

Gated Scene Specific YOLO: A Pruning and Gating Approach for Efficient Object Detection

Hector Acosta¹[0009–0003–6784–8882] and Jung Soon Ki¹[1111–2222–3333–4444]

Kyungpook National University, Daegu 41566, Republic of Korea lncs@springer.com
<http://www.springer.com/gp/computer-science/lncs>

Abstract. In the realm of object detection, the YOLO (You Only Look Once) architecture has marked a significant advancement, particularly for real-time applications. However, when deployed in varied and dynamic environments, especially on edge devices such as security cameras and traffic monitoring systems, the need for further optimization becomes apparent. Our work introduces "Gated Scene Specific YOLO," a novel adaptation of the YOLO architecture that incorporates a dynamic gating mechanism for enhanced efficiency. Traditional YOLO models, while robust, often process excessive data that may include redundant or irrelevant features for specific scenes. Our approach counters this by selectively activating neural pathways relevant to the input scene's unique attributes. This selective activation is achieved through an initial phase of dynamic gate generation during training, followed by a scene-specific analysis to determine gates that are consistently inactive. These identified gates are then permanently deactivated for the specific scene, allowing for a streamlined and efficient model tailored to the environment's characteristics. This method significantly reduces computational load while maintaining, and in some cases, improving the accuracy of the detection, making it particularly suitable for resource-constrained devices. Our experimental results show that Gated Scene Specific YOLO not only achieves notable improvements in processing speed but also maintains high detection accuracy, underscoring the potential of this approach to enhance real-time object detection in a variety of settings. This research contributes to the field by demonstrating a pragmatic approach to optimize deep learning models for specific operational contexts, paving the way for more adaptable and resource-efficient object detection solutions.

Keywords: Object Detection· YOLO Architecture· Scene-Specific Optimization· Gated Neural Networks· Model Pruning

1 Introduction

Object detection is a cornerstone of computer vision, impacting a wide array of applications from surveillance to autonomous driving. The evolution of deep learning has propelled advancements in this domain, particularly through the YOLO (You Only Look Once) architecture by Redmon [1]. Renowned for its

efficiency in real-time processing, YOLO is a testament to the power of modern object detection systems from high-powered computing environments to more performance-constrained devices. While YOLO excels in real-time processing across these settings, its adaptability still faces challenges, especially when deployed on edge devices where computational resources are inherently limited. To address these challenges, our research specifically focuses on YOLOv6 by Li et al. [4, 5], known for its hardware-focused design and efficiency in real-time applications, forming an ideal base for our enhancements. In these scenarios, even the slightest optimization can yield significant improvements in speed and efficiency. Addressing this limitation, our research introduces "Gated Scene Specific YOLO," a novel approach that integrates dynamic gating with scene-specific model pruning within the YOLO architecture. In the world of neural network optimization, model pruning emerges as a promising solution to alleviate computational intensity. This technique focuses on trimming redundant or less significant parameters from a neural network, thus streamlining its structure with minimal impact on performance. Our methodology advances this concept by employing a dynamic gating mechanism that adapts to the unique features of the input scene, enhancing the object detection process's efficiency without compromising accuracy. A key innovation in our approach is the use of Improved SemHash, a method initially introduced by Kaiser and Bengio [2] and further explored by Chen et al. [3]. This technique allows for the generation of binary gates during training, crucial for selectively activating or deactivating specific network filters in response to input variations. The process involves dynamically generating gates during training and then conducting an analysis phase for a specific scene. This analysis identifies gates that are consistently inactive and exports them for static application in real-world scenarios. As a result, the Gater Network, typically part of the backbone during training, can be excluded during actual deployment, relying instead on these statically determined gates for efficient and tailored object detection. Our research introduces a significant advancement in the YOLO architecture through the integration of a gating network, intelligently deactivating filters based on the unique features of each input scene. This novel approach, enhanced by the incorporation of Improved SemHash [2, 3], not only allows for precise network activity control but also brings considerable improvements in computational efficiency and detection accuracy. Extensive experimental validation, focusing on metrics such as FLOPs, FPS, and mAP@0.5:0.95, demonstrates the efficacy of our strategy as we observed a % reduction in FLOPs and an increase of % in FPS compared to the YOLO counterpart of our pruned model, all without compromising the robustness of detection as evidenced by stable mAP scores.

1.1 A Subsection Sample

Please note that the first paragraph of a section or subsection is not indented. The first paragraph that follows a table, figure, equation etc. does not need an indent, either.

Subsequent paragraphs, however, are indented.

Sample Heading (Third Level) Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Sample Heading (Fourth Level) The contribution should contain no more than four levels of

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	Lecture Notes	14 point, bold
1st-level heading	1 Introduction	12 point, bold
2nd-level heading	2.1 Printing Area	10 point, bold
3rd-level heading	Run-in Heading in Bold. Text follows	10 point, bold
4th-level heading	<i>Lowest Level Heading.</i> Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z$$

(1)

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 1).

Fig. 1. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal

Acknowledgments. A bold run-in heading in small font size at the end of the paper is used for general acknowledgments, for example: This study was funded by X (grant number Y).

Disclosure of Interests. It is now necessary to declare any competing interests or to specifically state that the authors have no competing interests. Please place the statement with a bold run-in heading in small font size beneath the (optional) acknowledgments¹, for example: The authors have no competing interests to declare that are relevant to the content of this article. Or: Author A has received research grants from Company W. Author B has received a speaker honorarium from Company X and owns stock in Company Y. Author C is a member of committee Z.

References

1. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)

¹ If EquinOCS, our proceedings submission system, is used, then the disclaimer can be provided directly in the system.