DNA: Deeply-supervised Nonlinear Aggregation for Salient Object Detection

Yun Liu, Ming-Ming Cheng, Xin-Yu Zhang, Guang-Yu Nie, and Meng Wang

Abstract—Recent progress on salient object detection mainly aims at exploiting how to effectively integrate multi-scale convolutional features in convolutional neural networks (CNNs). Many popular methods impose deep supervision to perform side-output predictions that are linearly aggregated for final saliency prediction. In this paper, we theoretically and experimentally demonstrate that linear aggregation of side-output predictions is suboptimal, and it only makes limited use of the side-output information obtained by deep supervision. To solve this problem, we propose Deeply-supervised Nonlinear Aggregation (DNA) for better leveraging the complementary information of various side-outputs. Compared with existing methods, it i) aggregates side-output features rather than predictions, and ii) adopts nonlinear instead of linear transformations. Experiments demonstrate that DNA can successfully break through the bottleneck of current linear approaches. Specifically, the proposed saliency detector, a modified U-Net architecture with DNA, performs favorably against state-of-the-art methods on various datasets and evaluation metrics without bells and whistles.

Index Terms—Salient object detection, saliency detection, deeply-supervised nonlinear aggregation.

I. INTRODUCTION

Salient object detection, also known as saliency detection, aims at simulating the human vision system to detect the most conspicuous and eye-attracting objects or regions in natural images [1], [2], [3]. The progress in saliency detection has been beneficial to a wide range of vision applications, including image retrieval [4], [5], visual tracking [6], scene classification [7], content-aware image/video processing [8], [9], thumbnail generation [10], video object segmentation [11], and weakly supervised learning [12], [13]. Although numerous models have been presented [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] and significant improvement has been made, it still remains an open problem to accurately detect complete salient objects in static images, especially in complicated scenarios.

Conventional saliency detection methods [2], [27], [28] usually design hand-crafted low-level features and heuristic priors, which are difficult to represent semantic objects and scenes. Recent advances on saliency detection mainly benefit from convolutional neural networks (CNNs) [29], [30], [31], [32], [33]. On the one hand, CNNs naturally learn multi-scale and multi-level feature representations in each layer due to the increasingly larger receptive fields and downsamped (strided) scales [34]. On the other hand, salient object detection requires multi-scale learning because of the various object/scene scales in intra- and inter-images [35], [36]. Therefore, current cutting-edge saliency detectors [15], [37], [38], [39], [17], [40], [41], [42], [43] mainly aim at designing complex network architectures to leverage multi-scale CNN features, e.g., the semantic meaningful information in the top sides and the complementary spatial details in the bottom sides.

Owing to the superiority of U-Net [25] (or FCN [44]) and HED [26] in multi-scale learning, many leading-edge saliency detectors add deep supervision onto U-Net networks [16], [38], [17], [18], [41], [45], [19], [46] (Fig. 1(d)). We note that these networks first predict multi-scale saliency maps using side-outputs. The generated multi-scale side-output predictions are then linearly aggregated, e.g., via a pixel-wise convolution (i.e., $1 \times 1$ convolution), to obtain the final saliency prediction which can thus combine the advantages of all side-output predictions. However, we theoretically and experimentally demonstrate that the linear aggregation of side-output predictions is suboptimal, and it makes limited use of the complementary multi-scale information implicated in side-output features. We provide detailed proofs in Section III.

Instead of linearly aggregating side-output predictions, we propose a nonlinear side-output aggregation method. Specifically, we concatenate the side-output features rather than side-output predictions and then apply nonlinear transformations to predict salient objects. We also impose deep supervision to side-output features for better optimization in the training phase, as illustrated in Fig. 1(e). In this way, the concatenated features can make better use of the multi-scale side-output features. We call the resulting method Deeply-supervised Nonlinear Aggregation (DNA). We apply DNA into a simply redesigned U-Net without bells and whistles. The proposed network performs favorably against all previous state-of-the-art salient object detectors with less parameters and faster speed. Our contributions are twofold:

- We theoretically and experimentally analyze the natural limitation of traditional linear side-output aggregation which can only make limited use of multi-scale side-output information.
- We propose Deeply-supervised Nonlinear Aggregation (DNA) for side-output features, whose effectiveness has been proved by introducing it into a simple network with less parameters and faster speed.
II. RELATED WORK

Salient object detection is a very active research field due to its wide range of applications and challenging scenarios. Early heuristic saliency detection methods extract hand-crafted low-level features and apply machine learning models to classify these features [47], [48], [49]. Some heuristic saliency priors are utilized to ensure the accuracy, such as color contrast [1], [2], center prior [50], [27] and background prior [51], [52]. With vast successes achieved by deep CNNs in computer vision, CNN-based methods have been introduced to improve saliency detection [53], [54], [55], [56], [57]. Region-based saliency detection [58], [59], [60], [61], [62], [20], [21] appeared in the early era of deep learning based saliency. These approaches view each image patch as a basic processing unit to perform saliency detection. More recently, CNN-based image-to-image saliency detection [15], [16], [37], [38], [39], [17], [40], [18], [41], [63], [42], [64], [65], [43], [66], [67], [45], [22], [23] has dominated this field by viewing saliency detection as a pixel-wise regression task and performing image-to-image predictions. Hence we mainly review CNN-based image-to-image saliency detection in the following.

Since saliency detection requires both high-level global information (existing in the top sides of CNNs) and low-level local details (existing in the bottom sides of CNNs), how to effectively fuse multi-level deep features is the main research direction [15], [37], [38], [39], [17], [40], [41], [42], [43], [66], [67], [45], [68], [69]. There are too many studies to list here, but the general trend of recent network designs is to become more and more complicated. We continue our discussion by briefly categorizing multi-scale deep learning into four classes: hyper feature learning, U-Net style, HED style, and U-Net + HED style. An overall illustration of this field is shown in Fig. 1.

