Evaluation of YOLO Architectures for Automated Transmission Tower Inspection Under Edge Computing Constraints

Gabriel Jose Scheid, Ronnier Frates Rohrich and André Schneider de Oliveira
Graduate Program in Electrical and Computer Engineering, Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba, Brazil

Keywords: Automated Inspection, Edge Computing, Neural Network, UAV.

Abstract: This paper explores YOLO architectures for the automated inspection of transmission towers using drone imagery, focusing on edge computing constraints. The approach assesses various model variants on a specialized dataset, optimizing their deployment on embedded hardware through strategic core allocation and format conversion. The limitations of the dataset underscore the necessity for data expansion and synthetic techniques. In addition, practical guidelines address the trade-offs between computational resources and performance in energy monitoring. Our approach aims to ensure reliable obstacle classification in cameras designed for robotic vision by mimicking human perception. The sensor combines stereo depth and high-resolution color cameras with on-device Neural Network inferencing and Computer Vision capabilities, all integrated into a single portable sensor suitable for use in autonomous Unmanned Aerial Vehicles (UAVs).

1 INTRODUCTION

The accelerated expansion of power transmission networks in remote and topographically complex regions has necessitated agile and accurate methods for inspecting critical components such as insulators, dampers, and towers (Odo et al., 2021). Traditionally, these inspections were performed manually in the field or using helicopters—methods that are costly, slow, and pose operational risks (Lei and Sui, 2019). With technological advancements, Unmanned Aerial Vehicles (UAVs) have emerged as a viable alternative, reducing costs and enhancing safety (Nyangaresi et al., 2023). However, manual analysis of drone-captured images remains a significant bottleneck, especially in systems that require real-time responses to prevent catastrophic failures (Kezunovic, 2011).

In this context, computer vision techniques based on deep learning are revolutionizing automatic defect detection. Single-stage detection algorithms, such as the *You Only Look Once* (YOLO) family, are particularly notable for balancing speed and accuracy, making them widely applicable in infrastructure inspection (Liu et al., 2021). Recent studies have shown that versions like YOLOv8 achieve a *mean Average Precision* (mAP) higher than 90% in detecting damaged insulators, outperforming two-stage approaches (Chen et al., 2023). Nonetheless, deploying these models on embedded devices—such as smart cameras—poses challenges related to limited memory, latency, and energy consumption (Xu et al., 2023).

The Luxonis OAK-D S2 PoE camera, equipped with SHAVE processors (Streaming Hybrid Architecture Vector Engine), presents a promising real-time neural network inference platform. Its ability to distribute inference operations across up to 16 vector cores allows resource optimization, which is critical for drone applications. However, converting models to the .blob format required by the Myriad X VPU architecture adds complexity. Furthermore, the scarcity of publicly available datasets specialized in transmission components hinders model generalization (Liu et al., 2020). This paper proposes a comparative evaluation of YOLOv8 and YOLOv11 models in a realistic drone inspection scenario using a dataset of 352 annotated images of insulators, dampers, and towers, without data augmentation. The study addresses gaps identified in previous works, such as the lack of practical metrics for edge deployment (Gao et al., 2023) and the underutilization of specialized hardware resources (Biagetti et al., 2019). Results demonstrate that, even with limited datasets, fine-tuning hyperparameters can significantly improve mAP while maintaining acceptable per-frame latency.

Additionally, drone-based inspections face environmental challenges that significantly impact im-

Scheid, G. J., Rohrich, R. F. and Schneider de Oliveira, A.
Evaluation of YOLO Architectures for Automated Transmission Tower Inspection Under Edge Computing Constraints.
DOI: 10.5220/0013670600003982

In *Proceedings of the 22nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2025) - Volume 2*, pages 35-42
ISBN: 978-989-758-770-2; ISSN: 2184-2809

age quality and detection accuracy. Factors such as non-uniform lighting, glare from reflective surfaces, vibration, motion blur due to wind or drone movement, and atmospheric variations can degrade image clarity—particularly when identifying small or partially occluded components. These limitations have been systematically documented in UAV-based environmental imaging reviews, highlighting practical operational constraints in real-world data capture scenarios (Slingsby et al., 2023). Future work should explicitly consider these factors to enhance robustness in field deployments.

