Residual Refinement Network with Attribute Guidance for Precise Saliency Detection

FENG LIN, WENGANG ZHOU, and JIAJUN DENG, University of Science and Technology of China, China
BIN LI and YAN LU, Microsoft Research Asia, China
HOUQIANG LI, University of Science and Technology of China, China

As an important topic in the multimedia and computer vision fields, salient object detection has been researched for years. Recently, state-of-the-art performance has been witnessed with the aid of the fully convolutional networks (FCNs) and the various pyramid-like encoder-decoder frameworks. Starting from a common encoder-decoder architecture, we enhance a residual refinement network with feature purification for better saliency estimation. To this end, we improve the global knowledge streams with intermediate supervisions for global saliency estimation and design a specific feature subtraction module for residual learning, respectively. On the basis of the strengthened network, we also introduce an attribute encoding sub-network (AENet) with a grid aggregation block (GAB) to guide the final saliency predictor to obtain more accurate saliency maps. Furthermore, the network is trained with a novel constraint loss besides the traditional cross-entropy loss to yield the finer results. Extensive experiments on five public benchmarks show our method achieves better or comparable performance compared with previous state-of-the-art methods.

CCS Concepts: • **Computing methodologies → Interest point and salient region detections;**

Additional Key Words and Phrases: Salient object detection, residual learning, deep intermediate supervision, attribute encoding

ACM Reference format:
Feng Lin, Wengang Zhou, Jiajun Deng, Bin Li, Yan Lu, and Houqiang Li. 2021. Residual Refinement Network with Attribute Guidance for Precise Saliency Detection. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 3, Article 81 (July 2021), 19 pages.
https://doi.org/10.1145/3440694

The work of H. Li was supported by NSFC under contract 61836011. The work of W. Zhou was supported in part by the National Natural Science Foundation of China under contracts 61822208 and 61632019, and in part by Youth Innovation Promotion Association CAS (No. 2018497).
Authors' addresses: F. Lin, W. Zhou (corresponding author), and J. Deng, University of Science and Technology of China, No.96 JinZhai Road, Baohe District, Hefei, China; emails: lin1993@mail.ustc.edu.cn, zhwg@ustc.edu.cn, dengjj@mail.ustc.edu.cn; B. Li and Y. Lu, Microsoft Research Asia, No. 5 Dan Ling Street, Haidian District, Beijing, China; emails: {libin, yanlu}@microsoft.com; H. Li (corresponding author), University of Science and Technology of China, No.96 JinZhai Road, Baohe District, Hefei, China; email: lihq@ustc.edu.cn.

1551-6857/2021/07-ART81 $15.00
https://doi.org/10.1145/3440694

1 INTRODUCTION

Salient object detection has attracted a vast amount of attention in recent years [1, 43, 59, 63, 65, 72]. A range of computer vision tasks such as image classification [44], object detection [8, 61], object tracking [18], image caption [6], and image retrieval [13, 15] benefit from this technique, which usually serves as the pre-processing step [14, 41]. Early traditional methods toward saliency detection commonly rely on hand-crafted features and local cues to highlight the region of interest [5, 25, 60]. Due to the lack of high-level semantic knowledge, these methods lead to a mass of distractors and defective saliency maps frequently, suffering from significant performance degradation in complicated scenes.

In the past few years, many works [19, 28, 40, 50] based on **convolutional neural networks (CNNs)** improve the performance of salient object detection by a large margin compared with those traditional algorithms. Owing to the capability of aggregating low-level cues and high-level knowledge, they alleviate the problems above and produce fine-grained estimation for object details. A significant fraction of the recent approaches employ an encoder-decoder structure as the basic framework, where the raw input flows through a top-down pathway and then generates a saliency map at the same resolution as the input image [12, 34, 59, 68]. Fully convolutional layers are widely adopted for end-to-end network training. Meanwhile, various elaborate modules are proposed to extract and strengthen image feature representations to pursue the higher performance [4, 33, 42, 71, 73].

In spite of the remarkable advance driven by feature learning, detecting salient objects in complex natural scenes is still a fairly challenging task. Due to the absence of the large-scale training data, how to leverage deep neural networks to extract discriminative features is a very popular problem in great demand. Usually, a universal framework for salient object detection consists of (*i*) a common pyramid-like backbone network, like VGGNet, as the feature encoder to extract deep global semantic information at a lower resolution, (*ii*) a convolutional subnetwork including several bilinear interpolation operations or deconvolutional layers as the feature decoder to restore the size of the enriched features above, and (*iii*) a pixel-wise predictor for generating saliency outputs. This type of structure has proven effective, but still has much room to improve. For instance, feature encoder goes deeper to obtain global semantic contexts by increasing the receptive fields, but at the same time, it also deteriorates local texture details because of the reduced spatial resolution of feature maps. Albeit previous works introduce attention mechanisms [4, 10, 54], design specific architectures [42, 66, 73], or turn to recurrent networks for precise prediction [7, 52, 53], how to utilize features extracted by networks efficiently remains to be studied.

In this article, we present an improved residual refinement network composed of four modules described below. First, inspired by residual learning [17], we adopt a coarse-to-fine solution where the network first generates a coarse saliency prediction at a low resolution, and then upsamples and refines it via well-designed residual learning with specifically designed deep supervisions. In this way, there is an elaborate division of labor among the lateral residual features and deep global contexts. Second, to help the decoder maintain more local cues, a **grid aggregation block (GAB)** is designed to combine complementary hierarchical features at different scales. Third, a light sub-network, named **attribute encoding network (AENet)**, is proposed to learn an attribute information vector as the guidance of the saliency predictor. Last but not least, to approximate the saliency outputs to the ground truth, we further introduce a novel constraint loss besides the traditional binary cross-entropy loss.

In summary, our major contributions are as follows:

- We enhance a refinement network that predicts saliency maps in a coarse-to-fine way progressively. Significant improvement is achieved by using explicit residual refinement and deep intermediate supervisions simultaneously.

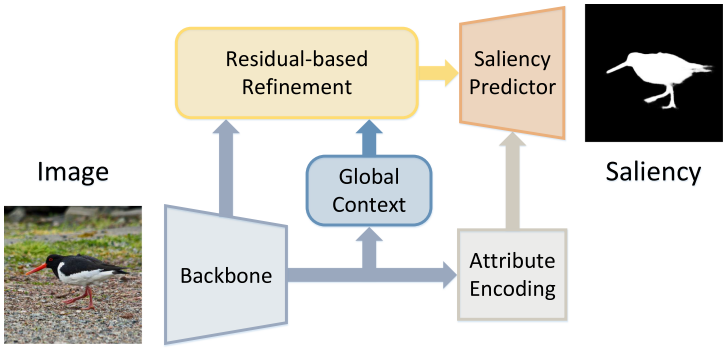


Fig. 1. The overview of our method. Colorful arrows indicate the flows of information among network mudules.

- Grid aggregation block is adopted for integrating local cues and global semantic contexts, and an attribute encoding network is employed to guide accurate saliency learning. Furthermore, a constraint loss is added to promote the performance without introducing extra cost of computational complexity.
- Experimental results on five public salient object detection benchmarks demonstrate our method achieves better or comparable performance compared with state-of-the-art methods.

2 RELATED WORK

Over the past decades, plenty of methods are proposed to tackle the salient object detection problem. In early years, many algorithms are based on hand-crafted features [2, 21–24, 36, 45]. Recently, with the development of deep learning, CNNs have made tremendous progress and become the mainstream in this task. Since our work follows the second paradigm, we only make an overview of the recent CNN-based works. And following the review, we give a brief explanation of the relationship between these works and our method.

Before Long et al. [39] proposed FCN in computer vision tasks, saliency detection frameworks were usually modified from a visual classification model. Wang et al. [50] combined local estimation and global search by generating object proposals and regressing their confidence to merge saliency maps. Li et al. [28] enclosed three scale inputs, including the target patch, neighboring regions, and the entire image, to extract a universal representation for saliency prediction. These methods pushed the frontier of saliency detection at that time, but they are time-consuming due to the repetition of inputs and can not avoid blurry edges and false detection.

