DOI 10.5281/twist.10049652#359

TWIST, 2025, Vol. 20, Issue 1, pp. 99-106

TWIST

Journal homepage: www.twistjournal.net

A Comparative Study of Pruning Strategies for EfficientNet and ResNet Architectures in Search and Rescue Applications

Onyekachi Njoku
Department of Electronic and Computer Engineering, University of Nigeria, Nsukka, Nigeria

Anioke Chidera*
Department of Electronic and Computer Engineering, University of Nigeria, Nsukka, Nigeria
[**Corresponding author*]

Imaga George
Department of Electronic and Computer Engineering, University of Nigeria, Nsukka, Nigeria

Arum Kingsley
Department of Statistics, University of Nigeria, Nsukka, Nigeria

Abstract
Nigeria's security challenges necessitate effective surveillance solutions. This study evaluates EfficientNet-B0, EfficientNet-B6, and ResNet-18 for Search and Rescue (SAR) operations using computer vision. We applied L1 Unstructured and Random Unstructured pruning techniques to assess each model's accuracy and computational efficiency. Through preprocessing diverse images and testing pre-trained models, our findings reveal that L1 unstructured pruning significantly improves processing times while preserving accuracy. Among the evaluated backbones, EfficientNet-B0 with L1 pruning emerged as the most efficient and accurate for SAR applications. This study offers valuable insights into selecting optimal computer vision models and pruning strategies for enhanced real-world surveillance.

Keywords
Pruning, Search, Rescue, Computer Vision, EfficientNet, ResNet

INTRODUCTION

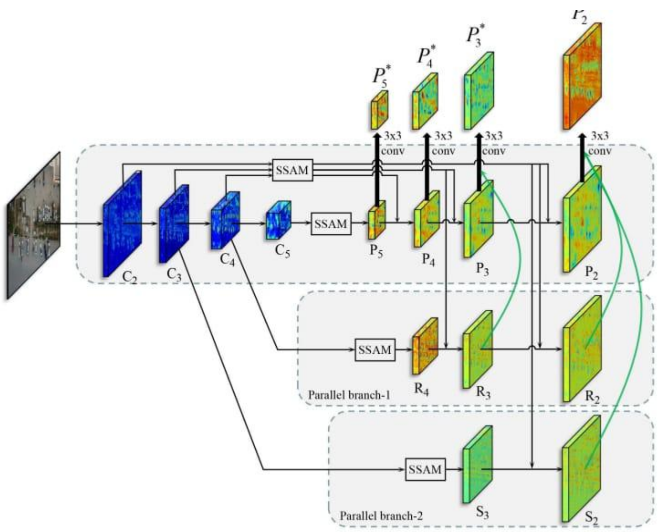
Nigeria's security architecture must be robust to effectively tackle rising security threats. UAVs equipped with sophisticated computer vision capabilities present a promising solution for addressing Nigeria's security challenges. UAV technology has found applications across various sectors, including agriculture (L. Wang et al., 2019), courier services (Eun et al., 2019), etc. Extensive research has been conducted in object detection (Krizhevsky et al., 2017; Liu et al., 2020; Szegedy et al., 2015), significantly enhancing UAV capabilities in various fields, particularly surveillance, through the implementation of deep convolutional neural networks (DCNNs)(Indolia et al., 2018). These advancements have facilitated the development of object detection models such as the YOLO series (Nnadozie, Casaseca-de-la-Higuera, et al., 2023; Nnadozie, Iloanusi, et al., 2023), R-CNN series (Girshick et al., 2014; Hsu et al., 2018; S. Ren et al., 2016; Zhang et al., 2019) and many others. These models have undergone rigorous testing on diverse image datasets such as Pascal VOC (Everingham et al., 2010) and MS COCO (Lin et al., 2014), achieving remarkable success. However, as new research progresses and more complex models are developed, they often become computationally demanding, resulting in increased processing times and requiring advanced processors.

