Revisiting Image Pyramid Structure for High Resolution Salient Object Detection

Taehun Kim[1][0000−0001−9322−9741], Kunhee Kim[1], Joonyeong Lee[1], Dongmin Cha[1], Jiho Lee[1], and Daijin Kim[1]

[1]Dept. of CSE, Pohang University of Science and Technology (POSTECH), Korea
{taehoon1018, kunkim, joonyeonglee, cardongmin, jiholee,
dkim}@postech.ac.kr
https://github.com/plemeri/InSPyReNet.git

Abstract. Salient object detection (SOD) has been in the spotlight recently, yet has been studied less for high-resolution (HR) images. Unfortunately, HR images and their pixel-level annotations are certainly more labor-intensive and time-consuming compared to low-resolution (LR) images and annotations. Therefore, we propose an image pyramid-based SOD framework, Inverse Saliency Pyramid Reconstruction Network (InSPyReNet), for HR prediction without any of HR datasets. We design InSPyReNet to produce a strict image pyramid structure of saliency map, which enables to ensemble multiple results with pyramid-based image blending. For HR prediction, we design a pyramid blending method which synthesizes two different image pyramids from a pair of LR and HR scale from the same image to overcome effective receptive field (ERF) discrepancy. Our extensive evaluations on public LR and HR SOD benchmarks demonstrate that InSPyReNet surpasses the *State-of-the-Art* (SotA) methods on various SOD metrics and boundary accuracy.

1 Introduction

While there are many successful works for SOD in low-resolution (LR) images, there are many demands on high-resolution (HR) images. One can argue that methods trained with LR datasets produce decent results on HR images by resizing the input size (Fig. 1a), but the quality in terms of the high-frequency details of prediction still remains poor in that way. Moreover, previous studies on HR prediction have been working on developing complex architectures and proposing laborious annotations on HR images [1–4] (Fig. 1b, c).

In this paper, we focus on only using LR datasets for training to produce high-quality HR prediction. To do so, we mainly focus on the structure of saliency prediction, which enables to provide high-frequency details from the image regardless of the size of the input. However, there is still another problem to be solved where the effective receptive fields (ERFs) [5] of HR images are different from the LR images in most cases. To alleviate the aforementioned issues, we propose two solid solutions which are mutually connected to each other.
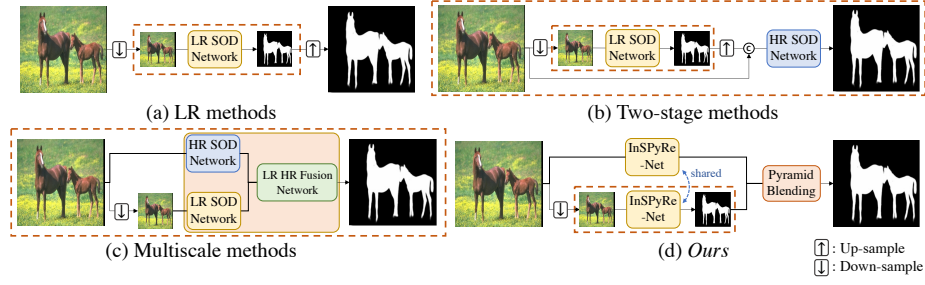
Fig. 1. Different approaches for HR SOD prediction. Areas denoted as a dashed box are trained with supervision. (a): Resizing HR input to LR, then up-sample. Works for any methods, lack of details. (b): Requires multiple training sessions, and HR datasets [1,2]. (c): Can overcome ERF discrepancy, but the architecture is complex, requires HR datasets [4]. (d): Works without HR dataset training. We predict multiscale results with single network and synthesize HR prediction with pyramid blending.

First is to design a network architecture which enables to merge multiple results regardless of the size of the input. Therefore, we propose Inverse Saliency Pyramid Reconstruction Network (InSPyReNet), which predicts the image pyramid of the saliency map. Image pyramid is a simple yet straightforward method for image blending [6], so we design InSPyReNet to produce the image pyramid of the saliency map directly. Previous works have already used image pyramid prediction, but results did not strictly follow the structure, and hence unable to use for the blending (Fig. 2). Therefore, we suggest new architecture, and new supervision techniques to ensure the image pyramid structure which enables stable image blending for HR prediction.

Second, to solve the problem of ERF discrepancy between LR and HR images, we design a pyramid blending technique for the inference time to overlap two image pyramids of saliency maps from different scales. Recent studies of HR SOD methods use two different scales of the same image, by resizing HR image to LR, to alleviate such problem [3,4], but the network should be complicated and large (Fig. 1c). Simply forwarding HR images to the InSPyReNet, or other LR SOD networks fail to predict salient region since they are not trained with HR images. Nevertheless, we notice the potential of enhancing details for high-quality details from HR prediction, even the result shows a lot of False Positives (HR prediction in Fig. 3). To combine the robust saliency prediction and details from LR and HR predictions, we blend the two image pyramids of saliency maps.

InSPyReNet *does not require HR training and datasets*, yet produces high-quality results on HR benchmarks. A series of quantitative and qualitative results on HR and LR SOD benchmarks show that our method shows SotA performance, yet more efficient than previous HR SOD methods in terms of training resources, annotation quality, and architecture engineering.
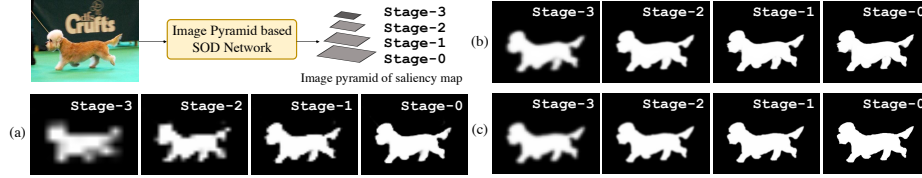
Fig. 2. Comparison of image pyramid based saliency map between (a) Chen *et al.* [7], (b) InSPyReNet, and image pyramid of (c) ground truth. Compared to the image pyramid of ground truth saliency map, Chen *et al.* shows distorted results especially for the higher stages (*e.g.*, `Stage-3`). However, our InSPyReNet shows almost identical results compared to the ground truth across each stage.

