

Original Paper

GreenCOD: A Green Camouflaged Object Detection Method

Hong-Shuo Chen¹, Yao Zhu¹, Suyu You², Azad M. Madni¹ and C.-C. Jay Kuo¹

¹*University of Southern California, Los Angeles, California, USA*

²*DEVCOM Army Research Laboratory, Adelphi, Maryland, USA*

ABSTRACT

We introduce GreenCOD, a green method for detecting camouflaged objects distinct in its avoidance of backpropagation techniques. GreenCOD leverages gradient boosting and deep features extracted from pre-trained Deep Neural Networks. Traditional camouflaged object detection approaches rely on complex deep neural networks, seeking performance improvements by backpropagation-based fine-tuning. However, such methods are typically computationally demanding and exhibit only marginal performance variations across different models. It raises the question of whether effective training can be achieved without backpropagation. In this direction, our work proposes a new paradigm that utilizes gradient boosting for COD. This approach significantly simplifies the model design, resulting in a system that requires fewer parameters and operations and maintains high performance compared to state-of-the-art deep learning models. Remarkably, our models are trained without backpropagation and achieve the best performance with fewer than 20G Multiply-Accumulate Operations. This new, more efficient paradigm opens avenues for further exploration in green, backpropagation-free model training. We make GreenCOD source code and on-device demo available at <https://greencod.ai/> for further research.

*Corresponding author: Hong-Shuo Chen, hongshuo@usc.edu

Keywords: Green Learning, Extreme Gradient Boosting (XGBoost), Camouflaged Object Detection

1 Introduction

The study of Camouflaged Object Detection (COD) stands at the forefront of computer vision research, delving into the challenge of identifying objects expertly concealed within their environments. COD transcends the limitations of traditional image segmentation [22, 12, 31, 14] by addressing the intricate task of detecting objects that seamlessly blend into their surroundings. This field tackles a range of camouflages, from the subtle color shifts in a chameleon to the strategic patterns of military uniforms and even the natural disguise of predators like lions in grasslands. The ability to detect such hidden entities has profound implications for various applications, pushing the boundaries of what computer vision can achieve.

The applications of COD are diverse and far-reaching. In wildlife conservation, for instance, it can be used for monitoring and studying naturally camouflaged animals, aiding in population tracking and behavioral research. Enhanced COD systems can improve surveillance and reconnaissance capabilities in military and defense, offering a tactical advantage in detecting camouflaged equipment or personnel. Effective COD in autonomous vehicles and robotics is crucial for navigating complex environments and ensuring safety and efficiency. Additionally, in healthcare, advanced COD techniques could assist in identifying subtle patterns in medical imagery [25], potentially aiding in early disease detection. Thus, the advancements in COD challenge our understanding of visual perception and unlock new possibilities across a spectrum of disciplines.

Recent progress in deep learning has significantly advanced the COD field, introducing an array of sophisticated methods [10, 32, 48, 21, 24, 28, 1, 26] and models dedicated to the precise identification of hidden objects. Central to these developments is the use of backpropagation in training deep neural networks. This fundamental algorithm, crucial for adjusting network weights based on error rates, has enabled the refinement of complex models to detect subtle and elusive camouflaged objects. These networks, characterized by their intricate structures and extensive backpropagation training processes, have achieved notable success in COD. However, this comes with a caveat. The reliance on backpropagation often means these systems demand high computational resources and involve complex designs, including extensive data processing and iterative adjustments for model fine-tuning. As a result, while models exhibit incremental improvements, they often do so with increased computational demands. It presents practical challenges, particularly in real-world scenarios where efficiency and resource management are vital. Additionally,

models trained with backpropagation can exhibit a black-box nature, where the internal decision-making processes are not transparent, posing challenges in interpretability.

A compelling question emerges: Can COD models be effectively trained without relying on backpropagation? Investigating this prospect could pave the way for developing more efficient and transformative models in the COD field. In a paradigm where backpropagation is absent, we unveil GreenCOD, a groundbreaking approach in the COD field that depends on gradient-boosting capabilities. At the heart of GreenCOD is the strategic employment of extreme gradient boosting (XGBoost), a variant of gradient boosting that excels in handling large-scale and complex data. Our method ingeniously integrates the power of XGBoost with the deep features extracted from pre-trained Deep Neural Networks (DNNs). GreenCOD applies a multi-scale analysis framework, leveraging the structured approach of gradient-boosting trees. The model works by analyzing layered images, beginning with a broad, coarse-level detection that identifies general areas of interest where camouflage might exist. It then progressively moves to finer scales, enhancing the details and improving the precision of the segmentation. This hierarchical processing allows GreenCOD to pinpoint camouflaged objects with impressive accuracy.

This innovative approach transcends the typical confines of back propagation-based models, offering a more interpretable and transparent learning trajectory. By doing so, GreenCOD sets a new precedent for future COD models, showcasing that high efficiency and environmental consciousness can go hand-in-hand without compromising detection capabilities. This paper addresses a primary concern: Can we develop a model that retains efficacy in COD tasks but is more efficient, interpretable, and environmentally friendly? With GreenCOD, we believe we have taken a significant step in that direction. Our code and data are publicly available at: <https://greencod.ai/>. We also provide an on-device demo to demonstrate the effectiveness of our method.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. The GreenCOD method is presented in Section 3. Experiments are shown in Section 4. Finally, concluding remarks are given in Section 5.

2 Related Work

2.1 Recent Approaches in COD

In recent years, various strategies have emerged to tackle the COD challenge. [10] laid the groundwork by introducing a foundational framework SINet dedicated to identifying camouflaged objects within images. Following this initiative, different network architectures and feature aggregation methods are proposed.

Network Architectures and Features Aggregation: The D²C-Net, introduced by [34], employs a dual-branch, dual-guidance, and cross-refine network to enhance detection performance. Similarly, [32] proposed the C²F-Net, a context-aware cross-level fusion network, to leverage contextual information for improved detection across different levels. [52] took a novel architectural approach by introducing the CubeNet, which features X-shape connections. For segmentation of camouflaged objects, [28, 27] utilized distraction mining in their PFNet. The exploration of neighbor connection and hierarchical information transfer, termed NCHIT, was discussed in the work of [40]. Additionally, [44] presented the TPRNet, a transformer-induced progressive refinement network. The feature aggregation and propagation network (FAPNet) was developed by Zhou *et al.* [46], while M. Zhang *et al.* [43] proposed Preynet, featuring a bidirectional bridging interaction module. The recent introduction of Camoformer by Yin *et al.* [38], which applies masked separable attention, demonstrates ongoing advancements in the field. Lastly, Ji *et al.* [15] highlighted the pursuit of optimization in this field through their efficient approach using deep gradient learning.

Uncertainty Methodology: In uncertainty exploration, [21] introduced JSCOD, an uncertainty-aware method for the joint detection of salient and camouflaged objects. Building on this concept, Liu *et al.* [23] proposed OCENet, a detection model that integrates aleatoric uncertainty. Further extending the application of uncertainty in detection methodologies, [36] focused on a transformer reasoning approach guided by uncertainty, named UGTR, to enhance the detection capabilities.

Texture, Edge, and Frequency Information:

Several methods have leveraged additional information, such as texture, edge, and boundary, to improve performance. TINet, introduced by [48], utilizes texture awareness through a texture-aware interactive guidance network and texture labels. Focusing on boundary awareness, [30] developed BAS, a segmentation network for mobile and web applications. Several methods have effectively employed edge information, including BSANet [47], BGNet [33], and the Edge-based reversible re-calibration network, ERRNet [16]. Each of them enhances detection performance through an edge-centric approach. Additionally, the exploration of frequency domain analysis by FNet [45] highlights the diversification of methodologies in this field. Furthermore, R. He *et al.* [13] demonstrated performance improvements using weakly-supervised learning with scribble annotations.

Diverse Methodologies:

Exploring a multifaceted strategy, [24] introduced Rank-Net, a novel approach designed to simultaneously localize, segment, and rank camouflaged objects, concurrently performing these tasks. In a different vein, [39] proposed a method incorporating mutual graph learning, specifically R-MGL, and S-MGL, to enhance detection and segmentation capabilities. Further diversifying

the field, Pang *et al.* [29] developed a mixed-scale triplet network, broadening the scope of methodological approaches. Additionally, Wu *et al.* [35] broke new ground with their source-free depth approach, enabling the reasoning of camouflaged objects in 3D space.