Deep Convolutional Neural Network (DCNN) architectures represent a cornerstone in the field of computer vision, revolutionizing the way we approach tasks such as image classification, object detection, and image segmentation. These architectures, ranging from classic designs like VGG (Nakada et al., 2017) and GoogLeNet (Szegedy et al., 2015) to more recent innovations like ResNeXt (Xie et al., 2017) and DenseNet (Huang et al., 2017), have significantly advanced the state-of-the-art in visual recognition tasks. Each architecture brings its own unique characteristics and strengths to the table, whether it is the simplicity and depth of VGG, the efficiency and scalability of MobileNet (Howard et al., 2017) or the dense connectivity of DenseNet. Moreover, the emergence of neural architecture search techniques has led to the discovery of novel architectures like NASNet (Zoph et al., 2018), tailored to specific tasks and datasets. With their ability to capture intricate features and patterns from raw pixel data, DCNN architectures continue to drive progress in computer vision, enabling breakthroughs in a wide range of applications across various domains.

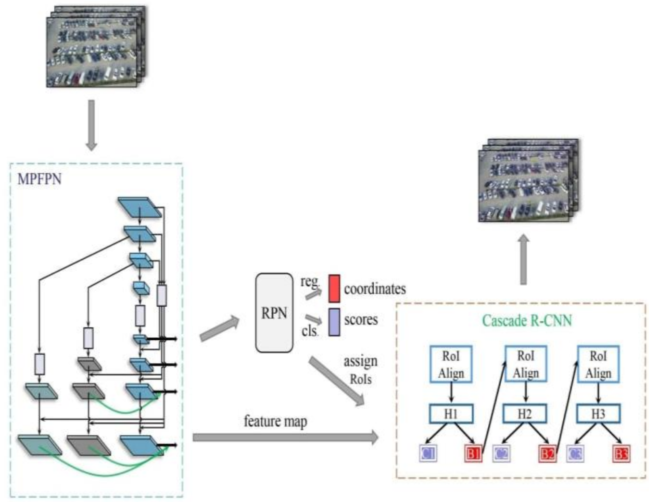
Tan & Le (2019) show the architecture of an EfficientNet in Fig. 2. It offers a compelling solution for reducing computational resources and processing time in deep learning tasks. Its innovative approach to compound scaling allows for efficient adjustment of the network's width, depth, and resolution, ensuring that the model achieves optimal performance with minimal computational overhead. By carefully balancing these scaling factors, EfficientNet architectures can achieve state-of-the-art accuracy while using significantly fewer parameters and computations compared to other architectures. This reduction in model complexity translates directly into faster inference times, making EfficientNet an ideal choice for scenarios where computational resources are limited or where real-time processing is essential. Moreover, the lightweight nature of EfficientNet makes it well-suited for deployment on edge devices and mobile platforms, enabling efficient and scalable solutions for a wide range of applications, from image classification to object detection and beyond. Therefore, EfficientNet emerges as a potent choice as the DCNN architecture for this research, particularly in the context of SAR operations where speed is paramount. Thakur & Hemanth (2021) addressed the crucial task of pedestrian detection in surveillance applications, recognizing its significance in various tasks such as person identification, counting, and tracking. Their research conducts an extensive evaluation of state-of-the-art algorithms on the MOT20 dataset (Dendorfer et al., 2020) and a custom dataset recorded using an Unmanned Aerial Vehicle (UAV). Three popular object detection models, namely Faster R-CNN, SSD, and YOLO, are tested, with YOLOv5 emerging as the top performer with 61% precision and 44% F-measure. The research done by Thakur & Hemanth (2021) underscores the importance of tailored object detection solutions for surveillance scenarios. Of all the DNN models for image recognition mentioned by Lyu et al. (2024), the ResNet is the most accurate, with a 3.57% error against a human error of 5% on the same image recognition tasks (Alom et al., 2018). This is why ResNet-18 is chosen alongside, EfficientNet-B0, and EfficientNet-B6 as suitable models employed in this research.

This paper builds upon pre-trained object detection models, introduces some adjustments to these object detection models and makes comparative analysis. In work done by Liu et al. (2020), whose model architecture is shown in Fig. 1a and 1b, two parallel branches were incorporated into the conventional FPN network to capture features that might have been missed in the deeper layers. Additionally, a cascade architecture was implemented within the Fast R-CNN stage to enhance the model's localization capabilities, leading to improved accuracy in object detection. This method by Liu et al. (2020) represents a novel approach distinct from many other solutions (Chen et al., 2019; Dai et al., 2017; Ma et al., 2019; Rabbi et al., 2020; Y. Ren et al., 2018; J. Wang et al., 2019; Zhuang et al., 2019) for small object detection, which often face challenges in adequately capturing features due to pixel limitations. However, in an attempt to increase the robustness of object detection, the model in Liu et al. (2020) increases processing time, which would not be ideal for search and rescue (SAR) operations, where rapid response and real-time processing are critical. To address this challenge and enhance efficiency, two variants of EfficientNet (EfficientNet-B0 and –B6) along with ResNet (ResNet-18) were pruned. By pruning these models, we aim to reduce their complexity while preserving their overall effectiveness, thus enabling faster inference without compromising performance.