To overcome the above drawback, FCNs are employed in this pixel-level recognition task. Hou et al. [19] introduced short connections between shallow and deep features with a fusion loss. Zhang et al. [67] adaptively aggregated multi-level features maps and detected salient objects with the combined features. Luo et al. [40] created a multi-resolution grid structure to encourage both local and global information to cooperate. Beyond that, a novel supervision was implemented to sharpen the boundaries of salient objects. Zhang et al. [66] proposed a gated bi-directional message passing module to combine semantic information and spatial local details, capturing richer contexts for accurate saliency maps. Wang et al. [54], Liu et al. [35], and Chen et al. [4] all exploited an encoder-decoder framework with attention weights to obtain saliency maps. To segment out the entire object with finer boundaries, Feng et al. [10] presented an attentive feedback module and introduce a boundary-enhanced loss. For a similar purpose, Wang et al. [55] devised a pyramid attention structure to concentrate on salient regions and then added an edge detection module to

offer boundary cues. Liu et al. [33] investigated from the point of the pooling technique, exploring the potentials of U-shape networks on this task, and they also leveraged edge detection to further improve the results. Similarly, to make full use of the external edge detection database, Wu et al. [58] proposed a **Cross Refinement Unit (CRU)**, which bidirectionally passes messages between the two tasks of salient object detection and edge detection. Their refinement framework stacked multiple CRUs to improve multi-level deep features. Xu et al. [64] modeled structural information of deep features and deep predictions into a unified CRF model and developed the mean-field approximation inference. A cascade CRFs architecture was designed to progressively integrate and refine multi-scale CNN features for salient object detection. Su et al. [49] proposed a boundary-aware network as well as an integrated successive dilation module to capture features with selectivity and invariance. Zhao et al. [70] used the complementary salient object information and salient edge information jointly to train an edge guidance network, preserving the salient object boundaries to improve the predicted saliency maps.

The above methods are dedicated to handling coarse boundaries, incoherent saliency regions, distracting noisy background, and so forth. In addition to the aforesaid works, the recurrent architecture is also popular in this field for its ability to incorporate prior knowledge, attend to regions of interest, or refine outputs progressively. Besides, some works try to overcome the inconsistency dilemma in saliency detection with the aid of the capsule network. We will not discuss them in detail and refer the readers to References [7, 27, 38, 52, 53] for a comprehensive understanding.

In this article, we build our network on an encoder-decoder framework. The common thread of References [4, 7, 12, 52, 58] and ours is all the models are built upon an effective backbone with redundancy in some way. But the previous methods and ours are quite different in several ways. To be specific, R^3Net [7] adopted a recurrent refinement framework, repeatedly using a same set of deep and shallow features as the input of their stacked refinement blocks, where the predicted saliency map at each stage was employed as an attention map to generate the saliency residual. Similarly, RFCN [52] also obtained salient object results via a recurrent fashion. RFCN employed an encoder-decoder network in each step, where the predicted foreground map in the last time served as the saliency prior. The convolutional encoder of RFCN took both the image and saliency prior as the input and the deconvolutional decoder refined the saliency prediction with the help of the saliency prior. Furthermore, RFCN used a two-stage training strategy with which the semantic segmentation data was exploited to pretrain the model. Different from RFCN using external semantic segmentation datasets, Refinet [12] utilized some assisted image segmentation algorithms to generate a series of segmentation hypotheses. The resultant segmentation regions, together with the initial coarse saliency map generated by a feature extraction stream, were fed into a convolutional fusion stream to produce the refined results. In addition to exploiting semantic segmentation prior, adding edge information was able to improve the performance in saliency refinement frameworks. SCRN [58] extracted two separate multi-level features for both salient object detection and edge detection. These two direction-specific feature integration operations aimed at simultaneous refinement for the two tasks. In contrast, we apply a residual learning scheme where a clear division of labor between global saliency estimation and residual refinement is achieved. We adopt a straightforward convolutional structure rather than a recurrent network for saliency refinement. From the perspective of network learning, our end-to-end model is trained with an effective assisted loss function on the salient object detection dataset. No more external data or extra algorithm processing is needed, and a simple network with elaborate modules is all we need to acquire satisfactory saliency.

Table 1. Network Parameters of VGG-16

Layer	Configurations
$Conv1\_1, ReLU$	$(3, 64, 3 \times 3, 1)$
$Conv1\_2, ReLU$	$(64, 64, 3 \times 3, 1)$
$MaxPool$	$(2 \times 2,\ 2)$
$Conv2\_1, ReLU$	$(64, 128, 3 \times 3, 1)$
$Conv2\_2, ReLU$	$(128, 128, 3 \times 3, 1)$
$MaxPool$	$(2 \times 2, 2)$
$Conv3\_1, ReLU$	$(128, 256, 3 \times 3, 1)$
$Conv3\_2, ReLU$	$(256, 256, 3 \times 3, 1)$
$Conv3\_3, ReLU$	$(256, 256, 3 \times 3, 1)$
$MaxPool$	$(2 \times 2, 2)$
$Conv4\_1, ReLU$	$(256, 512, 3 \times 3, 1)$
$Conv4\_2, ReLU$	$(512, 512, 3 \times 3, 1)$
$Conv4\_3, ReLU$	$(512, 512, 3 \times 3, 1)$
$MaxPool$	$(2 \times 2, 2)$
$Conv5\_1, ReLU$	$(512, 512, 3 \times 3, 1)$
$Conv5\_2, ReLU$	$(512, 512, 3 \times 3, 1)$
$Conv5\_3, ReLU$	$(512, 512, 3 \times 3, 1)$

The layer names are listed in the left column. The corresponding parameters of the convolutional layers are denoted in the form of ($in\_channel$, $out\_channel$, $kernel\_size \times kernel\_size$, $stride$), while the parameters of the pooling layers are denoted in the form of ($kernel\_size \times kernel\_size$, $stride$) in the right column. All convolutional layers are activated by the ReLU function. The last two original fully connected layers in VGG-16 are deprecated and are replaced with two custom blocks, i.e., *Conv6* and *Conv7*, as demonstrated in Table 3.

3 PROPOSED METHOD

3.1 Feature Encoder

3.1.1 Backbone. VGGNet has been proved effective in image recognition [48] where low-level textures and high-level knowledge are preserved in hierarchical features, which are highly helpful for both saliency detection and residual refinement. We adopt VGG-16 as the backbone network but discard the top two fully connected layers, which are replaced with two extra conventional blocks $\{Conv6, Conv7\}$. Each of the conventional blocks is composed of one 5×5 convolutional layer with a stride of 2 and three 5×5 convolutional layers with stride 1. The channel number of $\{Conv6, Conv7\}$ is 512 while output features are activated by the ReLU function. The backbone network configurations are listed in Table 1 and Table 3, where ConvX_Y is the Yth convolutional layer of the Xth block in VGG-16. Since we try to obtain a coarse preliminary saliency map, the deeper features help to locate saliency objects. Obviously, if the network starts off on the wrong foot, it is hard to rectify the error and get a satisfying result. So, we improve the deep feature with a global context block, as described in the next subsection.

3.1.2 Global Context Block (GCB). The bilinearly upsampled features derived from the layers $\{Conv5\_3, Conv6\_3, Conv7\_3\}$ are concatenated to maintain more global contexts at different **receptive fields (RFs)**, and then pass through a common 1×1 convolutional layer to reduce the channel number to 512. We feed this compact feature into GCB to enhance its ability of capturing

Table 2. Side-outputs and the Corresponding Referred Layers

Side-output	1	2	3	4
Layer	Conv4_3	Conv3_3	Conv2_2	Conv1_2

$ConvX\_Y$ is the Y_{th} layer of the X_{th} block.

the high-level semantic information. GCB is capable of modeling global context in visual recognition tasks [3]. As a simplified non-local block, GCB efficiently captures long-range dependency but is almost computationally free, with which the receptive field is enlarged equivalently so each pixel of the feature map establishes a relationship with each other, helping the network detect separated targets and resist background distractors. As a result, the enriched feature by GCB serves as the output of the feature encoder, which is used in the next step.

