DNA: Deeply-supervised Nonlinear Aggregation for Salient Object Detection

Yun Liu\textsuperscript{1} Deng-Ping Fan\textsuperscript{1} Guang-Yu Nie\textsuperscript{2} Xinyu Zhang\textsuperscript{1}
Vahan Petrosyan\textsuperscript{3} Ming-Ming Cheng\textsuperscript{1}

\textsuperscript{1} Nankai University \hspace{1cm} \textsuperscript{2} Beijing Institute of Technology \hspace{1cm} \textsuperscript{3} SuperAnnotate.AI

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 state-of-the-art 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. Code and data will be released upon paper acceptance.

1. 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, 9]. The progress in saliency detection has been beneficial to a wide range of vision applications, including image retrieval [12], visual tracking [42], scene classification [45], content-aware video compression [74], and weakly supervised learning [58, 59]. Although numerous models have been presented [31, 5, 70, 37, 21, 17] 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 [9, 24, 50] 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) [41, 34, 70, 57, 67, 27, 28, 72]. 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 downsampled (strided) scales. On the other hand, salient object detection requires multi-scale learning because of the various object/scene scales in intra- and inter-images. Therefore, current state-of-the-art saliency detectors [5, 64, 57, 67, 37, 56, 68, 55, 18] 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 [46] (or FCN [40]) and HED [61] in multi-scale learning, many state-of-the-art saliency detectors add deep supervision onto U-Net networks [70, 57, 37, 21, 68, 34, 17, 33, 22] (Figure 1(d)). We notice 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 thus can 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 3.

Based on this observation, we propose a new solution to this problem, namely Deeply-supervised Nonlinear Aggregation (DNA). Specifically, instead of linearly aggregating side-output predictions, we concatenate the side-output features and apply nonlinear transformations to predict salient objects, as illustrated in Figure 1(e). In this way, the concatenated features can make the most of the multi-scale side-output features. The experiments demonstrate that DNA can successfully break through the bottleneck of current linear aggregation. Specifically, we apply DNA into a simply redesigned U-Net without bells and whistles. The
The fused hyper features, called hypercolumn, are then used for final saliency 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 [40, 46], as shown in Figure 1(b). Many saliency detectors are of this type [54, 69, 41, 19, 56, 67, 31, 3]. 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 [61, 39, 38] were first presented for edge detection. Afterwards, similar ideas have been also introduced for saliency detection [5, 18]. 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 Figure 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 Figure 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 [70, 57, 37, 21, 68, 34, 17, 33, 22]. 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 state-of-the-art models have designed very complex feature fusion strategies for this [37, 68].

A full literature review of salient object detection is beyond the scope of this paper. Please refer to [2, 4, 10, 14] for more comprehensive surveys. 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.

3. Revisiting Linear Side-output Aggregation

Deep supervision and corresponding linear side-output prediction aggregation have been demonstrated to be effective in many vision tasks [61, 38, 37, 68]. 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 \( \{O_1, O_2, \ldots, O_N\} \), all of which are supervised by ground-truth maps (Figure 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

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

(1)

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

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

(2)

Similarly, we can compute side-output probability maps \( \{P_1, P_2, \ldots, P_N\} \).

**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

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

(3)

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

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

(4)

If a pixel \( p \) is positive, \( \text{MAE}(\hat{P})_p = |1 - \hat{P}(p)| = 1 - \hat{P}(p) \) and \( 1 - \max(P_i)_p \leq 1 - \hat{P}(p) \leq 1 - \min(P_i)_p \), so that \( \min(\text{MAE}(P_i)_p) \leq \text{MAE}(\hat{P})_p \leq \max(\text{MAE}(P_i)_p) \).

If the pixel \( p \) is negative, \( \text{MAE}(\hat{P})_p = |0 - \hat{P}(p)| = \hat{P}(p) \) and \( \min(P_i)_p \leq \hat{P}(p) \leq \max(P_i)_p \), so that \( \min(\text{MAE}(P_i)_p) \leq \text{MAE}(\hat{P})_p \leq \max(\text{MAE}(P_i)_p) \).

Note that we have \( \|w\|_1 = 1 \) here. Otherwise, \( \hat{P} \) 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

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

(2)

Similarly, we can compute side-output probability maps \( \{P_1, P_2, \ldots, P_N\} \).