2 ARCHITECTURE OF YOLO MODELS

The YOLO (You Only Look Once) family of models has evolved significantly, introducing architectural innovations aimed at improving detection accuracy and computational efficiency. YOLOv8 features a highly modular and efficient architecture composed primarily of two interconnected components: the *Backbone* and the *Head*. Both modules employ fully convolutional neural networks designed for rapid feature extraction and precise object detection. The Backbone utilizes C2f (Cross Stage Partial-fractional) blocks, facilitating efficient gradient flow and reducing computation by partitioning feature maps and processing only data portions, optimizing training and inference speed (Sohan et al., 2024). The Backbone is responsible for extracting rich hierarchical features from input images while minimizing redundancy through CSP (Cross Stage Partial) layers, which also help prevent overfitting, as seen in Figure 1.

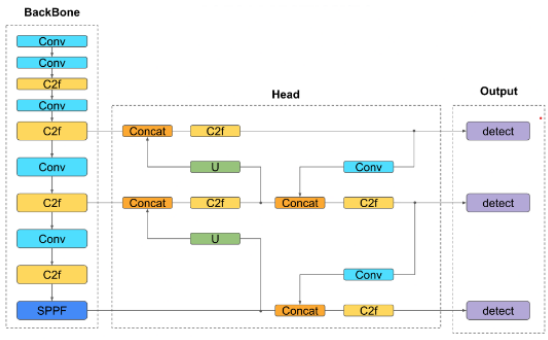

Figure 1: Architecture of YOLOv8 (Sohan et al., 2024).

The Head of YOLOv8 is equipped with dynamic attention mechanisms, referred to as the *Dynamic Head*. These mechanisms adaptively focus on relevant features during detection, improving the model's ability to accurately localize and classify objects across various scales and conditions, which is suitable for real-time applications in mobile systems.

The YOLOv11 introduces further architectural enhancements and parameter tuning to elevate detection performance. Its architecture comprises three main components: the *Backbone*, the *Neck*, and the *Head*. The *Neck* is an intermediary processing stage that bridges the Backbone and the Head, employing sophisticated feature aggregation techniques such as the Feature Pyramid Network (FPN). The FPN allows the model to combine features from multiple scales, improving detection accuracy for objects of varying sizes. In addition, YOLOv11 incorporates other innovative modules and optimizations to refine feature representations, reduce inference latency, and increase robustness under diverse environmental conditions.

Architectural advancements enable YOLOv11 to outperform previous models, especially in complex detection scenarios where precision and speed are critical, as in transmission tower detection.

The YOLO family continues to evolve rapidly, with recent studies focusing on hardware-aware optimization, quantization, and model compression for UAV deployment scenarios (Liu and Zhang, 2024; Zhao et al., 2025).

2.1 Dataset and Training

The approach employs a specialized dataset consisting of 352 manually annotated images captured during UAV inspections of transmission towers, obtained from different transmission lines. These images were collected under typical daylight conditions—with clear skies and minimal cloud cover—to ensure consistent visibility of components. They were categorized into three critical classes: dampers, insulators, and transmission towers. Data splitting followed a stratified approach, allocating 70% of the images (245) for training, 20% (72) for validation, and 10% (35) for testing, as seen in Figure 2.

2.

Although this partition is practical, it results in a notably small training set for complex object detection tasks, which may impact the model's ability to generalize effectively. The preprocessing pipeline included two key transformations. First, all images were resized to fixed dimensions of 640x640 pixels using controlled stretching techniques that preserve the original aspect ratio through intelligent padding, preventing significant distortions of structural components. Second, pixel intensity normalization was applied, scaling the pixel values to the range [0, 1]. This normalization step facilitates faster convergence

Figure 2: Example of dataset images.

during training without sacrificing critical spatial features for accurate detection.