3.2 Feature Decoder

3.2.1 Residual Refinement. It is well known that the deeper features encode global information in favor of object locating, while the shallower features preserve rich local cues that are conducive to refine details and sharpen boundaries. Based on this observation, we design a residual refinement network to improve saliency maps step-by-step. The framework is illustrated in Figure 3. The GCB feature is devoted to obtaining a preliminary saliency prediction (denoted as (a) in Figure 3) by a standard 3×3 convolutional layer (denoted as Conv0). The features from $\{Conv4\_3, Conv3\_3, Conv2\_2, Conv1\_2\}$ are selected as side-outputs for saliency refinement. Side-output $1 \sim 3$ are exploited in this section and side-output 4 is used in Section 3.2.3. For clarity, side-outputs are outlined in Table 2.

As observed in Figure 2, the global contexts produce global saliency prediction, then side-outputs concentrate on the tiny residual object details and edges, which are complementary to global saliency. Considering this procedure, we exploit multi-scale lateral features in a similar but also different fashion with FPNs. We adopt element-wise subtraction operation in place of summation in FPNs. As illustrated in Figure 3, The feature subtraction module subtracts the bilinearly upsampled feature generated in the previous step from the current side-output, while the feature concatenation module combines the current side-output and previous embedded features into an integrated embedding. At first step, the GCB feature serves as the inputs (denoted as (b) and (c) in Figure 3) of a feature subtraction module and a feature concatenation module, respectively, then the outputs are treated in the same way in the next step. We try to specify the uses of the features utilized in gradual refining steps. In a word, the main idea is to encourage global features and side-outputs to perform their own functions: By this feature integration, the coarse global saliency is obtained by deep global features and the refined residual is derivated from those residual-specific side-outputs that contain different information with global ones as much as possible. Furthermore, the side-output in each step is supposed to preserve different cues with those in the previous steps.

After residual-specific embedding, several stacked convolutional layers strengthen the feature further and produce the residual map afterward. The details of these units, i.e., ConvR_i $(i = \{1, 2, 3\})$, are listed in Table 3. The predicted residual is added to the bilinearly unsampled saliency map generated in the last step. The output in each step can be formulated as follows:

$$Sal_i = \begin{cases} conv(f_0), & \text{if } i = 0, \\ conv(s_i - Up(f_{i-1})) + Up(Sal_{i-1}), & \text{if } i > 0, \end{cases} \tag{1}$$

where s_i is side-output i, f_0 is the GCB feature, f_i is the concatenation of s_i and f_{i-1}, and Sal_i is the current saliency map in step i $(i \in \{0, 1, 2, 3\})$. Up denotes bilinearly upsample operation by a factor 2. To alleviate the considerable burden of feature learning and accelerate network

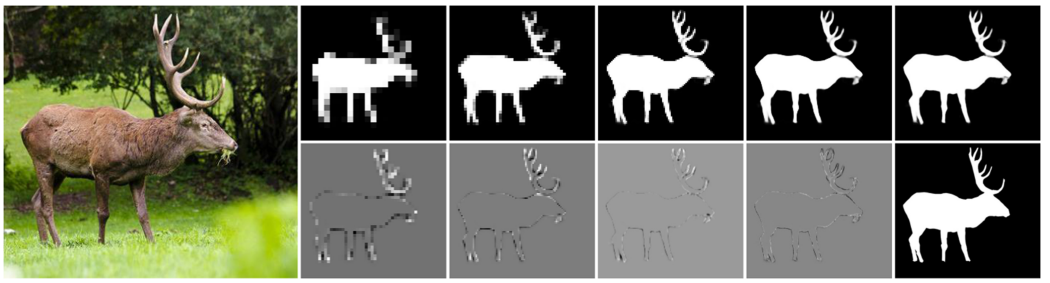


Fig. 2. Refinement on saliency maps in a step-by-step manner. The leftmost is an input image. Saliency maps from left to right in the first row are refined progressively. The improvement of local details and edges are illustrated in the second row where the increases of saliency values are marked by the lighter color (white) and the decreases are marked by the darker color (black), respectively. The ground truth is also shown at the end of the second row. To better understand the refinement procedure, we clarify the connections between Figure 2 and Figure 3 briefly. The first map in the first row is the upsampled output of the module Conv0 and the second to the fourth maps are the upsampled outputs of the element-wise summation modules following ConvR_1, ConvR_2, and ConvR_3, respectively. And the last saliency map in the first row is the final result of the saliency predictor.

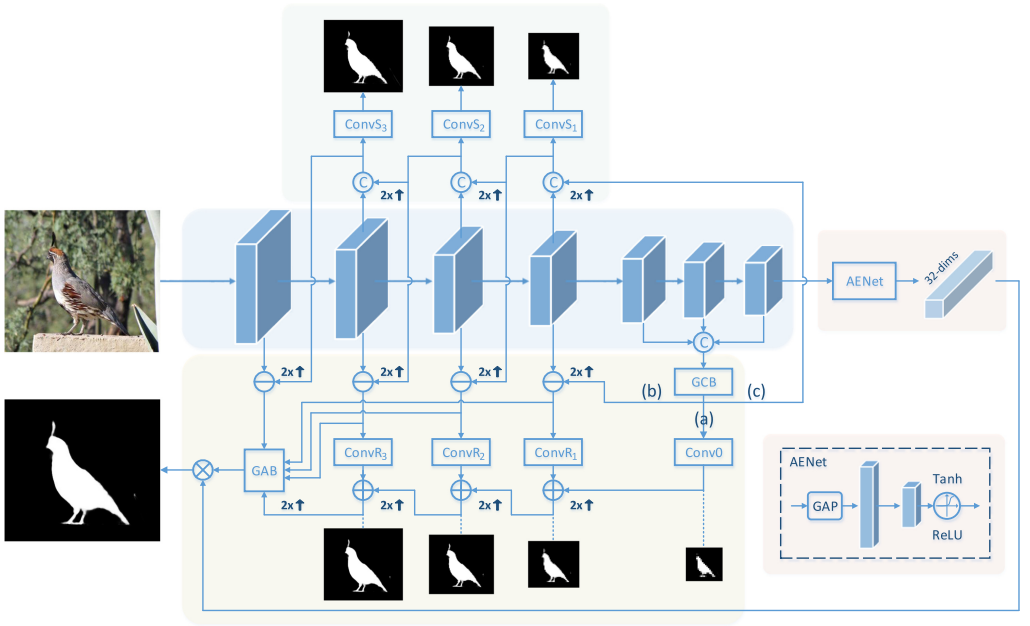


Fig. 3. The architecture of our proposed network. For an input image, we obtain the saliency map with a strategy of progressive refinement. The bottom half of the network represents the step-by-step refining procedure, the middle is the backbone network, and the top illustrates deep intermediate supervision. "+," "−," "×," and "C" denote the element-wise summation module, feature subtraction module, attribute-orientated saliency predictor, and concatenation operation, respectively. $(a) \sim (c)$ indicate the three flows of global context information that plays an important role in global saliency estimation, residual refinement, and intermediate supervision. Because other modules are identified by abbreviations, there is no more tautology here.

Table 3. The Network Configurations of Conv6, Conv7, ConvR$_i$, ConvS$_i$, and AENet

Conv6/Conv7	ConvR$_i$/ConvS$_i$	AENet
$(512, 5 \times 5, 2)$	$(C_i, 1 \times 1, 1)$	*Global Average Pooling*
$\{(512, 5 \times 5, 1), ReLU\} \times 3$	$\{(C_i, 3 \times 3, 1), ReLU\} \times 2$	$\{(512, 1 \times 1, 1), ReLU\}$
	$(1, 3 \times 3, 1)$	$(64, 1 \times 1, 1)$
		$Tanh, ReLU, Dropout^1(p = 0.3)$

The layer parameters of Conv6 and Conv7 are the same. Since ConvR$_i$ and ConvS$_i$ are identical except for the channel numbers, they are presented in the middle column, while AENet is displayed on the right. $(c, k \times k, s) \times d$ denotes d stacked convolutional layers with c channels, a kernel of $k \times k$, and an s stride. C_i is set to 64 in ConvR$_i$, but {512, 256, 128} for side-output {1, 2, 3}, respectively, in ConvS$_1$, ConvS$_2$, and ConvS$_3$.

convergence, the supervision is applied on all the saliency outputs, including the final result and the intermediate ones, which is similar with the implementation of the previous works [4, 19].