**Theorem 2.** The monotonically increasing mapping of \( \sigma(\|w\|_1 \cdot x) \) (\( \|w\|_1 > 0 \)) cannot change the ROC curve and AUC metric.

**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

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

(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

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

(7)

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

as shown in Figure 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

\[\text{AUC is the area under the ROC Curve.}\]
\[(\frac{1}{2}, \frac{1}{2}) \text{ to } (0, 0) \text{ continuously and monotonically. It is also obvious to see that } \{1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t))\} \text{ and } \{F(\varphi^{-1}(t)), G(\varphi^{-1}(t))\} \text{ are symmetric about the point } \left(\frac{1}{2}, \frac{1}{2}\right). \text{ Suppose the area under the curve } \{1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t))\} \text{ is } S_1, \text{ and the area under the curve } \{F(\varphi^{-1}(t)), G(\varphi^{-1}(t))\} \text{ is } S_2. \text{ By symmetry, we have } S_1 + S_2 = 1.

With the above conclusions, we can compute } S_2 \text{ as }
\[
S_2 = \int_{-\infty}^{1} G(\varphi^{-1}(t))dF(\varphi^{-1}(t))
= \int_{-\infty}^{+\infty} G(x)dF(x). \tag{8}
\]

Therefore, } S_2 \text{ is independent of the specific form of the function } \varphi(x), \text{ and } S_1 = 1 - S_2 \text{ is independent of } \varphi(x), \text{ too. Moreover, as } t \text{ ranges in } (0, 1), \text{ thus } \varphi^{-1}(t) \text{ ranges in } \mathbb{R}, \text{ we have}
\[
\{1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t)) : t \in (0, 1)\} = \{F(x), G(x) : x \in \mathbb{R}\} \tag{9}
\]

which is also independent of the form of } \varphi(x). \text{ When } F(x) \text{ and } G(x) \text{ are discrete, the set } \{1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t)) : t \in (0, 1)\} \text{ is discrete but still independent of } \varphi(x). \text{ Therefore, we can conclude that } \varphi(x) \text{ 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, \textit{i.e.}, linear aggregation with } \|w\|_1 = 1, \text{ has limited MAE results. From Theorem 2, we know the second step in Lemma 1, \textit{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 \text{ 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 4.2) for nonlinear regression to evaluate two well-known models: HED \[61\] and DSS \[18\], and the proposed DNA model. The results are summarized in Table 1. We can see significant improvement from linear regression to nonlinear regression. Based on this observation, this paper aims at designing \textit{a simple network with nonlinear side-output aggregation} for effective salient object detection.

4. Methodology

In this section, we will elaborate our proposed framework for salient object detection. We first introduce our base network in Section 4.1. Then, we present the deeply-supervised nonlinear aggregation in Section 4.2. An overall network architecture is illustrated in Figure 3.

4.1. Base Network

\textbf{Backbone net.} \text{To tackle the salient object detection, we follow recent studies \[7, 55, 18\] to use fully convolutional networks. Specifically, we use the well-known VGG16 network \[47\] as our backbone net, whose final fully connected layers are removed to serve for image-to-image translation. Salient object detection usually requires global information to judge which objects are salient \[9\], so enlarging the receptive field of the network would be helpful. To this end, we keep the final pooling layer as in \[18\] and replace the last two fully connected layers with convolution layers, one of which has the kernel size of } \(3 \times 3\) \text{ with } C_0^2 = 192 \text{ channels and another of which has the kernel size of } 7 \times 7 \text{ with } C_0^2 = 128 \text{ channels. Here, we use the } 3 \times 3 \text{ convolution layer to reduce the feature channels, because large kernel sizes (e.g., } 7 \times 7\) \text{ lead to much more parameters.}

There are five pooling layers in the backbone net. 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\} \text{ from bottom to top, respectively. We consider } S_6 \text{ 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 \[18, 61\], the side-output of each convolution block is con-