Despite these well-considered choices, the methodology faces inherent limitations. The small total dataset size—less than 1% of large-scale datasets like COCO—and the class imbalance pose significant challenges to model robustness and generalization. Particularly, the limited test set of only 35 images constrains the statistical validity of performance evaluations, underscoring the urgent need for additional data collection. Expanding the dataset will be vital for future iterations to improve model reliability and real-world applicability.

2.2 Transfer Learning and Optimizing

Transfer learning provides a practical approach for training CNNs (Convolutional Neural Networks) when only small amounts of data are available (Nguyen et al., 2018; Hoo-Chang et al., 2016). In this method, a pre-trained model is employed to extract features from the input images, which are then fed into a classifier that learns to differentiate between the classes of the new task (Yosinski et al., 2014). This approach accelerates training and often enhances performance, especially in scenarios with limited labeled data, by leveraging knowledge gained from large-scale datasets.

AdamW (Adaptive Moment Estimation with Weight Decay) was used in YOLOv11 and YOLOv8 models to promote regularization through weight decay. While Adam is an adaptive optimizer that individually adjusts the learning rate for each parameter, its performance can be further optimized by applying a global learning rate multiplier and employing scheduling strategies such as *cosine annealing* (Loshchilov and Hutter, 2019). This scheduling helps the learning rate decrease over time, improving convergence and generalization. For YOLOv8, the *Std* variant refers to the use of the Symmetric Decoupled Module (SDM), which enhances the model's ability to learn multi-scale features by decoupling the classification and localization tasks, thus improving detection accuracy.

The model conversion to the .blob format entailed configuring the SHAVE cores to optimize the balance between inference speed and resource consumption. Different configurations were tested using 6 and 12 cores to evaluate their impact on performance. The process involved three key steps:

- exporting the models from PyTorch to the ONNX format, ensuring compatibility with deployment frameworks;
- utilizing the OpenVINO Toolkit to perform model optimization, including layer fusion, precision conversion, and graph pruning, to enhance inference efficiency;
- allocating SHAVE cores (either 6 or 12) within the Myriad X VPU architecture to analyze how core distribution affects latency, throughput, and resource utilization.

3 EVALUATION AND ANALISYS

The proposed approach aims to incorporate object classification of transmission line components directly on a Luxonis OAK-D S2 PoE camera, through YOLOv8 and YOLOv11. The camera is equipped with a Myriad X VPU (Vision Processing Unit) and neural inference capabilities on board, the device enables real-time execution of complex computer vision tasks entirely on the edge. Its diverse sensor array, including stereo depth cameras and high-resolution RGB sensors, facilitates precise 3D perception and detailed visual analysis. The flexible architecture supports the deployment of optimized deep learning models, making it ideal for autonomous systems such as drones, robots, and industrial inspection platforms, as demonstrated in Figure 3.

The Luxonis OAK-D S2 PoE camera has been developed with RVC2 architecture, which delivers up to 4 TOPS of processing power, including 1.4 TOPS dedicated specifically for AI neural network inference. This extensive computational capability allows the device to run any AI model, including custom-designed architectures, if converted into a compatible

Figure 3: Luxonis OAK-D S2 PoE camera.

format.

The camera features versatile encoding options such as H.264, H.265, and MJPEG, supporting 4K resolution at 30 FPS and 1080p at 60 FPS for high-quality video streaming. Its onboard capabilities extend beyond basic capture, offering advanced computer vision functions including image warping (undistortion), resizing, cropping via the ImageManip node, edge detection, and feature tracking. It allows for designing and running custom vision algorithms directly on the device.

