

Modified Convolutional Architectures for Camouflaged Object Detection

Ivan Omelchenko^{a*}, Yuri Gordienko^a, Vladyslav Taran^a, Sergey Peresada^a,
Gert Jervan^b, Mairo Leier^b, Sergey Lyshevski^c, Sergii Stirenko^a

^a*National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”,
Kyiv, Ukraine*

^b*Tallinn University of Technology, Tallinn, Estonia*

^c*Rochester Institute of Technology, Rochester, New York, USA*

Received July 30, 2025; Revised January 29, 2026; Accepted February 6, 2026

Camouflaged object detection (COD) in natural environments is a challenging and understudied task in the field of computer vision. COD represents a critical frontier in applying artificial intelligence (AI) to environmental monitoring, infrastructure resilience, and conservation. This study presents an overview of modern approaches to COD, including convolutional neural networks (CNNs), transformer-based models, gradient-based methods, and graph-based frameworks and provides quantitative results of various frameworks that belong to these methods. The research outlines the difficulties associated with COD, such as variability in camouflage strategies, scarcity of well-annotated datasets, and computational complexity of existing models. Several benchmark datasets (COD10K, CAMO, NC4K, Military Personnel Data, and MoCA) are examined, each contributing unique properties to model development and evaluation. A modified framework Hybrid-SINet is proposed and evaluated using standard COD metrics, demonstrating improved performance (<https://www.kaggle.com/code/ivanomelchenkoim11/article-hybrid-sinet-benchmark> Hybrid-SINet benchmarking on Kaggle) over the baseline SINet model (<https://www.kaggle.com/code/ivanomelchenkoim11/article-sinet-benchmark> Baseline SINet benchmarking on Kaggle) under specific environmental conditions.

Keywords and phrases: Deep learning, object detection, camouflaged object detection, convolutional neural network, mean absolute error, environmental monitoring, infrastructure resilience monitoring, conservation.

AMS subject classification: 68T45.

1 Introduction

In the current world marked by environmental challenges, natural disasters, and rapid urbanization, the ability to detect camouflaged objects (COD) has become increasingly vital in a variety of industrial and social applications [7]. COD plays a critical role in agriculture [13], defect detection [5], medicine, and video analysis. In natural environments, COD models can be used for disaster management, search and rescue operations, autonomous machinery [8], and in warfare [26].

*Corresponding author. Email: i.omelchenko.ip04@kpi.ua

For example, during post-disaster rescue missions, identifying camouflaged survivors or hazardous debris concealed by natural elements is crucial.

Similarly, in smart cities, smart waste collection, automated scrap sorting, environmentally safe garbage detection, safe collection of concealed waste, or detection of hidden defects in infrastructure can prevent ecological damage and promote sustainability. These applications highlight the growing demand for intelligent systems capable of locating hidden or visually blended objects under complex real-world conditions.

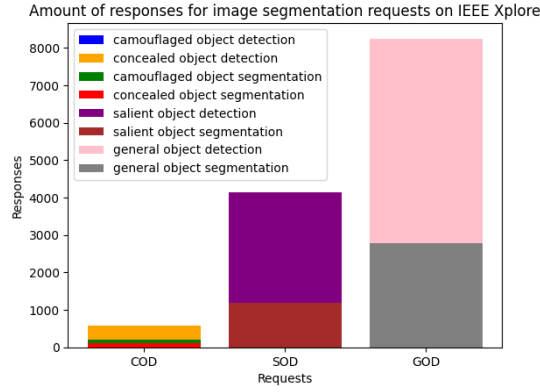
Camouflaged object detection (COD) is a subfield of Machine Learning (ML) that performs segmentation of targets that are concealed and blend into their surroundings [7]. The general approach applied in COD to identify a target is to assign each pixel i with $p_i \in [0, 1]$, where p_i represents the probability score of a pixel [5]. The closer the score is to 1, the higher the probability that the pixel belongs to a hidden object.

Despite its importance, COD remains one of the most challenging and understudied areas in computer vision. The claim is proven in Figure 1. On 9 July 2025, using the following COD-related keywords on platforms such as IEEE Xplore, Springer and Academia produced the illustrated results: *camouflaged object detection*, *concealed object detection*, *camouflaged object segmentation*, *concealed object segmentation*. These were compared with SOD-related requests using the keywords *salient object detection*, *salient object segmentation* and GOD-related results for the following formulations: *general object detection*, *general object segmentation*.

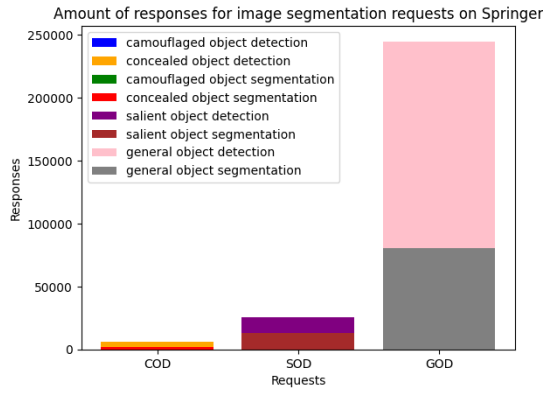
Unlike standard object detection, COD requires segmenting targets that are intentionally or naturally concealed and visually indistinguishable from their backgrounds [26]. This presents a unique computational problem where traditional algorithms often fail due to the subtle visual cues and intricate textures involved. While the human visual system can interpret context and semantic information to perceive such objects, artificial systems struggle with the ambiguity and variability inherent to camouflage. Thus, the following problems could be singled out when developing COD models [8, 26, 11, 7]:

- High variability in camouflage strategies, such as color matching, shadow blending, disruptive markings, and adaptive shapes across different natural and artificial environments. These are further complicated by diverse weather conditions, terrain types, and dynamic changes in visual scenes.
- Scarcity of annotated datasets that capture the complexity of camouflage in real-world settings. Existing datasets are often fragmented, domain-specific, and lack standardization, making it difficult to benchmark models or generalize across tasks.
- Computational and architectural complexity of existing models, which frequently rely on deep learning architectures that are difficult to train and inefficient in real-time or resource-constrained environments.