Hyper feature learning: Hyper feature learning [24], [70], [71] is the most intuitive way to learn multi-scale information, as illustrated in Fig. 1(a). Examples of this structure for saliency include [66], [37], [63], [42], [64], [72], [65], [73]. These models concatenate/sum multi-scale deep features from multiple layers of backbone nets [66], [37] or branches of the multi-stream nets [63], [42], [64]. The fused hyper features, called hypercolumn, are then used for final predictions.

U-Net style: It is widely accepted that the top layers of deep neural networks contain high-level semantic information, while the bottom layers learn low-level fine details. Therefore, a reasonable revision of hyper feature learning is to progressively fuse deep features from upper layers to lower layers [44], [25], as shown in Fig. 1(b). The top semantic features will combine with bottom low-level features to capture fine-grained details. The feature fusion can be a simple element-wise summation [44], a simple feature map concatenation (U-Net) [25], or complex designs based on them. Many saliency detectors are of this type [74], [75], [67], [76], [40], [39], [14], [77], [78]. Note that hyper feature learning and U-Net do not apply deep supervision, so they do not have side-outputs.

HED style: HED-like networks [26], [79], [80] were first presented for edge detection. Afterwards, similar ideas have been also introduced for saliency detection [15], [43]. HED-like networks add deep supervision at the intermediate sides to obtain side-output predictions, and the final result is a linear combination of all side-output predictions (shown in Fig. 1(c)). Unlike multi-scale feature fusion, HED performs multi-scale prediction fusion.

U-Net + HED style: These methods combine the advantages of both U-Net and HED. We outline this architectures in Fig. 1(d). Specifically, deep supervision is imposed at each of the convolution stage of U-Net decoder. Many recent saliency models fall into this category [16], [38], [17], [18], [41], [81], [45], [19], [82], [83], [84], [85], [23], [46]. They differ from each other by applying different fusion strategies. One notable similarity of these models is that the final prediction is produced by a linear aggregation of side-output predictions. Hence the multi-scale learning is achieved in two aspects: i) the U-Net aggregates multi-level convolutional features from top layers to bottom layers in an encoder-decoder form; ii) the multi-scale side-output predictions are further linearly aggregated for final prediction. Current research in this field mainly focuses on the first aspect, and top-performing models have designed very complex feature fusion strategies for this [17], [41].

A full literature review of salient object detection is beyond the scope of this paper. Please refer to [86], [87], [88] for more comprehensive surveys. In this paper, we focus on the second aspect of above U-Net + HED multi-scale learning: the multi-scale side-output aggregation. We find that the upper bound of traditional linear side-output prediction aggregation is limited to the side-output predictions. Hence we propose DNA to aggregate side-output features in the nonlinear way,
so that the aggregated hybrid features can make good use of the complementary multi-scale deep features. A streamlined diagram of our proposed DNA can be seen in Fig. 1(e). We demonstrate DNA can achieve superior performance with a very simple U-Net.

III. REVISITING LINEAR SIDE-OUTPUT AGGREGATION

Deep supervision and corresponding linear side-output prediction aggregation have been demonstrated to be effective in many vision tasks [26], [79], [17], [41]. This section analyzes the natural limitation of the linear-side-output aggregation from both theoretical and experimental perspectives. To the best of our knowledge, this is a novel contribution.

Suppose a deeply-supervised network has $N$ side-output prediction maps $\{O_1, O_2, \cdots, O_N\}$, all of which are supervised by ground-truth maps (Fig. 1(c)(d)). Without loss of generality, we assume the linear side-output aggregation is a pixel-wise convolution, i.e., $1 \times 1$ convolution. Hence, current linear side-output aggregation can be written as

\[
\hat{O} = \sum_{i=1}^{N} w_i \cdot O_i,
\]

where weights $w_i$ of pixel-wise convolution can be learned. Note that we have $w_i \geq 0$ here. Otherwise, $O_i$ would have negative effect to $\hat{O}$, so it should be excluded in the aggregation. To obtain the output saliency probability map, a standard sigmoid function $\sigma(x) = \frac{1}{1 + e^{-x}}$ should be applied to $\hat{O}$. The aggregated probability map becomes

\[
\hat{P} = \sigma(\hat{O}) = \sigma(\sum_{i=1}^{N} w_i \cdot O_i), \tag{2}
\]

Similarly, we can compute side-output probability maps $\{P_1, P_2, \cdots, P_N\}$.

Theorem 1. If $\|w\|_1 = 1$, the mean absolute error (MAE) of fused output $\hat{P}$ is limited by side-output predictions.

Proof. If $\|w\|_1 = 1$, it is natural to show

\[
\min(O_i) \leq \sum_{i=1}^{N} w_i \cdot O_i \leq \max(O_i), \tag{3}
\]

because $w_i \geq 0$ as discussed above. Since the sigmoid function $\sigma(x)$ is monotonically increasing, we have

\[
\min(P_i) \leq \hat{P} \leq \max(P_i). \tag{4}
\]

If a pixel $p$ is positive, we have $\text{MAE}(\hat{P})_p = |1 - \hat{P}(p)| = 1 - \hat{P}(p)$ and $1 - \max(P_i)_p \leq 1 - \hat{P}(p) \leq 1 - \min(P_i)_p$, so that $\min(\text{MAE}(P_i)_p) \leq \text{MAE}(\hat{P})_p \leq \max(\text{MAE}(P_i)_p)$ holds. If the pixel $p$ is negative, we have $\text{MAE}(\hat{P})_p = |0 - \hat{P}(p)| = \hat{P}(p)$ and $\min(P_i)_p \leq \hat{P}(p) \leq \max(P_i)_p$, so that $\min(\text{MAE}(P_i)_p) \leq \text{MAE}(\hat{P})_p \leq \max(\text{MAE}(P_i)_p)$ holds. Note that $w$ usually only has $N$ ($N \leq 6$ in VGG16 [89] and ResNet [90]) dimensions, so it is also difficult to make aforementioned left equality hold. Hence traditional linear aggregation is limited in terms of MAE metric. However, what we expect is to break through the limitation by making full use of multi-scale information. \hfill \Box

![Fig. 2. Probability (x axis) vs. the density of X' and Y' (y axis). TN: true negative; FN: false negative; TP: true positive; FP: false positive.](image)

\textbf{Lemma 1.} If $\|w\|_1 \neq 1$, traditional linear aggregation (as in Eq. (1) and Eq. (2)) is equivalent to first applying an aggregation with $\|w\|_1 = 1$ and then applying a monotonically increasing mapping.

