TRACER: Extreme Attention Guided Salient Object Tracing Network (Student Abstract)

Min Seok Lee[*], **WooSeok Shin**[*], **Sung Won Han**[†]
School of Industrial and Management Engineering, Korea University, Seoul, Korea
{karel, wsshin95, swhan}@korea.ac.kr

Abstract

Existing studies on salient object detection (SOD) focus on extracting distinct objects with edge features and aggregating multi-level features to improve SOD performance. However, both performance gain and computational efficiency cannot be achieved, which has motivated us to study the inefficiencies in existing encoder-decoder structures to avoid this trade-off. We propose TRACER which excludes multi-decoder structures and minimizes the learning parameters usage by employing attention guided tracing modules (ATMs).

Introduction

The performance of salient object detection (SOD) has improved by two approaches: improvement on the edge representation and discrepancy reduction during multi-level aggregation. These existing approaches improved SOD performance; however, they are incapable of simultaneously achieving the performance and computational efficiency. Therefore, to improve both performance and computational efficiency, this study focuses on reducing inefficiencies, which can develop in existing encoder-decoder structures, and applying adaptive pixel-wise weights to conventional loss functions.

Based on the existing encoder-decoder structures, we observed three inefficiencies. First, previous studies commonly employ deeper encoder representations for generating edge information; however, the representations require a large memory and cannot leverage refined edges in the encoder structure. Second, in multi-level aggregation, the existing methods do not consider the relative significance in each level. Finally, existing multi-decoder structures reduce multi-level distribution discrepancy, although, these structures cannot guarantee computational and memory efficiency.

In the process of applying adaptive pixel-wise weights to the loss function, conventional binary cross entropy (BCE) and IoU loss functions independently treat each pixel. However, the pixels adjacent to fine or explicit edges should be focused more than the pixels in the background or center of the salient object. Consequently, it is necessary to employ adaptive pixel-wise weights to delineate fine or explicit edge regions while excluding redundant areas.

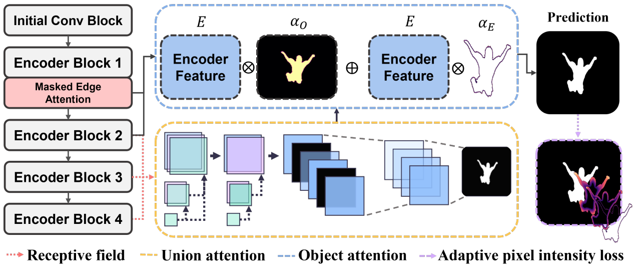

Figure 1: Overview of TRACER framework.

TRACER

Masked edge attention: The masked edge attention module enhances the edge features in low-level representations using a fast Fourier transform (FFT) and propagates the edge-refined representation to the second encoder block. Existing methods cannot leverage the explicit edges in the feature extraction phases because they require the outputs of deeper encoders to obtain the distinct edges. Using FFT, we compute the edge-refined representation X_E as follows: $X_E = X + \mathcal{RFB}(FFT^{-1}(f_r^H(FFT(E_1))))$. Here, E_1 denotes the first encoder block output and $f_r^H(\cdot)$ is a high-pass filter, which eliminates all frequencies except those in radius r. Because $FFT^{-1}(f_r^H(FFT(E_1)))$ contains the background noise when it is transformed from the frequency domain to the spatial domain, we eliminate noise using the receptive field operation $\mathcal{RFB}(\cdot)$ and generate explicit edge.
Union attention: A union attention module is designed to aggregate multi-level features and detect the more important context from both channel and spatial representations. Each encoder output $E_{i\in\{2,3,4\}}$ is aggregated to $X \in \mathbb{R}^{(32+64+128)\times H_2\times W_2}$. We discriminate the relatively significant channel-wise context and emphasize the spatial information based on complementary confidence scores obtained from the channel context.

$$\alpha_c = \sigma(softmax(\mathcal{F}_q(\widetilde{X})\mathcal{F}_k(\widetilde{X})^\top)\mathcal{F}_v(\widetilde{X})) \qquad (1)$$

In Eq. (1), $\tilde{X} \in \mathbb{R}^{C\times1\times1}$ is the channel-wise pooled representation, and $\mathcal{F}(\cdot)$ denotes the convolution operation using 1×1 kernel size. Context information is obtained by using the self-attention method and the softmax function to dis-

[*] These authors contributed equally.
[†] Corresponding author.

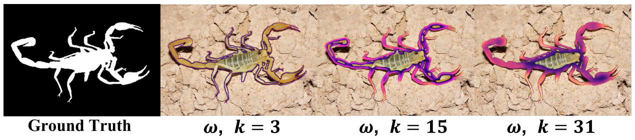

Figure 2: Pixel intensity ω visualization corresponding to the kernel size K.

criminate significant channels $\alpha_c \in \mathbb{R}^{C\times 1\times 1}$ with a sigmoid function. To refine the aggregated representation X, we apply confidence channel weight as follows: $X_c = (X \otimes \alpha_c) + X$. Subsequently, we retain confidence channels based on the distribution of α_c and the confidence ratio γ as follows:

$$\widetilde{X}_c = X_c \otimes mask \begin{cases} mask = 1, & \text{if } \alpha_c > F^{-1}(\gamma) \\ mask = 0, & \text{otherwise} \end{cases} \qquad (2)$$

