EHALEYO: Edge AI-Based High-Accuracy, Lightweight, Enhanced YOLOv11 for Real-Time Small UAV Detection in Complex Environments

Ali Kadhum Idrees
University of Applied Sciences and Arts
Dortmund, Germany
ali.idrees@fh-dortmund.de

Sara Kadhum Idrees
Universite Marie et Louis Pasteur-UTBM-CIAD
Belfort, France
sara.idrees@utbm.fr

Joseph Azar
Universite Marie et Louis Pasteur
Belfort, France
joseph.azar@univ-fcomte.fr

Raphaël Couturier
Universite Marie et Louis Pasteur
Belfort, France
raphael.couturier@univ-fcomte.fr

Franck Gechter
Universite Marie et Louis Pasteur-UTBM-CIAD
Belfort, France
franck.gechter@utbm.fr

Rolf Schuster
University of Applied Sciences and Arts
Dortmund, Germany
rolf.schuster@fh-dortmund.de

***Abstract*—The rapid development in civilian use of drones brings the focus on safety and security problems, particularly with respect to public space and airspace applications, demanding reliable detection in real time. Thus, as a remedy to these challenges, it has been identified that Edge AI comes as an attractive paradigm which facilitates resource-efficient and low-latency inference from data sources. To address these issues, we propose an Edge AI-Based High-Accuracy, Lightweight, Enhanced YOLOv11 (EHALEYO) for real-time small unmanned aerial vehicles (UAVs) detection in complex environments. EHALEYO incorporates depthwise convolution and integrates RepViT with Efficient Channel Attention (ECA) for enhanced feature extraction to increase feature representation and detection accuracy. Moreover, EHALEYO employed a lightweight LSDECD detection head to maintain the size of the model while preserving accuracy. Experimental evaluations demonstrate the efficiency of the proposed EHALEYO compared to existing methods. It reduces the model size by 41.5–62.1%, computational complexity by 33.9–53.6%, and storage footprint by 38.1–54.4% on the DUT dataset. Furthermore, EHALEYO outperforms all competing approaches on the VisDrone dataset, achieving superior performance with only 3.9 GFLOPs and 1.03M parameters. EHALEYO outperforms YOLOv11n on Raspberry Pi 5, achieving 43.5% higher FPS, 28.6% lower inference time, 30.3% lower per-frame latency, and 30.2% lower energy consumption per inference, allowing for effective real-time edge deployment.**

***Index Terms*—Edge AI, UAV Detection, Real-time Inference, Lightweight Deep Learning, YOLOv11, Complex Environments**

I. Introduction

A. Background and Context

As the drone industry and its associated supply chains mature, modern drones have become affordable, small-scale, highly flexible, and easy to operate. Drones are now widely used in aerial photography, fire rescue, light shows, and more. However, this proliferation has exposed sensitive areas, such as research and training bases and military facilities, to the risks of illegal intrusion and aerial reconnaissance, posing a significant threat to public safety and national security. Consequently, anti-drone technologies are essential for safeguarding public safety, national security, and the protection of key infrastructure. Effectively identifying drones is vital for fighting them, as it provides critical information for anti-drone systems to implement timely countermeasures. Existing tracking techniques, such as radio spectrum detection, radar detection and acoustic detection often require an expensive device and a specific environment. Radar detection is based on the reflection of electromagnetic waves from potential targets, however drones introduce many challenges due to their low flying altitude, their slow motion and their limited reflecting surface area [1].

Object detection has been greatly improved with recent advances in artificial intelligence. The deep learning algorithms automatically learn complex features from large datasets, which improve the accuracy of detection and the robustness to debris in various scenes. On the other hand, traditional object recognition methods, which combine hand-crafted features with machine learning classifiers [2] face the problems such as high computation cost and low efficiency, resulting in limited generalization. Their methods for real-time detection are also not doing very well, while deep learning can improve the speed of and accuracy in fastidious species. Deep learning-based algorithms are particularly good at detecting low-flying drones due to their small size and agile flight.