3.2.2 Deep Intermediate Supervision. Apart from the aforementioned supervision, we additionally append three intermediate supervision on side-outputs $1 \sim 3$. Following the concatenation module, a certain number of convolutional layers are simply stacked, which are denoted as ConvS$_i$ ($i = \{1, 2, 3\}$) in the top half of Figure 3. For simplification, the structure of ConvS$_i$ is almost identical to ConvR$_i$, whose detailed configurations are summarized in Table 3. These three ConvS$_i$ generate saliency maps of different sizes, corresponding to the resolution of side-outputs. The impact of deep intermediate supervision is twofold: first, they facilitate the training; second, and more importantly, they also guarantee the concatenated features we subtract from side-outputs in the residual learning procedure to capture global saliency information as much as possible. Deep intermediate supervision emphasizes the spirit of the proposed residual learning: For each step, global information are preserved in the last concatenated feature, while side-outputs are obsessed with details and boundaries of salient objects without distraction. Both the deep intermediate supervision and the proposed feature division constitute our residual refinement fashion. It is worth mentioning that these appended ConvS$_i$ help network training but introduce no computational complexity in inference stage.

3.2.3 Grid Aggregation Block (GAB). As we mentioned earlier, shallow features are utilized to compensate for saliency local details. But with the resolution rising, the improvement approaches to saturation. We find when the resolution is raised up to half of the input size, the residual refinement brings little difference on saliency maps (Figure 2). Since we have acquired a relatively acceptable saliency map, we can take advantage of this prior knowledge as well as the rich multi-level information derivated from the network to improve the result further. Inspired by GridNet [11], we design a novel grid aggregation block (GAB) to eliminate the residual refinement unit on side-output 4. As illustrated in Figure 4, the inputs of GAB consist of four side-outputs and upsampled saliency map generated in the last step. First, the concatenation of the highest-resolution feature and the saliency map is feed into GAB as the first feature stream. Streams keep the sizes and channel numbers constant, but vertical connections reduce or increase their resolutions while the channels remain unchanged. We employ a 1×1 convolutional layer to reduce channels of four streams to $\{32, 64, 64, 128\}$, with which GAB is fairly efficient compared with the original implementation of residual refinement. At the end of GAB, 1×1 convolution with 64 channels is performed for feature integration.

The basic idea of GridNet is to build a grid pattern [11], stimulating the interconnected feature streams to cooperate at various resolutions. Different from GridNet, GAB has multiple input

[1]Dropout is disabled in inference stage.

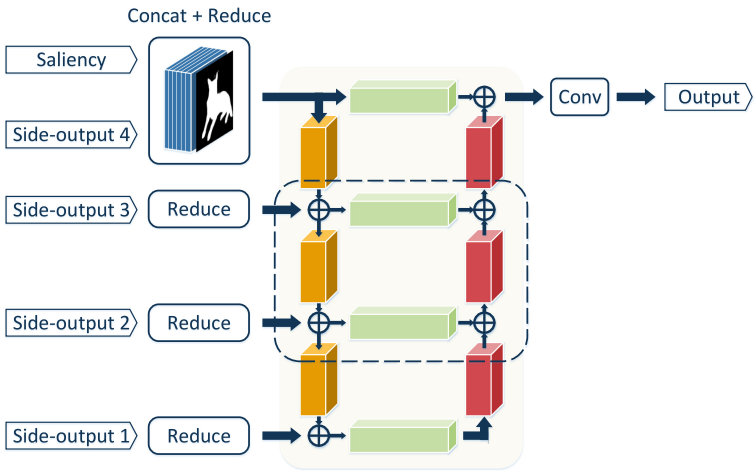


Fig. 4. GAB: green units are residual blocks, orange units are downsampling blocks with increased feature channels, and red units are upsampling blocks with decreased channel numbers. These three kinds of units are similar with those used in GridNet, except we do **not** use Batch Normalization due to limited batch size during training. *Concat* stands for feature concatenation. *Reduce* represents 1×1 convolution for channel reduction. For detailed schema of the zoom on the dashed box, we refer readers to Figure 5.

streams, including four different-scale side-outputs and one saliency prior as the attention weight. With the aid of the interconnected feature streams, sensitive shallow features enrich compact deep ones with low-level cues, and in turn, high-level semantics help to exclude the noise interference of the dense local characteristics. Information stream flows through two or three alternative pathways before feature summation, which increases the expressive power of the network. Beyond that, the saliency prior makes the network focus on the salient objects without the distraction of the complicated background.

3.3 Saliency Predictor

Different from existing works [31, 37, 69, 71] that contribute to design various specific units for more precise saliency prediction, we approach it from another perspective. Enlightened by the success of Mask R-CNN [16], where the detection head predicts the category-dependent instance masks, we design an exclusive saliency predictor for each category in the preliminary experiment. Unfortunately, none of the existing salient object detection datasets has category annotation. Furthermore, several salient objects of various kinds are likely to appear in a single image. Therefore, to predict saliency maps according to categories directly is nearly unreachable. From this perspective, we expect the network extracts the features of an input image with an eye to the salient object in the scene and predict the saliency maps adaptively. To this end, we implicitly encode category information into an attribute vector to make saliency prediction selectively, as described below.

Salient objects presented in an input image are regarded as the attributes of this image, which is encoded into an attribute information vector. To be specific, each pattern of the attribute vector represents a certain presence of objects in an image. For example, if a bee is busying itself about a peony in the image where both the bee and peony are annotated as salient objects, then the attribute vector is expected to partially differ from that of another example in which a butterfly is flitting about over the same peony. Unlike one-hot encoding in image classification, each

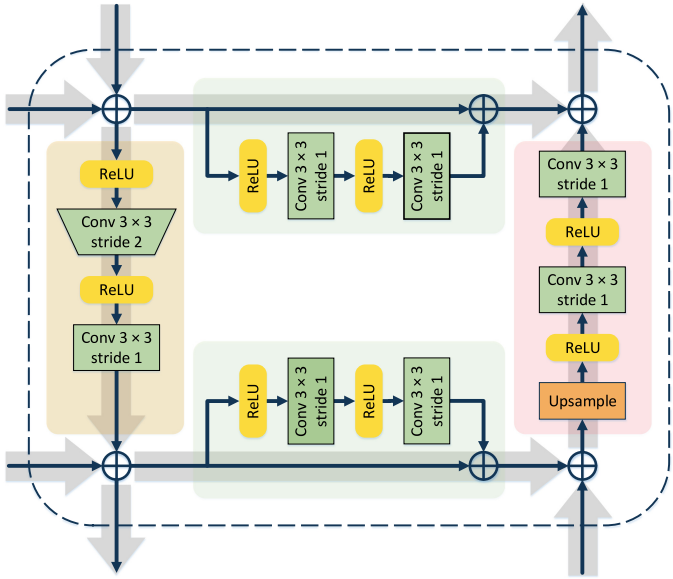


Fig. 5. Detailed schema of the zoom on the dashed box in Figure 4. Here, *Conv* $n \times n$ *stride* s denotes a convolutional layer with a kernel of $n \times n$ and stride s. *Upsample* is 2× bilinear interpolation. Symbol $\oplus$ represents element-wise summation. Gray arrows indicate the information streams.

dimension of our vector is independent and the dimension number is set to 64 for a large enough solution space to handle most kinds of realistic situations.