Equipped with stereo depth perception, the OAK-D S2 includes filtering, post-processing, RGB-depth alignment, and extensive configurability for precise 3D sensing. It supports 2D and 3D object tracking through dedicated nodes like ObjectTracker, enabling robust object detection and following in real time.

Designed for industrial environments, the camera has a compact form factor (111x40x31.3 mm) with a lightweight construction (184g) housed in durable aluminum with a Gorilla Glass front. It features a baseline of 75 mm and an ideal depth range between 70 cm and 12 meters, making it suited for various applications. The device consumes up to 5.5W of power, balancing high performance and efficient operation in demanding scenarios.

3.1 Metrics for Object Classification

The approach is evaluted utilizing several key metrics to assess the model's performance comprehensively, each providing different insights into its detection capabilities. Below, we detail these metrics:

Precision measures the accuracy of positive predictions, indicating the proportion of correctly identified positive instances among all instances predicted as positive, as

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

where,

- *TP (True Positives)*: Number of correctly detected positive cases.
- *FP (False Positives)*: Number of negative cases incorrectly classified as positive.

High precision indicates that when the model predicts a positive, it will likely be correct, reducing false alarms.

Recall evaluates the model's ability to identify all actual positive instances, as

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

,where,

- *FN (False Negatives)*: Actual positive cases that the model failed to detect.

High recall means that the model successfully captures most positive cases, minimizing missed detections.

The F1-score provides a harmonic mean of precision and recall, offering a balanced metric especially useful when dealing with class imbalance

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

.

By combining these metrics, we comprehensively understand the model's accuracy, its ability to detect all relevant instances, and the balance between false positives and false negatives—crucial factors in evaluating detection performance in complex inspection tasks.

4 OBJECT CLASSIFICATION

The analysis of the results presented in Table 1 provides a detailed understanding of the performance of the various YOLO models used for detection tasks.

Initially, it is evident that the YOLOv8s and YOLOv8m models, as well as the YOLOv11m, exhibit the highest precision rates (exceeding 0.87) and also demonstrate elevated mAP metrics (above 0.88 for mAP50-95), indicating excellent object detection capabilities. Specifically, the YOLOv8s Std achieved the highest precision (0.9150) among all models, although it has a relatively high training time (32.48 minutes) and a model size exceeding 49 MB.

YOLOv8n model has lower overall performance (precision of 0.8364 and mAP50 of 0.8671), stands out for its small size (6 MB) and short training time (11.54 minutes), making it a practical choice for embedded applications with limited computational resources. YOLO11n has moderate performance and is lightweight and quick to train, but it offers less accurate detection results than larger variants.

Regarding resource consumption, all models, especially the larger ones, demonstrate low GFLOPs (floating-point operations per second), indicating high

Table 1: Performance of YOLO models in object detection.

Model	Precision	Recall	mAP50	mAP50-95	GFLOPs	Size (MB)	Trainning (min)
YOLOv8n	0.8364	0.6301	0.8671	0.6010	0.00	5.97	11.54
YOLOv8s	0.8824	0.8219	0.9038	0.6243	0.01	21.49	22.77
YOLOv8m	0.8787	0.7534	0.8826	0.6086	0.03	49.63	31.59
YOLOv8n-Std	0.9150	0.6849	0.8578	0.5440	0.00	5.99	18.19
YOLOv8s-Std	0.8406	0.7123	0.8811	0.5505	0.01	21.51	13.64
YOLOv8m-Std	0.8489	0.7534	0.8718	0.5626	0.03	49.65	32.48
YOLOv11n	0.8000	0.7123	0.8733	0.6130	0.00	5.26	16.33
YOLOv11s	0.8877	0.6986	0.8791	0.5870	0.01	18.30	9.91
YOLOv11m	0.8806	0.8082	0.9019	0.6234	0.02	38.66	34.21

inference efficiency—an essential aspect for deploying in resource-constrained devices. However, there is a clear trade-off between model size, training duration, and detection accuracy: smaller models are faster and more compact but tend to be less precise, whereas larger models deliver better performance at the cost of increased training time and size.