B. Motivation

The powerful feature extraction capability of deep learning-based object detectors has enabled UAV identification to become an increasingly significant area of research. It is advantageous in efficiency and cost. But real-time computer

This work was conducted within the EMULATE project at Dortmund University of Applied Sciences and Arts, in collaboration with Université de Technologie de Belfort–Montbéliard (UTBM) and Université Marie et Louis Pasteur (UMLP), and forms part of the PhD thesis of Sara Kadhum Idrees.

vision detection of UAVs still has issues. For instance, when in a complex background, UAVs are easily missed or misrecognised, small UAVs will not be detected for most of the times, and it's difficult to strike a balance between detection cost and algorithmic performance. In the YOLO series, YOLOv11 is an excellent algorithm which has already achieved a lot in computer vision. The merging of deep learning with edge-aware devices is a enabling trend in various domains. Edge devices are particularly useful to execute drone detection models for applications such as drone surveillance where the system must be operational over long periods, since they offer low latency and low power consumption [1]. What's more, if local deep learning is deployed in these edge devices, the costs of transferring data and cloud computing will both decrease, providing an easier way to establish bigger surveillance systems. But deep learning models usually need a lot of processing power and memory, which can be a problem for edge-embedded systems with limited hardware. Also, models that are too complex might slow down real-time performance, but just making the model smaller often makes it less accurate at detecting things.

C. Contributions

This paper presents the following contributions: label=(iii)

1) We propose an Edge AI–based High-Accuracy, Lightweight, Enhanced YOLOv11 (EHALEYO) for real-time small UAV detection in complex environments. The main goal is to build a more lightweight model that can be deployed on an edge AI device while maintaining accurate real-time detection performance in complex environments.
2) We have included the newest lightweight CNN module in our model: RepViT, which, when used with the ECA attention mechanism to improve C3K2, makes the model better at representing more complicated situations and tasks that require deeper feature extraction.
3) We replace the standard convolutional layers in the backbone and neck networks with depthwise convolution layers. The proposed network reduces the number of network parameters without affecting its object detection capability.
4) We put an enhanced feature refinement through C3k2_RVB_ECA module, which improves multiscale detection accuracy. Strategic attention mechanisms enable better feature fusion and computational efficiency. Optimized architecture delivers superior small object detection while maintaining real-time performance.
5) Several experiments on Raspberry Pi 5 using the DUT [3] and VisDrone [4] datasets evaluated the proposed EHALEYO approach. EHALEYO outperforms other models in model size, GFLOPs, and detection accuracy. It surpasses YOLOv11n in FPS, inference time, per-frame latency, and energy consumption, making it favourably suited for real-time edge deployment.

II. Related work

Advanced anti-UAV systems for recognising tiny, rapidly moving drones have been developed in response to growing concerns about illegal UAV activities. By utilising AI advancements, deep learning-based object detection surpasses conventional techniques by facilitating more precise and efficient recognition of these challenging-to-detect drones. Object detection methods based on deep learning can be divided into two-stage region-based and one-stage regression-based solutions [wenxuan2022review]. The other one generates candidate regions before classifying objects and refining bounding boxes. The well-known algorithms in this research field include RCNN, Fast RCNN, Faster RCNN and Mask RCNN [5]–[8]. The one-stage methods, such as SSD [9] and YOLO [10], directly examine the entire image via a neural network to output classification and localization at the same time, while eliminating the need for prior candidate region generation. One-stage object detection method has its advantages that one needs simpler computation and less storage on parameters, resulting in a smaller model size and higher efficiency of the detection process, which makes it more suitable for real-time detection. The two-stage object detection model achieves better detection accuracy, but is computationally expensive, has many parameters and costs much more inference time. The YOLO algorithm is a successful one-stage object detector in terms of speed, accuracy, and complexity. Zhao et al. [39] proposed an improved underwater target detection method based on YOLOv7-Tiny, which takes advantage of the "noloss" feature of RepVGG in inference to further enhance network's feature extraction capacity. Zhao et al. introduced a lightweight detection network with multiple attentional mechanisms to address the challenges of object detection in aerial images. Furthermore, Zhu et al. [11] proposed a universal (MOT) approach to solve the occlusion problem. Lin et al. [12] has proposed a simple one-stage detector, RetinaNet combining with a pyramid feature network (FPN) and an anchor mechanism. All of these contributions are shaping the future of anti-UAV systems and making it easier to see them in a fuller spectrum of scenarios. Several other works are proposed to improve YOLOv11 based on the VisDrone dataset such as BMGS-YOLOv8 [13], LMAD-YOLO [14], RSW-YOLO [15], and Zhu et al. [16].