Without multiple category-specific subnetworks, we generate saliency maps by a quite light-weight predictor that comprises only one 1×1 convolutional layer (denoted as $\otimes$ in Figure 3), following GAB. The shape of its input f_p is $64 \times H \times W$. To take advantage of the guidance of encoded attribute information, we derivate the element-wise product $f_p^{'}$ of the attribute vector (denoted as V) and f_p by expanding the size of V, whose dimension is 64, to reach the same size as f_p through replication. The attribute-embedded $f_p^{'}$ is adopted as the new input of our saliency predictor. Because the attribute vector activates some input features and depresses the others along the channel dimension simultaneously, we extend the capability of our network to a great degree, which is consistent with the intention of investigating category-specific saliency prediction. Our attribute encoding is somewhat similar with recent SENet [20]. For each input example, SENet squeezes intermediate features into importance weights, then reweights features by channel-wise multiplication. From another angle, our method encodes global information into an attribute vector, guiding the network to learn which features should be used. The guideline information is independent of input features, which is derived from a special attribute encoding subnetwork.

Owing to the classification capability of the deep backbone network, our attribute **encoding network (AENet)** is reasonably simple. The complete structure is outlined in the right column of Table 3. We utilize two fully connected layers followed by two sequential activation functions, *Tanh* and *ReLU*. The latter scales the output to a range between 0 and 1 for feature suppression or excitation selectively. Dropout is exploited to avoid identical outputs for each input image during training. In the test phase, the dropout operation is deactivated.

3.4 Loss Functions

To obtain a finer prediction, we propose a novel constraint loss on predicted saliency maps, which is defined as follows:

$$L_{con} = e^{-\alpha \cdot x^2}, \tag{2}$$

where x refers to the output of our saliency predictor without nonlinear transformation by *sigmoid* function. α is a positive scaling weight, set to 0.25 in all experiments unless specifically mentioned. L_{con} approaches its minimum when x lies far from zero. In this case, L_{con} penalizes those equivocal pixels, which usually appear at the edges or fall in the regions with complicated textures. Finally, the formulation of the total loss is given as below:

$$L = L_{sal} + \sum_{k=0}^{K} L_{res}^{(k)} + \sum_{k=1}^{K} L_{int}^{(k)} + \beta \cdot L_{con}, \tag{3}$$

where L_{sal} represents a common cross-entropy loss between the final saliency map and the ground truth, L_{res} is the loss in residual learning, and L_{int} is used for deep intermediate supervision. K is set to 4 for the refining step $0 \sim 3$, and β regards to a hyper-parameter to balance cross-entropy losses and the proposed constraint loss. With repeated experimental verification, the network achieves the best performance when β is set to 0.5.

To verify the effectiveness of the constraint loss L_{con}, we investigate other training practices to improve precise saliency learning for comparison. Rethinking of focal loss [32], focal loss fairly remedies the imbalance of the positive and negative examples, and at the same time, down-weights easy examples, thus focuses training on those hard negative ones. For a similar purpose, another feasible strategy is to train the network with **online hard example mining (OHEM)** [47]. In Section 4.3.2, we compare the performance of ours with those based on OHEM, focal loss, as well as a variant of focal loss that is assigned $e^{-\gamma \cdot |x|}$ as the modulating factor of the cross-entropy loss.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets. We evaluate our method on five widely used public benchmarks: ECSSD [46], PASCAL-S [30], HKU-IS [29], DUT-OMRON [62], and DUT-S [51]. Initially, we give a brief description of these five benchmark datasets as follows: ECSSD contains 1,000 images with definite semantical meanings and complex structures as well. PASCAL-S is a well-collected subset of PASCAL VOC 2010 *val set* [9], and it contains 850 natural scenes picked on eye tracker. For HKU-IS, all of 4,447 images are carefully selected based on at least one of the following criteria: multiple disconnected salient objects, salient objects touching the image boundary, or low color contrast. DUT-OMRON has 5,168 challenging images in cluttered background. DUT-S is another large-scale complicated dataset including 10,553 images for training and 5,019 images for test. All the datasets above are elaborately annotated by human for pixel-wise saliency.

4.1.2 Evaluation Criteria. We evaluate our proposed method and other recent algorithms through three representative evaluation metrics: **precision-recall (PR)** curve, F-measure, and **mean absolute error (MAE)**. By binarizing the continuous saliency map with an incremental threshold in $[0, 255]$, we obtain the binary saliency map. Then PR curve is plotted by comparing the binary saliency map with the ground truth. F-measure is an overall performance measure calculated by precision, recall, as well as a balance weight β:

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}, \tag{4}$$

where β^2 is set to 0.3 to emphasize the precision. Given a predicted saliency map S and the corresponding ground truth G, MAE is computed by averaging the pixel-wise absolute difference between S and G:

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |S(i,j) - G(i,j)|, \tag{5}$$

where $S(i,j)$ is the continuous saliency value at position (i,j) and $G(i,j)$ is its ground truth. H and W denote the height and width of the saliency map S, respectively.

4.1.3 Implementation Details. As mentioned before, we employ the ImageNet [26] pre-trained VGG-16 as our backbone, and the parameters of other layers are initialized by random assignment with a zero mean and a variance of 0.0001. The network is optimized by SGD with a momentum 0.9, a weight decay of 0.0005, and a batch size at 10. We train the network with 60K iterations in total, while the initial learning rate is set to 0.001, which is decreased at 45K iterations and again at 55K iterations by the factor of 0.2. To avoid divergence, we utilize the warm-up method during training. DUT-S *train set* is chosen as the training set in which images are randomly horizontally flipped without extra data augmentation.

4.2 Performance Comparison

We evaluate our method on five widely used salient object detection datasets described above and compare against 15 previous state-of-the-art approaches: Amulet [67], DSS [19], RAS [4], DGRL [54], PAGR [69], BMPM [66], PiCANet [35], MLMSNet [56], AFNet [10], CPD [57], PoolNet [33], UnifiedCRF [64], BANet [49], TSPOANet [38], and EGNet [70]. We run the inference on a machine with a single NVIDIA GTX 1080Ti GPU and Intel(R) Xeon(R) E5-2640v4 (2.40 GHz) CPU at a speed of 25 FPS, where the size of the input image is 400×400. For a fair comparison, we exploit the released implementation with the recommended parameters or the pre-computed saliency maps provided by the authors. The results of all the methods are evaluated with the same evaluation code.[2] Note that the comparison does **not** take the ResNet-based frameworks into account. A more powerful backbone, auxiliary data, or post processing may be favorable for the performance, but it is out of the scope of this article.

Quantitative comparison is reported in Figure 6 and Table 4. From PR curves and F-measure curves, we can observe that our method is generally better than other methods, especially on ECSSD, PASCAL-S, HKU-IS, and DUT-S, while achieving comparable performance on DUT-OMRON with state-of-the-arts except at low level of recall ($recall < 0.7$). In terms of F-measure and MAE, our method outperforms the competing methods on the five datasets except for DUT-OMRON, especially, which has an advantage in MAE criteria by reason of our constraint loss. It is worth pointing out that PoolNet [33] performed joint training using the salient object detection dataset and an external edge detection dataset. So, we show the results of this method trained with only the salient object detection dataset and the numbers are provided by authors [33]. Moreover, the second-best method, i.e., EGNet [70], also optimized the salient edge detection and the salient object detection jointly to improve the results of salient object detection, which required a pre-processing procedure to generate the edge information from salient objects. In general, we train our model without needing extra edge data or pre-processing, surpassing or at least achieving comparable performance compared with previous state-of-the-art ones.

Figure 7 provides the visualized comparison of our method with respect to seven representative approaches. These examples are challenging because of complex background, low contrast,

[2] https://github.com/Andrew-Qibin/SalMetric.