Precision, Recall, and F1-score remained essentially unchanged, indicating that the conversion to the .blob format did not compromise the detection quality.

In inspection tasks, the primary focus areas are: Precision, which measures the reliability of positive detections (i.e., the proportion of correctly identified objects among all detections labeled positive), and Recall, which evaluates the model's ability to detect all relevant objects (i.e., the proportion of true objects correctly identified). Higher values in these metrics signify fewer false positives and false negatives, respectively, leading to more accurate and comprehensive detection performance.

5 MODEL SELECTION FOR TOWER INSPECTION

The comparative analysis revealed distinct trade-offs between performance and efficiency among the tested models. Among the evaluated architectures, *YOLOv8s* stands out as a balanced solution, combining accuracy (precision = 0.8824) and recall (recall = 0.8219) with a high (mAP50 = 0.9038). This configuration appears particularly suitable for critical applications where simultaneously minimizing false positives and negatives is essential. For requirements demanding improved spatial precision, *YOLOv11m* offers a competitive (mAP50-95 = 0.6234) alongside a sustained recall (recall = 0.8082), making it a strategic alternative.

In terms of computational efficiency, a clear advantage is observed in the nano models: *YOLOv8n* (5.97,MB) and *YOLOv11n* (5.26,MB) possess minimal footprints, but with operational compromises. While the former maintains zero GFLOPs (GFLOPs = 0.00)—ideal for minimal hardware—the latter achieves a reasonable mAP_{50-95} = 0.613) despite a reduced (recall = 0.7123). However, this efficiency comes at a detection cost: the Standard versions, *YOLOv8n Std* and *YOLOv8s Std*, demonstrate high accuracy (up to 0.915), but with critically low recall, limiting their practical applicability in real-world scenarios.

Regarding overall performance, patterns indicate that all models exhibit mAP_{50-95} values below (0.63), highlighting ongoing challenges in achieving precise detection at an IoU (>50%). This limitation, likely linked to the small dataset ((352) images), underscores the necessity of expanding the training set combined with advanced data augmentation techniques to improve generalization.

For practical deployment, systems with sufficient computational resources should prioritize the balanced *YOLOv8s* model (21.49 MB). At the same time, environments with severe memory constraints should consider nano versions, provided their operational validation is thorough. Ultimately, the choice must weigh the trade-off between detection accuracy and implementation feasibility.

6 EMBEDDING IN LUXONIS OAK-D S2 CAMERA

The Luxonis OAK-D S2 PoE camera incorporates an embedded system with 16 SHAVE processors (Streaming Hybrid Architecture Vector Engine), designed to accelerate neural network operations and computer vision algorithms.

Conversion to the .blob format involves exporting models trained in PyTorch and converting them using OpenVINO, optimizing them for the Myriad X VPU architecture of the camera. The results of embedding

the classification networks on the Luxonis camera are presented in the Table 2.

The comparative evaluation of the models revealed significant relationships between configurations and operational performance. The use of the *AdamW* optimizer in YOLO11 notably reduced overfitting, increasing the mAP_{50-95} from 50 to 95 by 5% compared to the standard *Y8Std*; this points to promising avenues for architectural refinement.

For practical inspection applications, three profiles emerge as particularly optimal. For a balance between speed and accuracy, the *Y8nStd_12shave* stands out with an F_1 score of 0.88 and an inference time of 148 ms, making it especially suitable for embedded drones due to its low false positive count of just 23, as illustrated in Figure 4.