III. The EHALEYO approach

A. DWConv Convolution

The conventional YOLOv11n model has a huge amount of parameters and its detection accuracy is relatively low. To tackle this issue, in this paper, we reduce the parameter size of the model pioneered by replacing all standard convolutions operations in backbone and neck by light DWConv operation. This method allows for even smaller model parameters and faster detection. The DWConv [17] is a special designed convolution operation whose architecture is shown in Fig.1. Unlike regular convolution that uses only one conv kernel for all channels, DWConv assigns a unique conv kernel to each

channel. This approach classifies the input feature maps and separately carries out convolution within each classification. With the number of groups N, parameters required for DWConv is just 1/N as many as those in traditional convolution. Therefore, the introduction of DWConv greatly reduces the number of model parameters, resulting in a more compact model.

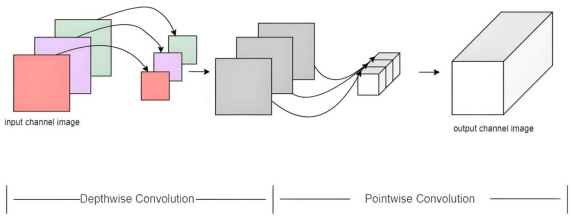


Fig. 1: The structure of DWConv

B. LSDECD:

LSDECD [18], a lightweight enhancement to the detection head. It is based on the LSCD design. Detailed enhanced convolution (DEConv) has replaced the standard convolutional layer, which was initially used for feature extraction. This refinement aims to enhance the detection head's capability to capture detailed information and improve detection accuracy. Fig. 2 presents the structure of the LSCDECD module. The DEConv improves the representation and generalization of the model by combining a priori information into the normal convolutional layers. This design allows DEConv to be equivalently converted to ordinary convolution through reparameterisation techniques without adding additional parameters or computational cost. This leads to improving the detail-capturing capability of the detector head. DEConv maximises the degree of lightweighting from the original LSCD module and achieves a significant improvement in model accuracy, which is considerably greater than that of the model without lightweighting. This improvement not only reduces the computational burden of the model but also plays an important role in maintaining or even improving detection performance.

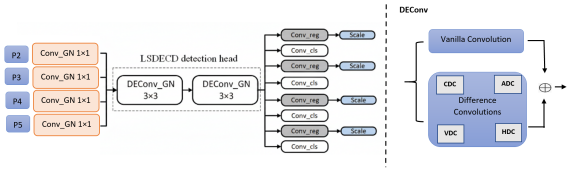


Fig. 2: Detailed structure of the LSDECD module in the detection head.

C. RVB-ECA:

The C3k2 module in YOLOv11 improves feature extraction through the use of variable convolutional kernels and channel separation, proving especially useful for intricate scenes and advanced feature extraction tasks. This strategy may augment the amount of parameters and processing resources necessary for detecting small targets, so diverging from the objective of a lightweight design. This study presents the lightweight Rep-Vit module [19], which enhances the MobileNetV3 model to resolve this issue. Figure 3 illustrates that the original MobileNetV3 block utilizes a 1x1 unfolded convolution alongside a 1x1 projection layer for interchannel interaction, while RepVit elevates the squeeze-and-excitation (SE) layer and implements structural reparameterization techniques to augment model learning during training.

Furthermore, the ECA attention mechanism is implemented to enhance model robustness by applying exponential moving averaging to historical data, effectively reducing the impact of noise or outliers and improving model stability without directly affecting performance metrics.

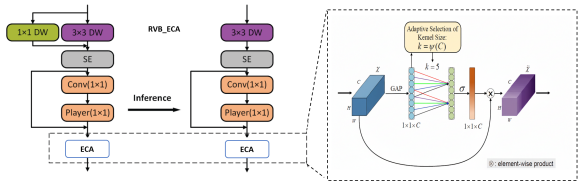


Fig. 3: Detailed structure of the RepVit-ECA module.

D. Network structure of EHALEYO:

Figure 4 illustrates the comprehensive structure of the suggested model. The input image is first subjected to feature extraction using the RVB module and the RVB-ECA module within the backbone network. The collected features are subsequently input into the neck feature fusion module to assimilate information across many scales. P2 incorporates an extra detection head in the neck, targeting a 160×160 feature map. This modification is specifically designed for detecting smaller targets and improving performance in scenarios where high-resolution feature maps are crucial.The retrieved features are ultimately processed by four lightweight detection heads to yield the target detection results for the small UAV.

