SAC-Net: Spatial Attenuation Context for Salient Object Detection

Xiaowei Hu\textsuperscript{1}, Chi-Wing Fu\textsuperscript{1,2}, Lei Zhu\textsuperscript{1}, and Pheng-Ann Heng\textsuperscript{1,2}

\textsuperscript{1}Department of Computer Science and Engineering, The Chinese University of Hong Kong
\textsuperscript{2}Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

Abstract

This paper presents a new deep neural network design for salient object detection by maximizing the integration of local and global image context within, around, and beyond the salient objects. Our key idea is to adaptively propagate and aggregate the image context with variable attenuation over the entire feature maps. To achieve this, we design the spatial attenuation context (SAC) module to recurrently translate and aggregate the context features independently with different attenuation factors and then attentively learn the weights to adaptively integrate the aggregated context features. By further embedding the module to process individual layers in a deep network, namely SAC-Net, we can train the network end-to-end and optimize the context features for detecting salient objects. Compared with 22 state-of-the-art methods, experimental results show that our method performs favorably over all the others on six common benchmark data, both quantitatively and visually.

1. Introduction

Salient object detection is an effective and useful preprocessing step in many computer vision tasks, e.g., object segmentation \cite{48} and tracking \cite{18}, video compression \cite{14} and abstraction \cite{58}, as well as saliency-aware image editing \cite{8} and texture smoothing \cite{62}. It is a fundamental problem in computer vision research and has been extensively studied in the past decade.

Early works attempt to detect salient objects based on low-level cues like contrast, color, and texture \cite{7, 21, 34}. However, relying on low-level cues is clearly inadequate to finding salient objects, which involve high-level semantics. Hence, most recent methods \cite{29, 35, 44, 56, 61} employ convolutional neural networks (CNNs) and take a data-driven approach to the problem by leveraging both high-level semantics and low-level details extracted from multiple CNN layers \cite{6, 10, 19, 20, 27, 28, 54, 55, 57}. However, since the convolution operator in CNN processes a local neighborhood in the spatial domain \cite{49}, existing methods tend to miss global spatial semantics in the results, e.g., they may misrecognize background noise as salient objects; see Section 4.2 for quantitative and qualitative comparisons.

Essentially, salient objects are key elements that stand out from the background. Such an inference process \cite{13} involves not only the local image context within and around the salient objects, but also the global image context, as well as suitable integration of the various context. Ideally, after extracting context features per image pixel, if we can connect all these features and let them communicate with every other over the spatial domain, we can optimize the feature integration for maximized performance. However, it is clearly infeasible in practice. Hence, we propose to propagate context features independently with different attenuation factors and learn to aggregate the resulting features adaptively; by then, our network can learn to detect salient objects by adaptively considering context features within, around, and even far from, the salient objects.

Figure 1 shows a challenging example with the associated attention maps learned in our network for integrating the various image context: long-range context aggregated...
with a small attenuation factor (c) to help locate the global background; medium-range context (d) to help identify the image regions of the same object; and short-range context aggregated with a large attenuation factor (e) to help locate the boundary between salient and non-salient regions.

To this end, we formulate the spatial attenuation context module, or SAC module for short, in a deep network to allow the image features in a CNN to propagate over variable spatial ranges by articulating different attenuation factors in the propagation. Our module has two rounds of recurrent translations to propagate and aggregate the image features. In each round, we propagate features independently using different attenuation factors towards different directions in the spatial domain; further, we formulate an attention mechanism to learn the weights to combine the aggregated features. Hence, we can adopt different attenuation factors (or influence ranges) for different image features. Furthermore, we deploy a SAC module in each layer of our network and predict a saliency map per layer based on the output from the SAC module and the convolutional features. Below, we summarize the major contributions of this work:

- First, we design the spatial attenuation context (SAC) module to recurrently propagate the image features over the whole feature maps with variable attenuation factors and learn to adaptively integrate the features through an attention mechanism in the module.
- Second, we adopt the SAC module in each layer of our network architecture to learn the spatial attenuation context in different layers, and train the whole network in an end-to-end manner for salient object detection.
- Lastly, we evaluate our method and compare it against 22 state-of-the-art methods on six common benchmark data. Results show that our method performs favorably over all the others for all the benchmark data.