2.2 Green Learning

The innovative framework of Green Learning, as introduced by [18], represents a paradigm shift in the computational strategies of modern artificial intelligence. Distinctly moving away from the reliance on deep learning methodologies, this approach pivots towards more computation-efficient machine learning techniques, thereby addressing the escalating resource demands of conventional AI systems.

At the core of Green Learning lies the strategic abandonment of backpropagation, a staple in traditional neural network training. Instead, it harnesses the potential of unsupervised feature extraction, utilizing either the Saab Transform [19] or its advanced iteration, the channel-wise Saab Transform [6]. This methodological transition facilitates more nuanced and efficient data processing, enabling the extraction of diverse features without the computational burden of backpropagation algorithms.

Further enhancing its efficacy, Green Learning employs sophisticated feature selection mechanisms, namely the discriminant feature test (DFT) and the relevant feature test (RFT) [37]. These techniques are instrumental in isolating a subset of discriminant features and are pivotal for the subsequent stages of model training. This selective approach ensures that only the most relevant and impactful features are carried forward, optimizing both the training process and the performance of the final model.

To train these discriminant features, Green Learning leverages various advanced algorithms, including XGBoost, Logistic Regression, SVM, and SLM [11]. Each of these methodologies brings unique strengths to the table, allowing for a flexible and robust training process tailored to the specific characteristics of the data set and the task at hand.

The hallmark of Green Learning is its operational efficiency, characterized by the absence of backpropagation and end-to-end training requirements. It reduces the computational load and enhances the framework’s scalability and applicability across various domains.

The practical applications of Green Learning have been demonstrated across various fields, showcasing its versatility and effectiveness. Notable examples include its role in deepfake detection [3, 2], where it has been instrumental in identifying and mitigating the spread of synthetic media. In the realm of geographic forensics [4, 5], Green Learning has provided new avenues for analyzing and interpreting geographic data with greater accuracy and efficiency. Additionally, its application in image forensics [49, 50, 51] and texture analysis

[42, 41] further underscores its potential in enhancing our understanding and processing of visual information.

In summary, Green Learning emerges as a transformative approach in artificial intelligence, offering a sustainable, efficient, and versatile data processing and analysis framework. It redefines the computational paradigms of AI and paves the way for more resource-efficient and scalable solutions across many applications.

3 GreenCOD Method

GreenCOD, which stands for Green Camouflaged Object Detection, is poised to revolutionize the COD field by forgoing the traditional reliance on back-propagation. It seeks to maintain high efficiency and performance standards while dramatically reducing the computational complexity typically measured by Multiply-Accumulate Operations (MACs) and the overall number of model parameters.

In our approach, we draw upon the strengths of the U-Net architecture. It is renowned for its adeptness in feature extraction across various scales and its capability to refine segmentation iteratively from broader strokes down to finer details. We have innovated upon this model by replacing the expansive pathway found on the right-hand side of U-Net with Extreme Gradient Boosting (XGBoost). This integration taps into XGBoost’s proficiency in identifying objects camouflaged within their surroundings.

A key benefit of GreenCOD is the circumvention of the exhaustive end-to-end training that deep learning models usually demand. Utilizing XGBoost contributes to a leaner model in terms of parameters and obviates the need for backpropagation in the training phase. This break from end-to-end training introduces a modular and adaptable methodology that differentiates our model from standard deep learning practices. To our knowledge, GreenCOD is the first to harness the power of XGBoost to detect concealed objects, marking a groundbreaking advancement in object detection.

In Figure 1, the proposed method integrates the power of deep learning with the robustness of gradient-boosted trees to achieve sophisticated COD. It adopts a multi-resolution approach, utilizing feature extraction and multi-scale XGBoost to effectively capture object hierarchies in images. Additionally, the method involves neighborhood construction to enhance context awareness during segmentation.

3.1 Feature Extraction

The initial phase of our process is the feature extraction stage, where the input image is resized to 672x672 and processed through the EfficientNetB4

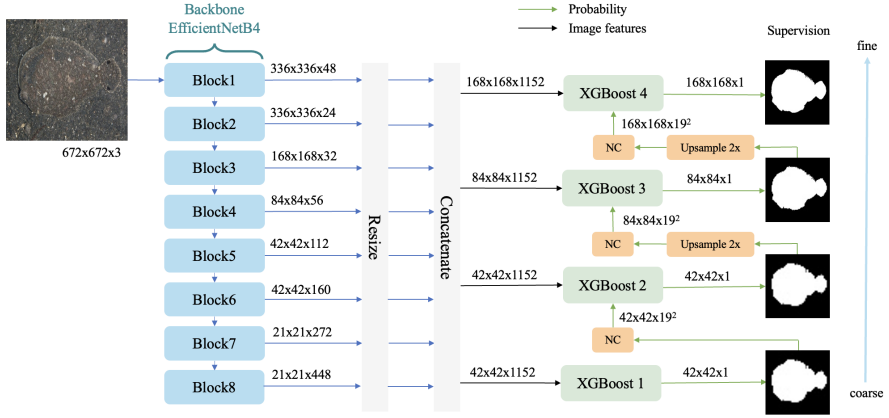


Figure 1: An overview of the GreenCOD method, where the input is an image of dimension $672 \times 672 \times 3$, and the output is a probability mask of dimension $168 \times 168 \times 1$. NC stands for Neighborhood Construction.

backbone. The EfficientNetB4 architecture is recognized for its exceptional ability to extract high-quality features and is considered cutting-edge in deep learning. As the image traverses through the sequence of eight blocks, labeled Block1 to Block8, it is processed by an array of convolutional, pooling, and normalization operations. This block progression allows the model to capture a comprehensive range of features—from the fine-grained details to the broader semantic aspects. Given that the backbone has been pre-trained on the expansive ImageNet database, we eliminate the need for further fine-tuning, thereby streamlining the model’s training process.

3.2 Concatenation and Resizing

We will bring the feature maps to uniform dimensions suitable for each processing stage once we derive the feature maps from the EfficientNetB4 backbone. Specifically, the input features of XGBoost 1 and XGBoost 2 are resized to dimensions of 42×42 . For the XGBoost 3, the maps are resized to 84×84 , while for XGBoost 4, they are resized to 168×168 . All features from Block 1 through Block 8, encompassing 1152 channels, are merged into a single cohesive structure. This standardization of the feature maps results in a comprehensive multi-resolution image representation spanning a range of scales and complexities. Such an arrangement is pivotal for the model’s proficiency in detecting and delineating objects and patterns of various sizes within the image.