In this paper, our primary focus is on reducing processing time, given the urgency of security situations, even at the expense of sacrificing a slight degree of accuracy. This implies that the impact of different pruning methods on the inference time of various computer vision architectures, specifically EfficientNet-B0, EfficientNet-B6, and ResNet-18 will be evaluated.



(a)



(b)

Fig. 1 Model architecture in Liu et al. (2020). (a) Architecture of Multi-branch Parallel FPN (b) Architecture of entire framework

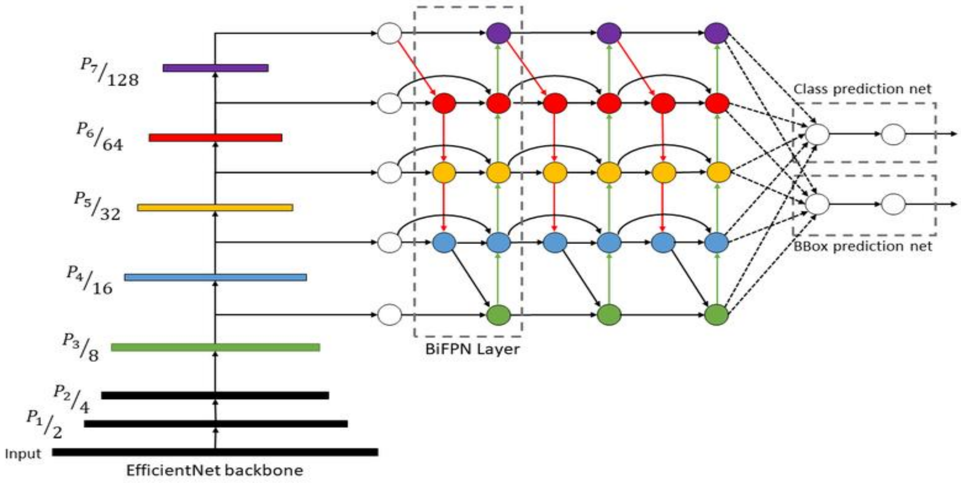


Fig. 2 EfficientNet Architecture (Buongiorno et al., 2022)

The main contributions of this paper can be summarized as:

(a) Model Selection and Preparation:
- Select pre-trained models of EfficientNet-B0, EfficientNet-B6, and ResNet-18 for comparative analysis.
- Implement appropriate preprocessing techniques for the input images to ensure consistency across all models.

(b) Implementation of Pruning Methods:
- Apply L1Unstructured pruning to each of the selected models.
- Apply Random Unstructured pruning to each of the selected models.
- Ensure the pruning process is correctly implemented and does not compromise the integrity of the models.

(c) Inference Time Measurement:
- Measure and record the inference time of each model before any pruning is applied.
- Measure and record the inference time of each model after L1Unstructured pruning.
- Measure and record the inference time of each model after Random Unstructured pruning.

The rest of this paper is organized as follows: Section 2 discusses the research method employed and the experiment done. The results of the experiments, comparison and analysis are discussed in section 3. The conclusion and recommendations are presented in section 4.

MATERIALS AND METHODS

This research evaluates the suitability of various computer vision architectures, specifically EfficientNet-B0, EfficientNet-B6, and ResNet-18, for surveillance applications in Nigeria, with a focus on Search and Rescue (SAR) operations. The most optimal model in terms of accuracy and computational efficiency was identified after applying pruning techniques in order to reduce the models' computational demands.

Model description

Four random images were downloaded from the internet to serve as the evaluation dataset. These images were chosen without regard to specific categories, quality, or resolution consistency, reflecting a practical scenario where diverse image types might be encountered in real-world applications. Each image underwent several preprocessing steps to ensure compatibility with the input requirements of the models. This included resizing the images to 128x128 pixels, center-cropping to 224x224 pixels, converting them to tensors, and normalizing them using ImageNet's standard mean and standard deviation values: mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225].

