FINet: Frequency Injection Network for Lightweight Camouflaged Object Detection

Weiyun Liang, *Graduate Student Member, IEEE*, Jiesheng Wu, Yanfeng Wu, Xinyue Mu, and Jing Xu, *Member, IEEE*

***Abstract*—Existing camouflaged object detection (COD) methods typically have large model parameters and computations, hindering their deployment in real-world applications. Although using lightweight backbones can help alleviate this problem, their weaker feature representation often leads to performance degradation. To address this issue, we observe that frequency information has shown effective for cumbersome networks, but its effectiveness for lightweight ones has not been thoroughly investigated. Biological studies indicate that the human visual system utilizes distinct neural pathways to respond to different frequency stimuli, contributing to specialization and efficiency. Motivated by this, we propose an efficient frequency injection module (FIM) to aid lightweight backbone features by separately injecting detailed high frequency and object-level low frequency cues at each stage. FIM can be used as a plug-and-play component in existing COD networks to enhance backbone features at a low cost. With FIM, our proposed frequency injection network (FINet) achieves competitive performance against most state-of-the-art methods with much faster speed (692FPS for the input size of 384×384) and fewer parameters (3.74 M).**

***Index Terms*—Camouflaged object detection, lightweight model, frequency information, feature fusion.**

I. Introduction

CAMOUFLAGED object detection (COD) aims to detect objects that visually blend into their surroundings, which has a wide range of downstream applications [1]. Recently, with the advent of large-scale benchmarks [1], [2], existing works mainly adopt deep models for COD, whose methods can be roughly divided into three categories, i.e., designing elaborate feature exploration modules [1], [3], [4], incorporating auxiliary tasks (e.g., classification [5], salient object detection (SOD) [6], and ranking [2]), and exploiting prior knowledge (e.g., texture [7], frequency [8], [9], [10], [11], and edge [12], [13], [14]).

While these methods have made remarkable achievements, their high performance often comes from large model parameters and computations, which hinders their deployment in real-world applications, especially on mobile devices. Unfortunately, there is a limited number of works that focus on lightweight COD frameworks, resulting in an urgent need to design efficient models for COD. In the well-established realm of SOD, one straightforward approach is to replace cumbersome backbones with lightweight ones, which often have weaker feature representation capability [15]. Current SOD works typically design elaborate multi-level, multi-scale, and attention methods [16], [17], [18], [19], [20] to enhance lightweight backbone features. However, since COD presents more significant challenges than SOD due to the intrinsic similarity between foreground and background, as well as ambiguous edges, discriminative camouflaged traces are difficult to capture [8]. Hence, solely exploring RGB domain features may require more complex designs and introduce more parameters to enhance the learning capability for discriminative features [1], [3], [4], [21], which may contradict the original intention of lightweight design.

Fortunately, we note that the frequency domain information contains invisible discriminative camouflaged clues, which has shown effective for cumbersome networks [22]. However, its effectiveness for lightweight ones has not been thoroughly investigated. The key issue lies in how to carefully aid lightweight backbone features with discriminative frequency cues for COD, while maintaining a lightweight design.

To address this issue, we observe the biological studies [23] indicate that the human vision system utilizes distinct neural pathways to respond to different frequency stimuli, enabling more effective handling of specific information types and ensuring efficiency. Motivated by this, we consider that processing high and low frequency bands separately can disentangle the complexity of frequency information and facilitate a more refined treatment of specific frequencies, thereby effectively exploring subtle discriminative cues. Specifically, we propose an efficient frequency injection module (FIM) for lightweight COD, which separately injects spatial detailed high frequency cues and object-level low frequency cues into RGB features to mine complementary camouflaged traces in two domains and strengthen lightweight backbone features. As illustrated in Fig. 1, with FIM, our proposed frequency injection network (FINet) only has 3.74 M parameters and 1.25 G FLOPs, and achieves 692FPS for the input size of 384 × 384 on a single NVIDIA RTX 3090 GPU, while matching the performance of most state-of-the-art (SOTA) methods.

Our main contributions can be summarized as follows:

Manuscript received 1 November 2023; revised 10 January 2024; accepted 13 January 2024. Date of publication 19 January 2024; date of current version 8 February 2024. This work was supported by the Natural Science Foundation of Tianjin City under Grant 21JCYBJC00110. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jeremy Thomas Reed. *(Weiyun Liang and Jiesheng Wu contributed equally to this work.) (Corresponding author: Jing Xu.)*

Weiyun Liang, Jiesheng Wu, Xinyue Mu, and Jing Xu are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China (e-mail: weiyunliang@mail.nankai.edu.cn; jasonwu@mail.nankai.edu.cn; muxinyue@mail.nankai.edu.cn; xujing@nankai.edu.cn).