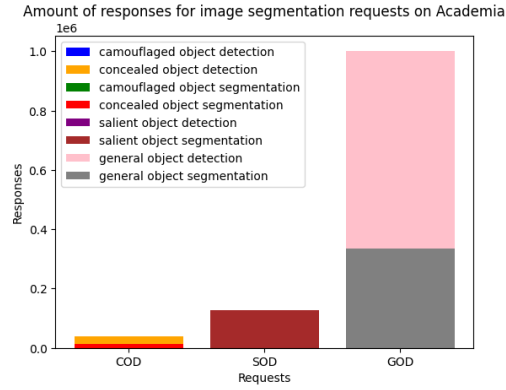
Given these challenges, the main aim of this study is to contribute to the advancement of COD systems by surveying and evaluating state-of-the-art (SOTA) techniques and introducing a novel hybrid model based on several components available in various neural network (NN) architectures. The work is structured around the following key stages: analysis of modern COD methods, examination of benchmark datasets, such as COD10K, CAMO, NC4K, Military Personnel Data, and MoCA, development of a modified hybrid model, evaluation of the proposed model using standard COD metrics.



(a) Image Segmentation-related studies on IEEE Xplore



(b) Image Segmentation-related studies on Springer



(c) Image Segmentation-related studies on Academia

Figure 1: Image segmentation-related results on various platforms

2 Background and related works

While COD is an understudied branch of Machine Learning, compared to its other directions, there is still a significant amount of research that is ongoing. This section offers a compilation of the most prominent modern approaches and the largest datasets used to train and test the frameworks.

2.1 Existing approaches

Some studies, such as [3] divide COD approaches into CNN-based and Transformer-based, probably, based on a model backbone, however, due to the fact that Transformer-based approaches sometimes employ CNN backbones and variability of CNN backbone-based methodologies, similarly to [8], we adopt the broader classification.

CNN-based approaches. Convolutional neural networks (CNN) identify targets using filters applied on images [18]. In these models, key parameters include padding, stride, activation functions such as sigmoid functions, and weight matrices. Input activations are convolved to produce feature maps used for downstream tasks.

There are three primary CNN-based strategies for COD: Attention-based, Two-stage, and Joint-learning approaches.

Attention-based strategies include notable models such as PFNet, which introduces Positioning and Focus Modules [9], $C^2F - Net$, which fuses cross-level and global context features [28], and UR-COD, which enhances boundary detection [14]. Among the two-stage approaches, SINet is well known for using separate search and identification modules [5]. Joint-learning strategies, early examples include ANET, uses classification and segmentation streams [22] to segment objects. Later models such as LSR introduced object ranking and localization [31], while ZoomNet focused on multiscale feature integration [27].

Transformer-based approaches [2] eliminate the need for recurrence by relying on attention mechanisms to model dependencies across sequences. These models use an encoder-decoder architecture, where the encoder maps an input sequence into a continuous representation, and the decoder generates an output sequence from it.

Self-attention is central to these models. It allows each position in the input to attend to all others, enabling more efficient computation and broader contextual understanding. Compared to CNNs, Transformer operations are more parallelizable and often more efficient.

The first transformer-based model applied to detection tasks was DETR [15], although it was not COD-specific. COD-dedicated Transformer models include OSFormer, which segments camouflaged instances; UGTR, which integrates uncertainty-aware reasoning modules [6]; FSPNet, which decodes hierarchical transformer features for local cue aggregation [32]; and CamoFormer, which uses separable attention in an end-to-end framework [3]. Notably, most of these models still use CNN backbones for feature extraction [9].

Deep gradient-based approaches. The Deep Gradient Network (DGNet) [8] represents a new perspective, proposing the separation of high- and low-level image features to better capture internal textures of camouflaged objects. This enables the network to rely more on internal gradients than on boundary cues. The method includes a context encoder and a texture encoder, which produce complementary features. These are combined using a process that involves group learning, projection strategies, and parallel residual operations. Together, they guide the decoder to generate accurate segmentations by focusing on inner-object patterns.

DGNet remains the primary representative of this approach. While it has influenced applications in other fields, such as Synthetic Aperture Radar (SAR) imaging [12], it remains relatively novel within COD with no further studies into this direction.

Graph-based approaches. Graph-based methods [17] aim to extract complementary information between detection and edge prediction tasks. These methods often treat camouflage detection as a joint problem, where both object regions and their boundaries must be accurately identified. Frameworks of this type employ various modules to decouple images into target and edge representations and perform segmentation by transforming them into sample-dependent

graphs.

The first Graph-Based approach was the MGL model [17], with the authors providing a single stage and recurrent implementations. While the method is novel, similarly to the deep gradient approach, follow-up studies in this direction exist, namely, introducing BGNet [29], which explicitly employs edge semantic information through the edge-aware module and enhances feature representation through context aggregation.

Overall, each approach has its advantages and drawbacks. CNN-based methods excel at hierarchical feature extraction using mature architectures but often struggle with modeling long-range dependencies and handling fine object variations. Transformer-based models improve global contextual awareness and parallelization, but demand large datasets and are computationally intensive. Deep Gradient-based approaches focus on subtle texture cues and reduce reliance on object boundaries, though they remain relatively new and less validated. Graph-based techniques leverage structural relationships and multi-task learning, offering fine-grained edge modeling, but they introduce greater complexity and rely on precise annotations.

2.2 COD-related datasets

The main COD-related datasets vary in scale, annotation types, and use.

COD10K. The COD10K dataset [5] was developed to serve as a comprehensive benchmark for advancing deep learning-based methods in camouflaged object detection (COD) and to facilitate the development of novel recognition models. It currently represents the largest publicly available COD dataset, providing 10,000 high-resolution images, 6,000 for training and 4,000 for testing each with dense hierarchical annotations. These include category-level labels, bounding boxes for task-specific object localization, and scene-level attributes such as occlusion and boundary complexity. Moreover, instance-level annotations are included to support object-level identification tasks [5]. The dataset adopts a superclass-subclass taxonomy, comprising five superclasses and 69 subclasses, yielding a ninefold increase in class diversity compared to its predecessor, CAMO. Additionally, COD10K includes a set of non-camouflaged images labeled with five superclasses but with a very limited subclass granularity.

CAMO. CAMO [22] was the first large-scale dataset specifically curated for COD research. It contains 1,250 high-quality images collected from an initial pool of approximately 3,000 images, spanning both natural and artificial camouflage scenarios across diverse environments. These images were annotated by external contributors and are characterized by complex target boundaries, frequent occlusions, disconnected object parts, overlapping entities, and visually distracting backgrounds. The dataset poses additional challenges related to target size, with many objects occupying less than 10% of the scene area. Furthermore, around 8% of the images (approximately 100) do not contain target objects, increasing the difficulty of the detection task. CAMO comprises 1,000 training and 250 testing images and includes 8 object categories within two broader domains: people and animals. CAMO-COCO [23], a related extension, introduces an equivalent number of non-camouflaged images to support comparative analysis.