The study employed pre-trained versions of EfficientNet-B0, EfficientNet-B6, and ResNet-18, obtained from popular deep learning frameworks such as PyTorch. These models were used in their original form as baselines for comparison, without any modifications to their architecture or weights. The primary performance metrics assessed were each model's top-1 accuracy and inference time, which are crucial for determining their suitability for SAR operations.

Two pruning techniques were applied to evaluate their effects on model size, computational complexity, and inference speed, The L1 Unstructured Pruning and the Random Unstructured Pruning. L1 Unstructured Pruning was performed globally across specific layers in all models, targeting 20% of weights with the lowest L1 norm for removal. This technique was implemented using PyTorch's 'torch.nn.utils.prune' module to systematically reduce the model's parameter count and enhance computational efficiency. In contrast, Random Unstructured Pruning was applied with the same 20% pruning rate, but the weights selected for removal were chosen randomly without considering their magnitude or impact on the model. This provided a contrasting approach to L1 Unstructured Pruning by introducing randomness.

Post-pruning, modifications were made to the output layers of the models, such as the fully connected layer in EfficientNet, to adapt to the reduced model size. However, no re-training was conducted after pruning; the models were evaluated immediately to observe the effects of pruning on performance metrics.

The experiments were conducted on a laptop with an Intel Core i3 processor and 12GB RAM, which provided a realistic context for computational efficiency in environments with limited resources. The experiments were implemented using the PyTorch framework, and inference times for each image were measured using Python's `time` module. Both pruned and unpruned models were tested individually, with processing times recorded before and after pruning to assess the computational gains achieved.

The comparison of results was observational, based on single-run tests for each model variant (pruned and unpruned). The primary metrics for comparison included inference time (in seconds) and the top-1 predicted class probability.

RESULTS

The images used for classification included those of a goldfish (GF), an African chameleon (AC), a great white shark (GWS), and a bucket (BKT). These images were selected to represent a diverse range of objects commonly encountered in real-world scenarios.The results of these experiments, including accuracy metrics and inference times, are summarized in Table 1 to Table 6 of the paper. By analyzing the performance of the pruned and compressed model across different image classes, we aimed to evaluate its effectiveness for image classification tasks relevant to SAR operations in Nigeria.

In the tables below, T1 represents the time taken to make predictions before changes were made to the pre-trained models. T2 represents the time taken to make predictions after changes were made. P1 represents the most probable object predicted by the model before changes with its probability while P2 represents the most probable object predicted by the model after changes with its probability. In this experiment, some predictions made after pruning (P2), specifically in experiments 2, 4, 5, and 6, produced outputs that were entirely different from the objects used for testing. The acronyms corresponding to these new predictions were defined within each of these experiments.

Experiment 1

-EfficientNet-B0

-L1Unstructured

Table 1 Summary of EfficientNet-B0 L1Unstructured pruned model testing

Image Size	T1(s)	T2(s)	P1	P2	Actual
370 x 234	0.2069	0.1689	GF-0.885	GF-0.870	GF
360 x 257	0.1659	0.1589	AC-0.766	AC-0.826	AC
1582 x 1186	0.2079	0.1879	GWS-0.581	GWS-0.384	GWS
474 x 603	0.1659	0.1599	BKT-0.951	BKT-0.954	BKT

Experiment 2

-EfficientNet-B0

-Random Unstructured

- PK – Pick
- MZ – Maze
- BIN – Binder
- CUP – Cup

Table 2 Summary of EfficientNet-B0 Random Unstructured pruned model testing

Image Size	T1 (s)	T2 (s)	P1	P2	Actual
370 x 234	0.1699	0.1649	GF-0.885	PK-0.020	GF
360 x 257	0.1759	0.1699	AC-0.766	MZ-0.011	AC
1582 x 1186	0.1939	0.1769	GWS-0.748	BIN-0.022	GWS
474 x 603	0.1739	0.1649	BKT-0.951	CUP-0.018	BKT

Experiment 3
-EfficientNet-B6
-L1Unstructured

Table 3 Summary of EfficientNet-B6 L1 Unstructured pruned model testing

Image Size	T1 (s)	T2 (s)	P1	P2	Actual
370 x 234	0.7756	0.7746	GF-0.899	GF-0.886	GF
360 x 257	0.8445	0.8245	AC-0.674	AC-0.643	AC
1582 x 1186	1.0654	0.8385	GWS-0.694	GWS-0.672	GWS
474 x 603	0.8675	0.8175	BKT-0.822	BKT-0.798	BKT