Yanfeng Wu is with the Department of Artificial Intelligence, Zhejiang Lab, Hangzhou 311121, China (e-mail: wuyanfeng@zhejianglab.com).

Source codes will be released at https://github.com/crrcoo/FINet.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LSP.2024.3356416, provided by the authors.

Digital Object Identifier 10.1109/LSP.2024.3356416

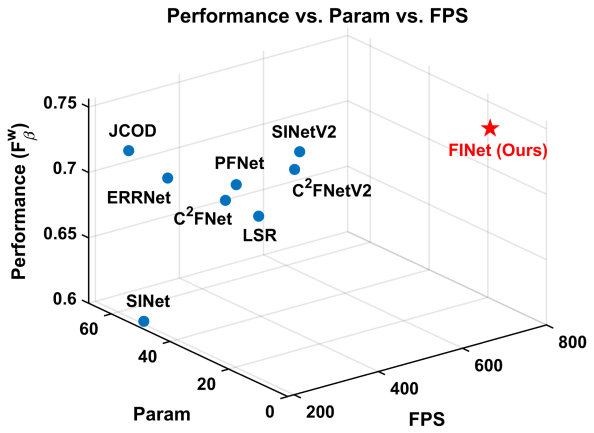

Fig. 1. Performance, parameters, and inference speed compared with state-of-the-art methods. The performance is measured using the weighted F-measure (F_β^w) on the CAMO dataset [5]. Our FINet achieves extremely high speed and superior performance with minimal parameters.

- We propose an efficient frequency injection network (FINet), which achieves 692FPS with only 3.74 M parameters, for lightweight COD. To the best of our knowledge, we are the first to investigate lightweight COD from a frequency perspective.
- A frequency injection module (FIM) is proposed to separately inject spatial detailed and object-level frequency cues into backbone features to mine camouflaged traces. FIM can be used as a plug-and-play component in current models to enhance the feature representation of backbone features with frequency cues at a low cost.
- Extensive experiments demonstrate that the proposed FINet achieves competitive performance against most SOTA methods on four benchmark datasets.

II. Methodology

A. Overview

The architecture of our proposed FINet is illustrated in Fig. 2. Specifically, given an input RGB image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, the backbone network extracts hierarchical features $\{\mathbf{F}_i\}_{i=1}^4 \in \mathbb{R}^{H_i\times W_i\times C_i}$, where H, W, C, and i represent height, width, channel, and the i-th stage, respectively. Meanwhile, following [24], we obtain the frequency domain feature $\mathbf{F}^{freq} \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 192}$ from $\mathbf{I}$, and partition it into high and low frequency bands, denoted as $\mathbf{F}^h$ and $\mathbf{F}^l \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 96}$, respectively. Then, FIM separately injects the high and low frequency features into $\mathbf{F}_i$ at each stage. Finally, the asymmetric decoder blocks (ADBs) gradually fuse the multi-stage features in a top-down manner and generate the final prediction.

B. Frequency Injection Module

As shown in Fig. 2, FIM consists of two modules, namely high frequency injection module (HFIM) and low frequency injection module (LFIM), designed to separately process high and low frequency bands, respectively. Unlike [8], which learns a global frequency prior for all stages, FIMs are used at each stage, enabling the adaptive learning of essential frequency bands for features at each stage to suit their distinct semantics. Due to the input frequency features are manually extracted [24], they unavoidably contain redundant information. Hence, both HFIM and LFIM adopt a compress and recover scheme to filter out redundant information and recalibrate prominent features. Inspired by [25], HFIM and LFIM both adopt a local and a global branches to enhance features from both local and global scales. Specifically, the FIM can be formulated as

$$\begin{aligned}
\dot{\mathbf{F}}_i &= \delta(\mathcal{B}(f_{1\times1}(\mathbf{F}_i)),\\
\mathbf{F}_i^H &= \mathcal{B}(f_{1\times1}(\mathrm{Cat}[\dot{\mathbf{F}}_i;\delta(\mathcal{B}(f_{1\times1}(\mathrm{Up}(\mathbf{F}^h))))])),\\
\mathbf{F}_i^L &= \mathcal{B}(f_{1\times1}(\mathrm{Cat}[\dot{\mathbf{F}}_i;\delta(\mathcal{B}(f_{1\times1}(\mathrm{Up}(\mathbf{F}^l))))])),\\
\ddot{\mathbf{F}}_i &= \mathcal{B}(f_{1\times1}(\delta(\mathrm{HFIM}(\mathbf{F}_i^H)\oplus\mathrm{LFIM}(\mathbf{F}_i^L)))),
\end{aligned} \tag{1}$$