2   Related Works

Salient Object Detection. Edge-Based Models are studied in SOD for better understanding of the structure of the salient object by explicitly modeling the contour of the saliency map. Methods with auxiliary edge estimator require additional edge GT, or extra training process with extra edge datasets. For instance, EGNet [8] has an additional edge estimation branch which is supervised with additional edge-only dataset. However, the effect of edge branch is limited to the encoder network (backbone), expecting better representation with robust edge information, because the estimated edge from the edge branch is not directly used to the detection. Also, LDF [9] designed an alternative representation for the edge information. They divided the saliency map into 'body' and 'detail', which corresponds to the edge part. Unlike EGNet, they utilized both 'body' and 'detail' for the saliency prediction in the inference stage. However, to achieve the disentanglement of 'body' and 'detail' components, it requires multiple training stages and ground truth generation.

Unlike auxiliary edge models, we embedded the image pyramid structure to the network for saliency prediction, which does not require additional training process nor extra datasets, and the decoder network is implicitly trained to predict the Laplacian of the saliency map, high-frequency details of the larger scales, which implicitly includes edge information. Thanks to this simple structure, we also do not require additional training stages.

Image Segmentation for HR Images. Pixel-wise prediction tasks such as SOD resize input images into a pre-defined shape (*e.g.*, 384×384) for batched and memory efficient training. This is plausible since the average resolution of training datasets are usually around 300 to 400 for both width and height. For example, the average resolution is 378×469 for ImageNet [10], and 322×372 for DUTS [11]. After training, resizing input images into a pre-defined shape is often required, especially when the input image is relatively larger than the pre-defined shape (Fig. 1a). However, down-sampling large images causes severe information loss, particularly for high-frequency details. We can overcome this problem by not resizing images, but current SotA SOD methods fail to predict
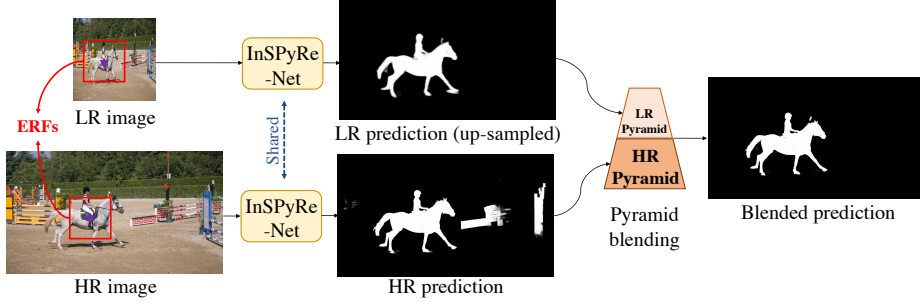
Fig. 3. Illustration of effective receptive field (ERF) [5] discrepancy between LR and HR images from InSPyReNet. LR prediction shows successful saliency prediction, but lack of details. While HR prediction shows better details but due to the ERF discrepancy (red boxes), it over detects objects. With pyramid blending, we can capture the global dependency from LR prediction while enhance local details from HR prediction at the same time. *Best viewed by zooming in.*

appropriate saliency map because they are neither trained with HR dataset nor designed to produce HR prediction. Most likely, the problem is the discrepancy between the effective receptive fields [5] of the same corresponding pixel from the original and resized images (Fig. 3).

CascadePSP [12] first tackled this problem in semantic segmentation by approaching HR segmentation by a refinement process. They trained their model with coarse segmentation masks as an input with a set of augmentation techniques, and used the model to refine an initial segmentation mask with multiple global steps and local steps in a recursive manner. However, they need an initial prediction mask from standalone models [13,14], which is definitely not resource-frendly. Zeng *et al.* [1] first proposed HR dataset for SOD task with a baseline model which consists of separate LR, HR and fusion networks. They combined global (GLFN) and local (LRN) information by two separate networks dedicated for each of them. Tang *et al.* [2] also designed LR and HR networks separately, where the branch for the HR (HRRN) gets an image and a predicted saliency map from the LR branch (LRSCN). PGNet [4] first proposed a standalone, end-to-end network for HR prediction by combining features from LR and HR images with multiple backbone networks.

Aforementioned methods require HR datasets for training, complex model architecture, multiple training sessions for submodules (Fig. 1b, c). Unlike previous methods, InSPyReNet does not require HR datasets for training, yet predicts fine details especially on object boundary.

Image Pyramid in Deep Learning Era. Studies of pixel-level prediction tasks have shown successful application of image pyramid prediction. LapSRN [15] first applied a Laplacian image prediction for Super Resolution task and since then, most end-to-end supervised super resolution methods adopt their structure. LRR [16] first applied a Laplacian image pyramid for the semantic seg-

mentation task in the prediction reconstruction process. Then, Chen *et al.* [7] adopted LRR prediction strategy for the SOD with reverse attention mechanism, and UACANet [17] extended self-attention mechanism with uncertainty area for the polyp segmentation. As the above methods have already proved that without any training strategy, we can expect the network to implicitly predict the image pyramid by designing the architecture. However, without extra regularization strategy for the supervision to follow the image pyramid structure rigorously, we cannot make sure that the Laplacian images from each stage truly contains high-frequency detail (Fig. 2).

We revisit this image pyramid scheme for prediction, and improve the performance by setting optimal stage design for image pyramid and regularization methods to follow pyramidal structure. Also, to the best of our knowledge, In-SPyReNet is the *first attempt to extend image pyramid prediction for multiple prediction ensembling by image blending technique.* This is because previous methods' Laplacian images did not strictly follow actual high-frequency detail. Rather, they focus more on correcting errors from the higher stages (Fig. 2). Unlike previous methods, we adopt scale-wise supervision (Fig. 4b), Stop-Gradient and pyramidal consistently loss (Sec. 3.2) for regularization which enables consistent prediction, and hence we are able to use blending technique by utilizing multiple results to facilitate more accurate results on HR benchmarks.

3 Methodology

3.1 Model Architecture

Overall Architecture. We use Res2Net [18] or Swin Transformer [19] for the backbone network, but for HR prediction, we only use Swin as a backbone. We provide a thorough discussion (Sec. 5) for the reason why we use only Swin Transformer for HR prediction.

From UACANet [17], we use Parallel Axial Attention encoder (PAA-e) for the multiscale encoder to reduce the number of channels of backbone feature maps and Parallel Axial Attention decoder (PAA-d) to predict an initial saliency map on the smallest stage (*i.e.*, Stage-3). We adopt both modules because they capture global context with non-local operation, and it is efficient thanks to the axial attention mechanism [20,21].