Experiment 4
-EfficientNet-B6
-Random Unstructured

- HC – Honey Comb
- SYG – Syringe
- SRW – Screw
- MSK – Match Stick

Table 4 Summary of EfficientNet-B6 Random Unstructured pruned model testing

Image Size	T1 (s)	T2 (s)	P1	P2	Actual
370 x 234	0.9205	0.8145	GF-0.899	HC-0.175	GF
360 x 257	0.8775	0.8305	AC-0.674	SYG-0.067	AC
1582 x 1186	0.8955	0.8295	GWS-0.694	SRW-0.031	GWS
474 x 603	0.8495	0.8295	BKT-0.822	MSK-0.048	BKT

Experiment 5
-ResNet-18
-L1Unstructured

- HMH – Hammer Head
- CCK – Cock
- OCH – Ostrich

Table 5 Summary of ResNet-18 L1Unstructured pruned model testing

Image Size	T1 (s)	T2 (s)	P1	P2	Actual
370 x 234	0.0890	0.1469	GF-1.000	HMH-0.106	GF
360 x 257	0.1299	0.1339	AC-0.994	CCK-0.107	AC
1582 x 1186	0.0820	0.1319	GWS-0.983	HMH-0.108	GWS
474 x 603	0.1039	0.1269	BKT-0.928	OCH-0.112	BKT

Experiment 6
-ResNet-18
-Random Unstructured

- ER – Electric Ray
- SR – Sting Ray

Table 6 Summary of ResNet-18 Random Unstructured pruned model testing

Image Size	T1 (s)	T2 (s)	P1	P2	Actual
370 x 234	0.1110	0.1269	GF-1.000	GWS-0.104	GF
360 x 257	0.1119	0.1419	AC-0.994	GWS-0.104	AC
1582 x 1186	0.0899	0.1449	GWS-0.983	ER-0.108	GWS
474 x 603	2.2687	0.1569	BKT-0.928	SR-0.104	BKT

The following graphical representations make comparisons between the experiments 1-6;

