <http://dx.doi.org/10.1155/2022/9221211>.
- [19] MUSA, A. et al. A lightweight cnn-based pothole detection model for embedded systems using knowledge distillation. IOS Press, 2022. ISSN 1879-8314. Disponível em: <http://dx.doi.org/10.3233/FAIA220281>.
- [20] NIENABER, S.; BOOYSEN, M.; KROON, R. Detecting potholes using simple image processing techniques and real-world footage. In: *Proc. 34th Annu. Southern African Transport Conf.* Pretoria, South Africa: CSIR International Convention Centre, 2015. p. 1–10.
- [21] JAKUBEC, M. et al. Comparison of cnn-based models for pothole detection in real-world adverse conditions: Overview and evaluation. *Applied Sciences*, v. 13, n. 9, 2023. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/13/9/5810>.
- [22] RAVI, N. et al. Sam 2: Segment anything in images and videos. *arXiv preprint*, 2024.
- [23] ZHAO, J.; ZHANG, Z.; QI, F. An intelligent image segmentation annotation method based on segment anything model. In: CRUZ, C.; ZHANG, Y.; GAO, W. (Ed.). *Intelligent Computers, Algorithms, and Applications*. Singapore: Springer Nature Singapore, 2024. p. 237–243. ISBN 978-981-97-0065-3.

- [24] REZA, M. A. et al. SegBuilder: A Semi-Automatic Annotation Tool for Segmentation . In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Los Alamitos, CA, USA: IEEE Computer Society, 2025. p. 8494–8503. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/WACV61041.2025.00823>).
- [25] MUMUNI, F.; MUMUNI, A. *Segment Anything Model for automated image data annotation: empirical studies using text prompts from Grounding DINO*. 2024. Disponível em: <https://arxiv.org/abs/2406.19057>).
- [26] ZHAO, H. et al. *Pyramid Scene Parsing Network*. 2017. Disponível em: <https://arxiv.org/abs/1612.01105>).
- [27] LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation . In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2015. p. 3431–3440. ISSN 1063-6919. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298965>).
- [28] SHAFIQUE, M. A.; MUNIR, A.; KONG, J. Deep learning performance characterization on gpus for various quantization frameworks. *AI*, MDPI AG, v. 4, n. 4, p. 926–948, out. 2023. ISSN 2673-2688. Disponível em: <http://dx.doi.org/10.3390/ai4040047>).
- [29] WU, H. et al. *Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation*. 2020. Disponível em: <https://arxiv.org/abs/2004.09602>).
- [30] MCKINSTRY, J. L. et al. Discovering low-precision networks close to full-precision networks for efficient inference. In: *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*. Vancouver, BC, Canada: IEEE, 2019. p. 6–9.
- [31] CUN, Y. L.; DENKER, J. S.; SOLLA, S. A. Optimal brain damage. In: *Proceedings of the 3rd International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1989. (NIPS'89), p. 598–605.
- [32] HAN, S. et al. Learning both weights and connections for efficient neural networks. In: *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*. Cambridge, MA, USA: MIT Press, 2015. (NIPS'15), p. 1135–1143.
- [33] ANWAR, S.; HWANG, K.; SUNG, W. Structured pruning of deep convolutional neural networks. *J. Emerg. Technol. Comput. Syst.*, Association for Computing Machinery, New York, NY, USA, v. 13, n. 3, fev. 2017. ISSN 1550-4832. Disponível em: <https://doi.org/10.1145/3005348>).
- [34] KUZMIN, A. et al. Pruning vs quantization: Which is better? 2023.
- [35] FANG, G. et al. DepGraph: Towards Any Structural Pruning . In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2023. p. 16091–16101. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01544>).
- [36] FAN, M. et al. Rethinking bisenet for real-time semantic segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, 2021. p. 9716–9725.
- [37] AHARON, S. et al. *Super-Gradients*. GitHub, 2021. Disponível em: <https://zenodo.org/record/7789328>).
- [38] ZHONG, Z. et al. *Random Erasing Data Augmentation*. 2017. Disponível em: <https://arxiv.org/abs/1708.04896>).
- [39] SANDRU, A.; GEORGESCU, M.-I.; IONESCU, R. T. *Feature-level augmentation to improve robustness of deep neural networks to affine transformations*. 2022. Disponível em: <https://arxiv.org/abs/2202.05152>).
- [40] LIU, Z. et al. Rethinking the value of network pruning. In: *International Conference on Learning Representations*. OpenReview, 2019. Disponível em: <https://openreview.net/forum?id=rJlnB3C5Ym>).