Refer to the stage design in Fig. 4, previous pyramid-based methods for pixel-level prediction [7,16] started with Stage-5, and ended at Stage-2. However, there are still two remaining stages to reconstruct for previous methods, which makes the reconstruction process incomplete in terms of the boundary quality. Thus, we claim that starting image pyramid from Stage-3 is sufficient, and should reconstruct until we encounter the lowest stage, Stage-0 for HR results. To recover the scale of non-existing stages (Stage-1, Stage-0), we use bi-linear interpolation in appropriate locations (Fig. 4).

We locate a self-attention-based decoder, Scale Invariant Context Attention (SICA), on each stage to predict a Laplacian image of the saliency map (Lapla-
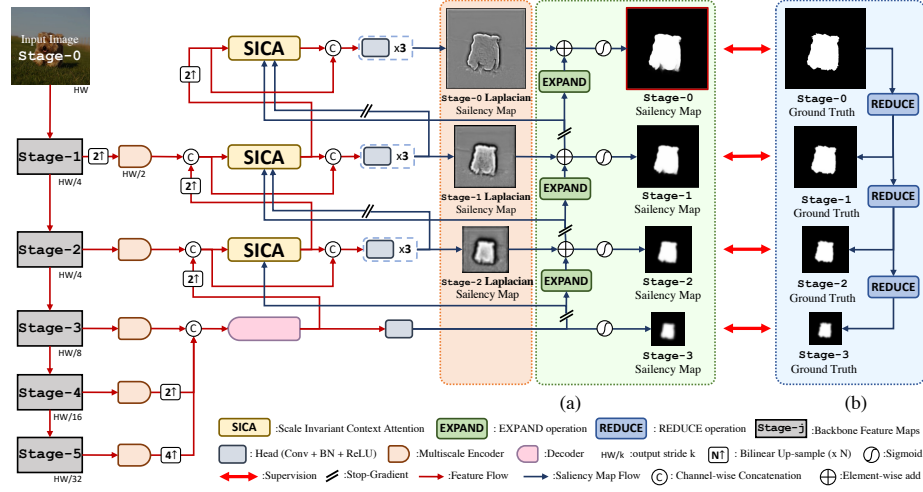
Fig. 4. The architecture of proposed InSPyReNet. (a) The initial saliency map from `Stage-3` and Laplacian saliency maps from higher-stages are combined with EXPAND operation to be reconstructed to the original input size. (b) The ground-truth is deconstructed to the smaller stages for predicted saliency maps from each stage by REDUCE operation.

cian saliency map). From the predicted Laplacian saliency maps, we reconstruct saliency maps from higher-stages to the lower-stages (Fig. 4a).

Scale Invariant Context Attention. Attention-based decoder for pixel-wise prediction shows great performance due to its non-local operation with respect to the spatial dimension [22,23]. However, when the size of the input image gets larger than the training setting (*e.g.*, 384×384), it usually fails to produce an appropriate result for the following reason. Because as the size of input image is large enough, there exist a train-inference discrepancies for a non-local operation which flattens the feature map according to the spatial dimension and does a matrix multiplication. For instance, the magnitude of the result from the non-local operation varies depending on the spatial dimension of the input image. Moreover, the complexity of non-local operation increases quadratically as the input size increases.

To this end, we propose SICA, a scale invariant context attention module for robust Laplacian saliency prediction. As shown in Fig. 5, the overall operation of SICA follows OCRNet [23]. We found that computing object region representation causes train-inference discrepancy, so we resize input feature maps $\mathbf{x}$ and context maps $\mathbf{c}$ according to the shape from training time (h, w). Because in the training step, images are already reshaped to the fixed shape, we do not have to resize them. For context maps, unlike OCRNet, we can only access to the saliency map which is insufficient, so we generate several context maps following [17]. Further details of the context map selection and equations may be found in the supplementary material. With SICA, we can compute Laplacian saliency
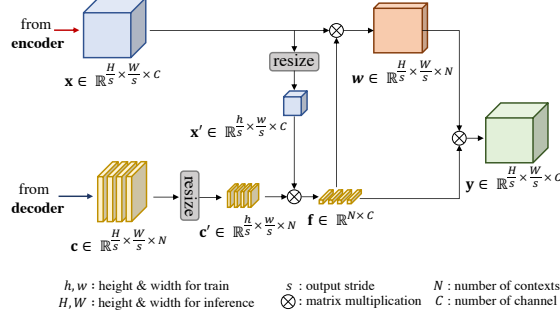
Fig. 5. Illustration of Scale Invariant Context Attention (SICA)

maps more precisely for HR images, and hence can apply pyramid blending for HR prediction (Sec. 3.3).

Inverse Saliency Pyramid Reconstruction. Laplacian pyramid [24] is an image compression technique that stores the difference between the low-pass filtered image and the original image for each scale. We can interpret the Laplacian image as a remainder from the low-pass filtered signal or, in other words, high-frequency details. Inspired by this technique, we revisit the image pyramid structure by designing our network to construct a Laplacian pyramid to concentrate on the boundary details and reconstruct the saliency map from the smallest stage to its original size. We start with the saliency map from the uppermost stage (`Stage-3`) for the initial saliency map and aggregate high-frequency details from the Laplacian saliency maps.

Formally, we denote the saliency map and Laplacian saliency map of the jth stage as S^j and U^j, respectively. To reconstruct the saliency map from the $j + 1$th stage to the jth stage, we apply EXPAND operation [24] as follows,

$$S_e^j(x,y) = 4 \sum_{m=-3}^{3} \sum_{n=-3}^{3} g(m,n) \cdot S^{j+1}\left(\frac{x-m}{2}, \frac{y-n}{2}\right) \qquad (1)$$

where $(x,y) \in \mathcal{I}^j$ are pixel coordinates and $\mathcal{I}^j$ is a lattice domain of `Stage-j`. Also, $g(m,n)$ is a Gaussian filter where the kernel size and standard deviation are empirically set to 7 and 1 respectively. To restore the saliency details, we add Laplacian saliency map from SICA as follows,

$$S^j = S_e^j + U^j. \qquad (2)$$

We repeat this process until we obtain the lowest stage, `Stage-0` as shown in Fig. 4a, and use it as a final prediction.

3.2 Supervision Strategy and Loss Functions

A typical way to supervise a network with multi-stage side outputs is to use bi-linear interpolation for each stage's prediction and compute the loss function

with the ground-truth. However, the predicted saliency map from higher-stage is small regarding its spatial dimension, and this may cause stage-scale inconsistency, especially for the boundary area of salient objects. Instead, we focus on *"Do what you can with what you have where you are"*. In fact, the saliency output from `Stage-3` cannot physically surpass the details from `Stage-2`, so we choose to provide each stage a suitable ground-truth. To do so, we create an image pyramid of the ground-truth (Fig. 4b).