where $f_{1\times1}$, $\mathcal{B}$, and δ denote the 1×1 convolutional layer, batch normalization layer, and GELU activation, respectively. $\mathrm{Cat}[\cdot;\cdot]$, $\mathrm{Up}(\cdot)$, and $\oplus$ indicate concatenation, bilinear interpolation, and element-wise addition, respectively. $\ddot{\mathbf{F}}_i \in \mathbb{R}^{H_i\times W_i\times C_i'}$ is the output of the FIM. Note that the input RGB and frequency features $\mathbf{F}_i$, $\mathbf{F}^h$, and $\mathbf{F}^l$ are first processed by three convolutions to adjust the feature channel to C_i', respectively. $\mathbf{F}_i^H$ and $\mathbf{F}_i^L$ are inputs of HFIM and LFIM.

1) High Frequency Injection Module: Since high frequency features contain more spatial details (e.g., texture differences and contours of camouflaged objects), HFIM compresses and recovers features along spatial dimension to filter out background interference and obtain subtle camouflaged traces. For the local branch, a depthwise separable convolution (DSConv) with a stride of r is adopted to reduce the spatial dimension, and a bilinear interpolation is used to restore the feature. While for the global branch, we adopt a global spatial attention mechanism to distribute more global information and generate the global attention map. The HFIM can be defined as

$$\begin{aligned}
\mathbf{M}_i^{local} &= \mathcal{B}(\hat{f}_{3\times3}(\mathrm{Up}(\delta(\mathcal{B}(\hat{f}_{3\times3}^r(\mathbf{F}_i^H)))))),\\
\mathbf{M}_i^{global} &= \mathbf{F}_i^H\otimes\sigma(\hat{f}_{3\times3}(\mathrm{Up}(\delta(\mathcal{B}(\hat{f}_{3\times3}^r(\tau(\mathbf{F}_i^H))))))),\\
\dot{\mathbf{F}}_i^H &= \mathcal{B}(f_{1\times1}(\mathbf{M}_i^{local}\oplus\mathbf{M}_i^{global})),
\end{aligned} \tag{2}$$

where τ and σ denote averaging the channels of the input feature and the sigmoid function, respectively. $\hat{f}_{k_1\times k_2}^r$ indicates the $k_1\times k_2$ DSConv with a stride of r, and r is a reduction ratio. The superscript r will be omitted without ambiguity if we have a stride of 1. For example, $\hat{f}_{3\times3}$ is the 3×3 DSConv with a stride of 1. $\mathbf{M}_i^{local}$, $\mathbf{M}_i^{global}$, and $\dot{\mathbf{F}}_i^H$ denote outputs of local branch, global branch, and HFIM, respectively.

2) Low Frequency Injection Module: Since low frequency features contain more object-level artifacts (e.g., object regions where color and texture change gently), LFIM focuses more on channel dimension to integrate various types of features and generate comprehensive object regions. Similar to HFIM, the LFIM adopts 1×1 convolutions to compress and recover the feature channel with a ratio of r. For the global branch, the global average pooling (GAP) is used to capture the global scale information. The LFIM is defined as

$$\begin{aligned}
\mathbf{N}_i^{local} &= \mathcal{B}(f_{1\times1}(\delta(\mathcal{B}(f_{1\times1}(\mathbf{F}_i^L))))),\\
\mathbf{N}_i^{global} &= \mathbf{F}_i^L\otimes\sigma(f_{1\times1}(\delta(\mathcal{B}(f_{1\times1}(\mathrm{GAP}(\mathbf{F}_i^L)))))),\\
\dot{\mathbf{F}}_i^L &= \mathcal{B}(f_{1\times1}(\mathbf{N}_i^{local}\oplus\mathbf{N}_i^{global})),
\end{aligned} \tag{3}$$