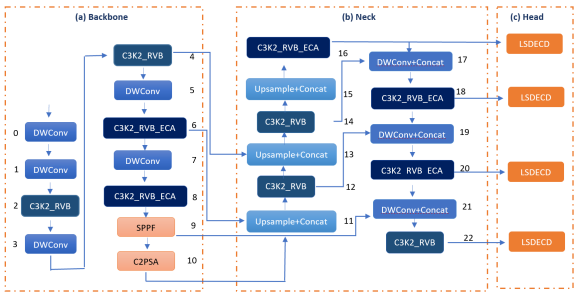


Fig. 4: The overall architecture of the proposed EHALEYO approach.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

This section shows experiments performed on the two different datasets to validate the effectiveness of the proposed

TABLE I: Hardware platform and parameters of the training environment.

Parameter	Configuration
CPU	Intel Xeon Gold 6226R @ 2.90GHz
GPU	Tesla T4 GPU
GPU memory size	15 GB
Deep learning architecture	torch-2.8.0+cu126 CUDA:0

TABLE II: Key parameters used during training the model.

Parameter	Setup
Epochs	300
Momentum	0.937
Learning Rate	0.01
Weight Decay	0.0005
Batch Size	32
Optimizer	SGD
Input Size	640×640

EHALEYO model. Two datasets are considered in this paper: DUT [3] and VisDrone [4]. ,

For the experimental environment and training processes, the Google Colab Pro computational platform was used in conducting this study. The experimental configuration included an Intel Xeon CPU and a Tesla T4 GPU, equipped with 12.7GB of system memory. All implementations were created in Python 3.12.12 using the PyTorch 2.8.0 deep learning framework with CUDA 12.6 speedup. The Tesla T4 GPU, which is based on the Turing architecture, had tensor cores that made it possible to train the proposed models quickly and with mixed precision. We set the input data image size to 640 × 640 to make the device do less work. Taking into account the balance between performance and the limited resources of the device. After many tries at debugging the parameters, the best parameter setup was found. Table I shows important parameter settings, while Table II shows the most important parameter settings for the EHALEYO model.

This study methodically examines the detection efficacy of the models through various performance metrics, including precision, recall, mAP0.5, mAP0.5:0.95, the quantity of parameter values for the model, model size, and operational efficiency. The following is the formula for calculating:

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

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

In evaluating the efficacy of a predictive model, we concentrate on various essential metrics: true positives (TPs), referring to samples predicted as positive by the model that are indeed positive; false positives (FPs), referring to samples predicted as positive by the model that are actually negative; and false negatives (FNs), referring to samples predicted as negative by the model that are actually positive. The precision-recall curve (P-R curve) is a crucial instrument that illustrates a model's performance over various thresholds. The region bounded by this curve, along the horizontal (recall) and vertical (precision) axes, is referred to as the average precision (AP). The mean average precision (mAP) is derived by averaging the AP values across all categories as follows:

$$AP = \int_0^1 \text{Precision}(Recall)\, d(Recall) \tag{3}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4}$$

