Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01386-3

ORIGINAL ARTICLE

Lightweight camouflaged object detection model based on multilevel feature fusion

Qiaoyi Li[1] · Zhengjie Wang[1] · Xiaoning Zhang[1] · Hongbao Du[1]

Received: 11 October 2023 / Accepted: 10 February 2024

Abstract

The intrinsic similarity between camouflaged objects and background environment impedes the automatic detection/segmentation of camouflaged objects, and novel network architectures for deep learning are promising to overcome this challenge and improve detection accuracy. However, these existing network architectures for distinguishing between camouflaged objects and their backgrounds do not account for the constraint of detection speed, which results in high computational complexity and the inability to meet the requirements of rapid detection. Therefore, based on the human visual system, this study proposes a single-stage lightweight camouflage object detection network using multilevel feature fusion, integrating features of various feature layers and receptive field sizes. Using three benchmark datasets for normal camouflaged objects, the lightweight network (LINet) model demonstrated an accuracy superior to those of six existing mainstream camouflaged object detection methods. Its detection speed, 126.3 frames per second, is significantly higher than those of the existing mainstream methods, enabling rapid detection with a maximum increase of 187.62%. The accuracy of LINet is the minimum and maximum for Resnet101 and Resnet152, respectively. These findings pave the way for diverse applications of camouflaged target detection algorithms.

Keywords Camouflaged object detection · Lightweight · Multilevel feature fusion · Detection speed · Feature extraction network

Introduction

The term "camouflage" was initially used to describe the behavior of certain species imitating the appearance, color, and other characteristics of their environment to hide from predators or hunt their prey [1]. For instance, certain insects and fish can change their bodily appearances to match the colors and patterns of their surrounding environments. This mechanism is utilized by humans in warfare and art. Soldiers and war equipment use camouflage or paint (i.e., artificial camouflage objects) to blend in with the surrounding environment for evading detection by humans and machines [2]. Artificial camouflage has been applied in entertainment and art (such as body painting). Figure 1a and b depicts camouflaged objects (insects and fish), whereas Fig. 1c and d depicts artificial camouflage (soldiers and body paintings).

Recently, camouflaged object detection (COD), i.e., identifying objects hidden in the background, has gained scholarly attention in the field of computer vision. In addition to its academic significance, COD has diverse applications, such as military target detection, medical diagnosis [3, 4], species discovery, and animal detection [5]. However, COD is highly challenging owing to the nature of camouflage, resulting in a high level of inherent similarity between the candidate object and background, which complicates the detection of camouflaged objects by humans and machines. As shown in Fig. 2, the boundaries of the two butterflies (target objects) blend with the bananas (background), rendering the COD more challenging compared to the traditional and salient object detection [6–10] or general object detection [11–13].

✉ Zhengjie Wang
wangzhengjie@bit.edu.cn

Qiaoyi Li
joe_li@bit.edu.cn

Xiaoning Zhang
xnzhang@bit.edu.cn

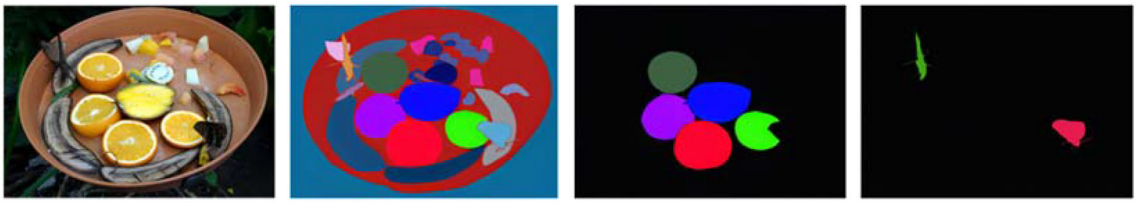
Hongbao Du
derek_dhb@bit.edu.cn

[1] School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China

Published online: 09 March 2024

(a) Camouflaging insects (b) Camouflaging fish (c) Military camouflage disguise (d) Body paint camouflage

Fig. 1 Instances of camouflage in the COD 10K dataset [3, 4]



(a) Input image (b) Panoramic segmentation target (c) Salient object target (d) Camouflaged object target

Fig. 2 Common detection tasks

In initial investigations, the majority of approaches employ basic features such as texture, edges, luminosity, and color to differentiate the camouflaged object from its surroundings [36–41]. Nevertheless, camouflage often disrupts the inherent features to deceive the observer, rendering these approaches comparatively less efficacious. To this end, deep-learning-based methods have been proposed for COD, which exhibit significant potential and can be classified into the following three approaches:

(1) Designing targeted network modules/architectures to effectively investigate the discriminative features of camouflaged objects and improve detection performance. For instance, C^2FNet [14] and UGTR [15]. This method requires in-depth adjustments and optimizations to the network, which increases the complexity in design and implementation.
(2) Incorporating auxiliary tasks into joint/multitask learning frameworks, such as classification tasks [2], edge extraction [16], salient object detection [17], and camouflaged object ranking [18]. Herein, valuable additional clues from shared features can be mined to significantly improve the feature representation of camouflaged targets, thereby enhancing the model's generalization ability and efficiency, and addressing data scarcity issues. However, the demand for computational and storage resources will be increased.
(3) A biomimetic approach wherein the predatory behavioral processes of animals in nature are simulated into design networks, such as SINet [3, 4] and PFNet [19]. Simulating complex natural behaviors can enhance the model's sensitivity and detection accuracy for camouflage targets. However, it requires a large amount of data and computational resources.