Proof. If $\|w\|_1 \neq 1$, we set $w = \tilde{w} = \|w\|_1$, so we have $\|\tilde{w}\|_1 = 1$. The computation of $\hat{P}$ becomes

\[
\hat{P} = \sigma(\sum_{i=1}^{N} \tilde{w}_i \cdot O_i), \tag{5}
\]

in which $\sigma(\|w\|_1 \cdot x)$ ($\|w\|_1 > 0$) is a monotonically increasing function in terms of $x$. \hfill \Box

\textbf{Theorem 2.} The monotonically increasing mapping of $\sigma(\|w\|_1 \cdot x)$ ($\|w\|_1 > 0$) cannot change the ROC curve and AUC metric$^1$.

Proof. Suppose the predicted scores of positive samples obey the distribution of $X \sim F(x)$, while the predicted scores of negative samples obey the distribution of $Y \sim G(x)$. We may assume $F$ and $G$ are continuous functions. $\varphi(x) = \sigma(k \cdot x)$ ($k > 0$) is a variant of sigmoid function, so we have $\varphi : \mathbb{R} \to (0, 1)$ and $\varphi$ is a monotonically increasing function. Let $X' = \varphi(X)$ and $Y' = \varphi(Y)$ be two transformed distributions. It is easy to show

\[
\mathbb{P}(X' \leq u) = \mathbb{P}(\varphi(X) \leq u) = \mathbb{P}(X \leq \varphi^{-1}(u)) = F(\varphi^{-1}(u)), \tag{6}
\]

and thus we can obtain $X' \sim F(\varphi^{-1}(x))$ and $Y' \sim G(\varphi^{-1}(x))$.

Let $t$ be a threshold, true positive rate (TPR) and false positive rate (FPR) can be computed as

\[
\text{TPR} = \frac{TP}{TP + FN} = \mathbb{P}(X' > t) = 1 - F(\varphi^{-1}(t)), \tag{7}
\]

\[
\text{FPR} = \frac{FP}{FP + TN} = \mathbb{P}(Y' > t) = 1 - G(\varphi^{-1}(t)),
\]

as shown in Fig. 2. Hence we can denote the ROC curve as $\{(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t))) : t \in (0,1)\}$. It is easy to show that as $t$ goes from $0$ to $1$ continuously, $(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t)))$ will change from $(1,1)$ to $(0,0)$ continuously and monotonically. It is also

$^1$AUC is the area under the ROC Curve.
obvious to see that \( \{(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t)))\} \) and \( \{(F(\varphi^{-1}(t)), G(\varphi^{-1}(t)))\} \) are symmetric about the point \( \left( \frac{1}{2}, \frac{1}{2} \right) \). Suppose the area under the curve \( \{(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t)))\} \) is \( S_1 \), and the area under the curve \( \{(F(\varphi^{-1}(t)), G(\varphi^{-1}(t)))\} \) is \( S_2 \). By symmetry, we have \( S_1 + S_2 = 1 \).

With the above conclusions, we can compute \( S_2 \) as

\[
S_2 = \int_0^1 G(\varphi^{-1}(t))dF(\varphi^{-1}(t))
\]

\[
= \int_{-\infty}^{+\infty} G(x)dF(x).
\]

Therefore, \( S_2 \) is independent of the specific form of the function \( \varphi(x) \), and \( S_1 = 1 - S_2 \) is independent of \( \varphi(x) \), too. Moreover, as \( t \) ranges in \((0, 1)\), thus \( \varphi^{-1}(t) \) ranges in \( \mathbb{R} \). We have

\[
\{(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t))) : t \in (0, 1)\}
\]

which is also independent of the form of \( \varphi(x) \). When \( \{F(x), G(x)\} \) are discrete, the set \( \{(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t))) : t \in (0, 1)\} \) is discrete but still independent of \( \varphi(x) \). Therefore, we can conclude that \( \varphi(x) \) cannot change the ROC curve and AUC metric.

Similar to the proof for Theorem 1, we can easily demonstrate that the first step in Lemma 1, i.e., linear aggregation with \( \|\tilde{w}\|_1 = 1 \), has limited MAE results. From Theorem 2, we know the second step in Lemma 1, i.e., a monotonically increasing mapping, cannot change the ROC curve and AUC value. Therefore, we can conclude that traditional linear aggregation with \( \|w\|_1 \neq 1 \) has limited improvement. Combined with Theorem 1, we can conclude that linear aggregation of side-outputs only has limited improvement.

Besides the theoretical proofs, we also perform experiments to compare linear aggregation versus nonlinear aggregation for salient object detection. To this end, we use the proposed nonlinear side-output feature aggregation (in Section IV-B) for nonlinear regression to evaluate two well-known models: HED [26] and DSS [43], and the proposed DNA model. The results are summarized in Table I. We can see significant improvement from linear regression to nonlinear regression. Based on this observation, this paper aims at designing a simple network with nonlinear side-output aggregation for effective salient object detection.

### IV. Methodology

In this section, we will elaborate our proposed framework for salient object detection. We first introduce our base network in Section IV-A. Then, we present the deeply-supervised nonlinear aggregation in Section IV-B. An overall network architecture is illustrated in Fig. 3.

#### A. Base Network

**Backbone network.** To tackle the salient object detection, we follow recent studies [63], [42], [43] to use fully convolutional networks. Specifically, we use the well-known VGG16 network [89] as our backbone network, whose final fully connected layers are removed to serve for image-to-image translation. Salient object detection usually requires global
information to locate the coarse positions of salient objects [2], so enlarging the receptive field of the network would be helpful. To this end, we keep the final pooling layer of VGG16 as in [43] and replace the last two fully connected layers with two convolution layers, one of which has the kernel size of $3 \times 3$ with $C^6_{(1)} = 192$ channels and another of which has the kernel size of $7 \times 7$ with $C^6_{(2)} = 128$ channels. Here, we use the $3 \times 3$ convolution layer to reduce the feature channels, because large kernel sizes (e.g., $7 \times 7$) lead to much more parameters.