First, we obtain the ground-truth G^j for `Stage-j` from G^{j-1} with REDUCE operation [24] as follows,

$$G^j(x,y) = \sum_{m=-3}^{3} \sum_{n=-3}^{3} g(m,n) \cdot G^{j-1}(2x+m, 2y+n). \tag{3}$$

From the largest scale, we deconstruct the ground-truth until we get ground-truths for each stage of our network.

For loss function, we utilize binary cross entropy (BCE) loss with pixel position aware weighting strategy $\mathcal{L}^{wbce}$ [25]. Moreover, to encourage the generated Laplacian saliency maps to follow the pyramid structure, we deconstruct S^{j-1} to the jth stage, $\tilde{S}^j$ by REDUCE operation. Then, we reinforce the similarity between S^j and reduced saliency map $\tilde{S}^j$ with pyramidal consistency loss $\mathcal{L}^{pc}$ as follows,

$$\mathcal{L}^{pc}(S^j, \tilde{S}^j) = \sum_{(x,y) \in \mathcal{I}^j} ||S^j(x,y) - \tilde{S}^j(x,y)||_1. \tag{4}$$

$\mathcal{L}^{pc}$ regularizes the lower-stage saliency maps to follow the structure of the image pyramid through the training process. We define the total loss function $\mathcal{L}$ as follows,

$$\mathcal{L}(S,G) = \sum_{j=0}^{3} \lambda_j \mathcal{L}^{wbce}(S^j, G^j) + \eta \sum_{j=0}^{2} \lambda_j \mathcal{L}^{pc}(S^j, \tilde{S}^j) \tag{5}$$

where η is set to 10^{-4} and $\lambda_j = 4^j$ for balancing the magnitude of loss across stages.

Finally, we include Stop-Gradient for the saliency map input of SICA and reconstruction process from higher-stages to force each stage saliency output to focus on each scale during training time and only affect each other in the inference time (Fig. 4). This strategy encourages the stage-wise ground-truth scheme by explicitly preventing the gradient flow from lower-stages affecting the higher-stages. Thus, supervisions with high-frequency details will not affect higher-stage decoder, which are intended only to have the abstract shape of the salient objects. While this strategy might affect the performance in terms of the multiscale scheme, we use feature maps from the different stages for multiscale encoder and SICA to compensate for this issue.

3.3 Pyramid Blending

While SICA enables saliency prediction for various image sizes, when the image gets larger, there still exists ERF discrepancies (Fig. 3). Thankfully, one very
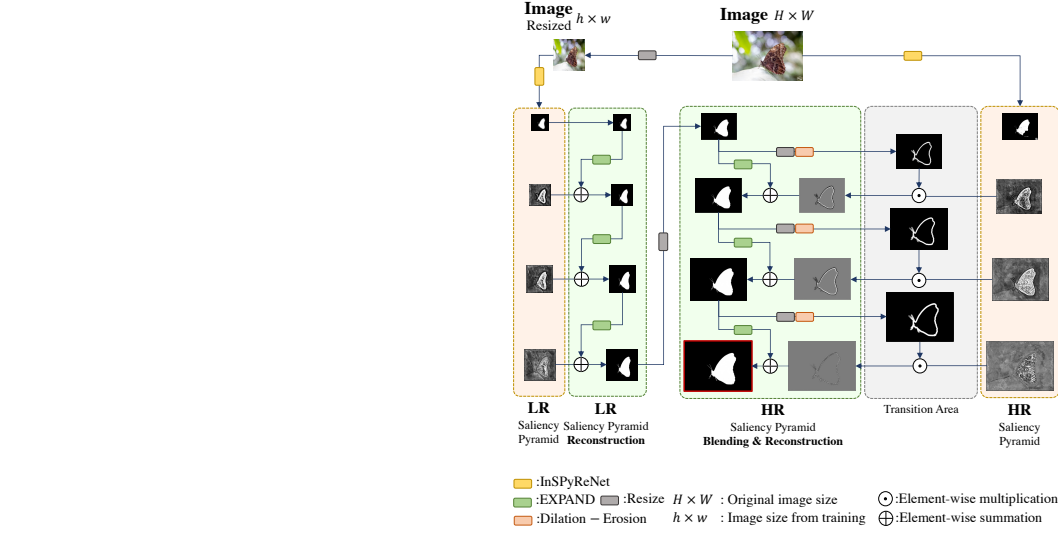
Fig. 6. Illustration of pyramid blending of InSPyReNet for HR prediction.

straightforward application for our saliency pyramid outputs is assembling multiple saliency pyramids from different inputs. We first generate saliency pyramids with InSPyReNet for original and resized images as shown in Fig. 6, namely LR and HR saliency pyramids. Then, instead of reconstructing the saliency map from the HR pyramid, we start from the lowest stage of the LR pyramid. Intuitively speaking, the LR pyramid is extended with the HR pyramid, so they construct a 7 stage saliency pyramid.

For the HR pyramid reconstruction, similar to [16], we compute the dilation and erosion operation to the previous stage's saliency map and subtract them to obtain the transition area for and multiply with the Laplacian saliency map. Transition area is used to filter out the unwanted noises from the HR pyramid, since the boundary details we need to apply should exist only around the boundary area. Unlike [16], it is unnecessary for the LR branch since we train InSPyReNet with methods in Sec. 3.2, results in the saliency pyramid are guaranteed to be consistent.