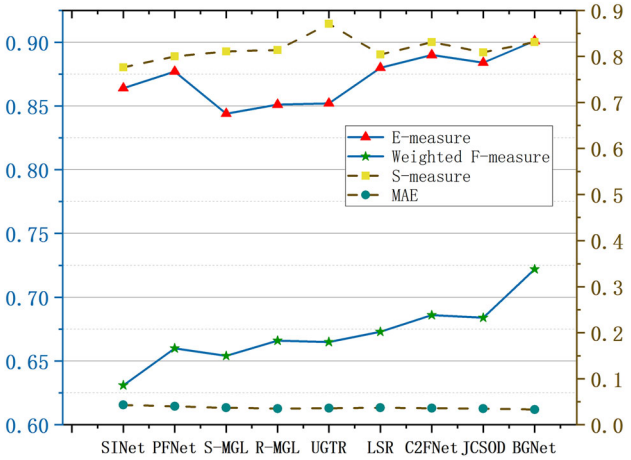


Fig. 3 Accuracy statistics of the camouflage detection models

Significant progress has been made using the aforementioned methods, such as SINet [3, 4] and BGNet [20]. Figure 3 shows the accuracy results for the COD10k test set, wherein the E-measure [22] increased from 0.864 to 0.901, S-measure [23] increased from 0.776 to 0.831, weighted F-measure [24] increased from 0.631 to 0.722, and mean absolute error (MAE) [25] decreased from 0.043 to 0.033. Evidently, the accuracy of the models increased; however, the following two major issues persisted:

(1) While improving the detection accuracy of the model, its complexity increased significantly. However, the detection time was neglected, due to which the requirements of fast detection in real-world applications were not met. For instance, the SINet algorithm [3, 4], a representative biomimetic method, primarily includes the following three modules: a receptive field module (RFM), partial decoder component (PDC), and search attention module. During algorithm implementation, RFM and PDC are called seven and two times, respectively, thereby significantly increasing the computational complexity. The mutual graph learning (MGL) algorithm [16] is a representative method that incorporates auxiliary tasks into the learning framework, which encodes the edge and object features together into the graph convolutional network and enhances feature representation using the graph interaction module. This increases the complexity of the model and the associated computational burden. UGTR [15] is a representative method that designs targeted network structures and integrates new components, such as the uncertainty quantification network (UQN), prototype transformer (PT), and uncertainty-guided transformer (UGT). These transform the deterministic mapping process of traditional COD models into an uncertainty-guided contextual reasoning process, thereby increasing the computational complexity.

(2) Primarily, Resnet50 [21] has been implemented as the backbone network to investigate model accuracy; however, comparisons between the schemes of deepening the feature extraction network for high accuracy and new network structures are insufficient [3, 4, 14–20]. Therefore, investigating the impact of various backbone networks on the accuracy and speed of COD models is crucial.

This study proposes a COD network based on multilevel feature fusion to reduce model complexity, achieve network lightweighting, and rapidly detect camouflaged targets. First, low-, medium-, and high-level features were extracted using a backbone network, and a dense connection strategy [26] was used to fuse features from different layers and preserve more information. Second, RFM [27] was introduced to extract and fuse the features of various receptive field sizes. Third, the multilevel features were fused with different feature layers, and receptive fields were fed into the decoder to obtain the predicted image. Finally, various backbone networks were compared to test the performance of the lightweight camouflaged target detection model.

Related works

Significant advances have been made with deep-learning-based COD models. Based on biology, a number of approaches [3, 4, 14, 19, 32–35] have been proposed. Several works come up with different perceptual systems that mimic human behavior vis-a-vis camouflaged objects. For instance, Rank-Net [33] divides the entire detection process into three stages: localization, segmentation and ranking. Inspired by humans attention coupled with the coarse-to-fine detection strategy, SegMaR [34] integrates Segment, Magnify and Reiterate in a multi-stage detection fashion. As can be seen above, Rank-Net [33] and SegMaR [34] are divided into several segments, which realizes the detection effect optimization process from coarse to fine. In addition, numerous researchers have attempted to improve camouflage target detection performance by simulating the predatory behavior of animals. For instance, SINet's framework is based on the search and recognition stages of animal predation comprising the following two main modules: the search module (SM) for searching camouflage objects and the recognition module (RM) for accurate detection. Furthermore, PFNet comprises the following two key modules: the positioning module (PM) to simulate the detection process during predation and focus module (FM) to execute the recognition process by focusing on blurred regions to improve the initial segmentation results. SINET and PFNet divide the camouflage target detection process into two stages. The candidate regions are generated in the first stage, and further localization is performed in the second stage to improve detection accuracy.

Contrastingly, this study simulates the human visual system and proposes a single-stage camouflage target detection framework to accelerate detection. The proposed method fuses features of different layers to obtain more distinguishable features, and introduces RFM [27] to simulate the sizes and eccentricities of receptive fields in the human visual system to enhance the fused features. Finally, the enhanced features are fed into the decoder and the final results are obtained.

Proposed method

Problem description

The COD model is represented by a function M_Θ parameterized by weights Θ. M_Θ accepts an image I as the input and generates a camouflage map $C \in [0,1]$. The objective is to learn Θ using a given labeled training dataset $\{I_i, C_i\}_{i=1}^{N}$, where I_i denotes a training image, C_i denotes the image label, and N denotes the number of training images.