There are five pooling layers in the backbone network. They divide the convolution layers into six convolution blocks, which are denoted as $\{S^1, S^2, S^3, S^4, S^5, S^6\}$ from bottom to top, respectively. We consider $S^6$ as the top valve that controls the overall contextual information that flows in the network. The resolution of the feature maps in each convolution block is half of the preceding one. Following [43], [26], the side-output of each convolution block is connected from the last layer of this block.

### Encoder-decoder network

Based on the backbone net, we build an encoder-decoder network that can be seen in Fig. 3. Concretely, we connect a $1 \times 1$ convolution layer to each of the convolution blocks $S^6$ and $S^5$ to adjust the number of channels (as shown in Table II). Then, we upsample the obtained feature maps from $S^6$ by two. The upsampled feature maps and resulting feature maps from $S^5$ are concatenated. To fuse the concatenated feature maps, two sequential convolution layers are used to generate the decoder side $S^5$. The decoder sides $\{S^4, S^3, S^2, S^1\}$ can be obtained in the same manner. For a clear presentation, we formulate the above process as follows:

$$
\begin{align*}
&\tilde{S}^i = \varphi(\text{Concat}(\phi_1(S^i), \phi_2(\tilde{S}^{i+1}))), \\
&\phi_1(\cdot) = \text{Conv}(\cdot), \\
&\phi_2(\cdot) = \text{Upsample(Conv}(\cdot)), \\
&\varphi(\cdot) = \text{ReLU(Conv}(\cdot)), \\
&\forall i \in \{1, 2, 3, 4, 5\}.
\end{align*}
$$

Note that we have $\tilde{S}^6 = S^6$, because $S^6$ is the last block in the encoder path and also the first block in the decoder path. In this way, the proposed encoder-decoder lets top contextual information flow into the lower layers, so the lower layers are expected to emphasize the details of salient objects in an image. Here, both two sequential convolution layers (\varphi(\cdot)) at the decoder side $S^i$ are with kernel size of $K_i \times K_i$ and output channels of $C_i$. We will discuss the parameter settings in detail in the experiment part.

### B. Deeply-supervised Nonlinear Aggregation

Instead of linearly aggregating side-output predictions at multiple sides as in previous literature [16], [38], [17], [18], [41], [45], [19], we propose to aggregate the side-output features in a nonlinear way. The proposed DNA module is displayed in the dotted box of Fig. 3. Specifically, we first adopt a $3 \times 3$ convolution for each $\tilde{S}^i$ to adjust the number of channels. Then, the feature maps are upsampled into the same size of the original image to generate **side-output features**. The side-output features can predict saliency maps using a simple $1 \times 1$ convolution. In the training phase, deep supervision is added for these predicted maps.

We concatenate all side-output features to construct hybrid features that contain rich multi-scale and multi-level information. One of the key ideas in our nonlinear aggregation is that we use asymmetric convolution that decomposes a standard two-dimensional convolution into two one-dimensional convolutions. That is to say, an $n \times n$ convolution is decomposed into two sequential convolutions with kernel sizes of $1 \times n$ and $n \times 1$. Here, the reasons why we use asymmetric convolution are twofold. On one hand, in the experiments, we find large kernel size in the DNA module can improve performance, and we believe it is because hybrid feature maps have large resolution, *i.e.*, the same resolution as the original image. On the other hand, convolutions with large kernel sizes are very time-consuming for large feature maps. According to the above analyses, we set $n = 7$ for asymmetric convolutions rather than small kernel sizes. Larger kernel sizes than $n = 7$ will only lead to little accuracy improvement while causing more computational load. The effectiveness of this choice has been validated in Section V-C where we try different settings of $n$ and asymmetric/standard convolutions. We use two groups of asymmetric convolutions, each of which consists of a $1 \times 7$ and a $7 \times 1$ convolution. With a $300 \times 300$ input image, the number of FLOPs (multiply-adds) for these asymmetric convolutions is 13.8G, while the number of FLOPs will be 60.4G if we use the standard two-dimensional $7 \times 7$ convolutions. At last, we connect a $1 \times 1$ convolution after the asymmetric convolutions to predict the final saliency maps.

In training, we adopt class-balanced cross-entropy loss [26] to supervise all side-output and final fused predictions. Since convolutions in the DNA module are followed by nonlinear activation (i.e., ReLU), the aggregation of side-output features is nonlinear. Although there are several nonlinear functions that can be used, such as ReLU, PReLU, and LeakyReLU, in this paper, we simply use the most common ReLU function to demonstrate the necessity of nonlinear side-output aggregation. The traditional linear side-output prediction aggregation can only linearly combine multi-scale predictions, while the proposed nonlinear side-output feature aggregation can make use of the complementary multi-scale features for final prediction and is thus more effective. With the simple encoder-decoder in Section IV-A, DNA performs favorably against previous methods. Note that previous methods [16], [38], [17], [18] usually present various network architectures, modules, and operations to improve performance, but in this paper, DNA only applies a simply-modified U-Net as base network.