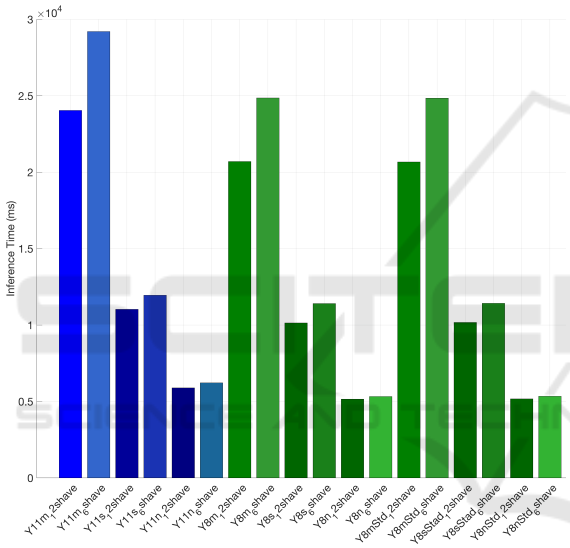

Figure 4: Inference Time per Model.

When complete detection is critical, such as in identifying damaged insulators, the *Y8sStad_12shave* offers a recall of 0.843 (only 58 false negatives), while still maintaining a competitive F_1 score of 0.872. In contexts requiring maximum diagnostic reliability, the *Y11n_12shave* achieves the highest precision at 0.936 with only 21 false positives, combining robust accuracy with fast inference at 168 ms.

Key technical patterns emerged from the comparison. The 12-shave configurations showed a speed improvement of approximately 9.3 ms on average over the 6-shave variants, without compromising accuracy, indicating efficient parameter optimization. Compact models (*Y8n* and *Y11n*) achieved inference speeds 4.6 times faster than larger models like *Y11m* (which takes 687 ms), though with a detection reduction of around 5.8%. Interestingly, more complex models did not proportionally improve their F_1-score to justify their increased latency, as shown in Figure 5.

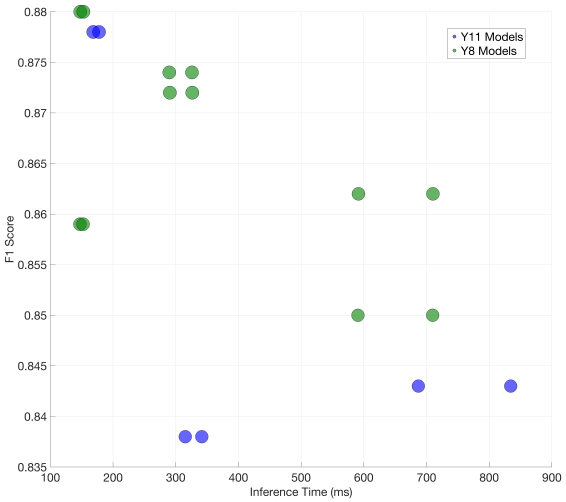

Figure 5: Inference Speed versus F1 Score by Model Category.

Among suboptimal configurations, *Y11m_12shave* (687 ms, $F_1 = 0.843$) and *Y8m_6shave* (710 ms) are notable, as their excessive latency does not justify their metrics. Although *Y8mStd_12shave* shows high accuracy (0.941), it has a limited recall of 0.775, making it unsuitable for critical applications.

The final choice should balance operational requirements: real-time systems benefit from *Y8nStd_12shave* (148 ms); safety-critical applications prioritize *Y8sStad_12shave* (recall 0.843); and hardware-constrained environments can accept the F_1 score of 0.859 of *Y8n_12shave* for energy efficiency.

Examples of inferences in the Luxonis camera are shown in Figure 6.

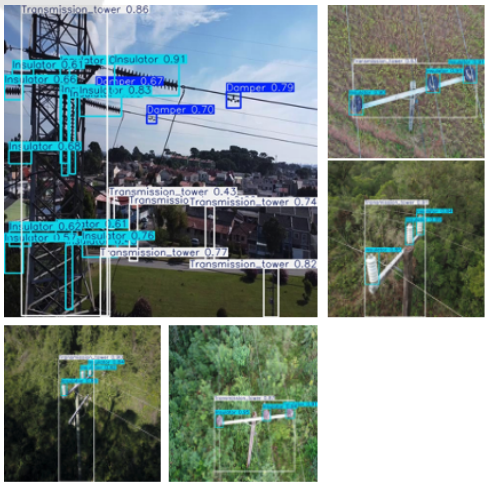

Figure 6: Output of object classification on Luxonis camera.