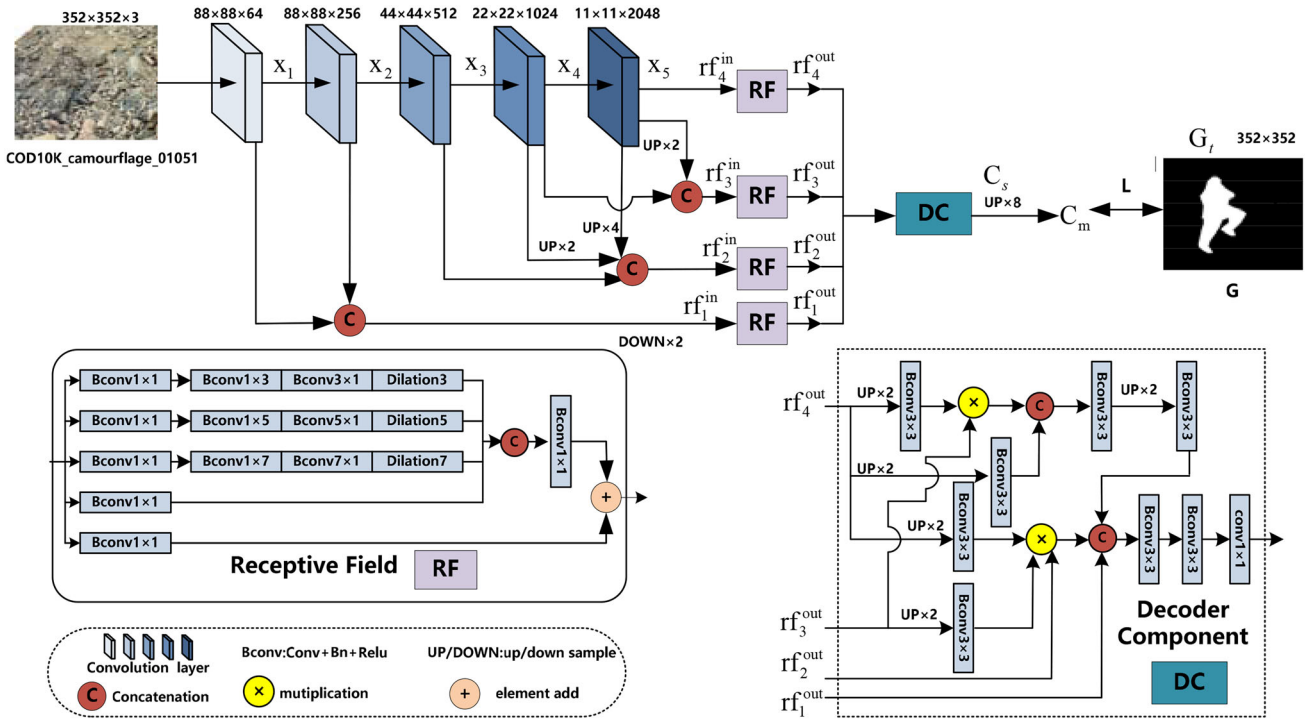


Fig. 4 The framework of lightweight camouflaged object detection algorithm

Overall architecture

Based on multilevel feature fusion, this study proposes a fast and single-branch COD framework called the lightweight network (LINet) that includes feature extraction, receptive field, and decoder modules. The feature extraction module extracts and uses features from various layers. RFM simulates the structures of receptive fields in the human visual system [27], thereby enhancing the feature extraction ability. The decoder module receives multilevel features and outputs feature maps (see Fig. 4).

Feature extraction module

The proposed model was designed based on Resnet50 [21], which is the most widely used backbone network for deep COD. Given an input image I of size H × W, the features were extracted at five levels, denoted as $\{x_i, i = 1, 2, 3, 4, 5\}$. Low-level features in shallow layers preserve spatial details for constructing object boundaries, while high-level features in deep layers retain semantic information for locating objects [42]. Thereafter, a dense connection strategy [26] was used to fuse information from different levels. To preserve spatial details for constructing object boundaries, the extracted low-level features $\{x_1, x_2\}$ were fused via concatenation, and a max-pooling operation was applied to halve the resolution and obtain feature rf_1^{in}. To preserve the semantic information of the target object, the high-level feature x_5 was upsampled using the bilinear interpolation method to increase its resolution by twofold and obtain the feature $x_5^{up\times2}$. Thereafter, the features$\{x_4, x_5^{up\times2}\}$ were fused via concatenation to obtain feature rf_3^{in}. To preserve more characteristic information, the extracted high-level features x_4 and x_5 were upsampled using bilinear interpolation by the factors of two and four to obtain the features $x_4^{up\times2}$ and $x_5^{up\times4}$, respectively. Furthermore, the features$\{x_3, x_4^{up\times2}, x_5^{up\times4}\}$ were fused by concatenation to obtain feature rf_2^{in}. Finally, fusion features $\{rf_1^{in}, rf_2^{in}, rf_3^{in}, rf_4^{in} = x_5\}$ retaining more distinguishing features were obtained.

RFM

After obtaining the candidate features $\{rf_1^{in}, rf_2^{in}, rf_3^{in}, rf_4^{in}\}$ using the feature extraction module, an improved RFM [27] simulating the human visual system was used to fuse the features with different receptive fields to generate the output features $\{rf_1^{out}, rf_2^{out}, rf_3^{out}, rf_4^{out}\}$. The internal structure of RFM can be divided into the following two parts: multibranch convolution layer with a different kernel number and tail expansion pooling or convolution layer. The former can obtain rich hierarchical features, whereas the latter can capture more contextual information in a larger area while maintaining the same number of parameters. Particularly, RFM consists of five branches $\{b_k, k = 1, 2, 3, 4, 5\}$. In each branch, the first convolutional layer has a size of 1 × 1

to reduce the number of channels to 32. Thereafter, branches b_3, b_4 and b_5 are connected to the following three additional convolutional layers: $1 \times (2k-3)$, $(2k-3) \times 1$, and a 3×3 layer with a dilation rate of $(2k-3)$. Branches b_3, b_4 and b_5 were fused using a concatenation operation, and their channel sizes were reduced to 32 using a 1×1 convolution operation, while the resolution remained equal to that of the input. Finally, after adding the branch b_1, the entire module was fed into a ReLU function, and the features $rf_{\mathrm{j}}^{\mathrm{out}}\{j = 1, 2, 3, 4\}$ were obtained.

Decoder module

After obtaining the candidate features $rf_{\mathrm{j}}^{\mathrm{out}}\{j = 1, 2, 3, 4\}$ using RFM, the camouflage map C_{s} can be computed using the decoder module as follows:

$$C_s = D\left(rf_j^{\mathrm{out}}\{j = 1, 2, 3, 4\}\right)$$

The obtained features rf_j^{out} were fed into the decoder module and a multiplication operation was used to minimize the gaps between features from multiple levels. Particularly, rf_2^{out} was set to f_2^c=rf_2^{out}. The feature $\{rf_j^{\mathrm{out}}, \mathrm{j} > 2\}$ was updated to f_j^c using element-wise multiplication with all the deeper features, performed as follows:

$$f_k^c = rf_j^{\mathrm{out}} \otimes \prod_{k=j+1}^{4} B\mathrm{Conv}\left(\mathrm{Up}\left(rf_k^{\mathrm{out}}\right)\right),\ j \in [2, 3]$$

where Bconv $(\cdot)$ is a sequential operation combining 3×3 convolution, batch normalization, and a ReLU activation function; and UP $(\cdot)$ is an upsampling operation with a ratio of 2^{k-j}. Finally, these discriminative features were combined using a concatenation operation to obtain the feature map C_{s}. The cross-entropy loss [9, 10] was considered as the loss function, formulated as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[G_i \ln \mathrm{Cm}_i + (1 - G_i)\ln(1 - \mathrm{Cm}_i)],$$

where N represents the number of samples, Cm is the object mask obtained by upsampling C_{s} to a resolution of 352×352, and G is the label.