---

**TABLE II**

<table>
<thead>
<tr>
<th>Side</th>
<th>$C_i$</th>
<th>$K_i \times K_i$</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Side-output 1</td>
<td>64</td>
<td>$3 \times 3$</td>
<td>1</td>
</tr>
<tr>
<td>Side-output 2</td>
<td>128</td>
<td>$3 \times 3$</td>
<td>1/2</td>
</tr>
<tr>
<td>Side-output 3</td>
<td>128</td>
<td>$5 \times 5$</td>
<td>1/4</td>
</tr>
<tr>
<td>Side-output 4</td>
<td>128</td>
<td>$5 \times 5$</td>
<td>1/8</td>
</tr>
<tr>
<td>Side-output 5</td>
<td>128</td>
<td>$5 \times 5$</td>
<td>1/16</td>
</tr>
<tr>
<td>Side output 6</td>
<td>-</td>
<td>-</td>
<td>1/32</td>
</tr>
</tbody>
</table>
set the momentum and weight decay to the typical values of\cite{61,40,53,54,59,62,63,64,72,84} to power parameters one multiplying poly of kernels. Since the deconvolution layers do not need training, initialized to 0. The upsampling operations are implemented Gaussian distribution with standard deviation 0.01. Biases are The weights of other layers are initialized from the zero-mean \cite{89} are initialized using the pretrained ImageNet model \cite{92}. framework. The convolution layers in the original VGG16 \cite{89} backbone. We implement our network using the well-known Caffe \cite{91}.

### A. Experimental Setup

**Implementation details.** The detailed configurations for $K_i$ and $C_i$ can be found in Table II. The large kernel size at top sides is helpful to accuracy. When $i = 1, 2, K_i \times K_i$ equals to $3 \times 3$; When $i = 3, 4, 5$, $K_i \times K_i$ equals to $5 \times 5$. The $C_i$ values for $i = 1, \cdots, 5$ are 64, 128, 128, 128 and 128, respectively. Since side-output prediction results have not been used, we remove these side-output prediction units in the test phase. However, we remain them in the training phase, because deep supervision can help the training and improve the accuracy of the final saliency prediction, as demonstrated in Section V-C.

We implement our network using the well-known Caffe \cite{91} framework. The convolution layers in the original VGG16 \cite{89} are initialized using the pretrained ImageNet model \cite{92}. The weights of other layers are initialized from the zero-mean Gaussian distribution with standard deviation 0.01. Biases are initialized to 0. The upsampling operations are implemented by deconvolution layers with frozen bilinear interpolation kernels. Since the deconvolution layers do not need training, we exclude them when computing the number of parameters. The network is optimized using SGD with learning rate policy of poly, in which the current learning rate equals the base one multiplying $\left(1 - \frac{\text{curr} \text{iter}}{\text{max} \text{iter}}\right)^{\text{power}}$. The hyper parameters $\text{power}$ and $\text{max} \text{iter}$ are set to 0.9 and 20000, respectively, so that the training takes 20000 iterations in total. The initial learning rate is set to $1 \times 10^{-7}$ that is the maximum value to keep the network from training exploding (i.e., larger values will cause the well-known "NaN" error). We follow previous saliency detection methods \cite{43,16,17,15,45,61,40,53,54,59,62,64,63,66,72,84} to set the momentum and weight decay to the typical values of 0.9 and 0.0005 \cite{93,89}, respectively. All experiments are performed on a TITAN Xp GPU.

**Datasets.** We extensively evaluate our method on six popular datasets, including DUTS \cite{94}, ECSSD \cite{95}, SOD \cite{96}, HKU-IS \cite{60}, THUR15K \cite{97} and DUT-O (or DUT-OMRON) \cite{51}. These six datasets consist of 15572, 1000, 300, 4447, 6232 and 5168 natural complex images with corresponding pixel-wise ground truth labeling. Among them, the DUTS dataset \cite{94} is a very recent dataset consisting of 10553 training images and 5019 test images in very complex scenarios. For a fair comparison, we follow recent studies \cite{40,17,42,37} to use DUTS training set for training and test on the DUTS test set (DUTS-TE) and other datasets.

**Evaluation criteria.** We utilize three evaluation metrics to evaluate our method as well as other recent salient object detectors, including max F-measure score ($F_{\beta}$), mean absolute error (MAE), and the weighted $F_{\beta}$-measure score \cite{98}.

Given a predicted saliency map with continuous probability values, we can convert it into binary maps with arbitrary thresholds and computing corresponding precision/recall values. Taking the average of precision/recall values over all images in a dataset, we can get many mean precision/recall pairs. F-measure is an overall performance indicator:

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

in which $\beta^2$ is usually set to 0.3 to emphasize more on precision. We follow recent studies \cite{67,43,41,75,17,14,15} to report max $F_{\beta}$ across different thresholds.

Given a saliency map $S$ and the corresponding ground truth
TABLE IV
Comparison between the proposed DNA and 16 competitors in terms of $F^\beta_\omega$-measure [98] on six datasets. The unit of the number of parameters (#Param) is million (M), and the unit of speed is frame per second (FPS). We report results on both VGG16 [89] backbone and ResNet-50 [90] backbone. The top three models in each column are highlighted in red, green and blue, respectively. For ResNet-50 based methods, we only highlight the top performance.