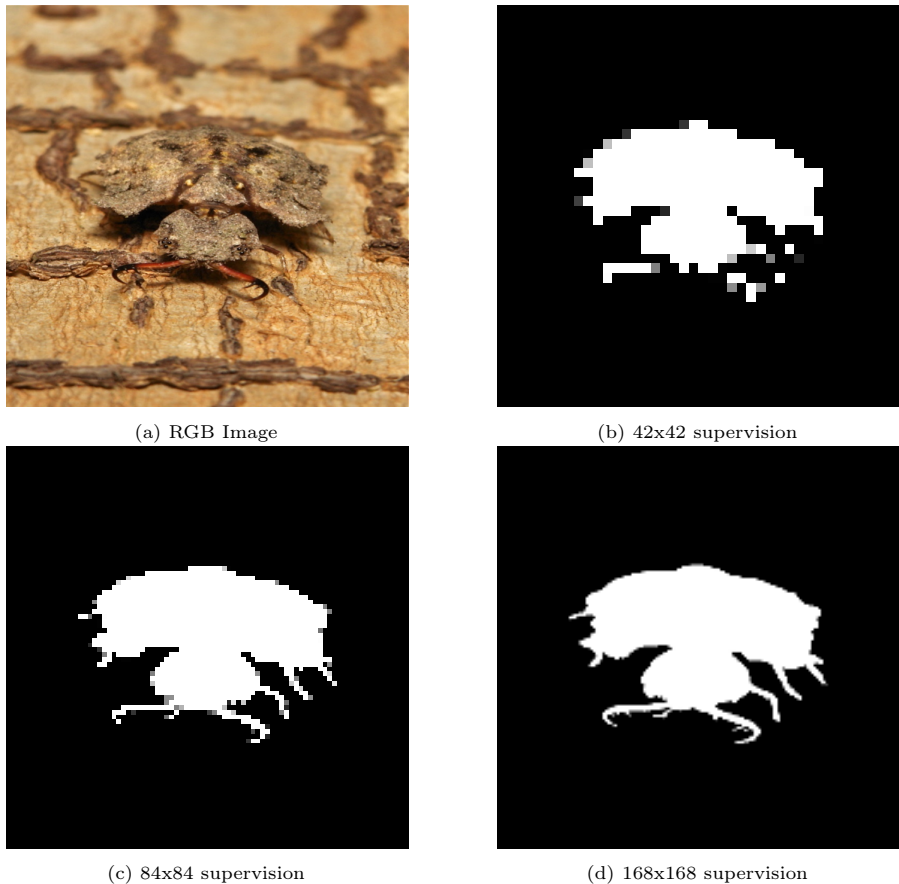


Figure 2: The illustration of the multi-scale supervision of each XGBoost

3.3 Multi-scale XGBoost

We delve into the sophisticated design of the XGBoost gradient-boosting framework, a technique favored for its effectiveness with structured data. In our innovative application, XGBoost is adapted to process image feature data derived from the previous concatenation of multi-scale feature maps. This multi-scale approach means the feature data is analyzed at various resolutions, each managed by a dedicated XGBoost model.

Our model is structured in a staged fashion, where each stage of XGBoost addresses a specific level of detail within the image. The process begins with XGBoost 1, which manages the broadest feature representation at a resolution of 42x42, setting the stage for the initial detection of camouflaged objects.

The following stages, XGBoost 2 and XGBoost 3, escalate in resolution to 42x42 and 84x84, respectively, progressively refining the detection accuracy and bringing the focus to subtler details of the camouflaged objects. XGBoost 4 is the terminating stage, which operates at the most refined resolution of 168x168, meticulously capturing the most intricate details for a comprehensive final detection. In Figure 2, we show the supervision at multiple scales, ranging from 42x42 to 168x168.

In the stages of XGBoost 2, 3, and 4, the methodology incorporates the predictions from the preceding XGBoost model, focusing exclusively on the discrepancies between these predictions and the actual ground truth. This approach is rooted in the core principles of boosting, where each model iteratively corrects the errors of its predecessor, thereby enhancing the overall predictive accuracy and reliability of the object detection process. This multi-scale approach ensures accurate and robust detection across various object sizes and complexities, strengthening the model's overall performance and reliability.

3.4 Neighborhood Construction (NC)

We examine a pivotal stage following each XGBoost analysis. The "Neighborhood Construction" phase is integral to our segmentation method, enhancing the model's context-aware capabilities. During this phase, the probabilities surrounding each pixel or region are aggregated, providing a richer dataset from which the model can draw more accurate segmentations. Such contextually enriched information is critical to increasing the precision with which the model delineates segmented areas, ensuring that objects and regions within the image are defined with clarity and correctness. The window size is a hyperparameter, and we set it to 19x19 in our experiment. Let's denote:

- $P(x, y)$ as the probability map output by the XGBoost model for a pixel at location (x, y) in the image. This map indicates the probability that each pixel belongs to a particular segment or class.
- W as the window size for the neighborhood, which is 19×19 in our case, leading to a total of 361 pixels in the neighborhood.
- $N_{x,y}$ as the neighborhood matrix formed around the pixel (x, y) , with dimensions equal to the window size W .

Given a pixel at location (x, y) , the neighborhood $N_{x,y}$ can be constructed by aggregating the probabilities of the pixels falling within the 19×19 window centered at (x, y) . Mathematically, this can be represented as follows:

$$N_{x,y} = \left\{ P(i, j) \mid i \in \left[x - \frac{W-1}{2}, x + \frac{W-1}{2} \right], j \in \left[y - \frac{W-1}{2}, y + \frac{W-1}{2} \right] \right\}.$$

This neighborhood matrix $N_{x,y}$ is then flattened into a vector $PF_{x,y}$ with dimension 361, which represents the new feature derived from the neighborhood for the pixel at (x, y) :

$$PF_{x,y} = \text{flatten}(N_{x,y}).$$

This feature vector $F_{x,y}$ is concatenated with other relevant features for the pixel at (x, y) , forming an enriched feature set used for the final segmentation prediction. The concatenation can be denoted as follows, where $IF_{x,y}$ represents other existing image features for the pixel:

$$F_{x,y} = [IF_{x,y} \parallel PF_{x,y}]$$

Our proposed approach to COD is a hybrid one, combining the strengths of the deep learning model with the gradient-boosted modeling. It harnesses the feature extraction capabilities of the EfficientNetB4 architecture, the layered analytical power of multi-scale XGBoost processing, and the contextual insights afforded by Neighborhood Construction. This integration enables the model to produce high-accuracy and high-resolution segmentations.

4 Experiments

4.1 Datasets

In our experiment, we maintain consistency with the methodology of previous experiments. Training is performed on a dataset that combines the CAMO [20] and COD10K [10] datasets, totaling 4040 images. Testing is carried out on two datasets: COD10K and NC4K [24]. The COD10K dataset contains 2026 images. The NC4K dataset is the largest dataset for testing, with 4121 images.

4.2 Evaluation Metrics

To benchmark the performance of our proposed method, we conducted a comprehensive comparison with the state-of-the-art methods employing identical evaluation metrics. The comparative analysis focused on several critical aspects including Mean Absolute Error (MAE), Structural measure, Enhanced-alignment Measure, and F-measure, where W and H are the width and height of the images respectively, $G(x, y)$ represents the pixel value of the Groundtruth at coordinates (x, y) , and $P(x, y)$ represents the pixel value of the prediction at coordinates (x, y) .

- The Mean Absolute Error (MAE) is computed as:

$$\mathcal{M} = \frac{1}{W \times H} \sum_x \sum_y |P(x, y) - G(x, y)| \quad (1)$$

The function $|P(x, y) - G(x, y)|$ computes the absolute difference between the corresponding pixel values of the two masks.

- The Structural measure [7] is given by:

$$S_\alpha = (1 - \alpha)S_o(P, G) + \alpha S_r(P, G), \quad (2)$$

where α serves to adjust the balance between the object-aware similarity S_o and the region-aware similarity S_r . Following the convention established in the original publication, we set α to a default value of 0.5.

- The Enhanced-alignment Measure [9] is computed as:

$$E_\phi = \frac{1}{W \times H} \sum_x \sum_y \phi[P(x, y), G(x, y)] \quad (3)$$

The function ϕ is the enhanced alignment matrix applied to the pixel values from masks P and G .

- The F-measure is given by:

$$F_\beta = \frac{(1 + \beta^2)\text{Precision} \times \text{Recall}}{\beta^2\text{Precision} + \text{Recall}}, \quad (4)$$

where the term $\beta^2 = 0.3$ gives more weight to the precision than the recall in the computation, as suggested in the previous work.

The comparative analysis results underscore our method’s efficacy and robustness, showcasing superior or comparable performance across the evaluated metrics.

4.3 Experiment results

Table 1 presents a comparative analysis of our proposed GreenCOD method against other leading-edge methods from recent literature, utilizing the COD10K dataset. The comparison includes explicitly models that operate under the computational threshold of 50G Multiply-Accumulate Operations (MACs) to ensure computational efficiency. Remarkably, our GreenCOD achieves the highest F-measure and the lowest Mean Absolute Error (MAE) with just 24.34 million parameters and 16.22 G MACs. This performance is notably superior to that of SegMaR, which requires 56.21 million parameters and 33.63 G MACs.

The favorable balance between performance and efficiency that GreenCOD offers illustrates its potential as a robust architecture worthy of further investigation. While GreenCOD does not secure the top spot in E-measure—where it ranks third, behind SegMaR and DGNet—it still demonstrates commendable overall efficacy.