2. Related Work

Rather than being comprehensive, we discuss mainly the methods on single-image salient object detection. Early methods use hand-crafted priors such as image contrast [21, 39], color [2, 36], texture [51, 52], and other low-level visual cues [15]; see [1] for a survey. Clearly, hand-crafted features are insufficient to capture high-level semantics, so methods based on them often fail for nontrivial inputs.

Recent works [29, 35, 44, 56, 61] exploit convolutional neural networks (CNN) to learn deep features for detecting salient objects. Zhao et al. [61] used a fully-connected CNN to find and combine global and local features to predict saliency maps. Wang et al. [44] developed a recurrent CNN with the prediction map from previous recurrent step as the guidance. Zhang et al. [56] adopted a dropout technique to learn deep uncertain convolutional features in the network to enhance its generalization capability. However, since these methods just take features at deep CNN layers, they tend to miss the details in the salient objects, which are captured mainly in the shallow layers.

Several recent works [10, 19, 20, 27, 54, 55, 57] enhance the detection quality by further integrating features in multiple CNN layers to simultaneously leverage more global and local context in the inference process. Among them, Li et al. [27] explored the semantic properties and visual contrast of salient objects, Hou et al. [19] created short connections to integrate features in different layers, while Zhang et al. [55] derived a resolution-based feature combination module and a boundary-preserving refinement strategy. Later, Deng et al. [10] adopted residual learning to alternatively refine features at deep and shallow layers. Zhang et al. [54] formulated a bi-directional message passing model to select features for integration. Zhang et al. [57] designed an attention-guided network to progressively select and integrate multi-level information. Li et al. [28] used a two-branch network to simultaneously predict the contours and saliency maps. Chen et al. [6] leveraged residual learning and reverse attention to refine the saliency maps. Although the detection quality keeps improving, the exploration of global spatial context, particularly in the shallow layers, is still heavily limited by the convolution operator in CNN, which is essentially a local spatial filter [49].

Very recently, Wang et al. [46] explored the global and local spatial relations in deep networks to locate salient objects and refine the object boundary. Liu et al. [33] aggregated the attended contextual features from a global/local view in feature maps of varying resolutions. Even the detection performance continues to improve on the benchmarks [26, 30, 37, 43, 51, 52], current methods may still miss local parts in salient objects and misrecognize noises in non-salient regions as salient objects. Beyond the recent works that emphasize the importance of reasoning spatial context for salient object detection, we leverage and selectively aggregate surrounding image context spatially in the same CNN layer by attentively allowing the context features to recurrently translate with varying attenuation factors.

3. Methodology

Figure 2 outlines the architecture of our spatial attenuation context network (SAC-Net), which takes a whole image as input and predicts the saliency map in an end-to-end manner. First, we use a CNN to generate feature maps in different resolutions and progressively propagate the image features at deep layers to feature maps at shallow layers to construct a feature pyramid [31]. After that, we use our SAC modules to harvest spatial attenuation context per layer and concatenate the module outputs with the corresponding convolutional features. Lastly, we predict a result per layer, upsample and merge it with the shallower-layer output, and take the result of the largest resolution as the
the weights to combine the recurrently-aggregated results in different resolutions from the input image using a convolutional neural network; (ii) construct a feature pyramid (in green) by successively upsampling the feature map at a deep layer and combining the upsampled result with the feature map at an adjacent shallower layer; (iii) use SAC modules (see Figure 4) to generate spatial attenuation context features for each layer; (iv) concatenate the outputs from the SAC modules with the convolutional features (in red); and (v) lastly, successively predict a saliency map at each layer and take the final saliency map of the largest resolution as the network output. In the figure, feature maps are indicated by blocks and thicker blocks of smaller sizes are higher-level features at deeper layers.

3.1. Spatial Attenuation Context Module

Figure 4 shows the architecture of the spatial attenuation context module, or SAC module, which takes a feature map as input and produces spatial attenuation context in the same resolution. As presented earlier, the spatial attenuation context contains image context aggregated by propagating local image context using varying attenuation factors via an attention mechanism; hence, we can disperse the local image context adaptively over the whole feature maps.

See again the SAC module in Figure 4. First, we use a 1 × 1 convolution on the input feature map to reduce the number of feature channels. Then, we adopt recurrent translations with varying attenuation factors (αk) to disperse the local image features in four different directions; see the illustration in Figure 3(b). After two rounds of recurrent translations, we adaptively disperse the local features over the 2D domain; see Figure 3(c). More importantly, we learn the weights to combine the recurrently-aggregated results via an attention mechanism in an end-to-end manner, so each pixel in the SAC module output can receive spatial context adaptively from its surroundings.