4   Experiments and Results

4.1   Experimental Settings

Implementation Details. We train our method with widely used DUTS-TR, a subset of DUTS [11] for training. We use Res2Net [18] or Swin Transformer [19] backbone which is pre-trained with ImageNet-1K or ImageNet-22K [10] respectively. Images are resized to 384×384 for training, and we use a random scale in a range of [0.75, 1.25] and crop to the original size, random rotation from -10
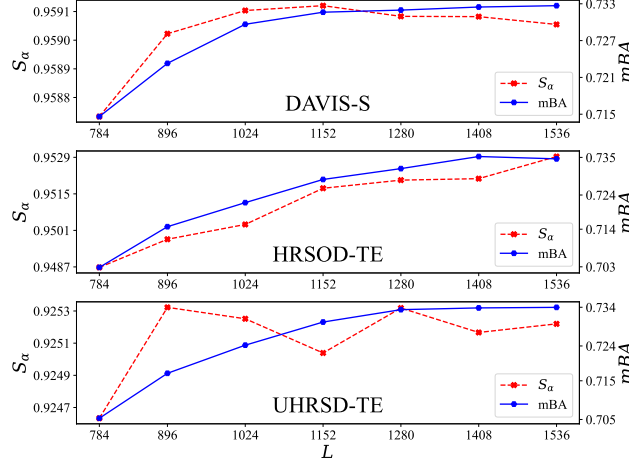
Fig. 7. Performance measure (S_α and mBA) of InSPyReNet with pyramid blending by changing L on three HR benchmarks.

to 10 degrees, and random image enhancement (contrast, sharpness, brightness) for the data augmentation. We set the batch size to 6 and maximum epochs to 60. We use Adam optimizer [26] with initial learning rate 1e-5, and follow the default PyTorch settings. Finally, we use poly learning rate decay for scheduling [13] with a factor of $(1 - (\frac{iter}{iter_{\max}})^{0.9})$ and linear warm-up for the first 12000 iterations.

Evaluation Datasets and Metrics. We evaluate our method on five LR benchmarks, DUTS-TE, a subset of DUTS for evaluation, DUT-OMRON [27], ECSSD [28], HKU-IS [29], and PASCAL-S [30]. Furthermore, we evaluate our method on three HR benchmarks, DAVIS-S [31], HRSOD-TE [1], and UHRSD-TE [4]. From [32], we report S-measure (S_α) [33], maximum F-measure ($F_{\max}$) [34], and Mean Absolute Error (MAE) [35]. Since F-measure requires a binary map, it is computed with thresholds in a range of [0, 255] and the maximum value is used for the evaluation. With the above metrics, we also report mean boundary accuracy (mBA) [36] for boundary quality measure.

4.2 Ablation Studies

Resizing Factor L. We use the resizing method for HR images from [36] for pyramid blending since current GPUs cannot deal with large sizes such as 4K images as is. So, we choose a maximum length of the shorter side of the image as L. For instance, if an input size is 1920×1080 and $L = 810$, then we resize the input into 1440×810. Moreover, we do not deploy pyramid blending process for inputs where the shorter side length is less than 512 because the difference between LR and HR pyramid is not enough for blending. We compare S_α and

Table 1. Ablation study of InSPyReNet (SwinB) with and without SICA and pyramid blending on HR benchmarks.

Resolution	Setting	DAVIS-S				HRSOD-TE				UHRSD-TE			
		$S_\alpha \uparrow$	$F_{\max} \uparrow$	MAE↓	mBA↑	$S_\alpha \uparrow$	$F_{\max} \uparrow$	MAE↓	mBA↑	$S_\alpha \uparrow$	$F_{\max} \uparrow$	MAE↓	mBA↑
w/o pyramid blending													
384×384	-	0.953	0.949	0.013	0.705	0.945	0.941	0.019	0.700	0.927	0.932	0.032	0.724
$L = 1280$	w/o SICA	0.396	0.602	0.497	0.504	0.373	0.416	0.530	0.512	0.242	0.395	0.645	0.506
$L = 1280$	w/ SICA	0.873	0.821	0.037	**0.774**	0.886	0.873	0.043	**0.750**	0.809	0.819	0.092	**0.751**
w/ pyramid blending													
$L = 1280$	w/o SICA	0.860	0.883	0.023	0.537	0.863	0.869	0.029	0.531	0.834	0.863	0.052	0.521
$L = 1280$	w/ SICA	**0.962**	**0.959**	**0.009**	0.743	**0.952**	**0.949**	**0.016**	0.738	**0.932**	**0.938**	**0.029**	0.741

mBA on three HR datasets by varying L from 784 to 1536 (Fig. 7). We choose $L = 1280$ since mBA almost converges after that.

SICA and pyramid blending. To demonstrate the necessity of SICA, we evaluate InSPyReNet with and without SICA. Since SICA only takes place when the input image is large enough to make train-inference discrepancy, we demonstrate results only on HR benchmarks. Please note that all evaluation is done with resizing method mentioned above, except for the LR resolution (Tab. 1).

In Tab. 1, InSPyReNet without SICA shows the worst performance, especially for the mBA. Since mBA only considers boundary quality, InSPyReNet with SICA and without pyramid blending shows the best performance in terms of mBA measure, yet shows poor results on other SOD metrics[1]. This is because even with SICA, InSPyReNet cannot overcome the discrepancy in effective receptive fields [5] between HR and LR images. For the setting without SICA, InSPyReNet with pyramid blending shows inferior results compared to the InSPyReNet without pyramid blending, meaning that the pyramid blending technique is meaningless without SICA since it worsen the results. Thus, SICA is crucial to be included in InSPyReNet, especially for the HR pyramid in the pyramid blending. Compared to the LR setting (*i.e.*, resizing into 384×384), using both SICA and pyramid blending shows better performance for all four metrics.

4.3   Comparison with State-of-the-Art methods

Quantitative Comparison. First, we compare InSPyReNet with 12 SotA LR SOD methods. In this experiment, we resize images same as for training. We either download pre-computed saliency maps or run an official implementation with pre-trained model parameters provided by the authors to evaluate with the same evaluation code for a fair comparison. Moreover, we re-implement Chen *et al.* [7], F^{3}Net [25], LDF [9], MINet [40], and PA-KRN [41] with same backbones we use to demonstrate how much the training settings affects the performance for other methods compared to InSPyReNet. We choose the above methods since

---

[1] This phenomenon shows that mBA itself cannot measure the performance of saliency detection, rather it only measures the quality of boundary itself.

Table 2. Quantitative results on five LR benchmarks. The first and the second best results for each metric are colored red and blue. ↑ indicates larger the better, and ↓ indicates smaller the better. † indicates our re-implementation.