In Table 2, our focus shifts from evaluating our proposed method against smaller models to benchmarking it alongside larger-scale models. This table is confined to models exceeding the computational complexity of 50G Multiply-Accumulate Operations (MACs). Although our model does not outperform the leading method, CamoFormer-C, it is essential to note that CamoFormer-C demands fourfold more parameters and a threefold increase in MACs compared to our model. Upon examining the Mean Absolute Error (MAE) and F-measure metrics, our model outperforms 11 of the 16 methods considered, all of which have significantly larger model sizes than ours. Regarding E-measure, our model surpasses 10 out of the 16 methods. Notably, our method substantially reduces MACs compared with R-GML, plummeting from 249.89G to 16.22G. This reduction translates to an energy consumption decrease by a factor of 15, emphasizing our model’s enhanced efficiency.

In Table 3, we extend the evaluation of our model to the NC4K dataset, currently the most extensive testing set, to assess our model’s ability to generalize across extensive conditions. Our model secures a second-place ranking in Mean Absolute Error (MAE), matching the performance of SegMaR while boasting a significantly smaller model size and fewer Multiply-Accumulate Operations (MACs). Introduced in 2023, DGNet leads the pack for models under 50 G MACs, with 19.22 million parameters and 2.77G MACs, achieving the best results. Nonetheless, our model stands out by offering greater interpretability. Moreover, it eliminates the need for end-to-end training of the entire model, thereby forgoing any requirement for backpropagation—an advantage that DGNet does not provide.

In Table 4, about the NC4K dataset, we assess our model alongside larger models with computational complexities exceeding 50G Multiply-Accumulate Operations (MACs). Our model demonstrates robustness by outscoring 7 of the 13 models in Mean Absolute Error (MAE), F-measure, and E-measure. This performance underscores the effectiveness of our model on the NC4K dataset, showcasing its capability to generalize successfully to larger datasets.

4.4 Visualization analysis

As illustrated in Figures 3 and 4, our attention is drawn to segmenting large concealed objects. In the first row, our model demonstrates exceptional detail in segmenting the camouflaged object, precisely identifying the butterfly with remarkable accuracy. The second row showcases the model’s capability to differentiate subtle details, such as the bird’s tail. The third row presents a

Table 1: Comparison of performance metrics between proposed and benchmark methods on the COD10K dataset. Only models with less than 50G Multiply-Accumulate Operations (MACs) were considered. The top-performing method for each metric on each dataset is highlighted in bold, while the second-best method is underscored.

| Model | Pub/Year | Input | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $M \downarrow$ | $E_\phi^{mn} \uparrow$ | Para. | MACs |
|-------------------|----------|------------------|---------------------|----------------------|----------------|------------------------|---------|--------|
| SINet [10] | CVPR'20 | 352 ² | 0.776 | 0.631 | 0.043 | 0.864 | 48.95M | 19.42G |
| C2FNet [32] | IJCAI'21 | 352 ² | 0.813 | 0.686 | 0.036 | 0.890 | 28.41M | 13.12G |
| TINet [48] | AAAI'21 | 352 ² | 0.793 | 0.635 | 0.042 | 0.861 | 28.56M | 8.58G |
| JSCOD [21] | CVPR'21 | 352 ² | 0.809 | 0.684 | 0.035 | 0.884 | 121.63M | 25.20G |
| LSR [24] | CVPR'21 | 352 ² | 0.804 | 0.673 | 0.037 | 0.880 | 57.90M | 25.21G |
| PFNet [28] | CVPR'21 | 416 ² | 0.800 | 0.660 | 0.040 | 0.877 | 45.64M | 26.54G |
| C2FNet-V2 [1] | TCSVT'22 | 352 ² | 0.811 | 0.691 | 0.036 | 0.887 | 44.94M | 18.10G |
| ERRNet [16] | PR'22 | 352 ² | 0.786 | 0.630 | 0.043 | 0.867 | 69.76M | 20.05G |
| TPRNet [44] | TVCJ'22 | 352 ² | 0.817 | 0.683 | 0.036 | 0.887 | 32.95M | 12.98G |
| FAPNet [46] | TIP'22 | 352 ² | 0.822 | 0.694 | 0.036 | 0.888 | 29.52M | 29.69G |
| BSANet [47] | AAAI'22 | 384 ² | 0.818 | 0.699 | 0.034 | 0.891 | 32.58M | 29.70G |
| SegMaR [17] | CVPR'22 | 352 ² | 0.833 | 0.724 | 0.034 | 0.899 | 56.21M | 33.63G |
| SINetV2 [8] | TPAMI'22 | 352 ² | 0.815 | 0.680 | 0.037 | 0.887 | 26.98M | 12.28G |
| CRNet [13] | AAAI'23 | 320 ² | 0.733 | 0.576 | 0.049 | 0.832 | 32.65M | 11.83G |
| DGNet-S [15] | MIR'23 | 352 ² | 0.810 | 0.672 | 0.036 | 0.888 | 7.02M | 2.77G |
| DGNet [15] | MIR'23 | 352 ² | 0.822 | 0.693 | 0.033 | 0.896 | 19.22M | 1.20G |
| GreenCOD-D3-1000 | - | 672 ² | 0.797 | 0.701 | 0.033 | 0.881 | 16.83M | 13.70G |
| GreenCOD-D3-10000 | - | 672 ² | 0.807 | 0.715 | 0.032 | 0.893 | 17.62M | 15.06G |
| GreenCOD-D6-1000 | - | 672 ² | 0.804 | 0.709 | 0.032 | 0.891 | 17.50M | 13.78G |
| GreenCOD-D6-10000 | - | 672 ² | 0.813 | 0.724 | 0.031 | 0.895 | 24.34M | 16.22G |

Table 2: Comparison of performance metrics between proposed and benchmark methods on the COD10K dataset. Only models with more than 50G Multiply-Accumulate Operations (MACs) were considered. The top-performing method for each metric on each dataset is highlighted in bold, while the second-best method is underscored.

| Model | Pub/Year | Input | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $M \downarrow$ | $E_\phi^{mn} \uparrow$ | Para. | MACs |
|-------------------|----------|------------------|---------------------|----------------------|----------------|------------------------|---------|---------|
| D2CNet [34] | TIE'21 | 320 ² | 0.807 | 0.680 | 0.037 | 0.876 | - | - |
| R-MGL [39] | CVPR'21 | 473 ² | 0.814 | 0.666 | 0.035 | 0.852 | 67.64M | 249.89G |
| S-MGL [39] | CVPR'21 | 473 ² | 0.811 | 0.655 | 0.037 | 0.845 | 63.60M | 236.60G |
| UGTR [36] | ICCV'21 | 473 ² | 0.818 | 0.667 | 0.035 | 0.853 | 48.87M | 127.12G |
| BAS [30] | arXiv'21 | 288 ² | 0.802 | 0.677 | 0.038 | 0.855 | 87.06M | 161.19G |
| NCHIT [40] | CVIU'22 | 288 ² | 0.792 | 0.591 | 0.046 | 0.819 | - | - |
| CubeNet [52] | PR'22 | 352 ² | 0.795 | 0.643 | 0.041 | 0.865 | - | - |
| OCENet [23] | WACV'22 | 480 ² | 0.827 | 0.707 | 0.033 | 0.894 | 60.31M | 59.70G |
| BGNet [33] | IJCAI'22 | 416 ² | 0.831 | 0.722 | 0.033 | 0.901 | 79.85M | 58.45G |
| PreyNet [43] | MM'22 | 448 ² | 0.813 | 0.697 | 0.034 | 0.881 | 38.53M | 58.10G |
| ZoomNet [29] | CVPR'22 | 384 ² | 0.838 | 0.729 | 0.029 | 0.919 | 32.38M | 95.50G |
| FDNet [45] | CVPR'22 | 416 ² | 0.840 | 0.729 | 0.030 | 0.919 | - | - |
| CamoFormer-C [38] | arXiv'23 | 384 ² | 0.860 | 0.770 | 0.024 | 0.926 | 96.69M | 50.77G |
| CamoFormer-R [38] | arXiv'23 | 384 ² | 0.838 | 0.724 | 0.029 | 0.916 | 54.25M | 78.85G |
| PopNet [35] | arXiv'23 | 512 ² | <u>0.851</u> | <u>0.757</u> | <u>0.028</u> | 0.910 | 188.05M | 154.88G |
| PFNet+ [27] | SCIS'23 | 480 ² | 0.806 | 0.677 | 0.037 | 0.884 | - | - |
| GreenCOD-D3-1000 | - | 672 ² | 0.797 | 0.701 | 0.033 | 0.881 | 16.83M | 13.70G |
| GreenCOD-D3-10000 | - | 672 ² | 0.807 | 0.715 | 0.032 | 0.893 | 17.62M | 15.06G |
| GreenCOD-D6-1000 | - | 672 ² | 0.804 | 0.709 | 0.032 | 0.891 | 17.50M | 13.78G |
| GreenCOD-D6-10000 | - | 672 ² | 0.813 | 0.724 | 0.031 | 0.895 | 24.34M | 16.22G |