Recurringly-attenuating translation. To optimize the dispersal of local context, we first formulate a parametric model to recurrently aggregate the image features with attenuation. Given the feature map after a 1 × 1 convolution (see Figure 4), we recurrently translate its features using different attenuation factors αk in four principal directions: left, up, right, and down. Moreover, to ensure manageable memory consumption, we set the number of feature channels in each recurrently-aggregated feature map as \( \left\lfloor \frac{256}{n} \right\rfloor \), where n is the number of different attenuation factors in the SAC module; see Table 3 for an experiment on n.

Denoting \( f_{i,j} \) as the feature at pixel \( (i, j) \) in a feature map, our recurrently-attenuating translation process propagates features progressively over the spatial domain using the following equation (typically in the up direction):

\[
\begin{align*}
\alpha_k & = \frac{n-k}{n} \quad (k \in \{1, 2, ..., n\}) \\
\beta & \in [0, 1] \quad \text{is a parameter in our recurrently-attenuating translation model.}
\end{align*}
\]

In Eq. (1), we recurrently aggregate image features by using \( r_{i,j}^{up} \), where a smaller \( \alpha_k \) (close to zero) allows the features to propagate over a longer distance, while a larger \( \alpha_k \) (close to one) limits the propagation, so the related local features affect a smaller local area; see again the illustration in Figure 3. Moreover, when \( r_{i,j}^{up} < 0 \), the first term in \( f_{i,j}^{up} \) will become zero, and \( \beta \) will be multiplied with \( r_{i,j}^{up} \). We define \( \beta \) in Eq. (1) to reduce the feature magnitude when it is negative. Since we learn the value of \( \beta \) for each feature channel,
we can introduce nonlinearities when aggregating the spatial context and express more complex relations among the local features. Note that in our experiments, we initialize $\beta$ as 0.1 for all the feature channels and learn it automatically during the network training process; in practice, we found that $\beta$ rarely goes beyond one in our experiments.

**Attention mechanism.** After recurrently-translating the input feature map using different attenuation factors in four directions, we will obtain $4n$ feature maps; see the feature maps with colored arrows in Figure 4. As discussed earlier, the long-range image context reveals global semantics, while the short-range context helps identify the boundary between salient and non-salient regions. To adaptively leverage the complementary advantages of these aggregated spatial context features, we formulate an attention mechanism to learn the weights for selectively integrating them.

As shown at the top left corner in Figure 4, we take the input feature map $F$ as the input to the attention mechanism and produce a set of unnormalized attention weights $\{A_1, A_2, ..., A_n\}$, each corresponding to a particular attenuation factor; superscript 1 indicates that these weights are for the first round of recurrent translations. Then, we apply the Softmax function (Eq. (3)) to normalize the weights and produce the attention weight maps $\{W_1^1, W_2^1, ..., W_n^1\}$ associated with different attenuation factors (see Figure 4):

$$\{A_1, A_2, ..., A_n\} = \mathcal{F}_{attention}(F; \theta), \text{ and } (2)$$

$$w_{i,j,k}^1 = \frac{\exp(a_{i,j,k}^1)}{\sum_k \exp(a_{i,j,k}^1)}, \text{ and } (3)$$

where $a_{i,j,k}^1 \in A_k$ is the unnormalized attention weight at pixel $(i,j)$ for attenuation factor $\alpha_k, w_{i,j,k}^1 \in W_k^1$ are the normalized attention weights, and $\theta$ denotes the parameters learned by $\mathcal{F}_{attention}$, which consists of two $3 \times 3$ convolution layers and one $1 \times 1$ convolution layer, and we apply the group normalization [50] and ReLU non-linear operation [23] after the first two convolution layers.

Next, we multiply $W_k^1$ with the corresponding context features aggregated after the recurrent translations:

$$f_{i,j} = \oplus_{k=1}^n \left[ (f_{i,j}^{\text{up}}(\alpha_k, \beta) \oplus f_{i,j}^{\text{down}}(\alpha_k, \beta)) \oplus f_{i,j}^{\text{left}}(\alpha_k, \beta) \oplus f_{i,j}^{\text{right}}(\alpha_k, \beta) \right] \times w_{i,j,k}^1, \text{ (4)}$$

where $\times$ denotes an element-wise multiplication, $\oplus$ denotes the concatenation operator, and $\oplus_{k=1}^n$ concatenates all the feature maps for different attenuation factors, after the feature maps are multiplied with the attention weights ($w_{i,j,k}^1$).