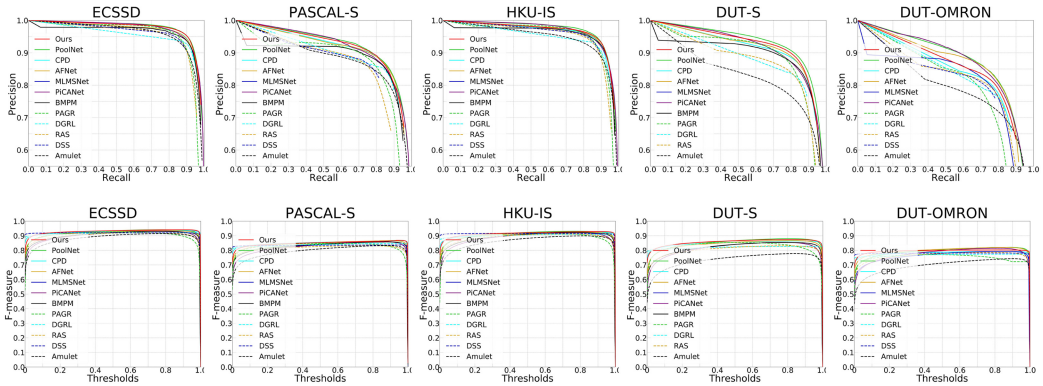


Fig. 6. Comparison of precision-recall (PR) curves (the first row) and F-measure curves (the second row) on five benchmarks. Here, the thresholds ($0 \sim 255$) on F-measure curves are normalized to $[0, 1]$.

Table 4. Performance Comparison with State-of-the-art Methods on Five Benchmarks

Methods	ECSSD		PASCAL-S		HKU-IS		DUT-S		DUT-OMRON	
	$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE
Amulet [67]	0.915	0.059	0.833	0.098	0.899	0.050	0.778	0.084	0.743	0.098
DSS [19]	0.921	0.052	0.836	0.094	0.916	0.040	-	-	0.781	0.063
RAS [4]	0.921	0.056	0.831	0.101	0.913	0.045	0.831	0.059	0.786	0.062
DGRL [54]	0.922	0.041	0.849	0.072	0.911	0.036	0.828	0.050	0.774	0.062
PAGR [69]	0.927	0.061	0.849	0.089	0.919	0.048	0.854	0.055	0.771	0.071
BMPM [66]	0.930	0.045	0.858	0.074	0.922	0.039	0.854	0.048	-	-
PiCANet [35]	0.932	0.047	0.861	0.078	0.922	0.042	0.855	0.053	0.815	0.068
MLMSNet [56]	0.930	0.045	0.858	0.074	0.922	0.039	0.854	0.048	0.793	0.064
AFNet [10]	0.935	0.042	0.866	0.070	0.926	0.036	0.867	0.045	0.820	0.057
CPD [57]	0.936	0.040	0.867	0.072	0.925	0.033	0.864	0.043	0.794	0.057
PoolNet [33]	0.937	0.045	0.858	0.078	0.928	0.035	0.876	0.043	0.817	0.058
UnifiedCRF [64]	0.928	0.049	0.858	0.089	0.920	0.039	-	-	0.802	0.057
BANet [49]	0.935	0.041	0.859	0.077	0.920	0.036	0.852	0.046	0.793	0.061
TSPOANet† [38]	0.887	0.052	0.825	0.075	0.880	0.039	0.799	0.048	0.703	0.063
EGNet [70]	0.941	0.044	0.863	0.076	**0.930**	0.034	**0.880**	0.043	**0.826**	0.056
Ours	**0.941**	**0.038**	**0.869**	**0.069**	0.929	**0.031**	0.874	**0.041**	0.803	**0.054**

$maxF_\beta$ denotes **max** F-measure for the best performance that methods can achieve. The best and the second-best results are highlighted in **red** and blue, respectively. † denotes that the work did **not** release codes or saliency maps, thus, we adopt the numbers provided in Reference [38].

multiple saliency, tiny instances, semantic ambiguity, and so on. Despite all these handicaps, it can be observed that our method produces more accurate saliency maps than others in the following aspects: (*a*) highlighted correct regions, (*b*) less missed detection, (*c*) lower false alarm, (*d*) sharper boundaries, and (*e*) clearer details. Nevertheless, there are still some unsatisfactory results shown in the last row. A possible reason is that the salient target (the snake) is presented in rich-texture background (the stone) and both of them are pretty similar in appearance. The existing methods can not handle this difficult case, which is left for the future work.

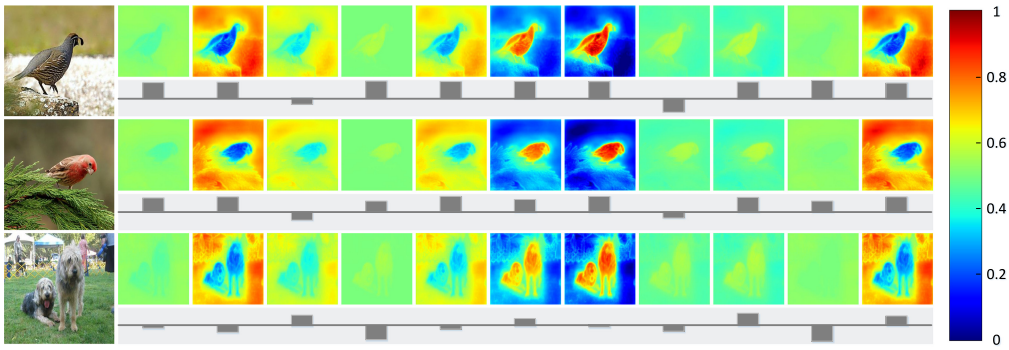


Fig. 7. Visualization of attribute-weighted features. The feature maps are processed with a pseudo-color enhancement for better visualization while the activations are scaled to $[0, 1]$. For each feature map, the corresponding dimension of the attribute vector is shown below. We normalize the quantitative values to present a striking contrast by subtracting the dimension-wise mean and dividing by the standard deviation.

Table 5. Ablation Analysis on ECSSD, PASCAL-S, HKU-IS, DUT-S, and DUT-OMRON

DIS	GAB	AE	ECSSD		PASCAL-S		HKU-IS		DUT-S		DUT-OMRON	
			$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE
			0.923	0.042	0.851	0.073	0.914	0.037	0.857	0.045	0.790	0.058
✓			0.933	0.040	0.859	0.071	0.919	0.035	0.866	0.043	0.798	0.055
✓	✓		0.936	0.040	0.863	0.070	0.921	0.034	0.868	0.042	0.801	0.054
✓	✓	✓	**0.941**	**0.038**	**0.869**	**0.069**	**0.929**	**0.031**	**0.874**	**0.041**	**0.803**	**0.054**

DIS: the deep intermediate supervision; **GAB**: the grid aggregation block; **AE**: the attribute encoding for saliency prediction.

4.3 Discussion

4.3.1 Ablation Analysis of Appended Modules. In this section, we study the impact of each module in the proposed network. In the beginning, we train a preliminary network with the **global context block (GCB)** as the baseline, where the missing **grid aggregation block (GAB)** is replaced by the same unit adopted on side-outputs 1~3. Since the **deep intermediate supervision (DIS)** and the **attribute encoding network (AENet)** are "plug and play" units, we simply abandon them for the basic baseline performance. As shown in Table 5, the growing performance is achieved along with the modules equipped on the network one-by-one.

As we can see in Table 5, after adding the intermediate supervision on side-outputs 1~3, we obtain a boost (0.5%~1.0% for F-measure) of the performance, which proves the effectiveness of our proposed residual refinement. As we mentioned in Section 3.2.3, the improvement is close to saturation when adopting residual refinement on side-output 4. GAB is proposed to further refine the saliency maps by (*a*) making information streams cooperate, (*b*) merging different-scale features, and (*c*) introducing attention mechanism. Here, we evaluate the influence of GAB as shown in Table 5: GAB raises F-measure by about 0.3% and lowers MAE by 0.001 on average. At last, we report the performance of the complete network equipped with AENet. The increased numbers demonstrate AENet plays an important role in our network, especially helpful to eliminate error between the predicted saliency map and the ground truth.