Table 3: Comparison of performance metrics between proposed and benchmark methods on the NC4K dataset. Only models with less than 50G Multiply-Accumulate Operations (MACs) were considered for computational efficiency. The top-performing method for each metric on each dataset is highlighted in bold, while the second-best method is underscored.

| Model | Pub/Year | Input | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $M \downarrow$ | $E_\phi^{mn} \uparrow$ | Para. | MACs |
|-------------------|----------|------------------|---------------------|----------------------|----------------|------------------------|---------|--------|
| SINet [10] | CVPR'20 | 352 ² | 0.808 | 0.723 | 0.058 | 0.871 | 48.95M | 19.42G |
| C2FNet [32] | IJCAI'21 | 352 ² | 0.838 | 0.762 | 0.049 | 0.897 | 28.41M | 13.12G |
| TINet [48] | AAAI'21 | 352 ² | 0.829 | 0.734 | 0.055 | 0.879 | 28.56M | 8.58G |
| JSCOD [21] | CVPR'21 | 352 ² | 0.842 | 0.771 | 0.047 | 0.898 | 121.63M | 25.20G |
| LSR [24] | CVPR'21 | 352 ² | 0.840 | 0.766 | 0.048 | 0.895 | 57.90M | 25.21G |
| PFNet [28] | CVPR'21 | 416 ² | 0.829 | 0.745 | 0.053 | 0.887 | 45.64M | 26.54G |
| C2FNet-V2 [1] | TCSVT'22 | 352 ² | 0.840 | 0.770 | 0.048 | 0.896 | 44.94M | 18.10G |
| ERRNet [16] | PR'22 | 352 ² | 0.827 | 0.737 | 0.054 | 0.887 | 69.76M | 20.05G |
| TPRNet [44] | TVCJ'22 | 352 ² | 0.846 | 0.768 | 0.048 | 0.898 | 32.95M | 12.98G |
| FAPNet [46] | TIP'22 | 352 ² | 0.851 | 0.775 | 0.047 | 0.899 | 29.52M | 29.69G |
| BSANet [47] | AAAI'22 | 384 ² | 0.841 | 0.771 | 0.048 | 0.897 | 32.58M | 29.70G |
| SegMaR [17] | CVPR'22 | 352 ² | 0.841 | 0.781 | 0.046 | 0.896 | 56.21M | 33.63G |
| SINetV2 [8] | TPAMI'22 | 352 ² | 0.847 | 0.770 | 0.048 | 0.903 | 26.98M | 12.28G |
| DGNet-S [15] | MIR'23 | 352 ² | 0.845 | 0.764 | 0.047 | 0.902 | 7.02M | 1.20G |
| DGNet [15] | MIR'23 | 352 ² | 0.857 | 0.784 | 0.042 | 0.911 | 19.22M | 2.77G |
| GreenCOD-D3-1000 | - | 672 ² | 0.815 | 0.756 | 0.049 | 0.884 | 16.83M | 13.70G |
| GreenCOD-D3-10000 | - | 672 ² | 0.823 | 0.766 | 0.047 | 0.892 | 17.62M | 15.06G |
| GreenCOD-D6-1000 | - | 672 ² | 0.820 | 0.763 | 0.047 | 0.891 | 17.50M | 13.78G |
| GreenCOD-D6-10000 | - | 672 ² | 0.827 | 0.772 | <u>0.046</u> | 0.893 | 24.34M | 16.22G |

Table 4: Comparison of performance metrics between proposed and benchmark methods on the COD10K dataset. Only models with more than 50G Multiply-Accumulate Operations (MACs) were considered. The top-performing method for each metric on each dataset is highlighted in bold, while the second-best method is underscored.

| Model | Pub/Year | Input | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $M \downarrow$ | $E_\phi^{mn} \uparrow$ | Para. | MACs |
|-------------------|----------|------------------|---------------------|----------------------|----------------|------------------------|---------|---------|
| R-MGL [39] | CVPR'21 | 473 ² | 0.833 | 0.740 | 0.052 | 0.867 | 67.64M | 249.89G |
| S-MGL [39] | CVPR'21 | 473 ² | 0.829 | 0.731 | 0.055 | 0.863 | 63.60M | 236.60G |
| UGTR [36] | ICCV'21 | 473 ² | 0.839 | 0.747 | 0.052 | 0.874 | 48.87M | 127.12G |
| BAS [30] | arXiv'21 | 288 ² | 0.817 | 0.732 | 0.058 | 0.859 | 87.06M | 161.19G |
| NCHIT [40] | CVIU'22 | 288 ² | 0.830 | 0.710 | 0.058 | 0.851 | - | - |
| OCENet [23] | WACV'22 | 480 ² | 0.853 | 0.785 | 0.045 | 0.902 | 60.31M | 59.70G |
| BGNet [33] | IJCAI'22 | 416 ² | 0.851 | 0.788 | 0.044 | 0.907 | 79.85M | 58.45G |
| PreyNet [43] | MM'22 | 448 ² | 0.834 | 0.763 | 0.050 | 0.887 | 38.53M | 58.10G |
| ZoomNet [29] | CVPR'22 | 384 ² | 0.853 | 0.784 | 0.043 | 0.896 | 32.38M | 95.50G |
| FDNet [45] | CVPR'22 | 416 ² | 0.834 | 0.750 | 0.052 | 0.893 | - | - |
| CamoFormer-C [38] | arXiv'23 | 384 ² | 0.883 | 0.834 | 0.032 | 0.933 | 96.69M | 50.77G |
| CamoFormer-R [38] | arXiv'23 | 384 ² | <u>0.855</u> | 0.788 | 0.042 | 0.900 | 54.25M | 78.85G |
| PopNet [35] | arXiv'23 | 512 ² | 0.861 | <u>0.802</u> | 0.042 | 0.909 | 188.05M | 154.88G |
| GreenCOD-D3-1000 | - | 672 ² | 0.815 | 0.756 | 0.049 | 0.884 | 16.83M | 13.70G |
| GreenCOD-D3-10000 | - | 672 ² | 0.823 | 0.766 | 0.047 | 0.892 | 17.62M | 15.06G |
| GreenCOD-D6-1000 | - | 672 ² | 0.820 | 0.763 | 0.047 | 0.891 | 17.50M | 13.78G |
| GreenCOD-D6-10000 | - | 672 ² | 0.827 | 0.772 | 0.046 | 0.893 | 24.34M | 16.22G |