<table>
<thead>
<tr>
<th>Methods</th>
<th>#Param</th>
<th>Speed</th>
<th>DUTS-TE</th>
<th>ECSSD</th>
<th>HKU-IS</th>
<th>DUT-O</th>
<th>SOD</th>
<th>THUR15K</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRFI [27]</td>
<td>18.40</td>
<td>1/19</td>
<td>0.507</td>
<td>0.619</td>
<td>-</td>
<td>0.494</td>
<td>0.528</td>
<td>0.508</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.629</td>
<td>0.692</td>
<td>0.616</td>
<td>0.523</td>
<td>0.550</td>
<td>0.538</td>
</tr>
<tr>
<td>DCE [60]</td>
<td>36.86</td>
<td>1/19</td>
<td>0.507</td>
<td>0.619</td>
<td>-</td>
<td>0.494</td>
<td>0.528</td>
<td>0.508</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.629</td>
<td>0.692</td>
<td>0.616</td>
<td>0.523</td>
<td>0.550</td>
<td>0.538</td>
</tr>
<tr>
<td>DCL [66]</td>
<td>66.24</td>
<td>1.4</td>
<td>0.632</td>
<td>0.782</td>
<td>0.770</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>DHS [45]</td>
<td>94.04</td>
<td>10.0</td>
<td>0.705</td>
<td>0.837</td>
<td>0.816</td>
<td>-</td>
<td>0.685</td>
<td>0.666</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>ELD [61]</td>
<td>43.09</td>
<td>1.0</td>
<td>0.607</td>
<td>0.783</td>
<td>0.743</td>
<td>0.593</td>
<td>0.634</td>
<td>0.621</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>RFCN [74]</td>
<td>134.69</td>
<td>0.4</td>
<td>0.586</td>
<td>0.725</td>
<td>0.707</td>
<td>0.562</td>
<td>0.591</td>
<td>0.592</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>NLD [67]</td>
<td>35.49</td>
<td>18.5</td>
<td>0.710</td>
<td>0.835</td>
<td>0.838</td>
<td>0.634</td>
<td>0.708</td>
<td>0.676</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>DSS [43]</td>
<td>62.23</td>
<td>7.0</td>
<td>0.700</td>
<td>0.832</td>
<td>0.821</td>
<td>0.643</td>
<td>0.698</td>
<td>0.662</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>Amulet [41]</td>
<td>33.15</td>
<td>9.7</td>
<td>0.657</td>
<td>0.839</td>
<td>0.817</td>
<td>0.626</td>
<td>0.674</td>
<td>0.650</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>UCF [75]</td>
<td>23.98</td>
<td>12.0</td>
<td>0.595</td>
<td>0.805</td>
<td>0.779</td>
<td>0.574</td>
<td>0.673</td>
<td>0.613</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>PICA [17]</td>
<td>32.85</td>
<td>5.6</td>
<td>0.745</td>
<td>0.862</td>
<td>0.847</td>
<td>0.691</td>
<td>0.721</td>
<td>0.688</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>C2S [14]</td>
<td>137.03</td>
<td>16.7</td>
<td>0.717</td>
<td>0.849</td>
<td>0.835</td>
<td>0.663</td>
<td>0.700</td>
<td>0.685</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>RAS [15]</td>
<td>20.13</td>
<td>20.4</td>
<td>0.739</td>
<td>0.855</td>
<td>0.850</td>
<td>0.695</td>
<td>0.718</td>
<td>0.691</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.666</td>
<td>0.666</td>
<td>0.584</td>
<td>0.669</td>
<td>0.624</td>
</tr>
<tr>
<td>DNA</td>
<td>20.06</td>
<td>25.0</td>
<td>0.797</td>
<td>0.897</td>
<td>0.889</td>
<td>0.729</td>
<td>0.755</td>
<td>0.723</td>
</tr>
</tbody>
</table>

G that are normalized to [0, 1], MAE can be calculated as

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |S(i,j) - G(i,j)|,$$  \hspace{1cm} (12)

where $H$ and $W$ are height and width, respectively. $S(i,j)$ denotes the saliency score at location $(i,j)$, similar to $G(i,j)$.

As demonstrated in [98], traditional evaluation metrics easily suffer from the interpolation flaw, dependency flaw, and equal-importance flaw. Hence the weighted $F^\beta_\omega$-measure score is proposed to amend these flaws. We follow [47], [76], [48], [17] to adopt $F^\beta_\omega$-measure as a metric with the default settings.

B. Performance Comparison

We compare our proposed salient object detector with 16 recent competitive saliency models, including DRFI [27], MDF [60], LEGS [59], DCL [66], DHS [45], ELD [61], RFCN [74], NLD [67], DSS [43], SRM [42], Amulet [41], UCF [75], BRN [40], PICA [17], C2S [14] and RAS [15]. Among them, DRFI [27] is the best-known non-deep-learning based model, and the other 15 models are all based on deep learning. We do not report MDF [60] results on the HKU-IS [60] dataset because MDF uses a part of HKU-IS for training. Due to the same reason, we do not report DHS [45] results on the DUT-O [51] dataset. Since SRM [42] and BRN [40] are built based on the ResNet-50 [90] backbone, we also report the results of the ResNet-50 version of the proposed DNA and PICA [17] for a fair comparison. All previous methods are tested using their publicly available code and the pretrained models released by the authors with default settings.

F-measure and MAE. Table III summarizes the numeric comparison in terms of $F$-measure ($F_\beta$) and MAE on six datasets. DNA can significantly outperform other competitors in most cases, which demonstrates its effectiveness. With VGG16 [89] backbone, the $F_\beta$ values of DNA are 2.8%, 1.2%, 1.4%, 1.4%, 0.6% and 1.0% higher than the second best method on the DUTS-TE, ECSSD, HKU-IS, DUT-O, SOD and THUR15K datasets, respectively. As can be seen, DNA also achieves the best performance in terms of MAE metric except on the SOD dataset where DNA performs slightly worse than PiCA [17]. Overall, PiCA [17] seems to achieve the second place. With the ResNet-50 backbone, DNA still performs better than previous competitors, indicating DNA is robust to different network architectures. Therefore, we suggest the future salient object detectors using nonlinear side-output aggregation instead of the traditional linear aggregation.

Weighted $F^\beta_\omega$-measure. The weighted $F^\beta_\omega$-measure is also a commonly-used saliency evaluation metric. In Table IV, we evaluate DNA and above-mentioned competitors using the $F^\beta_\omega$-measure. The VGG16 version of DNA achieves 5.2%, 3.5%, 3.9%, 3.4%, 3.4% and 3.2% higher $F^\beta_\omega$-measure than the second best method on the DUTS-TE, ECSSD, HKU-IS, DUT-O, SOD and THUR15K datasets, respectively. For ResNet-50 version, DNA achieves 3.6%, 1.4%, 2.2%, 2.6%, 1.7% and 1.8% better $F^\beta_\omega$-measure than previous competitors. Note that the network of DNA is very simple, making it easy to be followed and applied to other vision tasks.

Number of parameters and runtime. As shown in Table IV, DNA has fewer parameters, i.e., about 20M parameters with VGG16 backbone and 29M parameters with ResNet-50 backbone. DNA also runs faster than other methods, achieving 25fps with VGG16 and 12.8fps with ResNet-50.