NC4K. The NC4K dataset [31] was constructed as a benchmark for evaluating the generalization capabilities of COD models, specifically focusing on tasks such as Camouflaged Object Localization (COL) and Camouflaged Object Ranking (COR). Unlike COD10K and CAMO, NC4K does not include training data but is designed exclusively for testing purposes. It comprises 4,121 Internet-sourced images, each with corresponding ground-truth and instance-level annotations. The dataset does not provide class labels and is frequently used alongside training datasets such as COD10K or CAMO to comprehensively evaluate model performance across

diverse scenarios [31].

Military personnel data. The Military Personnel Data (CPD1K) dataset [30] was introduced to support the study of dense deconvolution networks in detecting camouflaged military personnel. This dataset addresses the specific needs of security and defense applications, which may not be adequately served by general-purpose COD datasets [26]. It comprises 1,000 images, evenly distributed across 20 camouflage-related classes. CPD1K remains one of the few publicly available COD datasets focused explicitly on military scenarios.

MoCA. The MoCA dataset [10] was developed to facilitate the study of motion-based segmentation of concealed objects in video data. As the most extensive video segmentation dataset for COD, it includes 141 video clips totaling 37,391 annotated frames, approximately seven times more content than earlier datasets such as that of Bideau and Learned-Miller [16]. Videos were sourced from YouTube and feature animals in their natural environments, with clip durations ranging from 1 to 79 seconds, sampled at 24 frames per second in HD resolution (720×1280). Annotations were provided at every fifth frame using bounding boxes and include both positional and temporal metadata [4]. The dataset also categorizes object motion into three types: locomotion, deformation, and static. MoCA covers 67 animal categories, offering a diverse and dynamic testbed for motion-based COD models.

3 Proposed model

3.1 Hybrid model

The present study proposes a novel hybrid model architecture to advance research in this area. The proposed Hybrid-SINet framework modifies the implementation of SINet [5], augmenting its foundational two-stage CNN pipeline. The novel contributions include the introduction of an edge attention mechanism, inspired by methodologies in [29, 26], and a dimensionality reduction strategy, motivated by principles in [8]. An overview of the Hybrid-SINet model architecture is presented in Fig. 2. For comparison, the architecture of a baseline SINet [5] model is dashed. Dimensional Reduction and Edge Integration, which will be discussed later, are enclosed to evade confusion as they are between Search and Identification operations, but are not present in baseline SINet.

Search module. In SINet [5] it incorporates a receptive field component designed to enhance the discriminative capacity of feature representations. For a given input image, a sequence of feature maps is extracted at multiple resolutions. These feature maps are categorized into low-level, middle-level, and high-level groups to capture varying spatial and semantic characteristics. The features are recombined by concatenation and downsampling. This module uses striding to ensure that image resolution is preserved between network layers.

Receptive field. SINet receptive is a modification of RFB-s, introduced by [20]. It consists of five branches, followed by three convolution layers: 1×1 , $(2k - 1) \times (2k - 1)$ and 3×3 . First four branches are concatenated and go through a 1×1 convolution. When the fifth branch is added, a ReLU function is applied to receive a result [5].

Edge attention module. The Edge Attention Module draws upon the Edge-Aware Module from BGNet [29], integrating both high-level and low-level features to mitigate the inclusion of non-object edge details. The proposed adaptation concatenates selected low-level and high-level features, as suggested in [26]. Channel dimensions of the input features are standardized through initial 1×1 convolutional layers. These are concatenated and passed through a sequence of convolutional operations to generate edge features. The process concludes with a

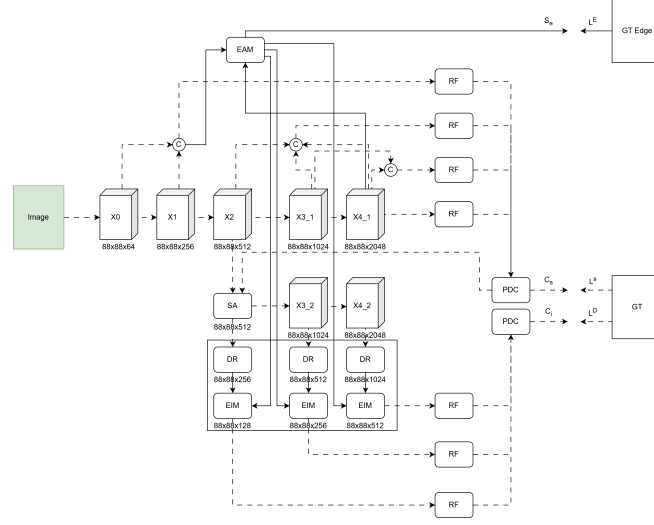


Figure 2: Overall architecture of the proposed Hybrid-SINet model. RF is Receptive Field, EAM is Edge Attention Module, DR is Dimensional Reduction Module, EIM is Edge Integration Module, all of which are discussed later. PDC are decoders, which process RF outputs through convolution and concatenation to receive SM and IM prediction maps and SA is Search Attention module which uses min-max mathematical normalization operations and convolutions on Gaussian kernel to get input features for IM from SM middle-level feature and prediction map.

sigmoid activation to enhance edge localization capabilities.

Dimensional reduction. Motivated by the design philosophy outlined in [8], which recommends separate treatment of low-level and high-level features, a Dimensional Reduction module is integrated to streamline computation. Features output from the Search Attention Module undergo dimensionality reduction via stacked convolutional blocks composed of convolution, batch normalization, and ReLU activations. The resulting lower-dimensional feature maps maintain spatial resolution while reducing channel complexity, thereby enhancing computational efficiency.

Edge guidance feature module. To further refine feature integration, the Edge Guidance Feature Module is employed. This component builds upon techniques described in [29] and aims to combine edge features from the Edge Attention Module with features reduced by the Dimensional Reduction module. Initially, edge and feature maps are fused through a series of element-wise operations, followed by convolution. The fused features are then enhanced using a global context mechanism involving global average pooling and channel-wise recalibration. This process aids in emphasizing relevant spatial features while suppressing noise.