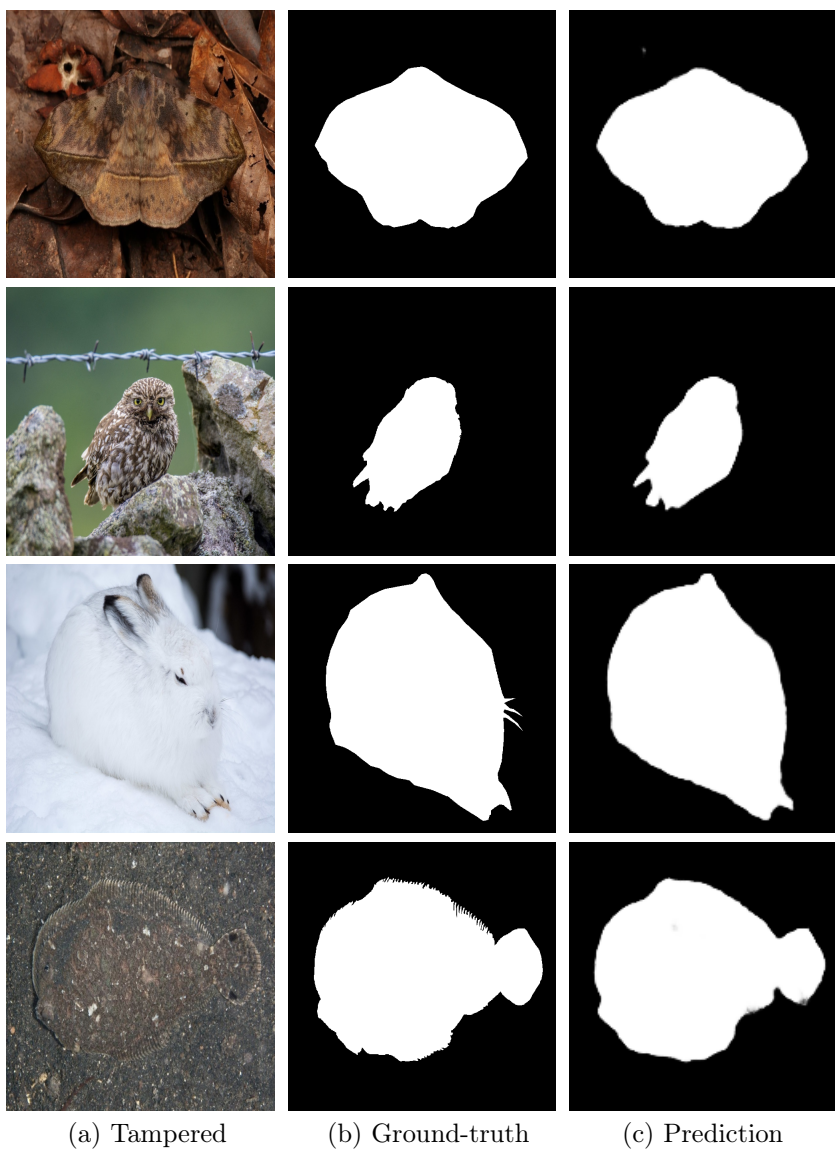


Figure 3: Illustration of mask predictions using the proposed GreenCOD. Easy images are taken from the COD10K test dataset. From left to right: (a) tampered images, (b) ground-truth masks, (c) prediction.



Figure 4: Illustration of mask predictions using the proposed GreenCOD. Difficult images are taken from the COD10K test dataset. From left to right: (a) tampered images, (b) ground-truth masks, (c) prediction.

challenging scenario: a rabbit immersed in snow, representing the complex conditions that could be encountered in everyday environments. Finally, in the fourth row, despite the fish being obscured by dust, our model successfully delineates its contours with high precision, highlighting the effectiveness of our approach in detecting concealed objects even with excellent boundaries.

4.5 Ablation Study

In this section, we present an ablation study to evaluate the contribution of each XGBoost model in a hierarchical coarse-to-fine architecture for COD. The architecture leverages XGBoost models that predict segmentation masks at corresponding resolutions. XGBoost 1 operates on the coarsest level (42x42), laying the groundwork for the segmentation. XGBoost 2 and 3 build upon this, providing mid-level refinements at resolutions of 42x42 and 84x84, respectively. XGBoost 4 delivers the final high-resolution mask (168x168x1). The segmentation performance is quantified using Mean Absolute Error (MAE) at each stage of the XGBoost integration.

In Table 5, The MAE decreases with each subsequent XGBoost model, indicating the importance of multi-scale feature integration for accurate COD. The initial coarse segmentation provided by XGBoost 1 is crucial for establishing the base structure of the mask. Each subsequent XGBoost model refines this structure by focusing on finer details, leading to a more accurate final segmentation. It suggests combining coarse prediction with high-level contextual information is critical to the model’s success.

Table 5: The MAE of each layer of XGBoost for different numbers of trees and depth.

| | 42x42 | 42x42 | 84x84 | 168x168 |
|------------|-----------|-----------|-----------|-----------|
| tree-depth | XGBoost 1 | XGBoost 2 | XGBoost 3 | XGBoost 4 |
| 1000-D3 | 0.041 | 0.036 | 0.034 | 0.033 |
| 10000-D3 | 0.039 | 0.035 | 0.033 | 0.033 |
| 1000-D6 | 0.040 | 0.035 | 0.032 | 0.032 |
| 10000-D6 | 0.038 | 0.035 | 0.032 | 0.031 |

In Table 6 examines the impact of input resolution on the MAE of the first layer of XGBoost. The results indicate that higher input resolutions generally lead to lower MAE, underscoring the importance of fine-grained input data for segmentation. The model captures more details as the input resolution increases, enhancing segmentation accuracy. The 672x672 resolution yields the best results, so we used this resolution for the remainder of the experiment. The 736x736 resolution does not provide any further improvement.

Table 7 presents the effect of different window sizes on the MAE of the second layer of XGBoost. The results show that increasing the window size

Table 6: The MAE of the first layer of XGBoost with different input resolution.

| input resolution | XGBoost 1 (42x42,1000-D3) |
|------------------|---------------------------|
| 352x352 | 0.044 |
| 416x416 | 0.042 |
| 672x672 | 0.041 |
| 736x736 | 0.041 |

Table 7: The MAE of the second layer of XGBoost with different window sizes.

| W window size | XGBoost 2 (42x42,1000-D3) |
|---------------|---------------------------|
| 3 | 0.0376 |
| 11 | 0.0357 |
| 19 | 0.0355 |
| 25 | 0.0354 |

improves the MAE, suggesting that larger windows enable the model to integrate contextual information better. This further refines the segmentation mask by capturing more surrounding details and reducing errors. We set $W = 19$ for the remainder of the experiment, as a window size of 25 does not provide much additional improvement.

Figure 5 illustrates the segmentation capabilities of a multi-scale XGBoost-based model at various stages within an ensemble learning framework. Subfigure 3a depicts the preliminary segmentation output from the first decision tree of the initial XGBoost model, providing a foundational understanding of the target structure with a coarse prediction. Progressing to Subfigure 3b, we observe the segmentation enhancements achieved by the same model’s hundredth tree, suggesting an iterative refinement within a single model’s scope. Further sophistication in the segmentation task is evident in Subfigure 3c, where the hundredth tree of the second XGBoost model likely captures more complex patterns, benefiting from an accumulation of learned features. The process culminates in Subfigure 3d, where the third XGBoost model’s hundredth tree presumably integrates the preceding models’ insights, offering the most detailed and precise delineation of the object of interest. Collectively, these subfigures demonstrate the sequential and additive nature of feature integration and decision-making in XGBoost ensembles, highlighting the intricate interplay between depth and breadth in learning representations for COD.

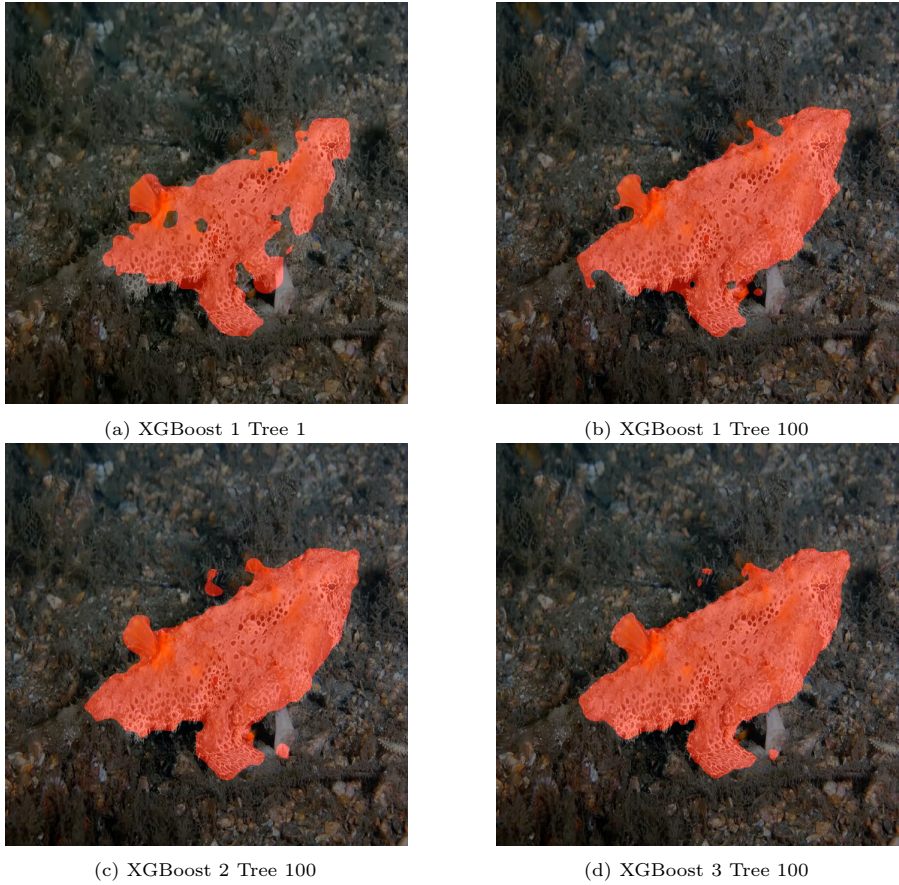


Figure 5: The illustration of the prediction of each XGBoost

4.6 Model Size and MACs computation

In this section, we detail the composition of the GreenCOD model in terms of its size (represented by the number of parameters) and its computational complexity (quantified through Multiply-Accumulate Operations (MACs)). XGBoost model size and MACs are computed by <https://hongshuo chen.com/XGBoost-calculator/>