Qualitative comparison. To visually exhibit the superiority of the proposed DNA over previous methods, we select some...
Fig. 4. Qualitative comparison between DNA and recent competitive methods. Here, GT represents ground-truth saliency maps.
TABLE V
Ablation studies. U-Net means the standard U-Net [25] with VGG16 backbone. If removing the DNA module and deep supervision, the proposed network in Fig. 3 becomes an encoder-decoder network that is called Encoder-Decoder. Encoder-Decoder w/ K3 replaces all the convolutions at the top sides of Encoder-Decoder with $3 \times 3$ convolutions. Encoder-Decoder w/ lin means we replace the DNA module in Fig. 3 with traditional linear aggregation in [26].

<table>
<thead>
<tr>
<th>Methods</th>
<th>DUTS-TE $F_\beta$</th>
<th>MAE</th>
<th>ECSSD $F_\beta$</th>
<th>MAE</th>
<th>HKU-IS $F_\beta$</th>
<th>MAE</th>
<th>DUT-O $F_\beta$</th>
<th>MAE</th>
<th>SOD $F_\beta$</th>
<th>MAE</th>
<th>THUR15K $F_\beta$</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Net</td>
<td>0.793</td>
<td>0.080</td>
<td>0.890</td>
<td>0.065</td>
<td>0.894</td>
<td>0.051</td>
<td>0.723</td>
<td>0.101</td>
<td>0.811</td>
<td>0.115</td>
<td>0.758</td>
<td>0.099</td>
</tr>
<tr>
<td>Encoder-Decoder w/ K3</td>
<td>0.766</td>
<td>0.101</td>
<td>0.869</td>
<td>0.081</td>
<td>0.876</td>
<td>0.064</td>
<td>0.687</td>
<td>0.129</td>
<td>0.778</td>
<td>0.131</td>
<td>0.736</td>
<td>0.112</td>
</tr>
<tr>
<td>Encoder-Decoder</td>
<td>0.831</td>
<td>0.053</td>
<td>0.911</td>
<td>0.052</td>
<td>0.916</td>
<td>0.057</td>
<td>0.754</td>
<td>0.073</td>
<td>0.830</td>
<td>0.117</td>
<td>0.789</td>
<td>0.077</td>
</tr>
<tr>
<td>Encoder-Decoder w/ lin</td>
<td>0.844</td>
<td>0.048</td>
<td>0.921</td>
<td>0.050</td>
<td>0.917</td>
<td>0.034</td>
<td>0.765</td>
<td>0.066</td>
<td>0.839</td>
<td>0.120</td>
<td>0.785</td>
<td>0.071</td>
</tr>
<tr>
<td>DNA w/o Deep Supervision</td>
<td>0.867</td>
<td>0.042</td>
<td>0.932</td>
<td>0.041</td>
<td>0.927</td>
<td>0.032</td>
<td>0.788</td>
<td>0.059</td>
<td>0.860</td>
<td>0.103</td>
<td>0.794</td>
<td>0.068</td>
</tr>
<tr>
<td>DNA</td>
<td>0.865</td>
<td>0.044</td>
<td>0.935</td>
<td>0.041</td>
<td>0.930</td>
<td>0.031</td>
<td>0.799</td>
<td>0.056</td>
<td>0.853</td>
<td>0.107</td>
<td>0.793</td>
<td>0.069</td>
</tr>
</tbody>
</table>

TABLE VI
Ablation studies for various parameter settings. The unit of the number of parameters (#Param) is million (M), and the unit of speed is frame per second (FPS).

<table>
<thead>
<tr>
<th>Methods</th>
<th>#Param</th>
<th>Speed</th>
<th>DUTS-TE $F_\beta$</th>
<th>MAE</th>
<th>ECSSD $F_\beta$</th>
<th>MAE</th>
<th>HKU-IS $F_\beta$</th>
<th>MAE</th>
<th>DUT-O $F_\beta$</th>
<th>MAE</th>
<th>SOD $F_\beta$</th>
<th>MAE</th>
<th>THUR15K $F_\beta$</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>18.49</td>
<td>27.0</td>
<td>0.859</td>
<td>0.044</td>
<td>0.932</td>
<td>0.041</td>
<td>0.928</td>
<td>0.031</td>
<td>0.796</td>
<td>0.057</td>
<td>0.855</td>
<td>0.105</td>
<td>0.790</td>
<td>0.069</td>
</tr>
<tr>
<td>#2</td>
<td>20.06</td>
<td>25.0</td>
<td>0.865</td>
<td>0.047</td>
<td>0.935</td>
<td>0.041</td>
<td>0.930</td>
<td>0.031</td>
<td>0.799</td>
<td>0.056</td>
<td>0.853</td>
<td>0.107</td>
<td>0.793</td>
<td>0.069</td>
</tr>
<tr>
<td>#3</td>
<td>27.88</td>
<td>22.7</td>
<td>0.866</td>
<td>0.043</td>
<td>0.936</td>
<td>0.041</td>
<td>0.930</td>
<td>0.031</td>
<td>0.799</td>
<td>0.056</td>
<td>0.861</td>
<td>0.106</td>
<td>0.792</td>
<td>0.069</td>
</tr>
<tr>
<td>#4</td>
<td>41.41</td>
<td>18.2</td>
<td>0.864</td>
<td>0.044</td>
<td>0.935</td>
<td>0.041</td>
<td>0.931</td>
<td>0.030</td>
<td>0.800</td>
<td>0.056</td>
<td>0.857</td>
<td>0.105</td>
<td>0.792</td>
<td>0.069</td>
</tr>
</tbody>
</table>

TABLE VII
Parameter settings for ablation studies in Table VI. The default setting in this paper is highlighted in dark.