With the attention weights learned to select and integrate the context features aggregated with different attenuation factors (see again Figure 1), our network can adaptively control the feature integration and allow the context features to be implicitly dispersed over varying spatial ranges.

**Completing the SAC module.** After concatenating the features, we complete the first round of recurrent translations in our SAC module and further apply a $1 \times 1$ convolution to reduce the feature channels. Then, we repeat the same process in the second round of recurrent translation using another set of attention weights $\{W_1^2, W_2^2, ..., W_n^2\}$, which are also learnt through the attention mechanism; see again Figure 4. After two rounds of recurrent translations, each pixel can obtain context features from the global domain adaptively aggregated with different attenuations; see

![Figure 4: The schematic illustration of the spatial attenuation context (SAC) module. We adopt two rounds of recurrent translations to propagate and aggregate image features. In each round, the colored arrows show the recurrent translation direction, while thicker (or thinner) arrows indicate stronger (or weaker) information propagation with less (or more) attenuation.](image-url)
We used six widely-used saliency benchmark datasets in our experiments: (i) ECSSD [51] has 1,000 natural images with many semantically meaningful but complex structures; (ii) PASCAL-S [30] has 850 images generated from the

Training parameters. We initialize the feature extraction part in our network (frontal blue blocks in Figure 2) using weights of ResNet-101 [16] trained on ImageNet [9], and initialize other network parts using random noise. Moreover, we use stochastic gradient descent to optimize the network with a momentum value of 0.9 and a weight decay of 0.0005, and we set the learning rate as $10^{-8}$, adjust it to be $10^{-9}$ after 13,000 training iterations, and stop the training after 20,000 iterations. Also, we horizontally flip the input images for data augmentation. Lastly, we train the network on a single NVidia Titan Xp GPU with a mini-batch size of one and update the weights in every ten training iterations.

Inference. We take the highest-resolution prediction as the overall result and refine the salient object boundary using fully-connected conditional random field (CRF) [22].

4. Experimental Results

4.1. Datasets and Evaluation Metrics

We used six widely-used saliency benchmark datasets in our experiments: (i) ECSSD [51] has 1,000 natural images with many semantically meaningful but complex structures; (ii) PASCAL-S [30] has 850 images generated from the
Figure 5: Visual comparison of saliency maps (c)-(h) produced by different methods. Apparently, our method produces more accurate saliency maps, where "*" indicates CRF is used as a post-processing step in the methods.

PASCAL VOC2010 segmentation dataset [11], where each image has several salient objects; (iii) SOD [38] has 300 images selected from the BSDS dataset [37], where the salient objects are typically of low contrast or closely contact with the image boundary; (iv) HKU-IS [26] has 4,447 images, where most images have multiple salient objects; (v) DUT-OMRON [52] has 5,168 high-quality images, each with one or more salient objects; and (vi) DUTS [43] has a training set of 10,553 images and a testing set (denoted as DUTS-test) of 5,019 images, where the images contain various number of salient objects with large variance in scale. Among the datasets, HKU-IS, DUT-OMRON, and DUTS provide a large number of test images captured under different situations, enabling more comprehensive comparisons among different methods. Moreover, we follow the recent works on salient object detection [33, 46, 54, 57] to train our network model using the training set of DUTS [43].

Next, we used three common metrics for quantitative evaluation: F-measure ($F_\beta$), structure measure ($S_m$) and mean absolute error (MAE), where a large $F_\beta$ or $S_m$ and a small MAE indicate a better result; see [12, 19] for their formulations. Also, we used the implementation of [12, 19] to compute $F_\beta$, $S_m$ and MAE for all results.

4.2. Comparison with the State-of-the-arts

We compared our method with 22 state-of-the-art methods; see the first column in Table 1. Among the methods, to detect salient objects, BSCA [40] and DRFI [21] use hand-crafted features, while others employ deep neural networks to learn features. For a fair comparison, we obtained their results either by using the saliency maps provided by the authors or by producing the results using their implementations with the released training models.

Quantitative comparison. Table 1 summaries the quantitative results compared with the 22 state-of-the-art methods in terms of $F_\beta$, $S_m$ and MAE on detecting salient objects in the six benchmark datasets. Our SAC-Net performs favorably against all the others for almost all the cases, regardless of whether CRF is used as a post-processing step. Especially, our method without CRF (SAC-Net) already achieves the best performance compared with all the other methods with CRF for most datasets.