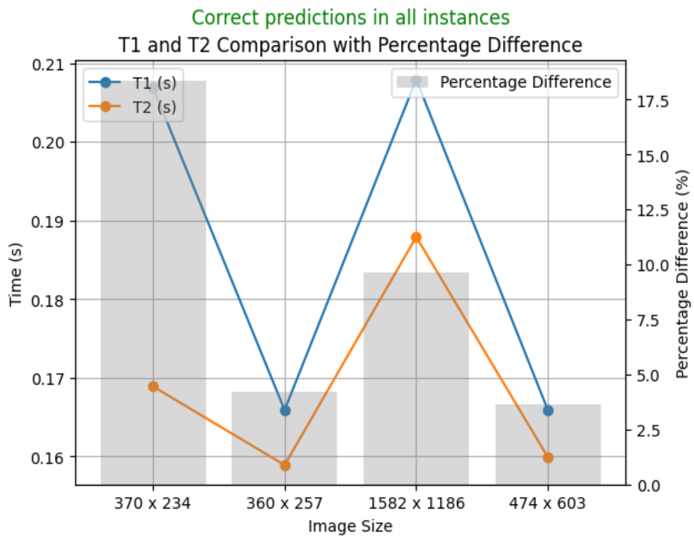


Fig. 3 Graphical outcome of experiment 1

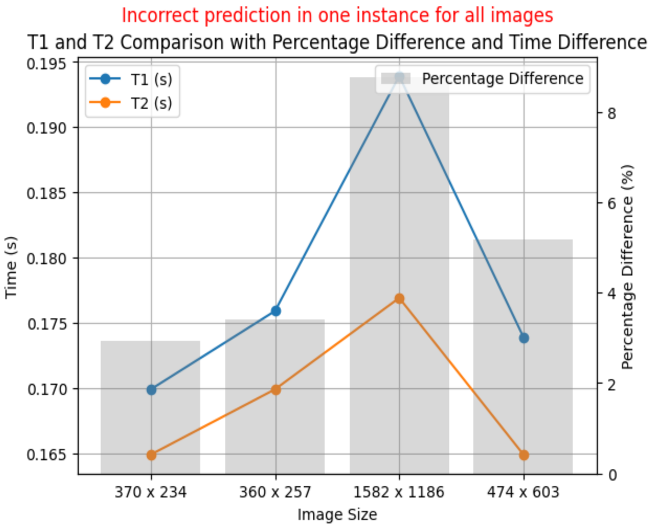


Fig. 4 Graphical outcome of experiment 2

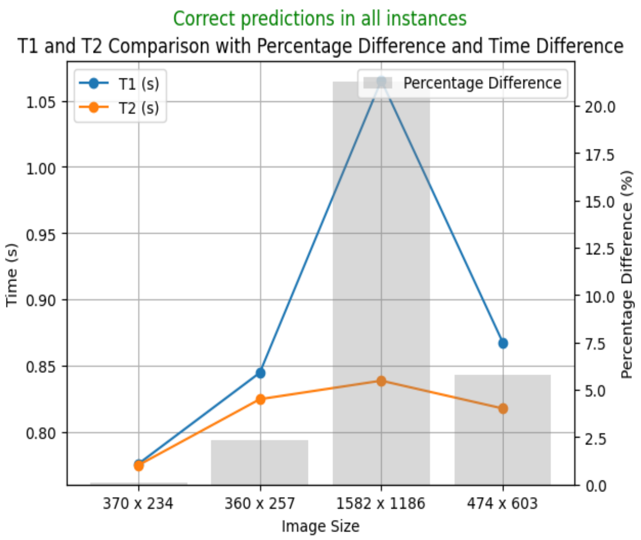


Fig. 5 Graphical outcome of experiment 3

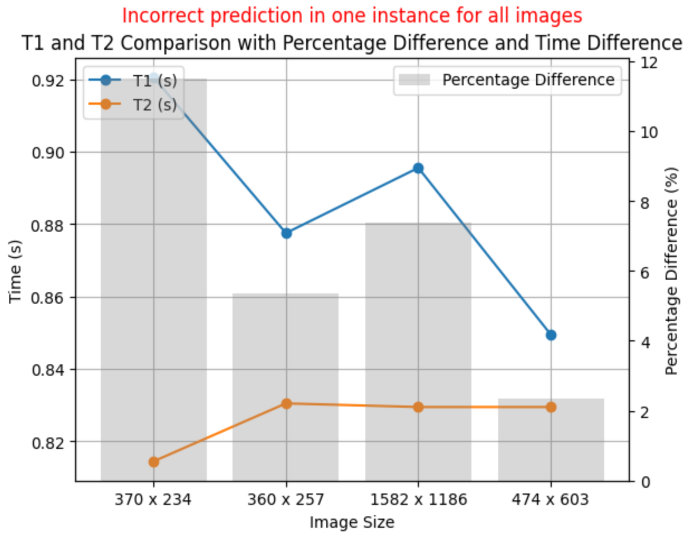


Fig. 6 Graphical outcome of experiment 4

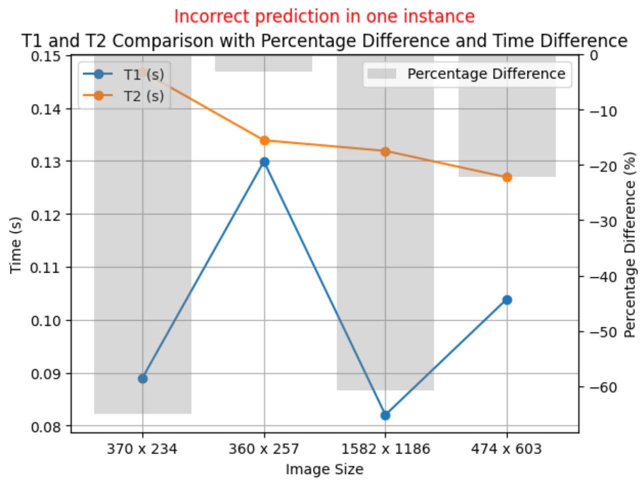


Fig. 7 Graphical outcome of experiment 5

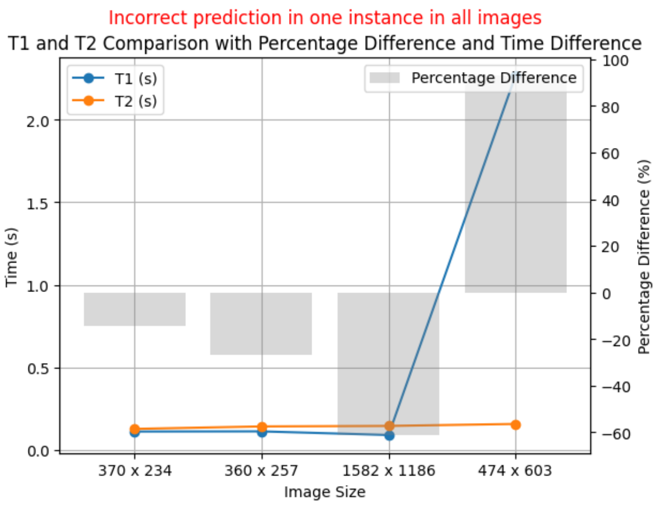


Fig. 8 Graphical outcome of experiment 6

DISCUSSION

Graphical results from experiments 1-4 show a consistent reduction in processing time after pruning (T2) compared to the initial time (T1). However, while both pruning methods improved processing time for the EfficientNet models, it can be seen from Table 2 and Table 4 that Random Unstructured pruning method significantly reduces accuracy of prediction. For example, in Table 2, the Image with size 370x234 is a Goldfish, however after pruning, the model predicted a Pick with a 2% probability. This trend of wrong predictions continued throughout the experiment and can also be seen for experiment 4.

For the pruned ResNet-18 models, neither processing time nor accuracy were improved. This is evident in experiments 5 and 6. Therefore, none of the pruning methods proposed in this paper are useful for the ResNet models.

L1 Unstructured pruning significantly enhances the performance of both EfficientNet-B0 and EfficientNet-B6 as shown in experiments 1 and 3. Although Random Unstructured and L1 Unstructured pruning methods improved the processing time for the EfficientNet models, only L1 Unstructured maintained accurate predictions after pruning. Therefore, we conclude that among the pruning techniques applied to EfficientNet and ResNet-18, L1 Unstructured on EfficientNet is the most effective in terms of performance and accuracy.

CONCLUSION

Overall, this research provides a structured approach to evaluating the performance of different computer vision backbones and pruning techniques for surveillance applications in Nigeria. By balancing model size, computational efficiency, and accuracy, this study aims to offer valuable insights for selecting the most suitable model architecture and pruning strategy for real-world SAR operations. L1 unstructured pruning on EfficientNet architectures significantly reduces processing time, making it highly suitable for object detection in Search and Rescue (SAR) operations. This pruning technique maintains the high accuracy of the model while streamlining its computational efficiency, ensuring that object detection tasks are performed swiftly and reliably. The ability to balance speed with precision is essential in SAR scenarios, where timely and accurate detection can be life-saving. Therefore, adopting L1 unstructured pruning for EfficientNet models can greatly enhance the effectiveness of SAR operations.