B. Experimental Results and Ablation experiment

1) Ablation experiments: The YOLOv11n model is further improved in this work for better target detection (See the improvements in Table III and training curves in Fig. 5). The model was observed to have a quick convergence of training losses and validation metrics around 100–300 cycles. This suggests an appropriate learning rate and steady performance, in particular with respect to the accuracy, recall, mAP@0.5, and mAP@0.5:0.95. The experimental results consistently show the influence of various architectural changes on a YOLOv11n baseline model. Each factor has a different impact on the trade-off between accuracy, model efficiency and computational cost. The DWConv (Depthwise Convolution) module's main goal is to improve efficiency by decreasing the number of model parameters by 40.8% (from 2.6M to 1.54M) and the computational complexity by 28.1% (from 6.4 to 4.6 GFLOPs). However, this efficiency comes at the expense of accuracy, as mAP@0.5 drops by 1.7% to 82.5% and recall drops by 2.0% to 74.6%. This shows the classic trade-off between efficiency and accuracy in designing lightweight models. The C3k2_RVB_ECA module is the most accurate of all the components. It increases mAP@0.5 by 5.8% to 90.0% and precision by 2.9% to 96.1%. The attention mechanism also significantly improves recall. It increases from 6.7% to 83.3%. This improvement increases the number of parameters slightly to 1.97M (24.2% less than the baseline), whilst maintaining an appropriate computational efficiency of 5.9 GFLOPs (7.8% less). The Detect_LSDECD head provided balanced improvements. It reaches the highest precision (96.4%) whilst keeping strong mAP@0.5 (89.3%) and recall (83.0%). It reduces the parameters to 1.82M (a 30.0% decrease). It provides a minimal computational overhead (6.1 GFLOPs, a 4.7% decrease). It shows that the architecture has been well optimized. Interestingly, combining DWConv with C3k2_RVB_ECA gives mixed results. The accuracy metrics show a drop compared to using C3k2_RVB_ECA alone, especially in recall (75.3% vs 83.3%). However, the parameter reduction (51.2% to 1.27M) and computational savings (23.4% to 4.9 GFLOPs) are significant. This implies possible interference between the efficiency-oriented DWConv and the accuracy-boosting attention mechanisms. The complete proposed model, EHALEYO, is the best combination of these components, striking an optimal balance across all metrics. Overall, the average detection accuracy of the EHALEYO model was 7.0% higher than that of the YOLOv11n model, significantly improving most of the metrics.

TABLE III: Ablation experiments were conducted using the proposed model.

Model	Prec(%)	Rec(%)	mAP0.5(%)	mAP50–95(%)	Params	GFLOPs
YOLO11n	93.2	76.6	84.2	53.3	2.6	6.4
DWConv	92.5	74.6	82.5	51.4	1.54	4.6
RVB_ECA	96.1	83.3	90.0	58.9	2.01	6.0
Detect_LSDECD	96.4	83.0	89.3	57.5	1.91	6.1
DWConv+RVB_ECA	92.1	75.3	89.7	54.8	1.27	4.9
Ours(Full)	**95.8**	**85.8**	**91.8**	**59.4**	**1.03**	**3.9**

TABLE IV: Performance comparison of different models on the DUT dataset

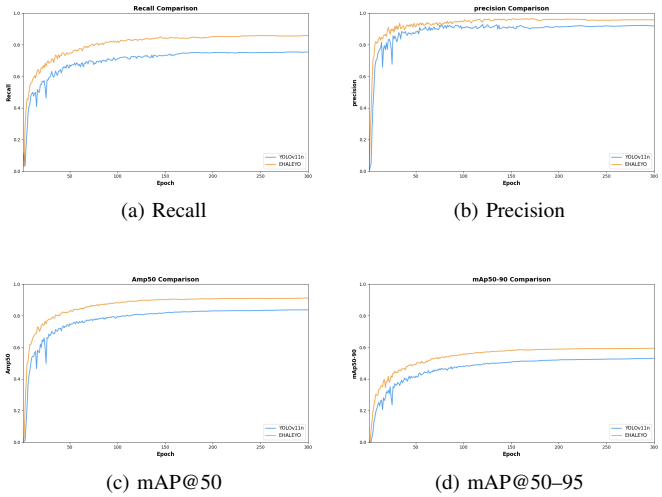
Models	Prec(%)	Rec(%)	mAP50(%)	mAP50-95(%)	Param/10^6	GFLOPs/G	size(MB)
YOLOv5	93.2	74.1	82.2	51.3	2.22	5.9	4.6
YOLOv8	92.9	74.1	82.8	52.1	2.72	6.9	5.6
YOLOv9t	93.4	73.9	82.4	53.3	1.76	6.7	4.2
YOLOv11	93.2	76.6	84.2	53.3	2.6	6.4	5.5
ET-YOLOv6n	78.9	72.1	90.6	58.8	3.2	7.9	-
SEB-YOLOv8s	95.9	83.1	90.5	61.5	12.3	-	25
EHALEYO	95.8	**85.8**	**91.8**	**59.4**	**1.03**	**3.9**	**2.6**