Identification module. The Identification Module serves to precisely localize camouflaged objects using features derived from the Search Module. The initial stage involves a partial decoder that generates coarse camouflage predictions. These are refined through subsequent processing stages, which incorporate both edge and high-level semantic features. An enhanced camouflage map is obtained using a guided attention mechanism based on Gaussian filtering and normalization. High-level features are decoded independently and then combined via a hierarchical refinement process. Element-wise operations, including multiplication and upsampling, are employed to progressively refine features across multiple scales. The final output is

produced by concatenating these refined feature maps, yielding a high-resolution segmentation of camouflaged objects.

3.2 Implementation details

Hybrid-Sinet has been implemented using PyTorch framework. The data, presented in the following sections, has been received by running discussed framework in Kaggle environment using GPU P100. The model was trained on a training data provided by a COD10K dataset and was benchmarked against testing branches of COD10K, CAMO, NC4K (which is for testing exclusively) and Military Personnel Data (or CPD1K), thus focusing on large datasets of static COD. The framework has been trained for 30 epochs, which takes about 2 hours on the platform with the denoted parameters.

The training parameters that are relevant for the process of training the discussed model are given as follows: learning rate is set to 1e-4, train size is 353, gradient clipping margin is 0.5, decay rate is 0.1, decay epoch is 30, batch size is 16.

3.3 Metrics and workflow

To objectively assess the performance of camouflaged object detection (COD) methods, we adopt a set of commonly used evaluation metrics, following prior works [8, 26]. The metrics include the Mean Absolute Error (MAE), the weighted F-measure (F_β^w), and the structural similarity measure (S_α). Each metric evaluates a different aspect of prediction quality, such as region-level accuracy, structural alignment, or pixel-wise discrepancy between a predicted saliency map and the corresponding ground truth mask.

Structural similarity measure (S_α): The structural similarity measure S_α quantifies the structural consistency between the predicted saliency map S_p and the ground truth map G . It is a convex combination of object-aware similarity S_o and region-aware similarity S_r , formulated as follows [26]:

$$S_\alpha = (1 - \alpha) \cdot S_o(S_p, G) + \alpha \cdot S_r(S_p, G), \quad (1)$$

where $\alpha \in [0, 1]$ is a balancing factor, typically set to 0.5, to equally weight both components. The term $S_o(S_p, G)$ captures global object-level similarity, while $S_r(S_p, G)$ measures structural correspondence across different image regions.

Weighted F-measure (F_β^w): The weighted F-measure F_β^w [19] evaluates the harmonic mean between weighted precision and weighted recall, thus accounting for class imbalance and spatial structure of the predicted map. It is defined as:

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

where β^2 is a weighting factor that determines the relative importance of precision versus recall. β is a nonnegative number, typically set to $\beta^2 = 0.3$ and is used to place greater emphasis on precision. Being an extension of the common F_β measure, it uses weighted versions of precision and recall, accounting the pixel-level importance using spatial weighting functions. This allows this function to be viable to work with non-binary ground truth and is noted to be able to account for the errors in a more correct way, processing their locations.

Mean absolute error (MAE): The Mean Absolute Error measures the average pixel-wise absolute difference between the predicted saliency map S_p and the ground truth mask G , and is defined as:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^W \sum_{y=1}^H |S_p(x, y) - G(x, y)|, \quad (3)$$

where W and H denote the width and height of the image, respectively. A lower MAE indicates a closer match between the predicted and ground truth maps.

These complementary metrics jointly provide a comprehensive evaluation framework, capturing both region-level similarity and pixel-level accuracy, and are widely adopted in benchmarking COD models.

4 Results and discussion

4.1 Comparison of baseline COD-related models

We present a comprehensive quantitative evaluation of state-of-the-art camouflaged object detection (COD) models in Table 1, utilizing the NC4K, CAMO, and COD10K benchmark datasets.

The evaluation metrics include previously discussed structural similarity measure (S_α), the weighted F-measure (F_β^w), and the Mean Absolute Error (MAE). The best results for each metric are highlighted in bold.

Metrics	PFNet	SINet	LSR	UGRT	DGNet	S-MGL	R-MGL	BGNet	FSPNet	ZoomNet	$C^2F - Net$	CamoFormer-C	CamoFormer-R	CamoFormer-S	CamoFormer-P
NC4K Dataset															
S_α	.829	.808	.840	.839	.857	.829	.833	.851	.879	.853	.836	.884	.857	.888	.892
F_β^w	.745	.723	.766	.734	.784	.731	.740	.788	.843	.818	.762	.833	.793	.840	.847
MAE	.053	.058	.048	.052	.042	.055	.052	.044	.035	.043	.049	.033	.041	.031	.030
CAMO Dataset															
S_α	.782	.745	.787	.785	.839	.772	.775	.812	.856	.820	.796	.860	.817	.876	.872
F_β^w	.695	.644	.696	.686	.769	.664	.673	.749	.830	.794	.719	.811	.756	.832	.831
MAE	.085	.092	.080	.086	.057	.089	.088	.073	.050	.066	.080	.051	.066	.043	.046
COD10K Dataset															
S_α	.800	.776	.804	.818	.822	.811	.814	.831	.851	.838	.813	.860	.838	.862	.869
F_β^w	.660	.631	.673	.667	.693	.655	.666	.722	.769	.766	.686	.767	.730	.772	.786
MAE	.040	.043	.037	.035	.033	.037	.035	.033	.026	.029	.036	.024	.029	.024	.023

Table 1: Quantitative comparison of various baseline COD-related models.

Overall, the Transformer-based CamoFormer family (specifically the S and P variants)

achieves state-of-the-art performance across all datasets and metrics, highlighting their effectiveness in modeling complex camouflage patterns. FSPNet, although not a Transformer-based model, consistently ranks among the top-performing baselines, suggesting that with appropriate architectural enhancements, CNN-based models can remain competitive. Among traditional CNNs, ZoomNet and LSR also yield commendable results in specific metrics. Additionally, models incorporating graph structures (BGNet) and gradient-based cues (DGNet) show solid performance, though they are consistently outperformed by Transformer-based architectures.

While Transformer-based models' ability to capture long-range dependencies and global context contributes to superior quantitative results compared to their competitors and may collectively emphasize the ongoing shift in COD model design from convolutional architectures toward Transformer-based frameworks, models of other approaches are still capable of achieving SOTA-like performance and still hold their relevancy in some niches of COD applications due to demands of Transformers and their lengthy training time needs. Thus, this study focused on developing a non-Transformer framework for such use cases.

4.2 Ablation study of modified components

In this subsection, the influence of various modifications that Hybrid-SINet introduces into a baseline SINet workflow is discussed. For this, the developed model is compared to the baseline SINet and various iterations of itself.