4.6.1 Model Size Analysis

In Table 8, the GreenCOD model integrates a convolutional neural network, EfficientNetB4, with four subsequent XGBoost models. A detailed distribution of parameters is as follows.

Table 8: Number of Parameters in GreenCOD Submodules

| Submodule | Number of Trees | Depth | Number of Parameters (%) |
|----------------|-----------------|-------|--------------------------|
| EfficientNetB4 | - | - | 16,742,216 (95.0%) |
| XGBoost 1 | 10000 | 3 | 220,000 (1.2%) |
| XGBoost 2 | 10000 | 3 | 220,000 (1.2%) |
| XGBoost 3 | 10000 | 3 | 220,000 (1.2%) |
| XGBoost 4 | 10000 | 3 | 220,000 (1.2%) |
| Total | - | - | 17,622,216 |

EfficientNetB4 Backbone: Constitutes the majority (95.0%) of the model’s parameters. With 16,742,216 parameters, it forms the parameter-intensive component of GreenCOD, highlighting the complexity inherent in convolutional neural networks.

XGBoost Models: Each model, from XGBoost 1 to 4, contains an identical number of parameters (220,000), cumulatively contributing to 4.8% of the total parameters. This uniformity indicates a scalable approach to segmentation across different resolutions without escalating parameter count.

Total Parameter Count: The entire GreenCOD model encompasses 17,622,216 parameters, with a significant proportion attributed to the CNN layers. Deep learning architectures rely heavily on convolutional filters for feature extraction. In the future, we will attempt to replace EfficientNet with other more efficient solutions to reduce the model size further.

4.6.2 Computational Complexity Analysis

In Table 9, the computational complexity for the GreenCOD model is assessed using MACs, which indicate the model’s efficiency during inference.

Table 9: MACs in GreenCOD Submodules

| Submodule | Size | Number of Trees | Depth | MACs (%) |
|----------------|------|-----------------|-------|------------------------|
| EfficientNetB4 | - | - | - | 13,503,446,880 (89.7%) |
| XGBoost 1 | 42 | 10000 | 3 | 70,560,000 (0.5%) |
| XGBoost 2 | 42 | 10000 | 3 | 70,560,000 (0.5%) |
| XGBoost 3 | 84 | 10000 | 3 | 282,240,000 (1.9%) |
| XGBoost 4 | 168 | 10000 | 3 | 1,128,960,000 (7.5%) |
| Total | - | - | - | 15,055,766,880 |

EfficientNetB4 Backbone: Dominates the computational process with 89.7% of the total MACs, amounting to 13,503,446,880 MACs. It reveals that the convolutional layers of the backbone are the primary contributors to the model’s computational load.

XGBoost Models: There is a notable increase in MACs from the coarsest model, XGBoost 1, to the finest, XGBoost 4. The former requires 70,560,000 MACs, while the latter necessitates 1,128,960,000 MACs, aligning with the increased resolution of the output masks.

Overall Computational Demand: The total MACs for GreenCOD amount to 15,055,766,880 (15.06G), lower than most deep learning methods.

4.7 On-device Demo for GreenCOD

We offer an on-device demo for our GreenCOD at <https://greencod.ai/demo>, utilizing the GreenCOD-D3-1000 model. The model is converted into a mobile-compatible format using ONNX and then run using ONNX.js on a web browser. Initially, the model is downloaded from the website (this only needs to be done once). Inference starts directly in the browser when you upload an image or take a photo with your phone. The results might be slightly different due to the model's conversion and device performance variations, but the core functionality remains the same.

Our GreenCOD demo provides several key benefits:

- **Privacy:**
 - Images are processed locally on your device, not uploaded to a server.
 - This approach helps protect your sensitive information from leaking.
- **Offline Capability:**
 - Once the model is loaded, it operates without an internet connection.
 - This is especially useful in remote areas where internet access is unavailable, such as during hiking trips.
- **Device Compatibility:**
 - The model runs on CPUs and uses a web-based interface.
 - It is accessible on any device with a web browser, including smartphones, tablets, and computers.
- **Eco-Friendliness:**
 - Inference is performed without servers or GPUs, reducing operational costs and environmental impact.

In summary, our GreenCOD demo ensures user privacy and offline capability. It promotes device compatibility and eco-friendliness, making it a versatile and sustainable solution for camouflaged object detection on the go.

5 Conclusion and Future Work

This research presents GreenCOD, an innovative methodology for COD that marries the efficiency of Extreme Gradient Boosting (XGBoost) with the robust deep feature extraction capabilities of Deep Neural Networks (DNNs). In the current landscape, the trend is to craft more complex DNN structures to improve detection efficacy. Yet, these approaches come with a significant computational load. In contrast, GreenCOD distinguishes itself by utilizing gradient boosting for detection, leading to a more streamlined model that demands fewer parameters and lower Multiply-Accumulate Operations (MACs) without compromising performance. A standout feature of GreenCOD is its ability to be trained effectively without the traditional reliance on backpropagation.

GreenCOD not only stands as an efficient approach in its current form but also signals potential for future explorations. Prospective studies may investigate the substitution of EfficientNet with alternative non-deep learning feature extraction methods to diminish the model size further. Additionally, there are expansive opportunities for applying GreenCOD in other domains, such as Salient Object Detection (SOD), Video COD, and Edge Detection, to broaden the scope of its applicability and impact.

6 Acknowledgments

This work was supported by the Army Research Laboratory (ARL) under agreement W911NF2020157. Computation for the work was supported by the University of Southern California's Center for Advanced Research Computing (carc.usc.edu).

References

- [1] G. Chen, S.-J. Liu, Y.-J. Sun, G.-P. Ji, Y.-F. Wu, and T. Zhou, "Camouflaged object detection via context-aware cross-level fusion", *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10), 2022, 6981–93.
- [2] H.-S. Chen, S. Hu, S. You, C.-C. J. Kuo, *et al.*, "Defakehop++: An enhanced lightweight deepfake detector", *APSIPA Transactions on Signal and Information Processing*, 11(2), 2022.
- [3] H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "Defakehop: A light-weight high-performance deepfake detector", in *2021 IEEE International conference on Multimedia and Expo (ICME)*, IEEE, 2021, 1–6.