Benchmark experiments

Experimental setup

Various annotated datasets have been released to promote the development of deep learning-based COD technology. The CHAMELEON dataset [28] consists of 76 images collected from the internet using "camouflaged animals" as keywords via the Google search engine. The CAMO dataset [2] contains 2500 images (2000 for training, 500 for testing) covering eight camouflage categories. The COD10K dataset [3, 4] was the first large-scale and challenging dataset to be constructed, consisting of 10,000 images covering 78 camouflage categories. The NC4K dataset [18] contains 4,121 images with additional localization and ranking annotations, facilitating the localization and ranking of camouflaged objects. The dataset used to evaluate the performance of LINet was trained with the following two different settings: (i) the CAMO default training set containing 1000 images, and (ii) the CAMO + COD10K default training set containing 4040 camouflage images. The accuracy and speed of the model were evaluated using the test sets NC4K, CAMO, and COD10K.

The following four renowned evaluation metrics were used in the experiment: MAE, E-measure (E_φ), S-measure (S_α), and weighted F-measure (F_β^{W}).

LINet was implemented using PyTorch and trained using the Adam optimizer [29]. During the training phase, the batch size and learning rate were set to 15 and 1e-4, respectively. The experiments were performed on an Intel(R) Xeon(R) Gold 6135 CPU at 3.40 GHz and RTX 2080TI platforms.

To demonstrate its effectiveness, the proposed approach was compared with the following six mainstream COD models: SINet [3, 4], PFNet [19], S-MGL [16], R-MGL [16], C2FNet [14], and BGNet [20]. To fairly compare the accuracy and speed of each model, the test results of the aforementioned methods were retrained and tested using a batch size of 15 while keeping the other settings constant using the author-provided open-source code. LSR [18] re-annotated and sorted the dataset, and JCSOD [17] introduced the DUTS training set [30] and PASCAL VOC 2007 dataset [31] to extract saliency and camouflage features during training. Simultaneously, the occupancy of memory resources was remarkably increased and novel datasets were introduced. Therefore, to ensure the fairness of the comparative experiment, LSR [18] and JCSOD [17] were not included in the comparison range.

Results and data analysis

Accuracy and speed analysis of the model with setting (i)

To measure the relative improvement in LINet compared to other mainstream COD models, the mean precision change rate metric $\mathrm{R_a}$ was proposed:

$$R_a = \frac{\sum_{\mathrm{i}}\sum_{\mathrm{j}}(1 - \frac{B_{ij}}{A_{ij}}) + \sum_{\mathrm{i}}(1 - \frac{D_i}{C_i})}{\mathrm{num_i} * \mathrm{num_j}}$$

Table 1 Comparison of the detection accuracy of the proposed method with six mainstream methods trained on the three benchmark datasets with setting (i)

Method	Pub./Year	CAMO-test				COD10K-test				NC4K			
		S_α	E_φ	F_β^W	M	S_α	E_φ	F_β^W	M	S_α	E_φ	F_β^W	M
SINet(i)	CVPR´20	0.752	0.781	0.597	*0.103*	**0.729**	*0.779*	*0.477*	*0.065*	**0.785**	*0.824*	*0.624*	*0.075*
PFNet(i)	CVPR´21	0.770	0.837	0.670	0.090	0.736	0.816	0.545	0.059	0.794	0.861	0.685	0.065
S-MGL(i)	CVPR´21	0.771	0.832	0.666	0.092	0.748	0.822	0.557	0.052	0.798	0.863	0.694	0.061
R-MGL(i)	CVPR´21	0.777	0.838	0.680	0.087	0.745	0.824	0.570	**0.051**	0.801	0.865	0.697	0.060
C^2FNet(i)	IJCAI´21	0.782	0.839	**0.688**	**0.084**	0.752	**0.831**	**0.582**	**0.051**	**0.806**	**0.869**	**0.709**	**0.059**
BGNet(i)	IJCAI´22	**0.799**	**0.846**	0.679	**0.084**	**0.766**	**0.831**	0.547	0.057	**0.816**	0.863	0.674	0.065
LNet(i)	Ours	*0.729*	*0.756*	*0.587*	*0.103*	*0.725*	0.783	0.497	0.059	*0.777*	*0.824*	0.639	0.072

The best and worst results are highlighted in bold and italics, respectively

Table 2 Comparison of detection speeds of the proposed method and six mainstream methods at the training setting (i)

Attribute method		SINet(i)	PFNet(i)	S-MGL(i)	R-MGL(i)	C^2FNet(i)	BGNet(i)	LNet(i)
FPS	CAMO-test	69.4	60.2	46.0	*42.4*	52.3	63.5	**112.0**
	COD10K-test	70.9	59.6	45.9	*42.0*	50.7	63.6	**119.8**
	NC4K	71.2	61.7	46.2	*43.4*	53.0	63.1	**114.2**

The best and worst results are highlighted in bold and italics, respectively

where j represents the precision indicators S_α, E_φ, F_β^W; i represents the test dataset; A_{ij} and B_{ij} represent the j-values corresponding to the other mainstream COD models and LINet detection model on the test dataset i, respectively; D_i and C_i represent the values of MAE precision indicator corresponding to the other mainstream COD models and LINet model on test dataset i, respectively; num_i and num_j represent the number of categories in the test dataset and j, respectively (see Table 1).

To measure the speed variations in LINet relative to other mainstream COD models, the following average speed change rate index R_v was proposed:

$$R_v = \frac{\sum_\text{i} (V_i - U_i)}{\sum_i U_i}$$

where V_i and U_i denote the frames per second (FPS) of LINet and other mainstream COD models on the test set i, respectively (see Table 2).