The quantitative results of such comparison is shown in Table 2. Both models have been trained on a dedicated subdivision of COD10K dataset and benchmarked against test subsets of COD10K, CAMO and the entirety of NC4K and Military Personnel Data. The training parameters of SINet are set to be similar to those proposed by the authors of its introduction paper in [5]. The training parameters of the iterations of proposed model are discussed in the previous section. Each iteration of Hybrid-SINet contains components introduced by the previous ones. Both models were run on Kaggle using the P100 GPU.

Overall, to improve performance of the baseline SINet model, the following steps were employed in the process of designing proposed framework:

1. **Addition of BGNet-inspired edge attention and integration of edge features into identification module (SINet + EAM + EIM).** The quantitative results show clear improvements in all metrics, besides some isolated cases of S measure decreasing for NC4K and CAMO datasets.
2. **Dimensional reduction of pre-edge integration high level features to decrease computational complexity of the model (Hybrid-SINet + DR).** Overall, the results received here are mixed and this module can be identified as a possible bottleneck, however, we still see marginal increases in, mainly, weighted F score and MAE across various datasets.
3. **Adjustment of the input and output dimensions of the edge integration and dimensional reduction components (Hybrid-Sinet + EIM II + DR II).** The results of the previously introduced model show the need for more throughout experimentation to refine the integration of the new components into the system, in order to find balance between performance of the model and decrease in computational complexity that dimensional reduction offers. From the results, we see an overall incremental improvement of the system, which suggests that the possible performance bottleneck introduced by the

previous modifications regarding worsened generalization abilities has been addressed to some extent.

4. **Adjustments to the structure of SINet workflow by introducing weights (Hybrid-SINet + Weights).** Inspired by the principle in [8] of separation of high- and low-level features, we introduce numerical weights to modules of some of the components. Although there is an incremental increase in some metrics, overall performance worsened, suggesting the need to take a different approach or weight adjustments.

Metrics					
	Baseline SINet	SINet + EAM + EIM	Hybrid-SINet + DR	Hybrid-SINet + EIM II + DR II	Hybrid-SINet + Weights
COD10K					
S_α	.740	.821	.815	.829	.820
F_β^w	.202	.250	.254	.250	.244
MAE	.148	.062	.066	.057	.057
NC4K					
S_α	.705	.681	.676	.688	.647
F_β^w	.504	.536	.537	.544	.487
MAE	.112	.106	.104	.103	.113
CAMO					
S_α	.627	.595	.574	.602	.557
F_β^w	.415	.481	.380	.417	.348
MAE	.161	.148	.151	.146	.158
Military Personnel Data					
S_α	.610	.659	.665	.660	.653
F_β^w	.178	.310	.333	.311	.322
MAE	.044	.022	.021	.023	.023

Table 2: Study of influence of different added components on performance of the developed model.

From the results in Table 2, Hybrid-SINet + EIM II + DR II shows the best overall quantitative increase compared to baseline SINet and the best generalization capabilities from all

iterations of the developed model, thus it was adopted as the current version of the Hybrid-SINet model that was discussed in the previous section and used in further comparisons.

Metric	Hybrid-SINet	SINet	BGNet	$C^2F - Net$	FSPNet
COD10K					
S_α	.829	.740	.844	.829	.756
F_β^w	.250	.202	.271	.256	.187
MAE	.057	.148	.066	.065	.059
NC4K					
S_α	.688	.705	.733	.717	.524
F_β^w	.544	.504	.548	.548	.330
MAE	.103	.112	.096	.099	.117
CAMO					
S_α	.602	.627	.643	.628	.525
F_β^w	.417	.415	.448	.436	.390
MAE	.146	.161	.140	.143	.141
Military Personnel Data					
S_α	.660	.610	.646	.621	.567
F_β^w	.311	.178	.237	.251	.190
MAE	.023	.044	.039	.039	.022

Table 3: Comparison of Hybrid-SINet and SINet across four datasets.

4.3 Comparison of Hybrid-SINet with baseline SINet and SOTA frameworks

The results presented in Table 3 offer a comparative comparison between the baseline SINet model [5], the proposed Hybrid-SINet modification of its architecture and some of SOTA CNN- and Graph-based COD frameworks. The following results have been received by running the models on Kaggle platform with GPU P100 using the training parameters, described in their respective studies, with the exception of the batch size of $C^2F - Net$ that had to be lowered to account for limitations of the Kaggle environment. From transformer-based models, FSPNet [32] was selected as one of the most advanced models of that type with publicly available code.

We present additional benchmark for these models to account for fair assessment of our model and existing SOTA analogs in context of Kaggle environment and, chiefly, to provide quantitative data on an additional dataset, CPD1K (or Military Personnel Data).

Similarly to the previous subsection, the non-Transformer models were trained on a training subset of COD10K and benchmarked against testing subsets of COD10K, CAMO and whole CPD1K and NC4K. For a fair assessment of the capabilities of FSPNet [32], we took a pretrained

model, provided by the model’s authors on the framework’s GitHub page. More details on the reasoning for such a choice are provided in the following subsection.

Evaluation metrics include: Structure Measure (S_α), Weighted F-measure (F_β^w), and Mean Absolute Error (MAE), each reflecting different aspects of segmentation quality.

Hybrid-SINet achieves stronger generalization on diverse and challenging datasets (COD10K, CAMO, NC4K), in comparison to a baseline, indicating its effectiveness in camouflaged object detection tasks. Hybrid-SINet tends to outperform SINet in precision-recall trade-offs (as shown by F_β^w) and average prediction error (MAE), whereas SINet maintains marginally better structural similarity (S_α) in select cases. In comparison to other SOTA non-Transformer frameworks, we note a particularly significant increase on quantitative metrics of the Military Personnel Data dataset measurements, which is also already visible within iterations of the model discussed in the previous subsection.

Metrics	Hybrid-SINet	FSPNet
COD10K		
S_α	.766	.753
F_β^w	.176	.105
MAE	.127	.075
NC4K		
S_α	.722	.524
F_β^w	.470	.259
MAE	.118	.153
CAMO		
S_α	.646	.504
F_β^w	.395	.254
MAE	.171	.181
Military Personnel Data		
S_α	.595	.509
F_β^w	.118	.050
MAE	.066	.052

Table 4: Comparison of CNN-based model (Hybrid-SINet) and Transformer-based model (FSPNet) in underfitting scenarios.