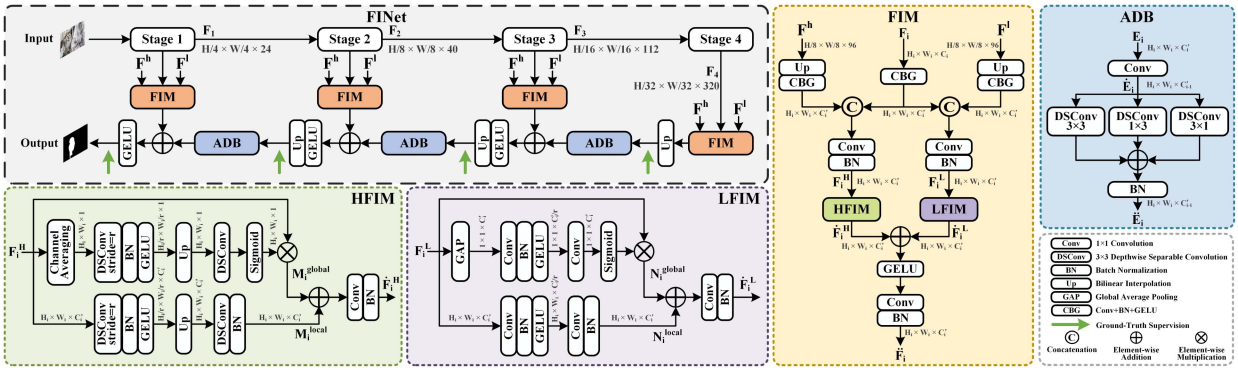

Fig. 2. Overall architecture of the proposed FINet, which comprises two components: frequency injection module (FIM) and asymmetric decoder block (ADB). The FIM mainly consists of two modules: high frequency injection module (HFIM) and low frequency injection module (LFIM). The high and low frequency features, denoted as $\mathbf{F}^h$ and $\mathbf{F}^l$, are extracted from the input image following [24].

where $\mathbf{N}_i^{local}$, $\mathbf{N}_i^{global}$, and $\dot{\mathbf{F}}_i^L$ denote outputs of local branch, global branch, and LFIM, respectively.

C. Asymmetric Decoder Block

According to [26], complementing square convolution kernels with asymmetric ones can strengthen the feature representation by exploring orthogonal characteristics. Motivated by this, we design a lightweight asymmetric decoder block (ADB) to improve the performance of the 3×3 DSConv with asymmetric DSConvs by adding few parameters, as illustrated in Fig. 2. Given the input feature $\mathbf{E}_i$, ADB is formulated as

$$\begin{aligned} \dot{\mathbf{E}}_i &= f_{1\times1}(\mathbf{E}_i), \\ \ddot{\mathbf{E}}_i &= \mathcal{B}(\hat{f}_{3\times3}(\dot{\mathbf{E}}_i) \oplus \hat{f}_{1\times3}(\dot{\mathbf{E}}_i) \oplus \hat{f}_{3\times1}(\dot{\mathbf{E}}_i)), \end{aligned} \tag{4}$$

where $\ddot{\mathbf{E}}_i$ is the output of the ADB. The 1×1 convolution adjusts the feature channel consistent with the next stage.

D. Loss Function

Following [1], [3], [4], we combine the weighted binary cross-entropy loss (L_{BCE}^w) and the weighted intersection over union loss (L_{IoU}^w). We supervise the final prediction of our model ($\mathbf{P}_1$) and the coarse prediction maps ($\{\mathbf{P}_i\}_{i=2}^4$). The loss function is defined as

$$L = \sum_{i=1}^{4} L_{BCE}^w(\mathbf{P}_i, \mathbf{G}) + \sum_{i=1}^{4} L_{IoU}^w(\mathbf{P}_i, \mathbf{G}), \tag{5}$$

where $\mathbf{G}$ is the ground-truth.

III. Experiments

Due to page limitations, additional results are provided in the supplemental file.

A. Experimental Settings

1) Datasets: Following [1], [3], we evaluate our model on four benchmark datasets, i.e., CHAMELEON [27], CAMO [5], COD10K [1], and NC4K [2].

2) Metrics: Following [4], [28], [29], we adopt four metrics to evaluate the performance, i.e., structure-measure (S_α), adaptive E-measure (E_ϕ^{ad}), weighted F-measure (F_β^w), and mean absolute error (M). The model parameters, FLOPs, and FPS are calculated following [3] and [30].

3) Training and Implementation Details: The pre-trained EfficientNet-B0 [31] backbone is used as the encoder. During the training phase, the input images are resized to 384×384 and augmented by random flipping, copping, rotation, and color enhancement. We utilize the Adam optimizer with a weight decay of 4e-8. The initial learning rate is 2.6e-4 and follows a cosine decay strategy. Our model is implemented with PyTorch framework and trained for 200 epochs on a single NVIDIA RTX 3090 GPU. The batch size is set to 32.