Figure 5 shows the detection accuracy and speed change rates of LINet relative to the other mainstream models when the training was set to (i). Evidently, LINet exhibits the highest accuracy drop rate of 10.33% and detection speed improvement rate of 121.79% compared to C^2FNet. Compared to SINet, LINet exhibits a slight accuracy improvement rate of 0.99% and detection speed improvement rate of 63.59%. Compared to the six mainstream camouflaged target

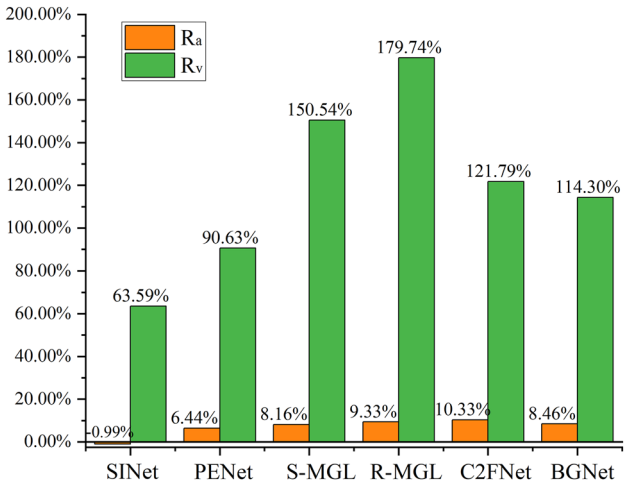


Fig. 5 Detection accuracy and speed change rate of LINet (i) compared to mainstream models

detection models, LINet significantly reduced the inference time while maintaining an insignificant decrease in accuracy. Hence, LINet is a promising solution to the problem of COD. Figure 6 shows a qualitative comparison between LINet and the six baselines when the training setting is (i). In a few challenging tasks (such as undefined boundaries, occlusion, and small objects), LINet exhibits instances of missed detections, such as the leg details in Fig. 6b and d and head details in Fig. 6f. However, to recognize relatively

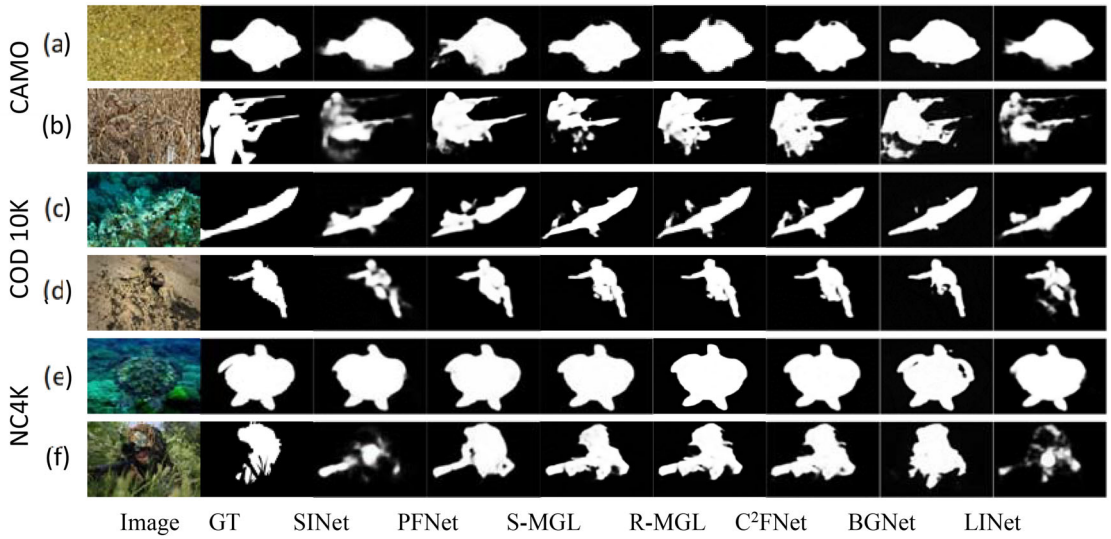


Fig. 6 Qualitative comparison of LINet (i) and mainstream models

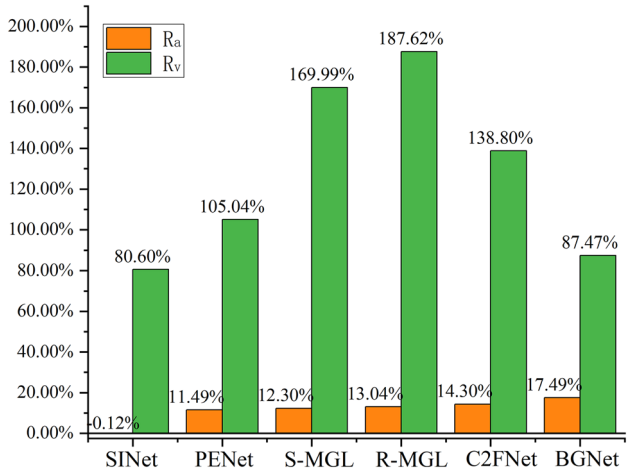


Fig. 7 Detection accuracy and speed variation rate of LINet (ii) compared to those of the mainstream models

normal camouflaged objects (such as (a), (c), and (e) in Fig. 6), LINet obtained more accurate positions compared to the best-performing model C^2FNet.

Precision and speed analysis of the model with setting (ii)

Figure 7 shows the detection accuracy and speed change rate of LINet relative to other mainstream models at the training setting (ii). Compared to BGNet, LINet exhibits the highest accuracy drop rate of 17.49% and speed increase rate of 87.47%. Compared to SINet, LINet exhibits a slight accuracy improvement rate of 0.12% and speed increase rate of 80.60%. Compared to the model trained with setting (i), the overall accuracy drop rate increased, indicating that LINet was more suitable for smaller data scenarios. However, the LINet detection time was significantly reduced, demonstrating the robustness of the proposed framework (see Tables 3, 4).

Figure 8 shows a qualitative comparison between LINet and the six baselines at the training setting (ii). To recognize relatively normal camouflaged objects (such as (a), (c), and (e) in Fig. 8), LINet performed similarly as BGNet that exhibited the best accuracy. However, for a few challenging tasks (such as (a), (c), and (e) in Fig. 8), the accuracy of the LINet model decreased noticeably. Compared with other mainstream models, the edge information detected by LINet is relatively blurry, indicating false detection. Figure 8f indicates that LINet identifies clear edge information of the firearms; however, it is not annotated in the labeled image.

Impact of various backbones on the detection accuracy and speed of LINet

The results of Table 5 indicate that when Resnet152 was used as the feature extraction network over Resnet50, LINet achieved the best performance in terms of accuracy, exhibiting an average accuracy improvement rate of 5.16%. However, the average detection speed decreased by 160.86%. Using Resnet152, the average accuracy of LINet was slightly lower than that of R-MGL having a similar detection speed, and the rate of decrease was 4.17%. Therefore, proposing novel model structures in the field of COD is crucial to improve model accuracy while maintaining detection speed.