We have to note that despite the fact that there are numerous studies that support the superiority of Transformer-based models over other approaches, the pretrained FSPNet [32] has underperformed in the conducted benchmarks. These results may have been influenced by the environment, as deploying models to other environments may introduce noise, not to mention that Kaggle, as discussed further, can provide much less impressive computational powers compared to the ones recommended for the model. As such, our benchmark is not representative of the performance of Transformers as a whole, but instead should be interpreted as a pretrained Transformer-based model underperforming in conditions of translating into this specific environment. Still, we see that the model maintained competitive MAE results, even outperforming CNN-based alternatives in some cases, despite overall performance issues.

4.4 *Transformer-based training in environment with limited resources and approach comparison*

In this subsection, we discuss possibilities of training Transformer-based models in environments with scarce resources and limited-time conditions. After that, one of the most advanced Transformer-based COD model, FSPNet [32], is compared to CNN-based model in underfitting scenario to further compare different approaches and their use cases.

As for Transformer-based model, FSPNet [32] has been trained for 100 epochs in the Kaggle environment. As discussed previously, training subset of COD10K was used. In the CNN- and the Graph-based approaches, there was little need to change training parameters. However, since Kaggle only supports up to 2 T4 GPUs, quite a few arguments had to be modified to allow for a satisfactory performance of a Transformer time-wise. While the authors note training time of 8 hours, which is comparable to some of the more advanced CNN-based frameworks, like ZoomNet [27], smaller count of available GPUs resulted in a decrease of an amount of processes equal to four times of the original. This resulted in a single training epoch taking more than 25 minutes to complete. To account for this, the image resolution had to be decreased from 384 to 224 (using Vision Transformer described by Touvron, Hugo, et al. [21]), which also enabled an increase of the batch size by up to 8 and the number of workers a data loader could create up to 4. These modifications allowed for higher speeds of a model, up to 15 minutes per epoch. While this is comparable to a training speed of $C^2F - Net$ with a lowered batch size, the latter framework concluded its training in just 30 epochs, which is about 6 hours of training. FSPNet, on the other hand, has been trained for 100 epochs, which took more than 20 hours.

Since the authors recommend to train the model for 200 epochs, a comparison to a similarly underfitted CNN-based model is provided, namely Hybrid-SINet that has been trained for 10 epochs out of 30, in the Table 4. For reference, benchmarks are available in Supplementary Material.

From Table 4, we see that Transformer-based framework maintained better MAE in some cases, while Hybrid-SINet showed overall better performance. It has to be noted that this test on FSPNet has been conducted with lower image resolution than recommended, which potentially influenced the results, but, as previously discussed, had to be done in order to speed up model training. Thus, the reasoning for selecting a pretrained model for conducting tests for the previous subsection could be provided based on this discussion.

From this comparison and previous discussions, we can conclude that Transformer-based models have the following advantages over the models of other approaches:

- In a stable environment, Transformer-based models vastly outperform the alternatives.
- Transformer-based operations are more efficient than convolutions.

However, they face the following shortcomings:

- Despite more efficient operations, Transformer-based models, as a whole, are more robust, resulting in greater complexity.
- Transformer-based models require more computational power to achieve reasonable performance speed.
- Transformer-based models require more epochs to train and may be more prone to underfitting, in comparison to the CNN-based alternatives. This may be relevant when taking their greater training demands into consideration.

- Transformer-based models are more reliant on a backbone, since allowed image resolutions and good data could prove to be more important for models more reliant on a powerful environment.

Overall, this study could provide the reasoning for relevance of selecting a non Transformer-based COD model for systems that have limited resources or can afford limited training time before deploying the model for use. While evaluation of faster SOD-specific Transformers has been done in COD scenarios, for instance, by Yin Bowen, et al. [3], they could approach quantitative results comparable to the Deep Gradient-based models, which is not a developed area of research in COD. Additionally, we note that DGNet has been designed to specifically capture feature dependencies in scenarios of high object obstruction, so it might show more precision in some edge cases, even with similar quantitative results. Meanwhile, modern CNN-based frameworks, such as ZoomNet [27] and Graph-based models, such as BGNet [29] confidently outperform SOD-specific Transformers for COD applications.

Transformer-based approaches are, on the other hand, preferable to use over the other frameworks in environments that could allow for complicated computations and when performance speed is not important, especially, during training.