Table 6. Performance Comparison of Various Practices on ECSSD, PASCAL-S, HKU-IS, DUT-S, and DUT-OMRON

	ECSSD		PASCAL-S		HKU-IS		DUT-S		DUT-OMRON	
	$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE	$maxF_\beta$	MAE
$baseline$	0.933	0.041	0.863	0.078	0.921	0.034	0.869	0.046	0.799	0.058
$OHEM$	0.938	0.041	0.863	0.077	0.925	0.034	0.866	0.047	0.796	0.059
$focal_{org}$	0.933	0.042	0.866	0.074	0.922	0.035	0.870	0.046	0.801	0.055
$focal_{var}$	0.934	0.042	0.864	0.075	0.922	0.034	0.871	0.044	0.800	0.055
L_{con}	**0.941**	**0.038**	**0.869**	**0.069**	**0.929**	**0.031**	**0.874**	**0.041**	**0.803**	**0.054**

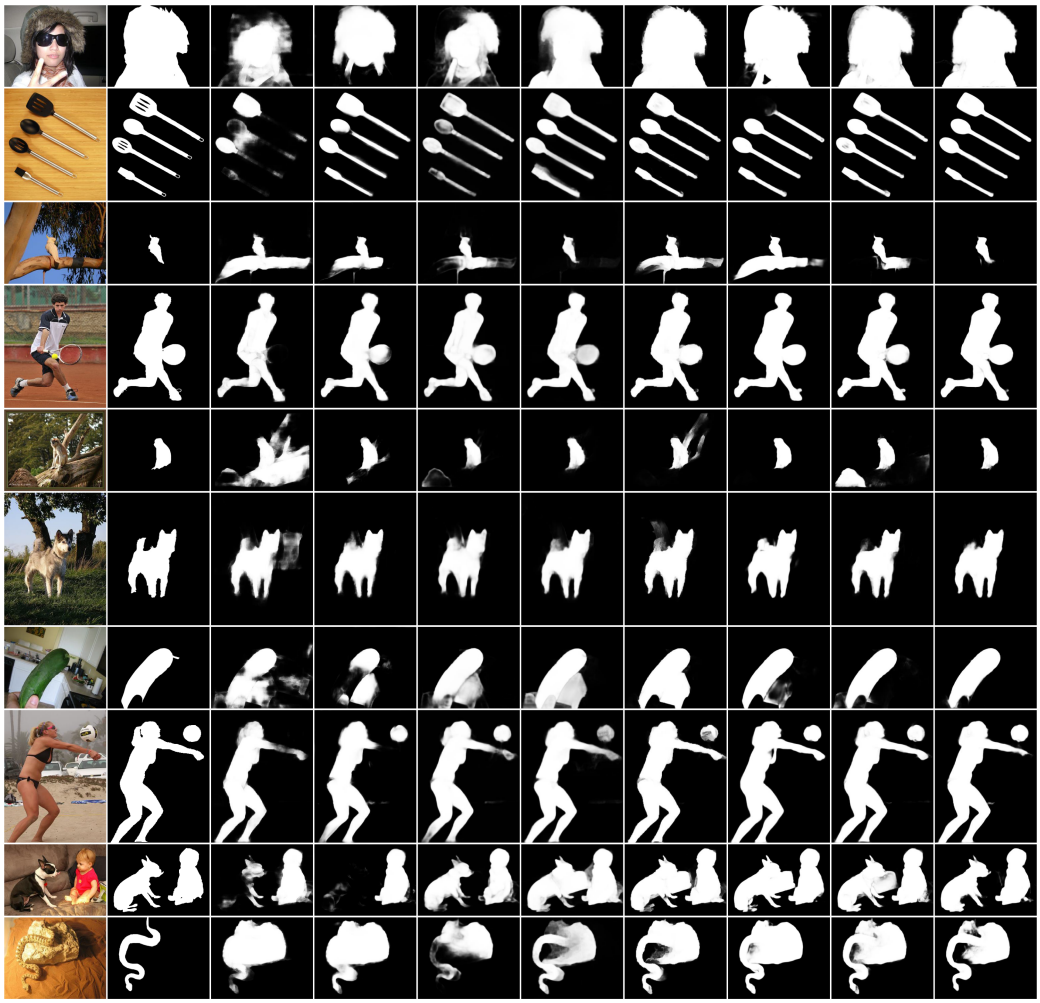
$baseline$ refers that we train the network without bells and whistles, $focal_{org}$ and $focal_{var}$ represent original focal loss and its variant mentioned in Section 3.4.

4.3.2 Additional Constraint Loss. Besides conventional cross-entropy loss, we add a regularization term L_{con} as an additional constraint loss. For comparison, we investigate three comparative training practices in addition to the baseline to give a better understanding of the proposed L_{con}. **Online hard example mining (OHEM)** is used for contrast, by which we emphasize the hard examples to improve the performance on borderline cases. OHEM has a similar effect with L_{con} on those uncertain pixels. The former only back-propagates losses with high values while the latter penalizes these borderline cases numerically. In the concrete implementation, we take the top k percent of pixel-wise losses to update the parameters of the network and traverse all values of k to pursue the most superior performance, which is reported in Table 6. Besides this, focal loss and its variant are adopted in training for the sake of comparison. Results are listed in Table 6. Our constraint loss achieves the best performance.

4.3.3 Visualization of Attribute-weighted Feature Maps. Figure 8 shows three visual examples: for each one, the top 11 weighted feature maps are exhibited in the upper row and the attentive value of the attribute vector for each corresponding feature map is plotted below. As can be seen, for the examples with quite different content (the second *vs.* the third), their attribute vectors are of great difference in certain dimensions, whereas the similar examples (the first *vs.* the second) share a similar distribution pattern, roughly. This meets our expectation that AENet exports discrepant information according to different salient objects. Additionally, the more distinct and correct feature maps are highlighted by the attribute vector; in contrast, the mussy and ambiguous ones are suppressed. As an instance, the first and the second map in the first row are reweighted by the attribute values. Since the attribute vector is nonnegative, the larger values reinforce the feature maps, resulting in larger activations. These feature maps are of greater benefit for the final results.

5 CONCLUSION

In this article, we present an improved residual refinement network to produce precise saliency prediction step-by-step. To achieve an explicit division of labor for global estimation and residual learning, we introduce the element-wise subtraction modules and deep intermediate supervision. For integration of multi-level contextual information, a grid aggregation block (GAB) is equipped on the network to merge four side-outputs and a predicted saliency map. To extend the capability of the network, we propose an attribute encoding network (AENet) to guide saliency predictor. We further add a novel constraint loss in training for more accurate results. Extensive experiments demonstrate that our method surpasses or compares favorably against state-of-the-art methods on all the five public benchmarks.



(a) Img (b) GT (c) Amulet (d) RAS (e) PAGR (f) PiCANet (g) AFNet (h) CPD (i) PoolNet (j) **Ours**

Fig. 8. Qualitative comparison of the proposed method with seven representative approaches. GT denotes the ground truth of the leftmost image. The saliency maps generated by each method are exhibited on the third to the last column.

REFERENCES

[1] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. 2015. Salient object detection: A benchmark. *IEEE Trans. Image Process.* 24, 12 (2015), 5706–5722.

[2] Ali Borji and Laurent Itti. 2012. Exploiting local and global patch rarities for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 478–485.

[3] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. 2019. GCNet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv preprint arXiv:1904.11492* (2019).

[4] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. 2018. Reverse attention for salient object detection. In *Proceedings of the European Conference on Computer Vision*. 234–250.

[5] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. 2014. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 3 (2014), 569–582.