The results of Table 6 indicate that when Resnet152 was used as the feature extraction network over Resnet50, LINet achieved the best performance in terms of accuracy, with an average accuracy improvement rate of 0.9%. However, the average detection speed decreased by 180.22%. Using

Table 3 The detection precision of the proposed method is compared to those of six other mainstream methods on three benchmark datasets at the training setting (ii)

Method	Pub./year	CAMO-test				COD10K-test				NC4K			
		S_α	E_φ	F_β^W	M	S_α	E_φ	F_β^W	M	S_α	E_φ	F_β^W	M
SINet(ii)	CVPR´20	0.723	*0.728*	*0.570*	*0.107*	**0.762**	*0.782*	*0.551*	*0.050*	0.803	0.829	0.674	0.065
PFNet(ii)	CVPR´21	0.767	0.818	0.673	0.089	0.798	0.875	0.657	0.039	0.826	0.884	0.743	0.053
S-MGL(ii)	CVPR´21	0.770	0.827	0.679	0.088	0.800	0.879	0.665	0.039	0.830	0.890	0.751	0.051
R-MGL(ii)	CVPR´21	0.774	0.830	0.681	0.086	0.802	0.882	0.672	0.038	0.832	0.893	0.754	0.050
C^2FNet(ii)	IJCAI´21	0.783	**0.847**	0.706	0.084	0.807	0.887	0.678	0.037	0.837	0.898	0.762	0.049
BGNet(ii)	IJCAI´22	**0.796**	0.846	**0.722**	**0.077**	**0.830**	**0.895**	**0.721**	**0.032**	**0.848**	**0.901**	**0.784**	**0.045**
LNet(ii)	Ours	*0.722*	0.732	*0.570*	**0.106**	*0.760*	0.791	0.553	0.049	*0.797*	*0.827*	*0.669*	*0.066*

The best and worst results are highlighted in bold and italics, respectively

Table 4 Comparison of the detection speeds (FPS) of the proposed method and those of the six state-of-the-art methods at the training setting (ii)

Attribute method		SINet(ii)	PFNet(ii)	S-MGL(ii)	R-MGL(ii)	C^2FNet(ii)	BGNet(ii)	LNet(ii)
FPS	CAMO-test	70.0	60.9	46.2	*42.8*	52.3	64.2	**124.9**
	COD10K-test	66.0	60.2	46.1	*43.1*	51.8	66.6	**123.0**
	NC4K	71.2	61.4	46.3	*44.2*	52.6	68.8	**126.3**

The best and worst results are highlighted in bold and italics, respectively

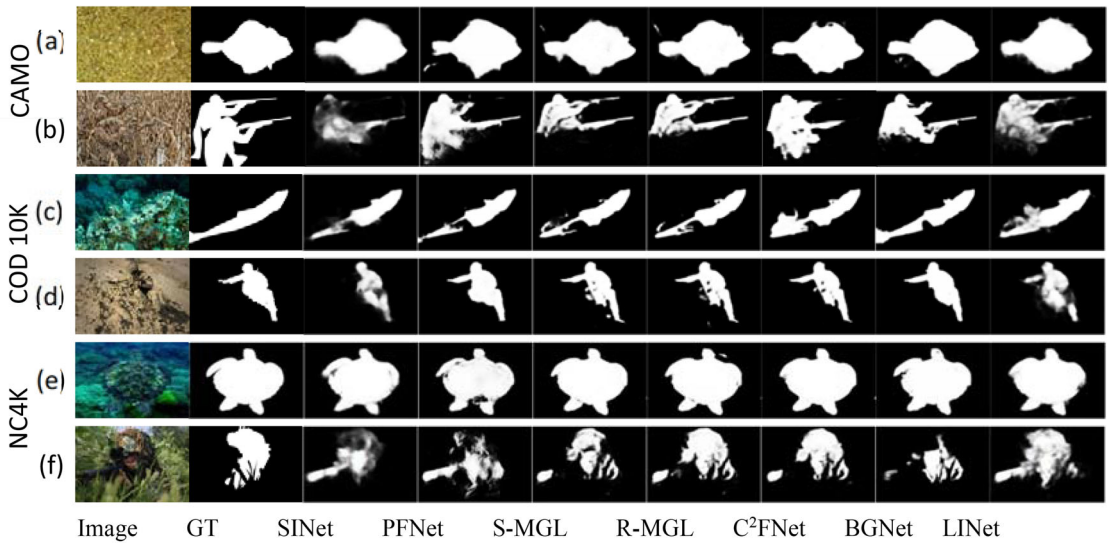


Fig. 8 Qualitative comparison of LINet (ii) with other mainstream models

Table 5 Comparison of detection accuracy and speed of LINet under various feature extraction networks at training setting (i)

Backbone	CAMO-test				COD10K-test				NC4K				Speed
	S_α	E_φ	F_β^W	M	S_α	E_φ	F_β^W	M	S_α	E_φ	F_β^W	M	FPS
Resnet50	*0.729*	*0.756*	*0.587*	*0.104*	*0.725*	*0.783*	*0.497*	*0.059*	*0.777*	*0.823*	*0.639*	*0.072*	**115.3**
Resnet101	0.740	0.775	0.606	0.098	0.731	0.785	0.506	0.057	0.783	0.830	0.648	0.070	57.4
Resnet152	**0.750**	**0.789**	**0.625**	**0.094**	**0.740**	**0.800**	**0.527**	**0.055**	**0.795**	**0.842**	**0.669**	**0.065**	**44.2**

The best and worst results are highlighted in bold and italics, respectively

Table 6 Comparison of the detection accuracy and speed of LINet under different feature extraction networks at the training setting (ii)

Backbone	CAMO-test				COD10K-test				NC4K				Speed
	S_α	E_φ	F_β^W	M	S_α	E_φ	F_β^W	M	S_α	E_φ	F_β^W	M	FPS
Resnet50	0.722	0.732	0.570	0.106	0.760	0.791	0.553	0.049	0.797	0.827	0.669	0.066	**124.7**
Resnet101	*0.720*	*0.726*	*0.556*	*0.110*	*0.753*	*0.774*	*0.521*	*0.053*	*0.793*	*0.813*	*0.646*	*0.071*	58.2
Resnet152	**0.737**	**0.759**	**0.591**	**0.103**	**0.761**	**0.792**	**0.550**	**0.050**	**0.802**	**0.831**	**0.672**	**0.066**	*44.5*

The best and worst results are highlighted in bold and italics, respectively

Table 7 Performance advantages of LINet

	Summary of design features	Performance advantages
LINet	1. Adopt the dense connection strategy [26] to fuse features at different levels, enhancing feature propagation and reducing the number of parameters	1. Compared to the detection speeds of mainstream algorithms, LINet can achieve a 187.62% increase in speed
	2. The RFM [27] module is employed to acquire rich hierarchical features, capturing more contextual information over a larger area while retaining the same number of parameters	2. For camouflaged objects with relatively regular patterns, the LINet model exhibits superior accuracy

Resnet101, LINet exhibited the worst performance in terms of accuracy. Therefore, in the field of COD, it may not be feasible to obtain high-precision models by increasing the depth of feature extraction network.