| Datasets | Metrics | HED [61] lin | DSS [18] lin | DNA lin |
|----------|---------|-------------|-------------|--------|
| DUTS-TE  | F_0    | 0.796       | 0.827       | 0.833  |
|          | MAE    | 0.079       | 0.057       | 0.055  |
|          |        | 0.844       | 0.865       |        |
| ECSSD    | F_0    | 0.892       | 0.911       | 0.915  |
|          | MAE    | 0.065       | 0.053       | 0.056  |
|          |        | 0.921       | 0.935       |        |
| HKU-IS   | F_0    | 0.893       | 0.912       | 0.916  |
|          | MAE    | 0.052       | 0.039       | 0.041  |
|          |        | 0.917       | 0.930       |        |
| DUT-O    | F_0    | 0.726       | 0.752       | 0.774  |
|          | MAE    | 0.100       | 0.078       | 0.066  |
|          |        | 0.765       | 0.799       |        |
| THU15K   | F_0    | 0.757       | 0.775       | 0.770  |
|          | MAE    | 0.099       | 0.083       | 0.074  |
|          |        | 0.785       | 0.793       |        |

The comparison of linear side-output prediction aggregation (\textit{i.e.}, lin) and nonlinear side-output feature aggregation (\textit{i.e.}, nonlin). The datasets and metrics will be introduced in Section 5.1. The linear aggregation of HED \[61\] and DSS \[18\] follows the original papers, and their nonlinear aggregation replaces linear aggregation with the proposed DNA.
Based on the backbone net, we build an encoder-decoder network that can be seen in Figure 3. Concretely, we connect a 1 × 1 convolution layer to each of the convolution blocks \( S^6 \) and \( S^5 \) to prepare for a proper number of channels. 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 similar manner. In this way, the proposed encoder-decoder lets top contextual information flow into the lower layers. Here, both two sequential convolution layers (\( \phi(\cdot) \)) at the decoder side \( S^i \) are with kernel size of \( K_i \times K_i \) and output channels of \( C_i \). 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. We will discuss the parameter settings in detail in the experiment part.

4.2. Deeply-Supervised Nonlinear Aggregation

Instead of linearly aggregating side-output predictions at multiple sides as in previous literature [70, 57, 37, 21, 68, 34, 17], we propose to aggregate the side-output features in a nonlinear way. The proposed DNA module is displayed in the dotted box of Figure 3. Specifically, we adopt a \( 3 \times 3 \) convolution for each \( S^i \) to prepare for proper 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, a \( n \times n \) 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. We use two groups of asymmetric convolutions, each of which consists of a \( 1 \times 7 \) and a \( 7 \times 1 \) convolution. The asymmetric convolutions have \( \frac{2n}{n} = \frac{2}{n} = \frac{2}{7} \) FLOPs (multiply-adds) of standard two-dimensional convolutions. At last, we connect a \( 1 \times 1 \) convolution after the asymmetric convolutions to predict the final output saliency maps.

In the training, we adopt class-balanced cross-entropy loss function [61] to supervise side-output predictions and the final fused prediction. Since each one-dimensional convolution is followed by nonlinear activation (e.g., ReLU), the aggregation of concatenated hybrid features is nonlinear. 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 thus is more effective. With the simple encoder-decoder described in Section 4.1, DNA performs favorably against previous methods. Note that previous methods [70, 57, 37, 21] usually present various network architectures, modules and operations to improve performance, but in this paper, the proposed DNA only applies a simply-modified U-Net as base network.

5. Experiments

5.1. Experimental Setup

Implementation Details. We implement the proposed network using the well-known Caffe [23] framework. The convolution layers contained in the original VGG16 [47] are...
initialized using the publicly available pretrained ImageNet model [11]. 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 bilinear interpolation kernels which are frozen in the training process. 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 \(\text{poly}\), in which the current learning rate equals the base one multiplying \((1 - \frac{\text{curr} \_\text{iter}}{\text{max} \_\text{iter}})^{\text{power}}\). 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 1e-7. The momentum and weight decay are set to 0.9 and 0.0005, respectively. All the experiments in this paper are performed on a TITAN Xp GPU.

Datasets. We extensively evaluate our method on six popular datasets, including DUTS [53], ECSSD [62], SOD [44], HKU-IS [29], THUR15K [8] and DUT-O (or DUT-OMRON) [63]. 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 [53] is a very recent dataset consisting of 10553 training images and 5019 test images in very complex scenarios. For fair comparison, we follow recent studies [56, 37, 55, 64] to use DUTS training set for training and test on the DUTS test set and other datasets.

Evaluation Criteria. We utilize three evaluation metrics to evaluate our method as well as other state-of-the-art salient object detectors, including max F-measure score, mean absolute error (MAE), and weighted \(F_\beta\)-measure score [43]. We follow [41, 18, 68, 69, 37, 31, 5, 37] to use the default settings for these metrics.