Here, $F^{-1}(\gamma)$ denotes γ quantile of α_c. We exclude an area of γ in the lower tail of the distribution α_c. Then, the refined input $\widetilde{X}_c$ is computed spatially to discriminate the salient object and generate the first decoder representation $D_0 \in \mathbb{R}^{1\times H_2\times W_2}$, as shown in Eq. (3).

$$D_0 = softmax(\mathcal{G}_q(\widetilde{X}_c)\mathcal{G}_k(\widetilde{X}_c)^\top)\mathcal{G}_v(\widetilde{X}_c) + \mathcal{G}_v(\widetilde{X}_c) \quad (3)$$

Here, $\mathcal{G}(\cdot)$ projects the input features to $\widetilde{X}_c \in \mathbb{R}^{1\times H_2\times W_2}$ using convolutional operation with 1×1 kernel size.

Object attention: To reduce the distribution discrepancy between encoder and decoder features $D \in \mathbb{R}^{1\times H\times W}$ using minimal parameters, we organize an object attention module as a decoder. To refine the salient object, the object weight α_O is calculated as follows: $\alpha_O = \sigma(D)$. However, α_O cannot always detect the entire object with explicit edge regions; thus, we generate a complementary edge weight α_E to cover the undetected regions. For each pixel in D, we reverse the detected areas and eliminate background noise corresponding to the denoising ratio d for missed region detection.

Adaptive pixel intensity loss: Pixels that are adjacent to fine or explicit edges require more attention compared to pixels in the background and center of the salient object. Thus, this study proposes adaptive pixel intensity loss, which applies the pixel intensity ω to each pixel as follows:

$$\omega_{ij} = (1-\lambda)\sum_{k\in K}\left|\frac{\sum\limits_{h,w\in A_{ij}} y^k_{hw}}{\sum\limits_{h,w\in A_{ij}} 1} - y_{ij}\right| y_{ij} \qquad (4)$$

Here, we aggregate adjacent pixels (h, w) around the target pixel A_{ij} by using multiple kernel size K and excluding weights outside the edges. In Fig. 2, if the target pixel consists of fine edges, multi-kernel aggregation is employed to allocate more intensity to the target pixel than to other pixels. λ is an overriding weight that penalizes when employing multi-kernel aggregation because hierarchical aggregation imposes more weights on the pixels at the explicit edges.

Experiments

Experimental setup: We performed the evaluation on public benchmark datasets: DUTS and DUT-OMRON. We trained TRACER using the DUTS-TR dataset and used the DUT-OMRON for testing, following existing studies (Wei, Wang, and Huang 2020; Wei et al. 2020). To measure TRACER performance, we used two evaluation metrics MAE and S-measure. The S-measure, which calculates the object-aware (S_o) and region-aware (S_r) structural similarities, was calculated as $S_m = \alpha \times S_o + (1-\alpha) \times S_r$, where α = 0.5.

Models	#Params	GFLOPs	MPE	DUTS-TE			DUT-OMRON		
				MAE	S_m	FPS	MAE	S_m	FPS
SCRN	25.25M	30.18	10.04m	.040	.885	41.29	.056	.837	41.52
F3Net	25.54M	32.86	7.24m	.035	.888	60.51	.053	.838	63.22
LDF	25.15M	31.02	7.05m	.034	**.892**	64.41	.052	.839	67.00
TR-R	25.28M	25.94	3.73m	.035	.890	145.48	.050	.845	154.38
TE2	11.09M	5.20	**2.46m**	**.030**	.891	**242.92**	**.047**	**.846**	**267.25**

Table 1: Comparison of TRACER effectiveness for ResNet based methods.

Experimental results: We compared TRACER framework with existing methods (Wu, Su, and Huang 2019; Wei, Wang, and Huang 2020; Wei et al. 2020), which showed outstanding performance. For a fair comparison, we measured minutes per epoch (MPE) and frames per second (FPS) for model training and inference times, respectively, under the same conditions. We adopted ResNet50 (TR-R) and EfficientNet b2 (TE2) as the backbone encoders. As listed in the Tab.1, the TE2 performed 2.9× to 4.1× faster than the existing methods on training each epoch and 3.8× to 6.4× faster on inference times. Indeed, the existing multi-decoder frameworks occupied 32.6% (SCRN), 38.2% (F3Net), and 34.6% (LDF) of total GFLOPs at the multi-decoder structures, respectively. In contrast, TR-R occupied 12.9%; thus, it shows TRACER framework improves the existing multi-decoder inefficiency.

Conclusion

We study the inefficiencies in the existing encoder-decoder structure, and we propose TRACER, which discriminates salient objects by employing ATMs. TRACER detects the objects and edges in both channel and spatial-wise representations using minimal learning parameters. To treat the relative importance of pixels, we propose an adaptive pixel intensity loss function. TRACER improves the performance and computational efficiency in comparison to the existing methods on the DUTS and DUT-OMRON datasets.

References

Wei, J.; Wang, S.; and Huang, Q. 2020. F^{3}Net: Fusion, Feedback and Focus for Salient Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12321–12328.

Wei, J.; Wang, S.; Wu, Z.; Su, C.; Huang, Q.; and Tian, Q. 2020. Label Decoupling Framework for Salient Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13025–13034.

Wu, Z.; Su, L.; and Huang, Q. 2019. Stacked cross refinement network for edge-aware salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 7264–7273.