Discussion

Based on the human visual system by blending various feature layers and receptive field sizes, this study proposes a single-stage lightweight camouflage target detection model. Unlike previous methods, this proposal provides a novel way to improve detection speed alongside accuracy by introducing biological ideas into the camouflage target detection model, which is essential for real-time applications. The performance advantages of LINet are summarized in Table 7.

The findings contribute to enhancing object detection algorithms by considering both accuracy and real-time performance. Furthermore, the following analyses can be drawn from the experimental results:

(1) Qualitatively, for challenging tasks (such as fuzzy boundaries and occlusions), the accuracy of LINet decreases. For relatively regular camouflaged objects, LINet model has better accuracy. This situation may be due to the fact that the feature extraction module cannot pay dynamic attention to boundary information, resulting in the loss of local information of the target. This problem would be quantitatively addressed by dividing the dataset based on the complexity of the boundary. Simultaneously, we continue to enhance the LINet model's ability to extract target boundary information from a biological standpoint and improve its efficiency in detecting complex boundary targets.

(2) When LINet uses Resnet50 as the feature extraction network over Resnet101, its accuracy increases. We believe that this situation arises because ResNet-101 has more hierarchies than ResNet-50, but sometimes not all hierarchies are beneficial for the detection of camouflaged targets. Hence, it may not be feasible to improve the detection accuracy by increasing the depth of the feature extraction network in camouflage target detection. Therefore, a feature extraction network would be designed or by adding pre-processing operations [43] for camouflage target detection to improve the detection accuracy without increasing the depth.

(3) When the model was trained with setting (ii), the overall accuracy decline rate increased compared to the model trained with setting (i), suggesting that LINet may be more suitable for smaller data scenarios. As a recent advancement, camouflage target detection remains relatively limited, necessitating the collection of more scene data for the model generalization test.

Conclusion

Contrary to the existing two-stage detection methods that simulate animal predation, we proposed a simple and effective single-stage LINet detection framework by integrating features of various feature layers and receptive field sizes. Considering the time constraints of the COD algorithm, we discussed the influence of various feature extraction networks

on the accuracy and speed of the LINet model. Experimental results indicate that compared to the detection speeds of the mainstream algorithms, that of LINet can be increased by 187.62%, with the maximum reduction of 17.49% in detection accuracy. The novel LINet model has demonstrated significant improvements in the efficiency of camouflaged object detection at the real-time level. Furthermore, the proposed method can be applied to scenarios requiring a fast detection of camouflaged targets.

Acknowledgements We would like to thank Dr. Yang Li who provided insightful feedback throughout the research process. We express our gratitude to Dr. Zhide Zhang for his advice on the experimental scheme.

Author contributions QL: conceptualization, methodology, data curation, writing—original draft preparation, visualization, investigation, validation and supervision.ZW: conceptualization, methodology, and supervision. XZ: data curation, writing—original draft preparation, visualization, investigation, validation, and writing—reviewing and editing. HD: writing—reviewing and editing.

Funding This work was supported by the national level Frontier Artificial Intelligence Technology Research Project [approval number 672020109].

Data availability Our code can be accessed publicly on the following website: http://github.com/justin-gif/li.

Declarations

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

Consent to participate All the authors agreed to participate in this paper.

References

1. Singh SK, Dhawale CA, Misra S (2013) Survey of object detection methods in camouflaged image. IERI Procedia 4:351–357. https://doi.org/10.1016/j.ieri.2013.11.050
2. Le TN, Nguyen TV, Nie Z et al (2019) Anabranch network for camouflaged object segmentation. Comput Vis Image Underst 184:45–56. https://doi.org/10.1016/j.cviu.2019.04.006
3. Fan DP, Ji GP, Sun G, et al. (2020) Camouflaged object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 2774–2784. https://doi.org/10.1109/CVPR42600.2020.00285.
4. Fan DP, Ji GP, Zhou T, et al. (2020) Pranet: parallel reverse attention network for polyp segmentation. In: Medical image computing and computer-assisted intervention—MICCAI. Proceedings of the part VI: 23rd International Conference, Lima, Peru, October 4–8, 2020 23. Springer International Publishing. p 263–273.
5. la Pérez-de Fuente R, Delclòs X, Peñalver E et al (2012) Early evolution and ecology of camouflage in insects. Proc Natl Acad Sci U S A 109:21414–21419. https://doi.org/10.1073/pnas.1213775110
6. Fan DP, Lin Z, Zhang Z et al (2021) Rethinking RGB-D salient object detection: Models, data sets, and large-scale benchmarks. IEEE Trans Neural Netw Learn Syst 32:2075–2089. https://doi.org/10.1109/TNNLS.2020.2996406
7. Li G, Xie Y, Lin L, et al. (2017) Instance-level salient object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p 247–256. https://doi.org/10.1109/CVPR.2017.34.
8. Wang W, Lai Q, Fu H et al (2022) Salient object detection in the deep learning era: an in-depth survey. IEEE Trans Pattern Anal Mach Intell 44:3239–3259. https://doi.org/10.1109/TPAMI.2021.3051099
9. Zhao JX, Cao Y, Fan DP, et al. (2019) Contrast prior and fluid pyramid integration for RGBD salient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 3922–3931. https://doi.org/10.1109/CVPR.2019.00405.
10. Zhao JX, Liu JJ, Fan DP, et al. (2019) EGNet: Edge guidance network for salient object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. p 8778–8787. https://doi.org/10.1109/ICCV.2019.00887.
11. Kirillov A, He K, Girshick R, et al. (2019) Panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 9396–9405. https://doi.org/10.1109/CVPR.2019.00963.
12. Liu L, Ouyang W, Wang X et al (2020) Deep learning for generic object detection: a survey. Int J Comput Vis 128:261–318. https://doi.org/10.1007/s11263-019-01247-4
13. Medioni G (2009) Generic object recognition by inference of 3-d volumetric. Object Categorization 87:1
14. Sun Y, Chen G, Zhou T, et al. 2021. Context-aware cross-level fusion network for camouflaged object detection. arXiv preprint arXiv:2105.12555. https://doi.org/10.24963/ijcai.2021/142.
15. Yang F, Zhai Q, Li X, et al. (2021) Uncertainty-guided transformer reasoning for camouflaged object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. p 4126–4135. https://doi.org/10.1109/ICCV48922.2021.00411.
16. Zhai Q, Li X, Yang F, et al. (2021) Mutual graph learning for camouflaged object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 12992–13002. https://doi.org/10.1109/CVPR46437.2021.01280.
17. Li A, Zhang J, Lv Y, et al. (2021) Uncertainty-aware joint salient object and camouflaged object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 10066–10076. https://doi.org/10.1109/CVPR46437.2021.00994.
18. Lv Y, Zhang J, Dai Y, et al. (2021) Simultaneously localize, segment and rank the camouflaged objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 11586–11596. https://doi.org/10.1109/CVPR46437.2021.01142.
19. Mei H, Ji GP, Wei Z, et al. (2021) Camouflaged object segmentation with distraction mining. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8768–8777. https://doi.org/10.1109/CVPR46437.2021.00866.