5.2. Performance Comparison

We compare our proposed salient object detector with 16 recent state-of-the-art saliency models, including DRFI [24], MDF [29], LEGS [52], DCL [30], DHS [34], ELD [27], RFCN [54], NLDF [41], DSS [18], SRM [55], Amulet [68], UCF [69], BRN [56], PiCA [37], C2S [31] and RAS [5]. Among them, DRFI [24] is the state-of-the-art non-deep-learning based method, and the other 15 models are all based on deep learning. We do not report MDF [29] results on the HKU-IS [29] dataset because MDF uses a part of HKU-IS for training. Due to the same reason, we do not report DHS [34] results on the DUT-O [63] dataset. For fair comparison, all these models are tested using their publicly available code and pretrained models released by the authors with default settings. We also report the results of the ResNet-50 [16] version of our proposed DNA.

Table 3 summarizes the numeric comparison in terms of \(F_\beta\) and MAE on six datasets. DNA can significantly outperform other competitors in most cases, which demonstrates its effectiveness. With VGG16 [47] 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, ECSSD, HKU-IS, DUT-O, SOD and THUR15K datasets, respectively. On the SOD dataset in terms of MAE metric, DNA performs slightly worse than the best result. PiCA [37] seems to achieves the second place overall. With the ResNet-50 backbone, DNA also performs better than previous competitors. DNA has less parameters, i.e., about 20M parameters for VGG16 backbone and 29M parameter for ResNet-50 backbone. DNA also runs faster than other methods, achieving 25fps for VGG16 backbone and 12.8fps for ResNet-50 backbone.

In Table 2, we evaluate DNA using the \(F_\beta\)-measure. The VGG16 version of DNA achieves 4.5%, 3.3%, 2.7%, 3.4%, 3.4% and 2.1% higher \(F_\beta\)-measure than other methods on DUTS-TE, ECSSD, HKU-IS, DUT-O, SOD and THUR15K datasets, respectively. For ResNet-50 version, DNA achieves 5.6%, 1.4%, 2.2%, 2.6%, 1.7% and 1.8% better \(F_\beta\)-measure than other competitors. We provide a qualitative comparison in Figure 4. For objects with various shapes and scales, DNA segments the entire objects with fine details (1-2 rows). DNA is also robust with complicated background (3-5 rows), multiple objects (6-7 rows) and confusing stuff (8 row).
Table 3. Comparison of the proposed DNA and 16 competitors in terms of the metrics of and ResNet-50 [16] backbone. The top three models in each column are highlighted in number of parameters (#Param) is million, and the unit of speed is frame per second (fps). We report results on both VGG16 [47] backbone and ResNet-50 [16] 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.