ACKNOWLEDGEMENTS

The authors appreciate the reviewers and editors whose contributions have immensely improved the quality of this article.

FUNDING INFORMATION

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

DECLARATION OF COMPETING INTERESTS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

1. Alom, Z., Taha, T. M., Yakopcic, C., Westberg, S., Hasan, M., Esesn, B. C. Van, Awwal, A. A. S., & Asari, V. K. (2018). The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. In *Preprint on ArXiV* (pp. 1–39).
2. Buongiorno, D., Caramia, D., Di Ruscio, L., Longo, N., Panicucci, S., Di Stefano, G., Bevilacqua, V., & Brunetti, A. (2022). Object Detection for Industrial Applications: Training Strategies for AI-Based Depalletizer. *Applied Sciences (Switzerland)*, *12*(22), 1–14. https://doi.org/10.3390/app122211581
3. Chen, C., Zhong, J., & Tan, Y. (2019). Multiple-oriented and small object detection with convolutional neural networks for aerial image. *Remote Sensing*, *11*(18), 1–23. https://doi.org/10.3390/rs11182176
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable Convolutional Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 764–773. https://doi.org/10.1109/ICCV.2017.89
5. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., & Leal-taix, L. (2020). MOT20 : A benchmark for multi object tracking in crowded scenes. *Preprint on ArXiV*, 1–7.
6. Eun, J., Song, B. D., Lee, S., & Lim, D. E. (2019). Mathematical investigation on the sustainability of UAV logistics. *Sustainability (Switzerland)*, *11*(21), 1–15. https://doi.org/10.3390/su11215932
7. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4
8. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference in COmputer Vision and Pattern Recognition*, 580–587.
9. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications* (Issue Preprint on ArXiv, pp. 1–10). https://doi.org/10.48550/arXiv.1704.04861
10. Hsu, S. C., Huang, C. L., & Chuang, C. H. (2018). Vehicle detection using simplified fast R-CNN. *2018 International Workshop on Advanced Image Technology, IWAIT 2018*, 1–3. https://doi.org/10.1109/IWAIT.2018.8369767
11. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243
12. Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, *132*, 679–688. https://doi.org/10.1016/j.procs.2018.05.069
13. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