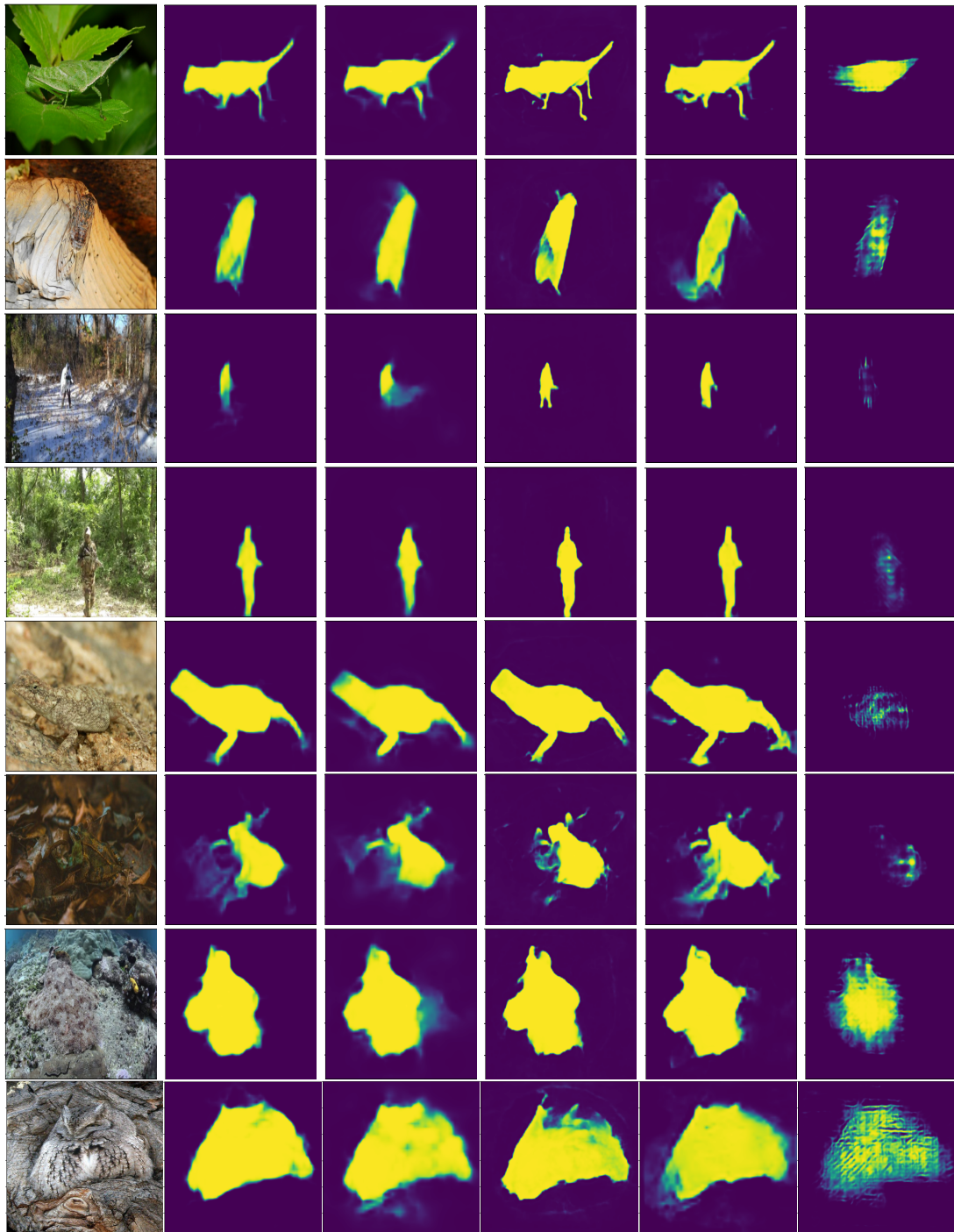
4.5 Visual comparison

In this subsection, visual comparison is demonstrated on how well a designed framework compares to SOTA analogs and a baseline model. It can be seen (Fig. 3) that the proposed Hybrid-SINet model can identify the same concealed objects as SINet and outperforms the baseline in some cases. For more examples, refer to the Supplementary Material.

5 Conclusions

In this work, qualitative and quantitative evaluations were proposed for some baseline models for camouflaged object detection (COD) on benchmark datasets including COD10K, CAMO, and NC4K. The hybrid architecture, Hybrid-SINet, was designed that combines the strengths of existing camouflaged object detection frameworks with targeted enhancements. Hybrid-SINet achieves superior performance in segmenting highly concealed or low-contrast targets, outperforming SINet and other state-of-the-art baselines in accuracy and visual coherence of predictions. The visualization results underscore the ability of the model to discern complex boundaries and subtle contrasts, even in highly cluttered or visually deceptive backgrounds.

Beyond empirical performance, the relevance of robust COD models extends far beyond academic evaluation. The capacity to detect concealed or camouflaged objects has direct implications for a wide array of industrial and societal applications. In disaster management and search-and-rescue operations, these models can assist in locating victims partially covered by debris or vegetation. In autonomous systems, such as self-driving vehicles, drones, or robotic arms, the precise detection of low-visibility hazards or objects is critical for safe navigation and decision-making. In the context of smart cities and infrastructure, Hybrid-SINet-like models can be integrated into systems for automated waste detection, smart waste collection, and scrap material sorting, enabling more efficient and sustainable resource management. The ability to detect concealed waste or foreign materials, whether submerged, partially buried, or disguised, contributes to environmentally safe garbage handling and reduces the risks to urban sanitation systems. Similarly, in infrastructure monitoring, identifying hidden cracks, corrosion, or struc-



(a) Original (b) Hybrid-SINet (c) SINet (d) BGNet (e) $C^2F - Net$ (f) FSPNet(pre-trained)

Figure 3: Visual comparison across datasets tested in Kaggle environment. Each two rows show images from the same dataset (in the following order: COD10K, Military Personnel Data, NC4K, CAMO), and each column displays the original image (left), Hybrid-SINet prediction (second image), and SINet prediction (third image), BGNet prediction (fourth), $C^2F - Net$ prediction (second rightmost) respectively and pretrained FSPNet prediction (rightmost).

tural anomalies early on can help prevent ecological damage, reduce maintenance costs, and prolong infrastructure lifespan.

The obtained results also resonate with broader research directions in computer vision. For instance, the performance of Hybrid-SINet in segmenting concealed objects under complex conditions provides a significant advancement for several critical computer vision domains. Its enhanced precision in discerning low-contrast targets and complex boundaries can directly improve object detection in infrared UAV imagery for search-and-rescue [1]. Furthermore, the models robust segmentation capabilities could be leveraged to generate higher-quality coarse annotations, accelerating the preparation of datasets for autonomous driving [25]. Finally, the architectures ability to parse cluttered, deceptive scenes is highly relevant for improving the accuracy and robustness of semantic segmentation in stereo vision systems for traffic analysis [24], where detecting obscured hazards is paramount for safety.

These results point to the feasibility and effectiveness of deploying advanced COD models in diverse, real-world environments. Future work will explore domain adaptation and lightweight deployment strategies to ensure scalability and integration into edge devices, making such technologies accessible and impactful in both high-resource and resource-constrained settings.

Supplementary material

The codes are available at:

- Proposed model, Hybrid-SINet code and benchmark on Kaggle:
<https://www.kaggle.com/code/ivanomelchenkoim11/article-hybrid-sinet-benchmark>
- Baseline SINet [5] benchmark on Kaggle (for Subsection 4.3):
<https://www.kaggle.com/code/ivanomelchenkoim11/article-sinet-benchmark>
- BGNet [29] benchmark on Kaggle (for Subsection 4.3):
<https://www.kaggle.com/code/ivanomelchenkoim11/article-bgnet-benchmark>
- $C^2F - Net$ [28] benchmark on Kaggle (for Subsection 4.3):
<https://www.kaggle.com/code/ivanomelchenkoim11/article-c2f-net-benchmark>
- Pretrained FSPNet [32] test on Kaggle (for Subsection 4.3):
<https://www.kaggle.com/code/ivanomelchenkoim11/article-fspnet-benchmark-pretrained>
- Hybrid-SINet underfitting case benchmark on Kaggle (for Subsection 4.4):
<https://www.kaggle.com/code/ivanomelchenkoim11/article-hybrid-sinet-underfitting>
- FSPNet [32] 100 epochs benchmark on Kaggle (for Subsection 4.4):
<https://www.kaggle.com/code/ivanomelchenkoim11/article-fspnet-benchmark>

The datasets are available at:

- COD10K [5] dataset on Kaggle (testing and training subsets were used in Section 4):
<https://www.kaggle.com/datasets/ivanomelchenkoim11/cod10k-dataset>
- Military Personnel Data (CPD1K) [30] dataset on Kaggle (dataset was used for testing in Section 4):
<https://www.kaggle.com/datasets/ivanomelchenkoim11/military-personnel-dataset-dataset/settings>

- NC4K [31] dataset on Kaggle (dataset was used for testing in Section 4):
<https://www.kaggle.com/datasets/ivanomelchenkoim11/nc4k-dataset>
- CAMO [22] dataset on Kaggle (testing subset was used in Section 4):
<https://www.kaggle.com/datasets/ivanomelchenkoim11/camo-dataset>

The licenses of datasets are set according to the original, or as 'unknown'. For paper citations and original sources, refer to the 'Description' or 'Provenance' section on given Kaggle links. For provided code links, paper citations and GitHub references are available in the first code cell for each earlier proposed model respectively.