| Methods       | #Param | Speed | DUTS-TE $F_β$ | ECSSD $F_β$ | HKU-IS $F_β$ | DUT-O $F_β$ | SOD $F_β$ | THUR15K $F_β$ |
|---------------|--------|-------|---------------|-------------|--------------|-------------|-----------|-------------|
|               |        |       | $MAE$         | $MAE$       | $MAE$        | $MAE$       | $MAE$     | $MAE$       |
| DRFI [24]     | -      | 1/8   | 0.649         | 0.154       | 0.777        | 0.161       | 0.774     | 0.146       | 0.652       | 0.138       | 0.704       | 0.217       | 0.670       | 0.150       |
| MDF [29]      | 56.86  | 1/19  | 0.707         | 0.114       | 0.807        | 0.138       | -         | -           | 0.680       | 0.115       | 0.764       | 0.182       | 0.669       | 0.128       |
| LEGS [52]     | 18.40  | 0.6   | 0.652         | 0.137       | 0.830        | 0.118       | 0.766     | 0.119       | 0.668       | 0.134       | 0.733       | 0.194       | 0.663       | 0.126       |
| DCL [30]      | 66.24  | 1.4   | 0.785         | 0.082       | 0.895        | 0.080       | 0.892     | 0.063       | 0.733       | 0.095       | 0.831       | 0.131       | 0.747       | 0.096       |
| DHS [34]      | 94.04  | 10.0  | 0.807         | 0.066       | 0.903        | 0.062       | 0.889     | 0.053       | -           | -           | 0.822       | 0.128       | 0.752       | 0.082       |
| ELD [27]      | 43.09  | 1.0   | 0.727         | 0.092       | 0.866        | 0.081       | 0.837     | 0.074       | 0.700       | 0.092       | 0.758       | 0.154       | 0.726       | 0.095       |
| RFCN [54]     | 134.69 | 0.4   | 0.782         | 0.089       | 0.896        | 0.097       | 0.892     | 0.080       | 0.738       | 0.095       | 0.802       | 0.161       | 0.754       | 0.100       |
| NLDF [41]     | 35.49  | 18.5  | 0.806         | 0.065       | 0.902        | 0.066       | 0.902     | 0.048       | 0.753       | 0.080       | 0.837       | 0.123       | 0.762       | 0.080       |
| DSS [18]      | 62.23  | 7.0   | 0.827         | **0.056**    | 0.915        | **0.056**   | 0.913     | **0.041**   | **0.774**   | **0.066**   | **0.842**   | **0.122**   | **0.770**   | **0.074**   |
| Amulet [68]   | 33.15  | 9.7   | 0.778         | 0.085       | 0.913        | 0.061       | 0.897     | 0.051       | 0.743       | 0.098       | 0.795       | 0.144       | 0.755       | 0.094       |
| UCF [69]      | 23.98  | 12.0  | 0.772         | 0.112       | 0.901        | 0.071       | 0.888     | 0.062       | 0.730       | 0.120       | 0.805       | 0.148       | 0.758       | 0.112       |
| PiCA [37]     | 32.85  | 5.6   | 0.837         | **0.054**    | 0.923        | **0.049**   | 0.916     | **0.042**   | 0.766       | 0.068       | 0.836       | **0.102**   | **0.783**   | **0.083**   |
| C2S [31]      | 137.03 | 16.7  | 0.811         | 0.062       | 0.907        | 0.057       | 0.898     | 0.046       | 0.759       | 0.072       | 0.819       | **0.122**   | **0.775**   | 0.083       |
| RAS [5]       | 20.13  | 20.4  | 0.831         | 0.059       | **0.916**    | 0.058       | **0.913** | 0.045       | **0.785**   | **0.063**   | **0.847**   | 0.123       | **0.772**   | **0.075**   |
| DNA           | 20.06  | 25.0  | 0.865         | 0.044       | 0.935        | 0.041       | 0.930     | 0.031       | 0.799       | 0.056       | 0.853       | 0.107       | **0.793**   | **0.069**   |

ResNet-50 [16] backbone

| Methods       | #Param | Speed | DUTS-TE $F_β$ | ECSSD $F_β$ | HKU-IS $F_β$ | DUT-O $F_β$ | SOD $F_β$ | THUR15K $F_β$ |
|---------------|--------|-------|---------------|-------------|--------------|-------------|-----------|-------------|
|               |        |       | $MAE$         | $MAE$       | $MAE$        | $MAE$       | $MAE$     | $MAE$       |
| SRM [55]      | 43.74  | 12.3  | 0.826         | 0.059       | 0.914        | 0.056       | 0.906     | 0.046       | 0.769       | 0.069       | 0.840       | 0.126       | 0.778       | 0.077       |
| BRN [56]      | 126.35 | 3.6   | 0.827         | 0.050       | 0.919        | 0.043       | 0.910     | 0.036       | 0.774       | 0.062       | 0.843       | **0.103**   | **0.769**   | 0.076       |
| PiCA [37]     | 37.02  | 4.4   | 0.853         | 0.050       | 0.929        | 0.049       | 0.917     | 0.043       | 0.789       | 0.065       | 0.852       | **0.103**   | **0.788**   | 0.081       |
| DNA           | 29.31  | 12.8  | **0.873**     | **0.040**    | **0.938**    | **0.040**   | **0.934** | **0.029**   | **0.805**   | **0.056**   | **0.855**   | 0.110       | **0.796**   | **0.068**   |

5.3. 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 [61] to obtain a deeply-supervised encoder-decoder, i.e., $ED$ w/ $lin$. The results are shown in Table 5. We can clearly see that nonlinear aggregation performs significantly better than linear aggregation, in terms of both $F_β$ and MAE.