14. Lin, T., Zitnick, C. L., & Doll, P. (2014). Microsoft COCO : Common Objects in Context. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland*, 740–755.
15. Liu, Y., Yang, F., & Hu, P. (2020). Small-Object Detection in UAV-Captured Images via Multi-Branch Parallel Feature Pyramid Networks. *IEEE Access*, *8*(2020), 145740–145750. https://doi.org/10.1109/ACCESS.2020.3014910
16. Lyu, Z., Yu, T., Pan, F., Zhang, Y., Luo, J., & Zhang, D. (2024). A survey of model compression strategies for object detection. *Multimedia Tools and Applications*, *83*(16), 48165–48236. https://doi.org/10.1007/s11042-023-17192-x
17. Ma, W., Guo, Q., Wu, Y., Zhao, W., Zhang, X., & Jiao, L. (2019). A novel multi-model decision fusion network for object detection in remote sensing images. *Remote Sensing*, *11*(7), 1–18. https://doi.org/10.3390/rs11070737
18. Nakada, M., Wang, H., & Terzopoulos, D. (2017). AcFR: Active Face Recognition Using Convolutional Neural Networks. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 35–40. https://doi.org/10.1109/CVPRW.2017.11
19. Nnadozie, E. C., Casaseca-de-la-Higuera, P., Iloanusi, O., Ani, O., & Alberola-López, C. (2023). Simplifying YOLOv5 for deployment in a real crop monitoring setting. *Multimedia Tools and Applications*, *0123456789*. https://doi.org/10.1007/s11042-023-17435-x
20. Nnadozie, E. C., Iloanusi, O. N., Ani, O. A., & Yu, K. (2023). Detecting Cassava Plants under Different Field Conditions Using UAV-Based RGB Images and Deep Learning Models. *Remote Sensing*, *15*(9), 1–18. https://doi.org/10.3390/rs15092322
21. Rabbi, J., Ray, N., Schubert, M., Chowdhury, S., & Chao, D. (2020). Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network. *Remote Sensing*, *12*(9), 1–25. https://doi.org/10.3390/RS12091432
22. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149.
23. Ren, Y., Zhu, C., & Xiao, S. (2018). Small object detection in optical remote sensing images via modified Faster R-CNN. *Applied Sciences (Switzerland)*, *8*(5), 1–11. https://doi.org/10.3390/app8050813
24. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594
25. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *36th International Conference on Machine Learning, ICML 2019*, 10691–10700.
26. Thakur, N., & Hemanth, J. (2021). Object Detection in Deep Surveillance. *Preprint on Research Square*, 1–33. https://doi.org/https://doi.org/10.21203/rs.3.rs-901583/v1 License:
27. Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C. C., & Lin, D. (2019). CARAFE: Content-aware reassembly of features. *Proceedings of the IEEE International Conference on Computer Vision*, 3007–3016. https://doi.org/10.1109/ICCV.2019.00310
28. Wang, L., Lan, Y., Zhang, Y., Zhang, H., Tahir, M. N., Ou, S., Liu, X., & Chen, P. (2019). Applications and prospects of agricultural unmanned aerial vehicle obstacle avoidance technology in China. *Sensors (Switzerland)*, *19*(3), 1–16. https://doi.org/10.3390/s19030642
29. Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 5987–5995. https://doi.org/10.1109/CVPR.2017.634
30. Zhang, X., An, G., & Liu, Y. (2019). Mask R-CNN with Feature Pyramid Attention for Instance Segmentation. *International Conference on Signal Processing Proceedings, ICSP*, 1194–1197. https://doi.org/10.1109/ICSP.2018.8652371
31. Zhuang, S., Wang, P., Jiang, B., Wang, G., & Wang, C. (2019). A single shot framework with multi-scale feature fusion for geospatial object detection. *Remote Sensing*, *11*(5), 1–20. https://doi.org/10.3390/rs11050594
32. Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8697–8710. https://doi.org/10.1109/CVPR.2018.00907