[6] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. 2018. Paying more attention to saliency: Image captioning with saliency and context attention. *ACM Trans. Multimedia Comput., Commun. Applic.* 14, 2 (2018), 1–21.
[7] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng. 2018. R3Net: Recurrent residual refinement network for saliency detection. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 684–690.
[8] Yuanyuan Ding, Jing Xiao, and Jingyi Yu. 2011. Importance filtering for image retargeting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 89–96.
[9] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2015. The Pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* 111, 1 (2015), 98–136.
[10] Mengyang Feng, Huchuan Lu, and Errui Ding. 2019. Attentive feedback network for boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1623–1632.
[11] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Tremeau, and Christian Wolf. 2017. Residual conv-deconv grid network for semantic segmentation. In *Proceedings of the British Machine Vision Conference*.
[12] Keren Fu, Qijun Zhao, and Irene Yu-Hua Gu. 2018. Refinet: A deep segmentation assisted refinement network for salient object detection. *IEEE Trans. Multimedia* 21, 2 (2018), 457–469.
[13] Yue Gao, Meng Wang, Dacheng Tao, Rongrong Ji, and Qionghai Dai. 2012. 3-D object retrieval and recognition with hypergraph analysis. *IEEE Trans. Image Process.* 21, 9 (2012), 4290–4303.
[14] Genliang Guan, Zhiyong Wang, Shaohui Mei, Max Ott, Mingyi He, and David Dagan Feng. 2014. A top-down approach for video summarization. *ACM Trans. Multim. Comput., Commun. Applic.* 11, 1 (2014), 1–21.
[15] Junfeng He, Jinyuan Feng, Xianglong Liu, Tao Cheng, Tai-Hsu Lin, Hyunjin Chung, and Shih-Fu Chang. 2012. Mobile product search with bag of hash bits and boundary reranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3005–3012.
[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. 2961–2969.
[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[18] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. 2015. Online tracking by learning discriminative saliency map with convolutional neural network. In *Proceedings of the International Conference on Machine Learning*. 597–606.
[19] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. 2017. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3203–3212.
[20] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7132–7141.
[21] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (1998), 1254–1259.
[22] Bowen Jiang, Lihe Zhang, Huchuan Lu, Chuan Yang, and Ming-Hsuan Yang. 2013. Saliency detection via absorbing Markov chain. In *Proceedings of the IEEE International Conference on Computer Vision*. 1665–1672.
[23] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. 2013. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2083–2090.
[24] Jiwhan Kim, Dongyoon Han, Yu-Wing Tai, and Junmo Kim. 2014. Salient region detection via high-dimensional color transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 883–890.
[25] Dominik A. Klein and Simone Frintrop. 2011. Center-surround divergence of feature statistics for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2214–2219.
[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 1097–1105.
[27] Jason Kuen, Zhenhua Wang, and Gang Wang. 2016. Recurrent attentional networks for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3668–3677.
[28] Guanbin Li and Yizhou Yu. 2015. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5455–5463.
[29] Guanbin Li and Yizhou Yu. 2016. Visual saliency detection based on multiscale deep CNN features. *IEEE Trans. Image Process.* 25, 11 (2016), 5012–5024.
[30] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, and Alan L. Yuille. 2014. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 280–287.

[31] Zun Li, Congyan Lang, Yunpeng Chen, Junhao Liew, and Jiashi Feng. 2019. Deep reasoning with multi-scale context for salient object detection. *arXiv preprint arXiv:1901.08362* (2019).
[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2980–2988.
[33] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. 2019. A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3917–3926.
[34] Nian Liu and Junwei Han. 2016. DHSNet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 678–686.
[35] Nian Liu, Junwei Han, and Ming-Hsuan Yang. 2018. PiCANet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3089–3098.
[36] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. 2010. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 2 (2010), 353–367.
[37] Yun Liu, Yu Qiu, Le Zhang, JiaWang Bian, Guang-Yu Nie, and Ming-Ming Cheng. 2018. Salient object detection via high-to-low hierarchical context aggregation. *arXiv preprint arXiv:1812.10956* (2018).
[38] Yi Liu, Qiang Zhang, Dingwen Zhang, and Jungong Han. 2019. Employing deep part-object relationships for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 1232–1241.
[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3431–3440.
[40] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin. 2017. Non-local deep features for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6609–6617.
[41] Xiongkuo Min, Guangtao Zhai, Ke Gu, and Xiaokang Yang. 2016. Fixation prediction through multimodal analysis. *ACM Trans. Multim. Comput., Commun. Applic.* 13, 1 (2016), 1–23.
[42] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. 2019. BASNet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7479–7489.
[43] Rong Quan, Junwei Han, Dingwen Zhang, Feiping Nie, Xueming Qian, and Xuelong Li. 2017. Unsupervised salient object detection via inferring from imperfect saliency models. *IEEE Trans. Multimedia* 20, 5 (2017), 1101–1112.
[44] Gaurav Sharma, Frédéric Jurie, and Cordelia Schmid. 2012. Discriminative spatial saliency for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3506–3513.
[45] Xiaohui Shen and Ying Wu. 2012. A unified approach to salient object detection via low rank matrix recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 853–860.
[46] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. 2015. Hierarchical image saliency detection on extended CSSD. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 4 (2015), 717–729.
[47] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 761–769.
[48] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[49] Jinming Su, Jia Li, Yu Zhang, Changqun Xia, and Yonghong Tian. 2019. Selectivity or invariance: Boundary-aware salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 3799–3808.
[50] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. 2015. Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3183–3192.
[51] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. 2017. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 136–145.
[52] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. 2016. Saliency detection with recurrent fully convolutional networks. In *Proceedings of the European Conference on Computer Vision*. Springer, 825–841.
[53] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. 2018. Salient object detection with recurrent fully convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 7 (2018), 1734–1746.
[54] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, and Ali Borji. 2018. Detect globally, refine locally: A novel approach to saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3127–3135.
[55] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven C. H. Hoi, and Ali Borji. 2019. Salient object detection with pyramid attention and salient edges. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1448–1457.

[56] Runmin Wu, Mengyang Feng, Wenlong Guan, Dong Wang, Huchuan Lu, and Errui Ding. 2019. A mutual learning method for salient object detection with intertwined multi-supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8150–8159.

[57] Zhe Wu, Li Su, and Qingming Huang. 2019. Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3907–3916.

[58] Zhe Wu, Li Su, and Qingming Huang. 2019. Stacked cross refinement network for edge-aware salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 7264–7273.

[59] Huaxin Xiao, Jiashi Feng, Yunchao Wei, Maojun Zhang, and Shuicheng Yan. 2018. Deep salient object detection with dense connections and distraction diagnosis. *IEEE Trans. Multimedia* 20, 12 (2018), 3239–3251.

[60] Yulin Xie, Huchuan Lu, and Ming-Hsuan Yang. 2012. Bayesian saliency via low and mid level cues. *IEEE Trans. Image Process.* 22, 5 (2012), 1689–1698.

[61] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*. 2048–2057.

[62] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. 2013. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3166–3173.

[63] Linwei Ye, Zhi Liu, Lina Li, Liquan Shen, Cong Bai, and Yang Wang. 2017. Salient object segmentation via effective integration of saliency and objectness. *IEEE Trans. Multimedia* 19, 8 (2017), 1742–1756.

[64] Xu Yingyue, Xu Dan, Hong Xiaopeng, Ouyang Wanli, Ji Rongrong, Xu Min, and Zhao Guoying. 2019. Structured modeling of joint deep feature and prediction refinement for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 3788–3797.

[65] Jun Zhang, Meng Wang, Liang Lin, Xun Yang, Jun Gao, and Yong Rui. 2017. Saliency detection on light field: A multi-cue approach. *ACM Trans. Multim. Comput., Commun. Applic.* 13, 3 (2017), 1–22.

[66] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. 2018. A bi-directional message passing model for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1741–1750.

[67] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. 2017. Amulet: Aggregating multi-level convolutional features for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 202–211.

[68] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. 2017. Learning uncertain convolutional features for accurate saliency detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 212–221.

[69] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. 2018. Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 714–722.

[70] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. 2019. EGNet: Edge guidance network for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 8779–8788.

[71] Ting Zhao and Xiangqian Wu. 2019. Pyramid feature attention network for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3085–3094.

[72] Yuan Zhou, Ailing Mao, Shuwei Huo, Jianjun Lei, and Sun-Yuan Kung. 2018. Salient object detection via fuzzy theory and object-level enhancement. *IEEE Trans. Multimedia* 21, 1 (2018), 74–85.

[73] Yunzhi Zhuge, Yu Zeng, and Huchuan Lu. 2019. Deep embedding features for salient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9340–9347.

Received April 2020; revised October 2020; accepted December 2020