Algorithms	Backbones	DUTS-TE			DUT-OMRON			ECSSD			HKU-IS			PASCAL-S		
		S_α ↑	F_{max} ↑	MAE↓	S_α ↑	F_{max} ↑	MAE↓	S_α ↑	F_{max} ↑	MAE↓	S_α ↑	F_{max} ↑	MAE↓	S_α ↑	F_{max} ↑	MAE↓
CNN backbone Models (ResNet, ResNext, Res2Net)																
PoolNet [37]	ResNet50	0.887	0.865	0.037	0.831	0.763	0.054	0.926	0.937	0.035	0.909	0.912	0.034	0.865	0.858	0.065
BASNet [38]	ResNet34	0.866	0.838	0.048	0.836	0.779	0.056	0.916	0.931	0.037	0.909	0.919	0.032	0.838	0.835	0.076
EGNet [8]	ResNet50	0.874	0.848	0.045	0.836	0.773	0.057	0.918	0.928	0.041	0.915	0.920	0.032	0.848	0.836	0.075
CPD [12]	ResNet50	0.869	0.840	0.043	0.825	0.754	0.056	0.918	0.926	0.037	0.905	0.911	0.034	0.848	0.833	0.071
GateNet [39]	ResNeXt101	0.897	0.880	0.035	0.849	0.794	0.051	0.929	0.940	0.035	0.925	0.932	0.029	0.865	0.855	0.064
†Chen *et al.* [7]	Res2Net50	0.890	0.869	0.040	0.834	0.769	0.061	0.931	0.943	0.035	0.921	0.927	0.034	0.871	0.862	0.060
†F³Net [25]	Res2Net50	0.892	0.876	0.033	0.839	0.771	0.048	0.915	0.925	0.040	0.915	0.925	0.030	0.856	0.842	0.065
†LDF [9]	Res2Net50	0.897	0.885	0.032	0.848	0.788	0.045	0.928	0.943	0.033	0.924	0.935	0.027	0.868	0.863	0.059
†MINet [40]	Res2Net50	0.896	0.883	0.034	0.843	0.787	0.055	0.931	0.942	0.031	0.923	0.931	0.028	0.865	0.858	0.060
†PA-KRN [41]	Res2Net50	0.898	0.888	0.034	0.853	0.808	0.050	0.930	0.943	0.032	0.922	0.935	0.027	0.863	0.859	0.063
Ours	Res2Net50	0.904	0.892	0.035	0.845	0.791	0.059	0.936	0.949	0.031	0.929	0.938	0.028	0.876	0.869	0.056
Transformer backbone Models (Swin, T2T-ViT)																
VST [42]	T2T-ViT-14	0.896	0.878	0.037	0.850	0.800	0.058	0.932	0.944	0.033	0.928	0.937	0.029	0.872	0.864	0.061
Mao *et al.* [43]	SwinB	0.917	0.911	0.025	0.862	0.818	0.048	0.943	0.956	0.022	0.934	0.945	0.022	0.883	0.883	0.050
†Chen *et al.* [7]	SwinB	0.901	0.883	0.034	0.860	0.810	0.052	0.937	0.948	0.030	0.928	0.935	0.029	0.876	0.868	0.058
†F³Net [25]	SwinB	0.902	0.895	0.033	0.860	0.826	0.053	0.937	0.951	0.027	0.932	0.944	0.023	0.868	0.864	0.059
†LDF [9]	SwinB	0.896	0.881	0.036	0.854	0.809	0.052	0.931	0.942	0.032	0.933	0.941	0.024	0.861	0.851	0.065
†MINet [40]	SwinB	0.906	0.893	0.029	0.852	0.798	0.047	0.935	0.949	0.028	0.930	0.938	0.025	0.875	0.870	0.054
†PA-KRN [41]	SwinB	0.913	0.906	0.028	0.874	0.838	0.042	0.941	0.956	0.025	0.933	0.944	0.023	0.873	0.872	0.056
Ours	SwinB	0.931	0.927	0.024	0.875	0.832	0.045	0.949	0.960	0.023	0.944	0.955	0.021	0.893	0.893	0.048

they provided source code with great reproducibility and consistent results. As shown in Tab. 2, our SwinB backbone model consistently shows outstanding performance across three metrics. Moreover, our Res2Net50 backbone model shows competitive results regarding its number of parameters.

Moreover, to verify the effectiveness of pyramid blending, we compare our method with SotA methods on HR and LR benchmarks (Tab. 3). Among HR methods, our method shows great performance among other methods, even though we use only DUTS-TR for training. Note that previous SotA HR methods show inferior results on LR datasets and vice versa, meaning that generalizing for both scales is difficult, while our method is robust for both scales. For instance, while PGNet trained with HR datasets (H, U) shows great performance on HR benchmarks, but shows more inferior results than other methods and even LR methods on LR benchmarks, while our method shows consistent results on both benchmarks. This is because LR datasets do not provide high-quality boundary details, while HR datasets lack of global object saliency.

Qualitative Comparison. We provide a visual comparison of our method in Fig. 8 and Fig. 9 on HR benchmarks. Overall, previous SotA methods are sufficient for detecting salient objects, but shows degraded results for complex scenes. Results show that InSPyReNet can produce accurate saliency prediction for the complex, fine details thanks to the pyramid blending. Moreover, even though we train our method only with LR dataset, DUTS-TR, InSPyReNet consistently shows accurate results compared to other methods.

Table 3. Quantitative results on three HR and two LR benchmarks. Beckbones; V: VGG16, R18: ResNet18, R50: ResNet50, S: SwinB. Datasets; D: DUTS-TR, H: HRSOD-TR, U: UHRSD-TR. The first and the second best results for each metric are colored red and blue. ↑ indicates larger the better, and ↓ indicates smaller the better. † indicates our re-implementation.