<table>
<thead>
<tr>
<th>No.</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Side 1</td>
<td>(3 x 3, 64)</td>
<td>(3 x 3, 64)</td>
<td>(3 x 3, 64)</td>
<td>(3 x 3, 64)</td>
</tr>
<tr>
<td>Side 2</td>
<td>(3 x 3, 128)</td>
<td>(3 x 3, 128)</td>
<td>(3 x 3, 128)</td>
<td>(3 x 3, 128)</td>
</tr>
<tr>
<td>Side 3</td>
<td>(3 x 3, 128)</td>
<td>(5 x 5, 128)</td>
<td>(5 x 5, 128)</td>
<td>(5 x 5, 128)</td>
</tr>
<tr>
<td>Side 4</td>
<td>(3 x 3, 128)</td>
<td>(5 x 5, 256)</td>
<td>(5 x 5, 256)</td>
<td>(5 x 5, 256)</td>
</tr>
<tr>
<td>Side 5</td>
<td>(3 x 3, 128)</td>
<td>(5 x 5, 256)</td>
<td>(5 x 5, 256)</td>
<td>(5 x 5, 512)</td>
</tr>
<tr>
<td>Side 6</td>
<td>(192, 128)</td>
<td>(192, 128)</td>
<td>(256, 256)</td>
<td>(256, 256)</td>
</tr>
</tbody>
</table>

representative images from various datasets to incorporate a variety of difficult circumstances, including complicated scenes, salient objects with thin structures, low contrast between foreground and background, multiple objects with different sizes, scenes with abnormal brightness, and etc. We display a qualitative comparison in Fig. 4 where we split the selected images into multiple groups, each of which is with several tags to describe its properties. Taking all circumstances into account, the proposed DNA can segment the right salient objects with coherent boundaries and connected regions, even in the complex, low-contrast, and abnormal scenes. This is the reason why DNA behaves better than other methods in the above quantitative comparison.

C. Ablation Studies
Nonlinear aggregation vs. linear aggregation. To demonstrate the effectiveness of nonlinear aggregation, we replace the DNA module in our network with the traditional linear side-output prediction aggregation [26] to obtain a deeply-supervised encoder-decoder, i.e., Encoder-Decoder w/ lin. The results are shown in Table V. We can clearly see that nonlinear aggregation performs significantly better than linear aggregation, in terms of both $F_\beta$ and MAE. A qualitative comparison between the linear side-output aggregation and nonlinear aggregation is shown in the 4th and 5th columns of Fig. 5. The superiority of nonlinear aggregation can be clearly observed in various complicated scenarios.

The proposed encoder-decoder vs. standard U-Net. If removing the DNA module and deep supervision, the proposed encoder-decoder is a simply modified version of U-Net [25]. First, we change the kernel size of all convolutions at top sides, i.e., $K_3 \times K_3, K_4 \times K_4$ and $K_5 \times K_5$, into $3 \times 3$. As displayed in Table V, the resulting model, Encoder-Decoder w/ K3, perform worse than the standard U-Net [25]. This could be because the proposed encoder-decoder has less feature channels and thus less parameters (U-Net has 31.06M parameters). Next, we use the default kernel size of $5 \times 5$ for top sides. The resulting model, Encoder-Decoder, performs better than U-Net. This demonstrates large kernel size at the top sides is important for better performance. We provide the qualitative comparison between Encoder-Decoder and U-Net in the 2nd and 3rd columns of Fig. 5. The proposed Encoder-Decoder can predict better saliency maps.

Encoder-decoder with or without deep supervision. In Table V, Encoder-Decoder w/ lin performs better than Encoder-Decoder, which can also be seen in the 3rd and 4th columns of Fig. 5. If removing the deep supervision in DNA, the resulting model (DNA w/o Deep Supervision) performs worse than DNA in most scenarios. Therefore, deep supervision can consistently improve the saliency prediction performance.

Parameter settings. To evaluate the effect of different parameter settings, we try various parameter settings in Table VII. For side 1-5, we report the settings of $(K_i \times K_i, C_i)$. For side 6, we report the settings of $C_6^{(1)}, C_6^{(2)}$. The evaluation results are summarized in Table VI. From the first and second experiment, we can see that large kernel sizes at top sides lead to better results, but the improvement is not as significant as in Table V where deep supervision is not used. From the third and
The asymmetric convolutions in DNA module.

In the second setting as our default parameters, better results. Considering the trade-off between the performance by increasing the convolution channels can generate slightly nonlinear aggregation in DNA with linear aggregation, this figure demonstrates the superiority of nonlinearity in saliency detection.

TABLE VIII
VARIOUS CONVOLUTION KERNEL SIZES IN THE DNA MODULE.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Metrics</th>
<th>$3 \times 3$</th>
<th>$5 \times 5$</th>
<th>$7 \times 7$</th>
<th>$1 \times 7$</th>
<th>$1 \times 9$</th>
<th>$9 \times 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUTS-TE</td>
<td>$F_{\beta}$</td>
<td>0.861</td>
<td>0.863</td>
<td>0.865</td>
<td>0.865</td>
<td>0.864</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.045</td>
<td>0.045</td>
<td>0.043</td>
<td>0.044</td>
<td>0.044</td>
<td></td>
</tr>
<tr>
<td>ECSSD</td>
<td>$F_{\beta}$</td>
<td>0.930</td>
<td>0.933</td>
<td>0.935</td>
<td>0.935</td>
<td>0.935</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.042</td>
<td>0.041</td>
<td>0.041</td>
<td>0.041</td>
<td>0.040</td>
<td></td>
</tr>
<tr>
<td>DUT-O</td>
<td>$F_{\beta}$</td>
<td>0.795</td>
<td>0.797</td>
<td>0.799</td>
<td>0.799</td>
<td>0.798</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.058</td>
<td>0.058</td>
<td>0.057</td>
<td>0.056</td>
<td>0.056</td>
<td></td>
</tr>
<tr>
<td>Speed (fps)</td>
<td></td>
<td>27.8</td>
<td>23.2</td>
<td>19.6</td>
<td>25.0</td>
<td>20.4</td>
<td></td>
</tr>
</tbody>
</table>

With a simply modified U-Net, DNA can reach new state-of-the-art under various metrics when compared with 16 recent saliency models. The proposed network also has less parameters and faster running speed, which demonstrate the effectiveness of DNA. In the future, we plan to apply DNA to further improve salient object detection and exploit it in other vision tasks that need multi-scale and multi-level information.

REFERENCES