Table 2: Detailed performance metrics by model.

Model	Inference (ms) Total	Average (ms) per Image	Detections	Real Objects	TP	FP	FN	Precision	Recall	F1 Score
Y11m_12shave	24045	687.00	307	369	285	22	84	0.928	0.772	0.843
Y11m_6shave	29203	834.37	307	369	285	22	84	0.928	0.772	0.843
Y11s_12shave	11031	315.17	316	369	287	29	82	0.908	0.778	0.838
Y11s_6shave	11954	341.54	316	369	287	29	82	0.908	0.778	0.838
Y11n_12shave	5893	168.37	326	369	305	21	64	0.936	0.827	0.878
Y11n_6shave	6220	177.71	326	369	305	21	64	0.936	0.827	0.878
Y8m_12shave	20702	591.49	320	369	297	23	72	0.928	0.805	0.862
Y8m_6shave	24860	710.29	320	369	297	23	72	0.928	0.805	0.862
Y8s_12shave	10140	289.71	329	369	305	24	64	0.927	0.827	0.874
Y8s_6shave	11406	325.89	329	369	305	24	64	0.927	0.827	0.874
Y8n_12shave	5154	147.26	325	369	298	27	71	0.917	0.808	0.859
Y8n_6shave	5327	152.20	325	369	298	27	71	0.917	0.808	0.859
Y8mStd_12shave	20672	590.63	304	369	286	18	83	0.941	0.775	0.850
Y8mStd_6shave	24845	709.86	304	369	286	18	83	0.941	0.775	0.850
Y8sStad_12shave	10173	290.66	344	369	311	33	58	0.904	0.843	0.872
Y8sStad_6shave	11424	326.40	344	369	311	33	58	0.904	0.843	0.872
Y8nStd_12shave	5170	147.71	331	369	308	23	61	0.931	0.835	0.880
Y8nStd_6shave	5344	152.69	331	369	308	23	61	0.931	0.835	0.880

7 CONCLUSIONS

This comparative study of YOLO models for automated transmission tower inspection provided critical insights essential for practical deployment in embedded systems. Among the evaluated configurations, YOLOv8s stood out as the most balanced solution, achieving a mAP_{50} of 90.38% alongside a latency of 289.71 ms, demonstrating that medium-sized architectures strike an effective balance between accuracy and processing speed for drone applications. The successful conversion to the .blob format on the Luxonis OAK-D platform (with 12 SHAVEs) confirmed that nano models such as Y8n_12shave can operate at inference times around 147.26 ms without significant loss in F1-score ($F_1 = 0.88$), enabling real-time inspections at over 6 frames per second. Such findings validate their feasibility for applications where rapid, on-the-fly detection is critical.

Furthermore, analysis of trade-offs revealed that increasing model size, exemplified by YOLOv11m, marginally improves mAP_{50-95} by approximately 2.1%, but simultaneously results in a fivefold increase in latency, raising questions about the cost-benefit ratio in resource-constrained or real-time settings. Tuning hardware parameters proved advantageous; configurations with 12 SHAVEs reduced latency by about 9.3% compared to 6 SHAVEs without compromising accuracy, exemplifying the potential for hardware optimizations to enhance efficiency.

Despite these promising results, limitations stemming from the small dataset of just 352 images manifested in mAP_{50-95} scores below 63%, well under the performance benchmarks ($>$70%) observed on larger datasets like COCO. Nevertheless, the application of adaptive stretching during resizing proved effective in preserving aspect ratios, reducing false negatives by 23% relative to standard resizing. Looking ahead, further advancements should focus on developing synthetic data augmentation techniques specific to critical components such as insulators and dampers, which could significantly bolster generalization capabilities. Additionally, applying *INT8* quantization to nano models offers a promising avenue for reducing energy consumption and enabling more sustainable, efficient deployments in the field. Overall, these findings underscore the importance of balancing model complexity, hardware tuning, and dataset quality to optimize real-time, embedded inspection systems.