Very importantly, our method outperforms all others on the three largest datasets (“HKU-IS”, “DUT-OMRON” and “DUTS-test”), which have more complicated salient object regions. This result demonstrates the strong capability of our method to deal with challenging inputs; see also the visual comparison results presented in Figures 5 and 6.

Visual comparison. Figures 5 and 6 present salient object detection results produced by various methods, including ours. From the figures, we can see that other methods (d)-(h) tend to include non-salient backgrounds or miss some salient details, while our SAC-Net is able to produce results (c) that are more consistent with the ground truths (b). Particularly, for challenging cases, such as (i) salient objects and non-salient background with similar appearance (see Figure 5), (ii) small salient objects (see 1st & 2nd rows in Figure 6), (iii) complex shapes (see 2nd row in Figure 5 and 1st & 4th rows in Figure 6), and (iv) multiple objects (see
2nd & 5th rows in Figure 5, and 1st, 2nd & 4th rows in Figure 6), our method can still predict more plausible saliency maps than the others, showing the robustness and quality of SAC-Net.

4.3. Evaluation on the Network Design

Component analysis. We performed an ablation study to evaluate the major components in SAC-Net. The first row of Table 2 shows the results from a basic model (FPN [31]) built with only the feature pyramid; see the green blocks in Figure 2. By having the SAC modules in the network to adaptively aggregate spatial context, we can see clear improvements on all the benchmark datasets as compared with the FPN results; see the first two rows in the table.

Compare with LSTM. The long short-term memory [17] (LSTM) is an efficient recurrent neural network to process sequence data by using a set of gates. The method has been extended to process 2D spatial information by some recent works on image classification [41] and saliency detection (s.t., PiCA [33]). We performed another experiment by adopting the LSTMs in four principal directions with two rounds of recurrent translations to replace our recurrently-attenuating translation model in the SAC module; in detail, we replaced the feature maps with colored arrows in Figure 4 by the LSTMs in corresponding directions.

The last row in Table 2 presents the LSTM results. Comparing with our results in the middle row, we can see that our method performs better for $F_{\beta}$, $S_m$ and MAE on all the benchmark data. We think the reason is that due to the limitation of the gate functions in LSTM [24], context features can only propagate over a short distance, thus limiting the dispersal of local context features in the spatial domain. On the other hand, the time complexity of computing LSTMs on 2D feature maps is very high; “with LSTM” took $\sim$213 hours to train the model, while our method took only $\sim$15 hours, which is more than 14 times faster.

Architecture analysis. To build our network, we empirically determine the value of $n$, which affects the number of attenuation factors and the number of feature channels in each aggregated feature map ($\frac{2^{n-1}}{n}$); see Figure 4. In general, a large $n$ allows the network to consider more variety of attenuation factors but each feature map would capture less information in return, since we keep the overall memory consumption to be manageable. Another parameter in our network is $\beta$, where we automatically learn its value for regulating the magnitude of the negative part in Eq. (1).

We evaluated our network on the three largest datasets (HKU-IS, DUT-OMRON, and DUTS-test) using different $n$ and learnable/fixed $\beta$. The results shown in Table 3 reveal that when we aggregate the image context using two different attenuation factors ($n=2$), we achieve better results than using only one single long-range aggregation ($n=1$).
The results further improve with larger \( n \) and roughly stabilizes when \( n \) reaches three, so we set \( n=3 \). On the other hand, comparing the results on the 3rd and last rows (both with \( n=3 \)) in Table 3, we can see that automatically learning and adjusting \( \beta \) gives better results than using a fixed \( \beta \).

### 4.4. Discussion

There has been a lot of works on exploiting spatial context in deep CNNs for image analysis. Dilated convolution [3, 53] takes context from larger regions by inserting holes into the convolution kernels, but the context information in use still has a fixed range in a local region. ASPP [4, 5] and PSPNet [59] adopt multiple convolution kernels with different dilated rates or multiple pooling operations with different scales to aggregate spatial context using different region sizes; however, their designed kernel or pooling sizes are fixed, less flexible, and not adaptable to different inputs. The non-local network [49] computes correlations between every pixel pair on the feature map to encode the global image semantics, but this method ignores the spatial relationship between pixels in the aggregation; for salient object detection, features of opposite semantics may, however, be important; see Figure 1. PSANet [60] adaptively learns attention weights for each pixel to aggregate the information from different positions; however, it is unable to capture the context on lower-level feature maps in high resolutions due to the huge time and memory overhead. Compared to these methods, our SAC-Net explores and adaptively aggregates context features implicitly with variable influence ranges; it is flexible, fast, and computationally friendly for efficient salient object detection.