20. Sun Y, Wang S, Chen C, et al. (2022) Boundary-guided camouflaged object detection. arXiv preprint arXiv:2207.00794. https://doi.org/10.24963/ijcai.2022/186.
21. He K, Zhang X, Ren S, et al. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p 770–778. https://doi.org/10.1109/CVPR.2016.90.
22. Fan DP, Gong C, Cao Y, et al. 2018. Enhanced-alignment measure for binary foreground map evaluation. arXiv preprint arXiv:1805.10421. https://doi.org/10.24963/ijcai.2018/97.
23. Fan DP, Cheng MM, Liu Y, et al. (2017) Structure-measure: a new way to evaluate foreground maps. In: Proceedings of the IEEE international conference on computer vision. p 4558–4567. https://doi.org/10.1109/ICCV.2017.487.
24. Margolin R, Zelnik-Manor L, and Tal A. (2014) How to evaluate foreground maps? In: Proceedings of the IEEE conference on computer vision and pattern recognition. p 248–255. https://doi.org/10.1109/CVPR.2014.39.
25. Perazzi F, Krähenbühl P, Pritch Y, et al. (2012) Saliency filters: Contrast based filtering for salient region detection. In: IEEE conference on computer vision and pattern recognition. p 733–740. https://doi.org/10.1109/CVPR.2012.6247743.
26. Huang G, Liu Z, Van Der Maaten L, et al. (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p 2261–2269. https://doi.org/10.1109/CVPR.2017.243.
27. Liu S, Huang D, and Wang Y. Receptive field block net for accurate and fast object detection, Computer Vision—ECCV 2018; 2018: 404–419. https://doi.org/10.1007/978-3-030-01252-6_24.
28. Skurowski P, Abdulameer H, Błaszczyk J, et al. (2018) Animal camouflage analysis: Chameleon database. Unpublished manuscript. 2: 7.
29. Kingma DP and Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980.
30. Wang L, Lu H, Wang Y, et al. (2017) Learning to detect salient objects with image-level supervision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p 3796–3805. https://doi.org/10.1109/CVPR.2017.404.
31. Everingham M, Van Gool L, Williams CKI et al (2010) The Pascal visual object classes (voc) challenge. Int J Comput Vis 88:303–338. https://doi.org/10.1007/s11263-009-0275-4
32. Yan J, Le TN, Nguyen KD et al (2021) Mirrornet: bio-inspired camouflaged object segmentation. IEEE Access 9:43290–43300. https://doi.org/10.1109/ACCESS.2021.3064443
33. Lv Y, Zhang J, Dai Y, et al. (2021) Simultaneously localize, segment and rank the camouflaged objects. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. p 11591–11601. https://doi.org/10.1109/CVPR46437.2021.01142.
34. Jia Q, Yao S, Liu Y, et al. (2022) Segment, magnify and reiterate: detecting camouflaged objects the hard way. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. p 4713–4722.https://doi.org/10.1109/CVPR52688.2022.00467
35. Pang Y, Zhao X, Xiang TZ, et al. (2022) Zoom in and out: a mixed-scale triplet network for camouflaged object detection. Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. p 2160–2170. https://doi.org/10.1109/CVPR52688.2022.00220
36. Bhajantri NU, Nagabhushan P (2006) Camouflage defect identification: a novel approach.9th International Conference on Information Technology (ICIT'06). IEEE. p 145–148.https://doi.org/10.1109/ICIT.2006.34
37. Feng X, Guoying C, Wei S (2013) Camouflage texture evaluation using saliency map. Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service. p 93–96. https://doi.org/10.1007/s00530-014-0368-y
38. Tankus A, Yeshurun Y (2001) Convexity-based visual camouflage breaking. Comput Vision Image Underst. 82(3):208–237. https://doi.org/10.1006/cviu.2001.0912
39. Xue F, Yong C, Xu S et al (2016) Camouflage performance analysis and evaluation framework based on features fusion. Multimed Tools Appl 75:4065–4082. https://doi.org/10.1007/s11042-015-2946-1
40. Li S, Florencio D, Zhao Y, et al. (2017) Foreground detection in camouflaged scenes. 2017 IEEE International Conference on Image Processing (ICIP). IEEE. p 4247–4251. https://doi.org/10.1109/ICIP.2017.8297083
41. Pike TW (2018) Quantifying camouflage and conspicuousness using visual salience. Methods Ecol Evol 9(8):1883–1895. https://doi.org/10.1111/2041-210X.13019
42. Zhao T, Wu X (2019) Pyramid feature attention network for saliency detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 3085–3094. https://doi.org/10.1109/CVPR.2019.00320
43. Aggarwal AK, Jaidka P (2022) Segmentation of crop images for crop yield prediction. Int J Biol Biomed 7:40–44