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.
The asymmetric convolutions in DNA module. In Table 7, we evaluate various convolution kernel sizes for the DNA model. Large convolution kernel sizes, e.g., $7 \times 7$, can achieve better performance. However, $7 \times 7$ convolution is very time-consuming as shown in Table 7, because the feature maps in DNA is with the same resolution as original images. Hence, we use asymmetric convolutions to achieve both large kernel size and fast speed.

6. Conclusion

Previous deeply-supervised saliency detection networks use linear side-output prediction aggregation. We theoretically and experimentally demonstrate that linear side-output aggregation is suboptimal and worse than nonlinear aggregation. Based on this observation, we propose the DNA module that aggregates multi-level side-output features in a nonlinear way. 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 computer vision tasks that need multi-scale and multi-level information.

Table 4. Ablation studies for various parameter settings. The unit of the number of parameters (#Param) is million, and the unit of speed is frame per second (fps).

| Methods | #Param | Speed | DUTS-TE | ECSSD | HKU-IS | DUT-O | SOD | THUR15K |
|---------|--------|-------|---------|-------|--------|-------|-----|---------|
| #1      | 18.49  | 27.0  | 0.859   | 0.932 | 0.041  | 0.928 | 0.931| 0.790  |
| #2      | 20.06  | 25.0  | 0.865   | 0.935 | 0.041  | 0.930 | 0.931| 0.790  |
| #3      | 27.88  | 22.7  | 0.866   | 0.936 | 0.041  | 0.930 | 0.931| 0.790  |
| #4      | 41.41  | 18.2  | 0.864   | 0.935 | 0.041  | 0.931 | 0.931| 0.790  |

Table 5. Ablation studies. U-Net means the standard U-Net [46] with VGG16 backbone. If removing the DNA module and deep supervision, the proposed network in Figure 3 becomes an encoder-decoder network that is called ED. ED w/ K3 replaces all the convolutions at the top sides of ED with $3 \times 3$ convolutions. ED w/ lin means we replace the DNA module in Figure 3 with traditional linear aggregation in [61].

Net [46]. First, we change the kernel size of all convolutions at top sides, i.e., $K_3 \times K_3$, $K_3 \times K_4$ and $K_5 \times K_5$, into $3 \times 3$. As displayed in Table 5, the resulting model, ED w/ K3, perform worse than the standard U-Net [46]. This could be because the proposed encoder-decoder has less feature channels and thus less parameters (U-Net has 31.06M parameters). Nest, we use the default kernel size of $5 \times 5$ for top sides. The resulting model, ED, performs better than U-Net. This demonstrates large kernel size at the top sides is important for better performance.

Encoder-decoder with or without deep supervision. In Table 5, we can also find deep supervision can consistently improve the encoder-decoder networks.

Parameter settings. To evaluate the effect of different parameter settings, we try various parameter settings in Table 6. The results are summarized in Table 4. 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 5 where deep supervision is not used. From the third and fourth experiments, we find that introducing more parameters by increasing the convolution channels can generate slightly better results. Considering the trade-off between performance, number of parameters and speed, we choose the second setting as our default parameters.
References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *IEEE CVPR*, pages 1597–1604, 2009. 1, 2

[2] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE TIP*, 12(24):5706–5722, 2015. 2

[3] N. D. Bruce, C. Catton, and S. Janjic. A deeper look at saliency: Feature contrast, semantics, and beyond. In *IEEE CVPR*, pages 516–524, 2016. 2

[4] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *IEEE TPAMI*, 2018. 2

[5] S. Chen, X. Tan, B. Wang, and X. Hu. Reverse attention for salient object detection. In *ECCV*, 2018. 1, 2, 6, 7

[6] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li. DISC: Deep image saliency computing via progressive representation learning. *IEEE TNNLS*, 27(6):1135–1149, 2016. 2

[7] X. Chen, A. Zheng, J. Li, and F. Lu. Look, perceive and segment: Finding the salient objects in images via two-stream fixation-semantic CNNs. In *IEEE ICCV*, pages 1050–1058, 2017. 2, 4

[8] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu. Salientshape: Group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014. 6

[9] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015. 1, 2, 4

[10] R. Cong, J. Lei, H. Fu, M.-M. Cheng, W. Lin, and Q. Huang. Review of visual saliency detection with comprehensive information. *IEEE TCSVT*, 2018. 2

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255, 2009. 6

[12] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu. Visual-textual joint relevance learning for tag-based social image search. *IEEE TIP*, 22(1):363–376, 2013. 1