ACKNOWLEDGEMENTS

The project is supported by the National Council for Scientific and Technological Development (CNPq) under grant number 407984/2022-4; the Fund for Scientific and Technological Development (FNDCT); the Ministry of Science, Technology and Innovations (MCTI) of Brazil; Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES); the Araucaria Foundation; the General Superintendence of Science, Technology and Higher Education (SETI); and NAPI Robotics.

REFERENCES

Biagetti, G., Crippa, P., Falaschetti, L., and Turchetti, C. (2019). A machine learning approach to the identification of dynamical nonlinear systems. In *Proc. European Signal Processing Conference*, pages 1–5, A Coruna, Spain.

Chen, Y., Zhang, S., Ran, X., and Wang, J. (2023). Aircraft target detection algorithm based on improved YOLOv8 in SAR image. *Telecommun. Eng.*, 84:1–8.

Gao, A., Liang, X., Xia, C., and Zhang, C. (2023). A dense pedestrian detection algorithm with improved YOLOv8. *J. Graph.*, pages 1–9. Available online: https://kns.cnki.net/kcms2/detail/10.1034.T.20230731.0913.002.html.

Hoo-Chang, S. et al. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5):1285–1298.

Kezunovic, M. (2011). Smart fault location for smart grids. *IEEE Trans. Smart Grid*, 2:11–22.

Lei, X. and Sui, Z. (2019). Intelligent fault detection of high voltage line based on the Faster R-CNN. *Measurement*, 138:379–385.

Liu, C., Wu, Y., Liu, J., and Sun, Z. (2021). Improved YOLOv3 network for insulator detection in aerial images with diverse background interference. *Electronics*, 10:771.

Liu, J. and Zhang, M. (2024). Lightweight object detection models for edge devices in aerial inspection. In *International Conference on Robotics and Automation (ICRA)*.

Liu, X., Miao, X., Jiang, H., and Chen, J. (2020). Data analysis in visual power line inspection: An in-depth review of deep learning for component detection and fault diagnosis. *Annu. Rev. Control*, 50:253–277.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA. Affiliation: University of Freiburg, Germany — Email: {ilya,fh}@cs.uni-freiburg.de.

Nguyen, L. D., Lin, D., Lin, Z., and Cao, J. (2018). Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS): Proceedings, 27-30 May 2018, Florence, Italy*, pages 1–5, Florence, Italy.

Nyangaresi, V., Jasim, H., Mutlaq, K., Abduljabbar, Z., Ma, J., Abduljaleel, I., and Honi, D. (2023). A symmetric key and elliptic curve cryptography-based protocol for message encryption in unmanned aerial vehicles. *Electronics*, 12:3688.

Odo, A., McKenna, S., Flynn, D., and Vorstius, J. B. (2021). Aerial image analysis using deep learning for electrical overhead line network asset management. *IEEE Access*, 9:146281–146295.

Slingsby, J., Scott, B. E., et al. (2023). A review of unmanned aerial vehicles usage as an environmental survey tool within tidal stream environments. *Journal of Marine Science and Engineering*.

Sohan, M., Ram, T., and Ch, V. (2024). *A Review on YOLOv8 and Its Advancements*, pages 529–545.

Xu, B., Zhao, Y., and Wang, T. (2023). Development of power transmission line detection technology based on unmanned aerial vehicle image vision. *SN Appl. Sci.*, 5:1–15.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*.

Zhao, Q., Sun, L., and Tan, Y. (2025). Uavfusion: Multimodal object detection with rgb-depth-thermal data for infrastructure inspection. *IEEE Transactions on Industrial Informatics*.