Acknowledgements

This research was supported by the Office of Naval Research (ONRG) and US National Academy of Science (US NAS) (IMPRESS-U initiative, No. STCU-7125) in the part of exploratory research on the new robust machine learning approaches for object detection and classification; and by the National Research Fund of Ukraine under grant 2025.06/0100 in the part of the development of neural network models for object detection, and by the Ministry of Education and Sciences of Ukraine (MESU), grant 2715r, as part of the search for advanced deep learning methods based on the requirements posed by devices on Edge Computing and Edge Intelligence layers. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONRG, US NAS, and NRFU.

References

- [1] Andrii Polukhin, Yuri Gordienko, Gert Jervan, and Sergii Stirenko. Object detection for rescue operations by high-altitude infrared thermal imaging collected by unmanned aerial vehicles. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 490504. Springer, 2023
- [2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:2017
- [3] Bowen Yin, Xuying Zhang, Deng-Ping Fan, Shaohui Jiao, Ming-Ming Cheng, Luc Van Gool, and Qibin Hou. Camoformer: Masked separable attention for camouflaged object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024
- [4] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7177-7188, 2021
- [5] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2777-2787, 2020
- [6] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Ao Luo, Hong Cheng, and Deng-Ping Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4146-4155, 2021
- [7] Fengyang Xiao, Sujie Hu, Yuqi Shen, Chengyu Fang, Jinfa Huang, Chunming He, Longxiang Tang, Ziyun Yang, and Xiu Li. A survey of camouflaged object detection and beyond. *arXiv preprint arXiv:2408.14562*, 2024
- [8] Ge-Peng Ji, Deng-Ping Fan, Yu-Cheng Chou, Dengxin Dai, Alexander Liniger, and Luc Van Gool. Deep gradient learning for efficient camouflaged object detection. *Machine Intelligence Research*, **20**(1):92-108, 2023
- [9] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8772-8781, 2021

- [10] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *Proceedings of the Asian conference on computer vision*, 2020
- [11] Jialun Pei, Tianyang Cheng, Deng-Ping Fan, He Tang, Chuanbo Chen, and Luc Van Gool. Osformer: One-stage camouflaged instance segmentation with transformers. In *European conference on computer vision*, pages 19-37, Springer, 2022
- [12] Ke Li, Di Wang, Zhangyuan Hu, Wenxuan Zhu, Shaofeng Li, and Quan Wang. Unleashing channel potential: Space-frequency selection convolution for sar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17323-17332, 2024
- [13] Liqiong Wang, Jinyu Yang, Yanfu Zhang, Fangyi Wang, and Feng Zheng. Depth-aware concealed crop detection in dense agricultural scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17201-17211, 2024
- [14] Mingchen Zhuge, Xiankai Lu, Yiyu Guo, Zhihua Cai, and Shuhan Chen. Cubenet: X-shape connection for camouflaged object detection. *Pattern Recognition*, 127:108644, 2022
- [15] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213-229. Springer, 2020
- [16] Pia Bideau and Erik Learned-Miller. A detailed rubric for motion segmentation. *arXiv preprint arXiv:1610.10033*, 2016
- [17] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12997-13007, 2021
- [18] Rahul Chauhan, Kamal Kumar Ghanshala, and RC Joshi. Convolutional neural network (cnn) for image detection and recognition. In *2018 first international conference on secure cyber computing and communication (ICSCCC)*, pages 278-282. IEEE, 2018
- [19] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248-255, 2014
- [20] Songtao Liu, Di Huang, et al. Receptive field block net for accurate and fast object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 385-400, 2018
- [21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Her'è J'egou. Training data-efficient image transformers distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020
- [22] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabran network for camouflaged object segmentation. *Computer vision and image understanding*, **184**:45-56, 2019
- [23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'ar, and C Lawrence 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*, pages 740-755. Springer, 2014
- [24] Vlad Taran, Nikita Gordienko, Yuriy Kochura, Yuri Gordienko, Alexandr Rokovyi, Oleg Alienin, and Sergii Stirenko. Performance evaluation of deep learning networks for semantic segmentation of traffic stereo-pair images. In *Proceedings of the 19th international conference on computer systems and technologies*, pages 7380, 2018
- [25] Vlad Taran, Yuri Gordienko, Alexandr Rokovyi, Oleg Alienin, and Sergii Stirenko. Impact of ground truth annotation quality on performance of semantic image segmentation of traffic conditions. In *International conference on computer science, engineering and education applications*, pages 183193. Springer, 2019
- [26] Yang Liu, Cong-qing Wang, and Yong-jun Zhou. Camouflaged people detection based on a semi-supervised search identification network. *Defence Technology*, **21**:176-183, 2023
- [27] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan 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*, pages 2160-2170, 2022
- [28] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware cross-level fusion network for camouflaged object detection. *arXiv preprint arXiv:2105.12555*, 2021
- [29] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xiang. Boundary-guided camouflaged object detection. *arXiv preprint arXiv:2207.00794*, 2022
- [30] Yunfei Zheng, Xiongwei Zhang, Feng Wang, Tieyong Cao, Meng Sun, and Xiaobing Wang. Detection of people with camouflage pattern via dense deconvolution network. *IEEE Signal Processing Letters*, **26**(1):29-33, 2018
- [31] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *Proceedings of the IEEE/CVF conference on*

- computer vision and pattern recognition*, pages 11591-11601, 2021
- [32] Zhou Huang, Hang Dai, Tian-Zhu Xiang, Shuo Wang, Huai-Xin Chen, Jie Qin, and Huan Xiong. Feature shrinkage pyramid for camouflaged object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5557-5566, 2023