[13] C. Gong, D. Tao, W. Liu, S. J. Maybank, M. Fang, K. Fu, and J. Yang. Saliency propagation from simple to difficult. In *IEEE CVPR*, pages 2531–2539, 2015. 2

[14] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine*, 35(1):84–100, 2018. 2

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *IEEE CVPR*, pages 447–456, 2015. 2

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016. 3, 6, 7

[17] S. He, J. Jiao, X. Zhang, G. Han, and R. W. Lau. Delving into salient object subitizing and detection. In *IEEE ICCV*, pages 1059–1067, 2017. 1, 2, 5

[18] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2019. 1, 2, 4, 6, 7

[19] P. Hu, B. Shuai, J. Liu, and G. Wang. Deep level sets for salient object detection. In *IEEE CVPR*, pages 2300–2309, 2017. 2

[20] X. Hu, Y. Liu, K. Wang, and B. Ren. Learning hybrid convolutional features for edge detection. *Neurocomputing*, 313:377–385, 2018. 2

[21] M. A. Islam, M. Kalash, and N. D. Bruce. Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects. In *IEEE CVPR*, pages 7142–7150, 2018. 1, 2, 5

[22] S. Jia and N. D. Bruce. Richer and deeper supervision network for salient object detection. *arXiv preprint arXiv:1901.02425*, 2019. 1, 2

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678, 2014. 5

[24] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *IEEE CVPR*, pages 2083–2090, 2013. 1, 2, 6, 7

[25] Z. Jiang and L. S. Davis. Submodular salient region detection. In *IEEE CVPR*, pages 2043–2050, 2013. 2

[26] C. Lang, J. Feng, S. Feng, J. Wang, and S. Yan. Dual low-rank pursuit: Learning salient features for saliency detection. *IEEE TNNLS*, 27(6):1190–1200, 2016. 2

[27] G. Lee, Y.-W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *IEEE CVPR*, pages 660–668, 2016. 1, 2, 6, 7

[28] G. Li, Y. Xie, L. Lin, and Y. Yu. Instance-level salient object segmentation. In *IEEE CVPR*, pages 247–256, 2017. 1

[29] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *IEEE CVPR*, pages 5455–5463, 2015. 2, 6, 7

[30] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *IEEE CVPR*, pages 478–487, 2016. 2, 6, 7

[31] X. Li, F. Yang, H. Cheng, W. Liu, and D. Shen. Contour knowledge transfer for salient object detection. In *ECCV*, pages 355–370, 2018. 1, 2, 6, 7

[32] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. DeepSaliency: Multi-task deep neural network model for salient object detection. *IEEE TIP*, 25(8):3919–3930, 2016. 2

[33] Z. Li, C. Lang, Y. Chen, J. Liew, and J. Feng. Deep reasoning with multi-scale context for salient object detection. *arXiv preprint arXiv:1901.08362*, 2019. 1, 2

[34] N. Liu and J. Han. DHSNet: Deep hierarchical saliency network for salient object detection. In *IEEE CVPR*, pages 678–686, 2016. 1, 2, 5, 6, 7

[35] N. Liu and J. Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *IEEE TIP*, 27(7):3264–3274, 2018. 2

[36] N. Liu, J. Han, T. Liu, and X. Li. Learning to predict eye fixations via multiresolution convolutional neural networks. *IEEE TNNLS*, 29(2):392–404, 2018. 2

[37] N. Liu, J. Han, and M.-H. Yang. PiCaNet: Learning pixel-wise contextual attention for saliency detection. In *IEEE CVPR*, pages 3089–3098, 2018. 1, 2, 3, 5, 6, 7
[38] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang. Richer convolutional features for edge detection. *IEEE TPAMI*, 2019. 2

[39] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai. Richer convolutional features for edge detection. In *IEEE CVPR*, pages 5872–5881, 2017. 2

[40] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE CVPR*, pages 3431–3440, 2015. 1, 2

[41] Z. Luo, A. K. Mishra, A. Achkar, J. A. Eichel, S. Li, and P.-M. Jodoin. Non-local deep features for salient object detection. In *IEEE CVPR*, pages 6609–6617, 2017. 1, 2, 6, 7

[42] V. Mahadevan and N. Vasconcelos. Saliency-based discriminant tracking. In *IEEE CVPR*, 2009. 1

[43] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps? In *IEEE CVPR*, pages 248–255, 2014. 6