4) Comparing State-of-The-Art Methods: We compare our proposed method with 17 SOTA methods, i.e., SINet [1], PFNet [28], R-MGL [12], LSR [2], JCOD [6], UGTR [32], C^{2}FNet [21], DTC-Net [7], ERRNet [13], SINetV2 [3], C^{2}FNetV2 [4], ZoomNet [33], R-MGL_V2 [34], FEDER [35], FSPNet [36], SAM [37], and TinyCOD [29]. Note that TinyCOD is the first tiny model for COD [29].

B. Comparison With State-of-The-Art Methods

1) Quantitative Comparison: As shown in Table I, our FINet only has about 15% parameters and 10% FLOPs of SINetV2 [3], and achieves about 1.8 times faster. Our FINet outperforms SINetV2 [3] by 0.2%, 1.9%, 0.6%, and 0.3% in terms of S_α, E_ϕ^{ad}, F_β^w, and M on COD10 K, respectively. Compared with the lightweight model TinyCOD [29], FINet has about 1 M fewer parameters and 0.15 G fewer FLOPs, and outperforms it on both CAMO, COD10 K, and NC4K. The reason might be explained that TinyCOD [29] just reinforces RGB features, which needs more parameters to improve the model learning capability. While our FINet surpasses 13 SOTA methods, it still falls behind some of the recent complex models, e.g., FEDER [35] and FSPNet [36]. This suggests the ongoing need for research in lightweight COD methods.

2) Qualitative Comparison: The qualitative comparison results are shown in Fig. 3. For these challenging scenes, our model obtains superior results, while other methods often incompletely predict objects (e.g., rows 1 and 4), distract by background objects (e.g., rows 2 and 3), or struggle to fully detect multiple objects (e.g., row 5).

TABLE I
COMPARISON WITH SOTA METHODS

Method	Publication	Param	FLOPs	FPS	CHAMELEON (76)				CAMO (250)				COD10K (2,026)				NC4K (4,121)			
					$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$
SINet [1]	CVPR'20	48.9M	19.5G	184	0.869	0.899	0.740	0.044	0.751	0.834	0.606	0.100	0.776	0.867	0.631	0.043	0.808	0.883	0.723	0.058
PFNet [28]	CVPR'21	46.5M	26.6G	227	0.882	0.942	0.810	0.033	0.782	0.855	0.695	0.085	0.800	0.868	0.660	0.040	0.829	0.894	0.745	0.053
R-MGL [12]	CVPR'21	73.7M	378.6G	-	0.893	0.923	0.812	0.031	0.776	0.848	0.673	0.088	0.814	0.864	0.666	0.035	0.833	0.884	0.741	0.052
LSR [2]	CVPR'21	28.7M	17.5G	316	0.890	0.936	0.822	0.030	0.787	0.859	0.696	0.080	0.804	0.883	0.673	0.037	0.840	0.904	0.766	0.048
JCOD [6]	CVPR'21	55.9M	30.3G	197	0.891	0.943	0.833	0.030	0.800	0.872	0.728	0.073	0.809	0.882	0.684	0.035	0.842	0.906	0.771	0.047
UGTR [32]	ICCV'21	37.6M	121.7G	-	0.888	0.920	0.796	0.031	0.785	0.861	0.686	0.086	0.818	0.850	0.667	0.035	0.839	0.890	0.747	0.052
C^2FNet [21]	IJCAI'21	26.4M	13.2G	221	0.888	0.932	0.828	0.032	0.796	0.865	0.719	0.080	0.813	0.886	0.686	0.036	0.838	0.901	0.762	0.049
DTC-Net [7]	TMM'22	53.5M	-	-	0.876	-	0.773	0.039	0.778	-	0.667	0.084	0.790	-	0.616	0.041	-	-	-	-
ERRNet [13]	PR'22	67.7M	20.1G	372	0.917	0.875	0.787	0.039	0.779	0.855	0.679	0.085	0.786	0.845	0.630	0.043	0.827	0.892	0.737	0.054
SINetV2 [3]	TPAMI'22	24.9M	12.3G	387	0.888	0.930	0.816	0.030	0.820	0.884	0.743	0.070	0.815	0.864	0.680	0.037	0.847	0.901	0.770	0.048
C^2FNetV2 [4]	TCSVT'22	25.2M	18.2G	377	0.893	0.947	0.845	0.028	0.799	0.869	0.730	0.077	0.811	0.890	0.691	0.036	0.840	0.903	0.772	0.048
ZoomNet [33]	CVPR'22	32.4M	102.1G	-	0.902	0.952	0.845	0.023	0.820	0.883	0.752	0.066	0.838	0.893	0.729	0.029	0.853	0.907	0.784	0.043
R-MGL_V2 [34]	TIP'23	-	-	-	0.892	-	0.825	0.029	0.774	-	0.684	0.085	0.813	-	0.682	0.034	0.831	-	0.751	0.051
FEDER [35]	CVPR'23	42.1M	36.0G	-	0.887	0.943	0.834	0.030	0.802	0.877	0.738	0.071	0.822	0.901	0.716	0.032	0.847	0.913	0.789	0.044
FSPNet [36]	CVPR'23	273.8M	283.3G	-	0.908	0.945	0.851	0.023	0.856	0.919	0.799	0.050	0.851	0.900	0.735	0.026	0.879	0.923	0.816	0.035
SAM [37]	ICCV'23	635.6M	2736.6G	-	0.727	0.736	0.639	0.081	0.684	0.689	0.606	0.132	0.783	0.800	0.701	0.049	0.767	0.778	0.696	0.078
TinyCOD [29]	ICASSP'23	4.72M	1.40G	-	0.887	0.931	0.814	0.030	0.822	0.890	0.752	0.066	0.811	0.877	0.678	0.036	0.843	0.903	0.766	0.047
FINet (Ours)	-	**3.74M**	**1.25G**	**692**	**0.883**	**0.929**	**0.808**	**0.031**	**0.828**	**0.890**	**0.752**	**0.065**	**0.817**	**0.883**	**0.686**	**0.034**	**0.847**	**0.904**	**0.771**	**0.047**