2) Comparison Results with Other Baseline Methods: We conducted a series of comparison experiments (see Table IV), comparing the proposed EHALEYO with several traditional YOLO models as well as models that currently perform better on the DUT dataset, such as SEBYOLOv8s [20] and ET-YOLOv6n [21], which are trained under the same conditions. It can be seen from the results in Table IV that the proposed EHALEYO outperforms the other models, achieving mostly the highest accuracy of 95.8%, superpassing most other models, which range from 78.9% (ET-YOLOv6n) to 93.4% (YOLOv9t). It is slightly less than SEB-YOLOv8s, which is 95.9%. EHALEYO introduce better results regarding a recall of 85.8%, exceeding the other models, which range from 72.1% (ET-YOLOv6n) to 83.1% (SEB-YOLOv8s). The mAP50 metric indicates that EHALEYO scores 91.8%, which is significantly higher than other methods that range from 82.2% (YOLOv5) to 90.6% (ET-YOLOv6n). As shown in Table IV EHALEYO achieves the best results, with 1.03M Params, 3.9G FLOPs, and a model size of 2.6MB. Thus, EHALEYO excels in both detection accuracy and lightweight design. Figure 5 indicates that the proposed EHALEYO outperforms the other models for accuracy on a similar DUT dataset.

TABLE V: Performance comparison of different models on the VisDrone dataset

Models	Prec(%)	Rec(%)	mAP50(%)	mAP50-9(%)	Param/10^6	GFLOPs
YOLOv8	42.3	33.3	33.2	19.2	2.72	6.9
YOLOv11	42.1	33.2	33.1	19.1	2.6	6.4
BMGS-YOLOv8 [13]	-	-	30.7	17.3	2.79	-
LMAD-YOLO [14]	39.3	29.4	26.8	14.9	-	-
RSW-YOLO [15]	41.4	33.4	30.9	17.6	11.86	38.4
Zhu et al. [16]	42.7	31.0	31.1	17.2	2.37	11.9
EHALEYO	**46.7**	**35.1**	**35.7**	**20.9**	**1.03**	**3.9**

As shown in Table V, EHALEYO outperforms other methods on the VisDrone dataset in both accuracy and efficiency. It achieves the highest recall (35.1%), precision (46.7%), mAP@50 (35.7%), and mAP@50–95 (20.9%). It is obviously outperforming baseline and enhanced YOLO variants. These improvements are accomplished with only 3.9 GFLOPs and 1.03 M parameters, which are incredibly lower than those of competing models. This indicates that EHALEYO supplies an incredible efficiency–accuracy trade-off, making it well-suited for deployment in resource-constrained edge environments.



(a) Recall (b) Precision

(c) mAP@50 (d) mAP@50–95

Fig. 5: Training curves comparison between EHALEYO and YOLOv11. (a) Recall; (b) Precision; (c) mAP@50; (d) mAP@50–95.

3) Edge-Embedded Device Experiment: We successfully deployed the trained UAV drone detection model on the Raspberry Pi 5 (CPU: Quad-core 64-bit Broadcom BCM2712 Arm Cortex-A76 @ 2.4 GHz, RAM: LPDDR4X memory of 16 GB) and conducted detection tests. Fig. 6 compares the original YOLOv11n with EHALEYO, both deployed on the Raspberry Pi 5, for UAV drone detection. The results show that the proposed EHALEYO model achieves higher confidence levels and improved localization accuracy. We execute EHALEYO and YOLOv11n models on Raspberry Pi 5 with 640×640 input to show the efficiency of the proposed EHALEYO model (see Table VI).

TABLE VI: Performance comparison of EHALEYO and YOLOv11n models on Raspberry Pi 5 with 640×640 input

Metric	EHALEYO	YOLOv11n	Improvement (%)
Latency per frame (ms)	273.3	392.3	30.3% lower
FPS	3.66	2.55	43.5% higher
Estimated energy per inference (J)	1.64	2.35	30.2% lower

EHALEYO outperforms YOLOv11n on the Raspberry Pi 5 by increasing throughput to 3.66 FPS (43.5% improvement) and decreasing latency by 30.3%. Additionally, it exhibits superior efficiency for real-time edge deployment, reducing energy consumption per inference by 30.2%.