[44] V. Movahedi and J. H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 49–56, 2010. 6

[45] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang. Region-based saliency detection and its application in object recognition. *IEEE TCSVT*, 24(5):769–779, 2014. 1

[46] O. Ronneberger, P. Fischer, and T. Brox. U-Net: convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 1, 2, 8

[47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 4, 5, 6, 7

[48] J. Su, J. Li, C. Xia, and Y. Tian. Selectivity or invariance: Boundary-aware salient object detection. *arXiv preprint arXiv:1812.10066*, 2018. 2

[49] H. R. Tavakoli, F. Ahmed, A. Borji, and J. Laaksonen. Saliency revisited: Analysis of mouse movements versus fixations. In *IEEE CVPR*, pages 6354–6362, 2017. 2

[50] N. Tong, H. Lu, X. Ruan, and M.-H. Yang. Saliency object detection via bootstrap learning. In *IEEE CVPR*, pages 1884–1892, 2015. 1

[51] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien. Real-time salient object detection with a minimum spanning tree. In *IEEE CVPR*, pages 2342–2349, 2016. 2

[52] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *IEEE CVPR*, pages 3183–3192, 2015. 2, 6, 7

[53] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan. Learning to detect salient objects with image-level supervision. In *IEEE CVPR*, pages 136–145, 2017. 6

[54] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016. 2, 6, 7

[55] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu. A stage-wise refinement model for detecting salient objects in images. In *IEEE ICCV*, pages 4019–4028, 2017. 1, 2, 4, 6, 7

[56] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji. Detect globally, refine locally: A novel approach to saliency detection. In *IEEE CVPR*, pages 3127–3135, 2018. 1, 2, 6, 7

[57] W. Wang, J. Shen, X. Dong, and A. Borji. Salient object detection driven by fixation prediction. In *IEEE CVPR*, pages 1711–1720, 2018. 1, 2, 5

[58] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE CVPR*, pages 1568–1576, 2017. 1

[59] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang. Revisiting dilated convolution: A simple approach for weakly and semi-supervised semantic segmentation. In *IEEE CVPR*, pages 7268–7277, 2018. 1

[60] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang. What is and what is not a salient object? learning salient object detector by ensembling linear exemplar regressors. In *IEEE CVPR*, pages 4321–4329, 2017. 2

[61] S. Xie and Z. Tu. Holistically-nested edge detection. *ICCV*, 125(1-3):3–18, 2017. 1, 2, 3, 4, 5, 7, 8

[62] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *IEEE CVPR*, pages 1155–1162, 2013. 6

[63] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *IEEE CVPR*, pages 3166–3173, 2013. 2, 6

[64] Y. Zeng, H. Lu, L. Zhang, M. Feng, and A. Borji. Learning to promote saliency detectors. In *IEEE CVPR*, pages 1644–1653, 2018. 1, 2, 6

[65] D. Zhang, J. Han, and Y. Zhang. Supervision by fusion: Towards unsupervised learning of deep salient object detector. In *IEEE ICCV*, pages 4048–4056, 2017. 2

[66] J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *IEEE CVPR*, pages 9029–9038, 2018. 2

[67] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang. A bi-directional message passing model for salient object detection. In *IEEE CVPR*, pages 1741–1750, 2018. 1, 2

[68] P. Zhang, D. Wang, H. Lu, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *IEEE ICCV*, pages 202–211, 2017. 1, 2, 3, 5, 6, 7

[69] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin. Learning uncertain convolutional features for accurate saliency detection. In *IEEE ICCV*, pages 212–221, 2017. 2, 6, 7

[70] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang. Progressive attention guided recurrent network for salient object detection. In *IEEE CVPR*, pages 714–722, 2018. 1, 2, 5

[71] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *IEEE CVPR*, pages 1265–1274, 2015. 2

[72] L. Zhu, H. Ling, J. Wu, H. Deng, and J. Liu. Saliency pattern detection by ranking structured trees. In *IEEE ICCV*, pages 5467–5476, 2017. 1

[73] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *IEEE CVPR*, pages 2814–2821, 2014. 2

[74] F. Zund, Y. Pritch, A. Sorkine-Hornung, S. Mangold, and T. Gross. Content-aware compression using saliency-driven image retargeting. In *ICIP*, pages 1845–1849, 2013. 1