| Algorithms | Backbone | Train Datasets | HR benchmarks | | | | | | | | | | | | LR benchmarks | | | | | |
			DAVIS-S				HRSOD-TE				UHRSD-TE				DUTS-TE			DUT-OMRON		
			$S_\alpha\uparrow$	$F_{max}\uparrow$	MAE↓	mBA↑	$S_\alpha\uparrow$	$F_{max}\uparrow$	MAE↓	mBA↑	$S_\alpha\uparrow$	$F_{max}\uparrow$	MAE↓	mBA↑	$S_\alpha\uparrow$	$F_{max}\uparrow$	MAE↓	$S_\alpha\uparrow$	$F_{max}\uparrow$	MAE↓
†Chen et al. [7]	S	D	0.934	0.925	0.018	0.697	0.915	0.907	0.032	0.684	0.915	0.919	0.034	0.712	0.901	0.883	0.034	0.860	0.810	0.052
†F³Net [25]	S	D	0.931	0.922	0.017	0.681	0.912	0.902	0.034	0.674	0.920	0.922	0.033	0.708	0.902	0.895	0.033	0.860	0.826	0.053
†LDF [9]	S	D	0.928	0.918	0.019	0.682	0.905	0.888	0.036	0.672	0.911	0.913	0.038	0.702	0.896	0.881	0.036	0.854	0.809	0.052
†MINet [40]	S	D	0.933	0.930	0.017	0.673	0.927	0.917	0.025	0.670	0.915	0.917	0.035	0.694	0.906	0.893	0.029	0.852	0.798	0.047
†PA-KRN [41]	S	D	0.944	0.935	0.014	0.668	0.927	0.918	0.026	0.653	0.919	0.926	0.034	0.673	0.913	0.906	0.028	0.874	0.838	0.042
PGNet [4]	S+R18	D	0.935	0.931	0.015	0.707	0.930	0.922	0.021	0.693	0.912	0.914	0.037	0.715	0.911	0.903	0.027	0.855	0.803	0.045
Zeng et al. [1]	V	D,H	0.876	0.889	0.026	0.618	0.897	0.892	0.030	0.623	-	-	-	-	0.824	0.835	0.051	0.762	0.743	0.065
Tang et al. [2]	R50	D,H	0.920	0.935	0.012	0.716	0.920	0.915	0.022	0.693	-	-	-	-	0.895	0.888	0.031	0.843	0.796	0.048
PGNet [4]	S+R18	D,H	0.947	0.948	0.012	0.716	0.935	0.929	0.020	0.714	0.912	0.915	0.036	0.735	0.912	0.905	0.028	0.858	0.803	0.046
PGNet [4]	S+R18	H,U	0.954	0.956	0.010	0.730	0.938	0.939	0.020	0.727	0.935	0.930	0.026	0.765	0.861	0.828	0.038	0.790	0.727	0.059
Ours	S	D	0.962	0.959	0.009	0.743	0.952	0.949	0.016	0.738	0.932	0.938	0.029	0.741	0.931	0.927	0.024	0.875	0.832	0.045
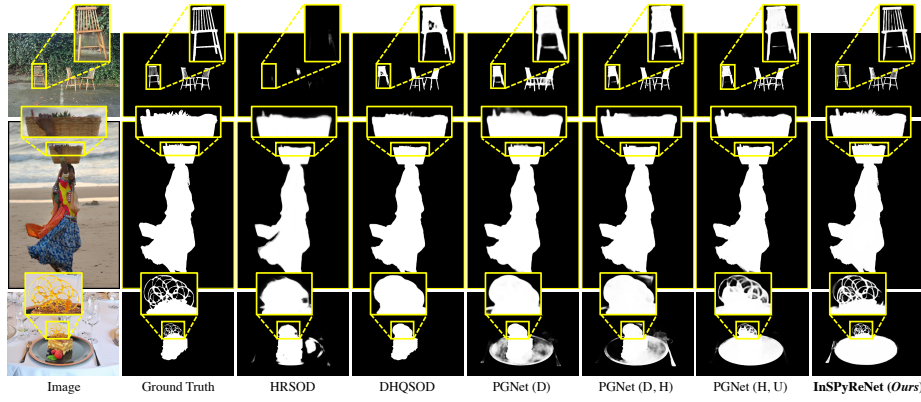
Fig. 8. Qualitative results of InSPyReNet (SwinB) compared to SotA HR methods on HRSOD-TE. *Best viewed by zooming in.*
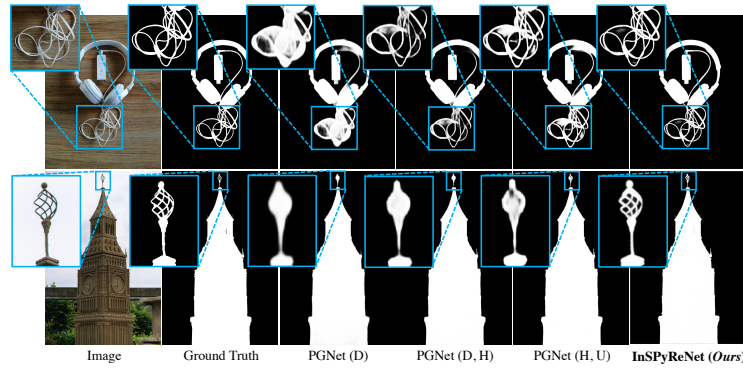

Fig. 9. Qualitative results of InSPyReNet (SwinB) compared to PGNet on UHRSD-TE. *Best viewed by zooming in.*
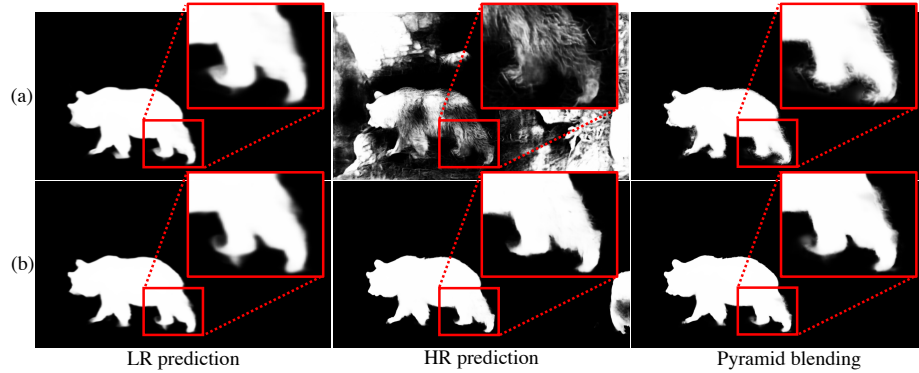
LR prediction   HR prediction   Pyramid blending

Fig. 10. Visual comparison of LR, HR prediction, and pyramid blended results of InSPyReNet with (a) Res2Net50 and (b) SwinB backbones.

5   Discussion and Conclusion

Weakness: Backbone Network. We do not use Res2Net50 for HR benchmark due to the following reason. As shown in Fig. 10, HR prediction from Res2Net50 backbone produces saliency map with numerous unnecessary artifacts. This is because CNN backbones are vulnerable to its ERF size, which is highly dependent on its training dataset. Unlike traditional CNN backbones, there are many works to minimize the above issue such as Fast Fourier Convolution [44], or ConvNeXt [45]. We found that those methods are helpful for reducing such artifacts for HR prediction, but not enough for detail reconstruction. However, Vision Transformers like SwinB have larger ERFs and consist of non-local operation for regarding global dependencies, which are suitable for our method. Thus, even the HR prediction shows some False Positives (second column, second row in Fig. 10), we can easily remove them while enhance boundary details via pyramid blending.