V. Conclusion and Perspectives

This work introduces an Edge AI-Based High-Accuracy, Lightweight, Enhanced YOLOv11 (EHALEYO) for real-time small UAVs detection in complex environments. It enhances



(a) EHALEYO (b) YOLOv11n

(c) EHALEYO (d) YOLOv11n

Fig. 6: Visual comparison between EHALEYO (left) and YOLOv11n (right). Each row shows one pair of corresponding images.

feature extraction by replacing YOLOv11's backbone convolution with c3k2-RVB-ECA. EHALEYO also replaced the standard convolution layers with DWConv to reduce the number of parameters and keep the accuracy at an acceptable level. The addition of a P2 detection layer to EHALEYO increases the accuracy for small, distant targets, whereas a lightweight LSDECD head reduces the parameter count and computational load. Real Experiments on the Raspberry Pi 5 using DUT and VisDrone datasets demonstrate superior accuracy and efficiency. EHALEYO outperforms YOLOv11n in terms of FPS, per-frame latency, and energy consumption per inference, offering an effective edge AI–ready solution for UAV detection in complex environments. The future work would consider using a more advanced feature extraction method to improve the recall rate and accuracy of small objects. We also plan to design efficient network architectures and lightweight frequency domain filters to further decrease the computational complexity of our model.

ACKNOWLEDGMENT

This work was conducted as part of project EMULATE funded by the European Union and the German Federal Ministry of Economics and Climate Action under research grant 13IPC012.

REFERENCES

[1] W. Yu and K. Mo, "Yolo-gcof: A lightweight low-altitude drone detection model," *IEEE Access*, 2025.

[2] Q. Guo, N. Liu, Z. Wang, and Y. Sun, "Review of deep learning based object detection algorithms," *J. Detect. Control*, vol. 45, pp. 10–20, 2023.

[3] J. Zhao, J. Zhang, D. Li, and D. Wang, "Vision-based anti-uav detection and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 323–25 334, 2022.

[4] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 11, pp. 7380–7399, 2021.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[8] C. Kantor, B. Rauby, L. Boussioux, E. Jehanno, and H. Talbot, "Overcam: Gradient-based localization and spatial attention for confidence measure in fine-grained recognition using deep neural networks," 2020.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[10] 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*, 2016, pp. 779–788.

[11] Z. Cao, J. Li, D. Zhang, M. Zhou, and A. Abusorrah, "A multi-object tracking algorithm with center-based feature extraction and occlusion handling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 4464–4473, 2022.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[13] Y. Ning and J. Chen, "Target detection algorithm for uav aerial images based on bmgs-yolov8," in *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*. IEEE, 2025, pp. 1088–1092.

[14] X. Xing, F. Luo, L. Wan, K. Lu, Y. Peng, and X. Tian, "Lmad-yolo: A vehicle image detection algorithm for drone aerial photography based on multi-scale feature fusion," *PLoS One*, vol. 20, no. 7, p. e0328248, 2025.

[15] H. Wang, J. Shang, X. Wang, Q. Zhang, X. Wang, J. Li, and Y. Wang, "Rsw-yolo: A vehicle detection model for urban uav remote sensing images," *Sensors*, vol. 25, no. 14, p. 4335, 2025.

[16] H. Zhu and X. Xie, "Enhanced yolov11n for small object detection in uav imagery: higher accuracy with fewer parameters," *Scientific Reports*, 2026.

[17] Y. Cao, D. Pang, Q. Zhao, Y. Yan, Y. Jiang, C. Tian, F. Wang, and J. Li, "Improved yolov8-gd deep learning model for defect detection in electroluminescence images of solar photovoltaic modules," *Engineering Applications of Artificial Intelligence*, vol. 131, p. 107866, 2024.

[18] X. Li, G. Li, Z. Zhang, and H. Sun, "Research on improved algorithms for cone bucket detection in formula unmanned competition," *Sensors*, vol. 24, no. 18, p. 5945, 2024.

[19] A. Wang, H. Chen, Z. Lin, J. Han, and G. Ding, "Repvit: Revisiting mobile cnn from vit perspective," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 15 909–15 920.

[20] A. Fang, S. Feng, B. Liang, and J. Jiang, "Real-time detection of unauthorized unmanned aerial vehicles using seb-yolov8s," *Sensors*, vol. 24, no. 12, p. 3915, 2024.

[21] S. Feng, X. Pu, Y. Hu, Q. Zeng, and X. Wang, "Et-yolov6: Tiny object detection algorithm on uav targets based on star operation and cross-level fusion mechanism," 2025.