Lastly, we also analyzed the failure cases, for which we found to be highly challenging, also for the other state-of-the-art methods. For instance, our method may fail for (i) multiple salient objects in very different scales (see Figure 7 (top)), where the network may regard the small objects as non-salient background; (ii) dark salient objects (see Figure 7 (middle)), where there are insufficient context to determine whether the regions are salient or not; and (iii) salient objects over a complex background (see Figure 7 (bottom)), where high-level scene knowledge is required to understand the image.

*We will release our code, the trained models, and the predicted saliency maps on all the benchmark datasets upon the publication of this work.*

### 5. Conclusion

This paper presents a novel saliency detection network based on the spatial attenuation context. Our key idea is to recurrently propagate and aggregate image context with different attenuation factors and to integrate the aggregated features using weights learnt from an attention mechanism. Using our model, local image context can adaptively propagate over different ranges, and we can leverage the complementary advantages of these context to improve the saliency detection quality. In the end, we evaluated our method on six common benchmark datasets and compared it extensively with 22 state-of-the-art methods. Experimental results clearly show that our method performs favorably over all the others, both visually and quantitatively. In the future, we plan to explore the potential of our SAC module design for instance-level salient object detection and enhance its capability for detecting salient objects in videos.

### References

[1] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, 2015.
[38] V. Movahedi and J. H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In CVPRW, pages 49–56, 2010. 5, 6
[39] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In CVPR, pages 733–740, 2012. 2
[40] Y. Qin, H. Lu, Y. Xu, and H. Wang. Saliency detection via cellular automata. In CVPR, pages 110–119, 2015. 5, 6
[41] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio. ReNet: A recurrent neural network based alternative to convolutional networks. arXiv preprint arXiv:1505.00393, 2015. 7
[42] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In CVPR, pages 3183–3192, 2015. 5
[43] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan. Learning to detect salient objects with image-level supervision. In CVPR, pages 136–145, 2017. 2, 5, 6
[44] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In ECCV, pages 825–841, 2016. 1, 2, 5
[45] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu. A stage-wise refinement model for detecting salient objects in images. In CVPR, pages 4019–4028, 2017. 5
[46] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji. Detect globally, refine locally: A novel approach to saliency detection. In CVPR, pages 3127–3135, 2018. 2, 5, 6, 7
[47] W. Wang, J. Shen, X. Dong, and A. Borji. Salient object detection driven by fixation prediction. pages 1711–1720, 2018. 5
[48] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In CVPR, pages 3395–3402, 2015. 1
[49] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In CVPR, pages 7794–7803, 2018. 1, 2, 8
[50] Y. Wu and K. He. Group normalization. In ECCV, pages 3–19, 2018. 4
[51] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In CVPR, pages 1155–1162, 2013. 2, 5
[52] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In CVPR, pages 3166–3173, 2013. 2, 5, 6
[53] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In ICLR, 2016. 8
[54] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang. A bi-directional message passing model for salient object detection. In CVPR, pages 1741–1750, 2018. 1, 2, 5, 6
[55] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In ICCV, pages 202–211, 2017. 1, 2, 5
[56] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin. Learning uncertain convolutional features for accurate saliency detection. In ICCV, pages 212–221, 2017. 1, 2, 5
[57] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang. Progressive attention guided recurrent network for salient object detection. In CVPR, pages 714–722, 2018. 1, 2, 5, 6, 7
[58] H. Zhao, X. Mao, X. Jin, J. Shen, F. Wei, and J. Feng. Real-time saliency-aware video abstraction. The Visual Computer, 25(11):973–984, 2009. 1
[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In CVPR, pages 2881–2890, 2017. 8
[60] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia. PSANet: Point-wise spatial attention network for scene parsing. In ECCV, pages 267–283, 2018. 8
[61] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In CVPR, pages 1265–1274, 2015. 1, 2
[62] L. Zhu, X. Hu, C.-W. Fu, J. Qin, and P.-A. Heng. Saliency-aware texture smoothing. IEEE Transactions on Visualization and Computer Graphics, 2018. 1