- Denotes the results are unavailable. The results of sam [37] are obtained from [38].

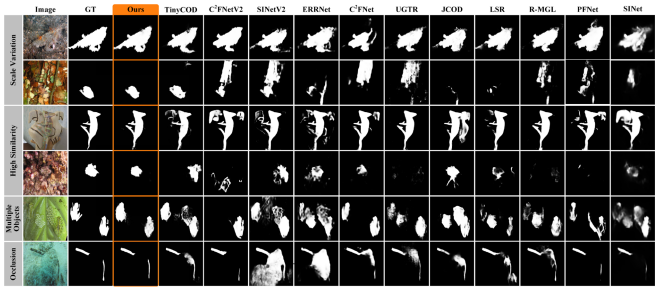

Fig. 3. Qualitative comparison with SOTA methods.

TABLE II
ABLATION STUDIES OF THE MAIN COMPONENTS

Model	FIM		ADB	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	HFIM	LFIM		$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$
(a)			✓	0.822	0.880	0.740	0.070	0.808	0.870	0.669	0.037	0.842	0.899	0.761	0.049
(b)		✓	✓	0.820	0.885	0.744	0.069	0.812	0.879	0.679	0.036	0.843	0.903	0.767	0.048
(c)	✓		✓	0.826	0.887	0.750	0.067	0.812	0.873	0.677	0.036	0.846	0.903	0.767	0.047
(d)	✓	✓		0.826	0.887	0.752	0.068	0.812	0.875	0.675	0.036	0.845	0.902	0.766	0.048
(e)	✓	✓	✓	**0.828**	**0.890**	**0.752**	**0.065**	**0.817**	**0.882**	**0.686**	**0.034**	**0.847**	**0.904**	**0.771**	**0.047**

C. Ablation Study

The results of ablation studies are listed in Table II. Model (e) is our final model, i.e., FINet.

1) Significance of the FIM: To evaluate the significance of FIM, HFIM, and LFIM, we remove each of them and get model (a), (b), and (c), respectively. The results show that removing either HFIM or LFIM causes a performance degradation, and removing both of them leads to the most significant performance drop, demonstrating the effectiveness of FIM, HFIM, and LFIM. It can be observed that removing HFIM causes S_α to decrease by 0.8% and 0.4% on CAMO and NC4K, respectively, while the removal of the LFIM only reduces S_α by 0.2% and 0.1%. This suggests that both high and low frequency features contribute to COD, with high frequency information being more effective.