- [4] H.-S. Chen, K. Zhang, S. Hu, S. You, and C.-C. J. Kuo, “Fake Satellite Image Detection via Parallel Subspace Learning (PSL)”, in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2022, 1502–6.
- [5] H.-S. Chen, K. Zhang, S. Hu, S. You, and C.-C. J. Kuo, “Geo-defakehop: High-performance geographic fake image detection”, *arXiv preprint arXiv:2110.09795*, 2021.
- [6] Y. Chen, M. Rouhsedaghat, S. You, R. Rao, and C.-C. J. Kuo, “Pixel-hop++: A small successive-subspace-learning-based (ssl-based) model for image classification”, in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, 3294–8.
- [7] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, “Structure-measure: A new way to evaluate foreground maps”, in *Proceedings of the IEEE international conference on computer vision*, 2017, 4548–57.
- [8] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, “Concealed object detection”, *IEEE transactions on pattern analysis and machine intelligence*, 44(10), 2021, 6024–42.
- [9] D.-P. Fan, G.-P. Ji, X. Qin, and M.-M. Cheng, “Cognitive vision inspired object segmentation metric and loss function”, *Scientia Sinica Informationis*, 6(6), 2021.
- [10] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, “Camouflaged object detection”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, 2777–87.
- [11] H. Fu, Y. Yang, V. K. Mishra, and C.-C. J. Kuo, “Subspace learning machine (SLM): Methodology and performance evaluation”, *Journal of Visual Communication and Image Representation*, 2024, 104058.
- [12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn”, in *Proceedings of the IEEE international conference on computer vision*, 2017, 2961–9.
- [13] R. He, Q. Dong, J. Lin, and R. W. Lau, “Weakly-supervised camouflaged object detection with scribble annotations”, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37, No. 1, 2023, 781–9.
- [14] Y.-T. Hu, H.-S. Chen, K. Hui, J.-B. Huang, and A. G. Schwing, “Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 3105–15.
- [15] G.-P. Ji, D.-P. Fan, Y.-C. Chou, D. Dai, A. Liniger, and L. Van Gool, “Deep gradient learning for efficient camouflaged object detection”, *Machine Intelligence Research*, 20(1), 2023, 92–108.
- [16] G.-P. Ji, L. Zhu, M. Zhuge, and K. Fu, “Fast camouflaged object detection via edge-based reversible re-calibration network”, *Pattern Recognition*, 123, 2022, 108414.

- [17] Q. Jia, S. Yao, Y. Liu, X. Fan, R. Liu, and Z. Luo, “Segment, magnify and reiterate: Detecting camouflaged objects the hard way”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 4713–22.
- [18] C.-C. J. Kuo and A. M. Madni, “Green learning: Introduction, examples and outlook”, *Journal of Visual Communication and Image Representation*, 90, 2023, 103685.
- [19] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, “Interpretable convolutional neural networks via feedforward design”, *Journal of Visual Communication and Image Representation*, 60, 2019, 346–59.
- [20] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, “Anabranched network for camouflaged object segmentation”, *Computer vision and image understanding*, 184, 2019, 45–56.
- [21] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, “Uncertainty-aware joint salient object and camouflaged object detection”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 10071–81.
- [22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context”, in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, Springer, 2014, 740–55.
- [23] J. Liu, J. Zhang, and N. Barnes, “Modeling aleatoric uncertainty for camouflaged object detection”, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, 1445–54.
- [24] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D.-P. Fan, “Simultaneously localize, segment and rank the camouflaged objects”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 11591–601.
- [25] V. Magouliantis, C. A. Alexander, and C.-C. J. Kuo, “A Comprehensive Overview of Computational Nuclei Segmentation Methods in Digital Pathology”, *arXiv preprint arXiv:2308.08112*, 2023.
- [26] Y. Mao, J. Zhang, Z. Wan, Y. Dai, A. Li, Y. Lv, X. Tian, D.-P. Fan, and N. Barnes, “Generative transformer for accurate and reliable salient object detection”, *arXiv preprint arXiv:2104.10127*, 2021.
- [27] H. Mei, X. Yang, Y. Zhou, G. Ji, X. Wei, and D. Fan, “Distraction-aware camouflaged object segmentation”, *SCIENTIA SINICA Informationis (SSI)*, 2023.
- [28] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, “Camouflaged object segmentation with distraction mining”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 8772–81.

- [29] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, “Zoom in and out: A mixed-scale triplet network for camouflaged object detection”, in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2022, 2160–70.
- [30] X. Qin, D.-P. Fan, C. Huang, C. Diagne, Z. Zhang, A. C. Sant’Anna, A. Suarez, M. Jagersand, and L. Shao, “Boundary-aware segmentation network for mobile and web applications”, *arXiv preprint arXiv:2101.04704*, 2021.
- [31] 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, 779–88.
- [32] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, “Context-aware cross-level fusion network for camouflaged object detection”, *arXiv preprint arXiv:2105.12555*, 2021.
- [33] Y. Sun, S. Wang, C. Chen, and T.-Z. Xiang, “Boundary-guided camouflaged object detection”, *arXiv preprint arXiv:2207.00794*, 2022.
- [34] K. Wang, H. Bi, Y. Zhang, C. Zhang, Z. Liu, and S. Zheng, “D²C-Net: A Dual-Branch, Dual-Guidance and Cross-Refine Network for Camouflaged Object Detection”, *IEEE Transactions on Industrial Electronics*, 69(5), 2021, 5364–74.
- [35] Z. Wu, D. P. Paudel, D.-P. Fan, J. Wang, S. Wang, C. Demonceaux, R. Timofte, and L. Van Gool, “Source-free depth for object pop-out”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, 1032–42.
- [36] F. Yang, Q. Zhai, X. Li, R. Huang, A. Luo, H. Cheng, and D.-P. Fan, “Uncertainty-guided transformer reasoning for camouflaged object detection”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 4146–55.
- [37] Y. Yang, W. Wang, H. Fu, C.-C. J. Kuo, *et al.*, “On supervised feature selection from high dimensional feature spaces”, *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.
- [38] B. Yin, X. Zhang, Q. Hou, B.-Y. Sun, D.-P. Fan, and L. Van Gool, “Camo-former: Masked separable attention for camouflaged object detection”, *arXiv preprint arXiv:2212.06570*, 2022.
- [39] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, and D.-P. Fan, “Mutual graph learning for camouflaged object detection”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 12997–3007.
- [40] C. Zhang, K. Wang, H. Bi, Z. Liu, and L. Yang, “Camouflaged object detection via neighbor connection and hierarchical information transfer”, *Computer Vision and Image Understanding*, 221, 2022, 103450.

- [41] K. Zhang, H.-S. Chen, Y. Wang, X. Ji, and C.-C. J. Kuo, “Texture analysis via hierarchical spatial-spectral correlation (HSSC)”, in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, 4419–23.
- [42] K. Zhang, H.-S. Chen, X. Zhang, Y. Wang, and C.-C. J. Kuo, “A data-centric approach to unsupervised texture segmentation using principle representative patterns”, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, 1912–6.
- [43] M. Zhang, S. Xu, Y. Piao, D. Shi, S. Lin, and H. Lu, “Preynet: Preying on camouflaged objects”, in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, 5323–32.
- [44] Q. Zhang, Y. Ge, C. Zhang, and H. Bi, “TPRNet: camouflaged object detection via transformer-induced progressive refinement network”, *The Visual Computer*, 2022, 1–15.
- [45] Y. Zhong, B. Li, L. Tang, S. Kuang, S. Wu, and S. Ding, “Detecting camouflaged object in frequency domain”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 4504–13.
- [46] T. Zhou, Y. Zhou, C. Gong, J. Yang, and Y. Zhang, “Feature aggregation and propagation network for camouflaged object detection”, *IEEE Transactions on Image Processing*, 31, 2022, 7036–47.
- [47] H. Zhu, P. Li, H. Xie, X. Yan, D. Liang, D. Chen, M. Wei, and J. Qin, “I can find you! boundary-guided separated attention network for camouflaged object detection”, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, No. 3, 2022, 3608–16.
- [48] J. Zhu, X. Zhang, S. Zhang, and J. Liu, “Inferring camouflaged objects by texture-aware interactive guidance network”, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, No. 4, 2021, 3599–607.
- [49] Y. Zhu, X. Wang, H.-S. Chen, R. Salloum, and C.-C. J. Kuo, “A-pixelhop: A green, robust and explainable fake-image detector”, in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, 8947–51.
- [50] Y. Zhu, X. Wang, H.-S. Chen, R. Salloum, and C.-C. J. Kuo, “Green Steganalyzer: A Green Learning Approach to Image Steganalysis”, *arXiv preprint arXiv:2306.04008*, 2023.
- [51] Y. Zhu, X. Wang, R. Salloum, H.-S. Chen, C.-C. J. Kuo, et al., “RGID: A Robust and Green GAN-Fake Image Detector”, *APSIPA Transactions on Signal and Information Processing*, 11(2), 2022.
- [52] M. Zhuge, X. Lu, Y. Guo, Z. Cai, and S. Chen, “CubeNet: X-shape connection for camouflaged object detection”, *Pattern Recognition*, 127, 2022, 108644.