Future Work and Conclusion. Starting from previous works with Laplacian pyramid prediction [7, 16], we have shown that InSPyReNet shows noticeable improvements on HR prediction without any HR training datasets or complex architecture. In a series of experiments, our method shows great performance on HR benchmarks while robust again LR benchmarks as well. Although we only utilize a concept of pyramid-based image blending for merging two pyramids with different scales, we hope our work can extend to the multi-modal input such as RGB-D SOD or video SOD with temporal information.

References

1. Zeng, Y., Zhang, P., Zhang, J., Lin, Z., Lu, H.: Towards high-resolution salient object detection. In: ICCV. (2019) 7234–7243 1, 2, 4, 10, 13
2. Tang, L., Li, B., Zhong, Y., Ding, S., Song, M.: Disentangled high quality salient object detection. In: ICCV. (2021) 3580–3590 1, 2, 4, 13
3. Zhang, P., Liu, W., Zeng, Y., Lei, Y., Lu, H.: Looking for the detail and context devils: High-resolution salient object detection. IEEE TIP **30** (2021) 3204–3216 1, 2
4. Xie, C., Xia, C., Ma, M., Zhao, Z., Chen, X., Li, J.: Pyramid grafting network for one-stage high resolution saliency detection. arXiv preprint arXiv:2204.05041 (2022) 1, 2, 4, 10, 13
5. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. NeurIPS **29** (2016) 1, 4, 11
6. Burt, P.J., Adelson, E.H.: A multiresolution spline with application to image mosaics. ACM TOG **2** (1983) 217–236 2
7. Chen, S., Tan, X., Wang, B., Hu, X.: Reverse attention for salient object detection. In: ECCV. (2018) 234–250 3, 5, 11, 12, 13, 14
8. Zhao, J.X., Liu, J.J., Fan, D.P., Cao, Y., Yang, J., Cheng, M.M.: Egnet: Edge guidance network for salient object detection. In: ICCV. (2019) 8779–8788 3, 12
9. Wei, J., Wang, S., Wu, Z., Su, C., Huang, Q., Tian, Q.: Label decoupling framework for salient object detection. In: CVPR. (2020) 13025–13034 3, 11, 12, 13
10. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115** (2015) 211–252 3, 9
11. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: CVPR. (2017) 136–145 3, 9
12. Wu, Z., Su, L., Huang, Q.: Cascaded partial decoder for fast and accurate salient object detection. In: CVPR. (2019) 3907–3916 4, 12
13. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR. (2017) 2881–2890 4, 10
14. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017) 4
15. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. (2017) 624–632 4
16. Ghiasi, G., Fowlkes, C.C.: Laplacian pyramid reconstruction and refinement for semantic segmentation. In: ECCV. (2016) 519–534 4, 5, 9, 14
17. Kim, T., Lee, H., Kim, D.: Uacanet: Uncertainty augmented context attention for polyp segmentation. In: ACM MM. (2021) 2167–2175 5, 6
18. Gao, S.H., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.: Res2net: A new multi-scale backbone architecture. IEEE TPAMI **43** (2021) 652–662 5, 9
19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021) 5, 9
20. Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T.: Axial attention in multidimensional transformers. arXiv preprint arXiv:1912.12180 (2019) 5
21. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In: ECCV. (2020) 108–126 5

22. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: CVPR. (2019) 3146–3154 6
23. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: ECCV. (2020) 173–190 6
24. Burt, P., Adelson, E.: The laplacian pyramid as a compact image code. IEEE Transactions on Communications **31** (1983) 532–540 7, 8
25. Wei, J., Wang, S., Huang, Q.: F^{3}net: Fusion, feedback and focus for salient object detection. In: AAAI. Volume 34. (2020) 12321–12328 8, 11, 12, 13
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In Bengio, Y., LeCun, Y., eds.: ICLR. (2015) 10
27. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: CVPR. (2013) 3166–3173 10
28. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended cssd. IEEE TPAMI **38** (2015) 717–729 10
29. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: CVPR. (2015) 5455–5463 10
30. Li, Y., Hou, X., Koch, C., Rehg, J.M., Yuille, A.L.: The secrets of salient object segmentation. In: CVPR. (2014) 280–287 10
31. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR. (2016) 724–732 10
32. Wang, W., Lai, Q., Fu, H., Shen, J., Ling, H., Yang, R.: Salient object detection in the deep learning era: An in-depth survey. IEEE TPAMI (2021) 1–1 10
33. Fan, D.P., Cheng, M.M., Liu, Y., Li, T., Borji, A.: Structure-measure: A new way to evaluate foreground maps. In: ICCV. (2017) 4548–4557 10
34. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: CVPR. (2009) 1597–1604 10
35. Perazzi, F., Krähenbühl, P., Pritch, Y., Hornung, A.: Saliency filters: Contrast based filtering for salient region detection. In: CVPR. (2012) 733–740 10
36. Cheng, H.K., Chung, J., Tai, Y.W., Tang, C.K.: Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In: CVPR. (2020) 8890–8899 10
37. Liu, J.J., Hou, Q., Cheng, M.M., Feng, J., Jiang, J.: A simple pooling-based design for real-time salient object detection. In: CVPR. (2019) 3917–3926 12
38. Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., Jagersand, M.: Basnet: Boundary-aware salient object detection. In: CVPR. (2019) 7479–7489 12
39. Zhao, X., Pang, Y., Zhang, L., Lu, H., Zhang, L.: Suppress and balance: A simple gated network for salient object detection. In: ECCV. (2020) 35–51 12
40. Pang, Y., Zhao, X., Zhang, L., Lu, H.: Multi-scale interactive network for salient object detection. In: CVPR. (2020) 9413–9422 11, 12, 13
41. Xu, B., Liang, H., Liang, R., Chen, P.: Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection. In: AAAI. (2021) 3004–3012 11, 12, 13
42. Liu, N., Zhang, N., Wan, K., Shao, L., Han, J.: Visual saliency transformer. In: ICCV. (2021) 4722–4732 12
43. Mao, Y., Zhang, J., Wan, Z., Dai, Y., Li, A., Lv, Y., Tian, X., Fan, D.P., Barnes, N.: Transformer transforms salient object detection and camouflaged object detection. arXiv preprint arXiv:2104.10127 (2021) 12
44. Chi, L., Jiang, B., Mu, Y.: Fast fourier convolution. In: NeurIPS. Volume 33. (2020) 4479–4488 14

45. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022) 14