2) Effectiveness of the ADB: We remove ADBs in our FINet and get model (d). Comparing the results of model (d) and (e), the ADB improves E_ϕ^{ad} by 0.3%, 0.7%, and 0.2% on CAMO, COD10 K, and NC4K, respectively. This indicates the effectiveness of the ADB.

D. Feature Visualizations of the FIM

As shown in Fig. 4, we utilize Grad-CAM [42] to visualize the features in the FIM. Seeing the red boxes in Fig. 4, the HFIM concentrates on regions with distinct textures or color compared to the surroundings, such as the vertical patterns of the moth and the fin of the flounder. The LFIM focuses more on regions inside the object with relatively consistent patterns. The FIM combines the attention areas of both HFIM and LFIM, and focuses on the entire object. The above analysis demonstrates the effectiveness of the FIM.

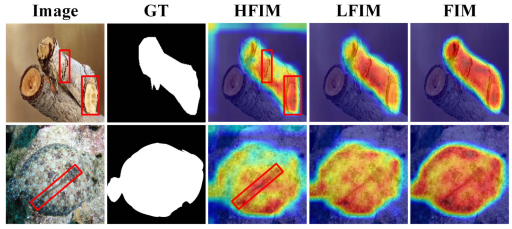

Fig. 4. Feature Visualizations of the FIM.

TABLE III
UNIVERSALITY OF THE FIM

Method	Backbone	Param	FLOPs	COD10K (2,026)				NC4K (4,121)			
				$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{ad}\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$
SINet [1]	ResNet-50 [39]	48.9M	19.5G	0.776	**0.867**	0.631	0.043	0.808	0.883	0.723	0.058
SINet-FIM	ResNet-50 [39]	49.1M	19.7G	**0.797**	0.859	**0.645**	**0.041**	**0.834**	**0.893**	**0.745**	**0.052**
SINetV2 [3]	Res2Net-50 [40]	24.9M	12.3G	0.815	0.864	**0.680**	0.037	0.847	0.901	0.770	0.048
SINetV2-FIM	Res2Net-50 [40]	25.0M	12.4G	**0.815**	**0.865**	0.679	**0.036**	**0.849**	**0.901**	**0.772**	**0.047**
TinyCOD [29]	TinyNet-A [41]	4.72M	1.40G	0.811	0.877	0.678	0.036	0.843	0.903	0.766	0.047
FINet (Ours)	TinyNet-A [41]	4.59M	1.38G	**0.817**	**0.886**	**0.690**	**0.034**	**0.847**	**0.907**	**0.775**	**0.045**

E. Universality of the FIM

To evaluate the universality of our FIM, we conduct two experiments and the results are listed in Table III. First, we add FIMs after backbones of SINet [1] and SINetV2 [3], and obtain SINet-FIM and SINetV2-FIM, respectively. Comparing their results, adding FIM boosts the performance at a very low cost, e.g., SINet-FIM improves F_β^w by 1.4% and 2.2% on COD10 K and NC4K respectively, and only increases 0.4% parameters of SINet [1]. This demonstrates that our FIM can be efficiently used as a plug-and-play component in other models. Second, we replace our backbone with the TinyNet-A [41] used by TinyCOD [29]. It can be observed that our model has fewer parameters and FLOPs while outperforms TinyCOD [29] by a large margin. The above observations indicate the generalization potential of FIM for other backbones.

IV. CONCLUSION

In this letter, we propose an efficient frequency injection network, namely FINet, for lightweight COD. FIM separately injects detailed and object-level frequency cues into RGB features to strengthen lightweight backbone features. Extensive experiments demonstrate that our FINet achieves competitive performance against most SOTA methods, with lower model complexity and much faster speed.

REFERENCES

[1] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2774–2784.

[2] Y. Lv et al., "Simultaneously localize, segment and rank the camouflaged objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11586–11596.

[3] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6024–6042, Oct. 2022.

[4] G. Chen, S.-J. Liu, Y.-J. Sun, G.-P. Ji, Y.-F. Wu, and T. Zhou, "Camouflaged object detection via context-aware cross-level fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6981–6993, Oct. 2022.

[5] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabranch network for camouflaged object segmentation," *Comput. Vis. Image Understanding*, vol. 184, pp. 45–56, 2019.

[6] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10066–10076.

[7] W. Zhai, Y. Cao, H. Xie, and Z.-J. Zha, "Deep texton-coherence network for camouflaged object detection," *IEEE Trans. Multimedia*, vol. 25, pp. 5155–5165, 2023.

[8] Y. Zhong, B. Li, L. Tang, S. Kuang, S. Wu, and S. Ding, "Detecting camouflaged object in frequency domain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4494–4503.

[9] J. Lin, X. Tan, K. Xu, L. Ma, and R. W. Lau, "Frequency-aware camouflaged object detection," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 19, no. 2, pp. 1–16, 2022.

[10] R. Cong, M. Sun, S. Zhang, X. Zhou, W. Zhang, and Y. Zhao, "Frequency perception network for camouflaged object detection," in *Proc. ACM Int. Conf. Multimedia*, 2023, pp. 1179–1189.

[11] C. Xie, C. Xia, T. Yu, and J. Li, "Frequency representation integration for camouflaged object detection," in *Proc. ACM Int. Conf. Multimedia*, 2023, pp. 1789–1797.

[12] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, and D.-P. Fan, "Mutual graph learning for camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12992–13002.

[13] G.-P. Ji, L. Zhu, M. Zhuge, and K. Fu, "Fast camouflaged object detection via edge-based reversible re-calibration network," *Pattern Recognit.*, vol. 123, 2022, Art. no. 108414.

[14] J. Wu, W. Liang, F. Hao, and J. Xu, "Mask-and-edge co-guided separable network for camouflaged object detection," *IEEE Signal Process. Lett.*, vol. 30, pp. 748–752, 2023.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[16] Y. Liu, X.-Y. Zhang, J.-W. Bian, L. Zhang, and M.-M. Cheng, "SAM-Net: Stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3804–3814, 2021.

[17] N. Huang, Q. Jiao, Q. Zhang, and J. Han, "Middle-level feature fusion for lightweight RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 6621–6634, 2022.

[18] L. Tang, B. Li, Y. Wu, B. Xiao, and S. Ding, "Fast: Feature aggregation for detecting salient object in real-time," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 1525–1529.

[19] S. Kuang, S. Meng, B. Xiao, L. Tang, and B. Li, "Rethinking two-B-real net for real-time salient object detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 2005–2009.

[20] B. Li, Z. Sun, L. Tang, and A. Hu, "Two-b-real net: Two-branch network for real-time salient object detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 1662–1666.

[21] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, "Context-aware cross-level fusion network for camouflaged object detection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1025–1031.

[22] P. Li, X. Yan, H. Zhu, M. Wei, X.-P. Zhang, and J. Qin, "FindNet: Can you find me? Boundary-and-texture enhancement network for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 6396–6411, 2022.

[23] M. Shim et al., "Influence of spatial frequency and emotion expression on face processing in patients with panic disorder," *J. Affect. Disord.*, vol. 197, pp. 159–166, 2016.

[24] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the frequency domain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1740–1749.

[25] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," 2015, *arXiv:1506.04579*.

[26] X. Ding, Y. Guo, G. Ding, and J. Han, "ACNet: Strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1911–1920.

[27] P. Skurowski, H. Abdulameer, J. Błaszczyk, T. Depta, A. Kornacki, and P. Kozieł, "Animal camouflage analysis: Chameleon database," *Unpublished manuscript*, vol. 2, no. 6, 2018, Art. no. 7.

[28] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, "Camouflaged object segmentation with distraction mining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8768–8777.

[29] H. Xing, S. Gao, H. Tang, T. Q. Mok, Y. Kang, and W. Zhang, "TINYCOD: Tiny and effective model for camouflaged object detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[30] Y.-H. Wu, Y. Liu, J. Xu, J.-W. Bian, Y.-C. Gu, and M.-M. Cheng, "MobileSal: Extremely efficient RGB-D salient object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10261–10269, Dec. 2022.

[31] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[32] F. Yang et al., "Uncertainty-guided transformer reasoning for camouflaged object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4126–4135.

[33] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, "Zoom in and out: A mixed-scale triplet network for camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2150–2160.

[34] Q. Zhai et al., "MGL: Mutual graph learning for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 1897–1910, 2023.

[35] C. He et al., "Camouflaged object detection with feature decomposition and edge reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22046–220 55.

[36] Z. Huang et al., "Feature shrinkage pyramid for camouflaged object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 5557–5566.

[37] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.

[38] L. Tang, H. Xiao, and B. Li, "Can SAM segment anything? When SAM meets camouflaged object detection,"2023, *arXiv:2304.04709*.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[40] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[41] K. Han, Y. Wang, Q. Zhang, W. Zhang, C. Xu, and T. Zhang, "Model Rubik's cube: Twisting resolution, depth and width for tinynets," in*Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 19353–19364.

[42] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.