PiCANet: Learning Pixel-wise Contextual Attention for Saliency Detection

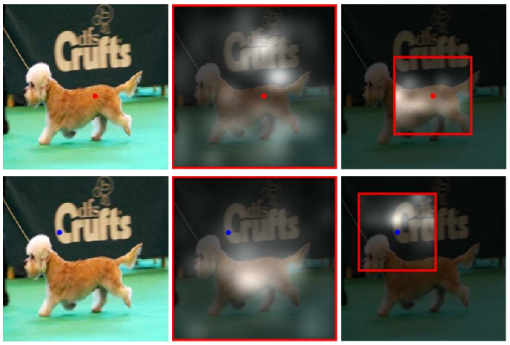
Nian Liu[1] Junwei Han[1*] Ming-Hsuan Yang[2,3]
[1]Northwestern Polytechincal University [2]University of California, Merced [3]Google Cloud
{liunian228, junweihan2010}@gmail.com mhyang@ucmerced.edu

Abstract

Contexts play an important role in the saliency detection task. However, given a context region, not all contextual information is helpful for the final task. In this paper, we propose a novel pixel-wise contextual attention network, i.e., the PiCANet, to learn to selectively attend to informative context locations for each pixel. Specifically, for each pixel, it can generate an attention map in which each attention weight corresponds to the contextual relevance at each context location. An attended contextual feature can then be constructed by selectively aggregating the contextual information. We formulate the proposed PiCANet in both global and local forms to attend to global and local contexts, respectively. Both models are fully differentiable and can be embedded into CNNs for joint training. We also incorporate the proposed models with the U-Net architecture to detect salient objects. Extensive experiments show that the proposed PiCANets can consistently improve saliency detection performance. The global and local PiCANets facilitate learning global contrast and homogeneousness, respectively. As a result, our saliency model can detect salient objects more accurately and uniformly, thus performing favorably against the state-of-the-art methods.

1. Introduction

Saliency detection aims at modeling human visual attention mechanism to detect distinct regions or objects, on which people likely focus their eyes in visual scenes. Contextual information plays an essential role in this visual task. As one of the earliest pioneering computational saliency models, Itti *et al*. [12] calculate the feature difference between each pixel and its surrounding regions as the contrast to infer saliency. Numerous methods have been subsequently developed [7, 4, 15] that utilize local or global contexts as the reference to evaluate the contrast of each image location (i.e., local or global contrast). These models aggregate visual information at all the locations of the referred context region into a contextual feature to infer contrast.



(a) (b) (c)

Figure 1. Example of learned global and local pixel-wise contextual attention maps. (a) shows the original image and two example pixels, i.e., the red dot on the foreground dog and the blue dot on the background. (b) and (c) show the learned global and local contextual attention maps for the two pixels, respectively. The brightness of each location indicates the magnitude of its attention weight. The red boxes indicate the referred context regions.

Recently, convolutional neural networks (CNNs) have been introduced into saliency detection to learn effective contextual representation. Specifically, several methods [18, 24, 47] first directly use CNNs to extract features from multiple image regions with varying contexts and subsequently combine these contextual features to infer saliency. Some other models [17, 19, 23, 22, 37, 9, 26, 45, 46, 38] adopt fully convolutional networks (FCNs) [25] for feature representation at each image location and generate saliency map in a convolutional way. In these models, the first school extracts contextual features from each input image region, while the second one extracts features at each image location from its corresponding receptive field.

However, all the existing models utilize context regions holistically to construct contextual features, in which the information at every contextual location is integrated. Intuitively, for a specific image pixel, not all of its contextual information contribute to its final decision. Some related regions are usually more useful, while other noisy responses should be discarded. For example, for the red dot pixel in the first row in Figure 1, we need to compare it with the background to infer its global contrast. If we want to

*Corresponding author

2575-7075/18 $31.00 © 2018 IEEE
DOI 10.1109/CVPR.2018.00326

check whether it belongs to the foreground dog for uniformly highlighting the whole dog, we need to refer to other parts of the dog. While for the blue dot pixel in the second row, we need to refer to the foreground dog and other parts of the background, respectively. Thus, if we can identify relevant context regions and construct informative contextual feature for each pixel, better decisions can be made. Nevertheless, this important issue has not been addressed by existing methods.

To address the problem discussed above, in this paper we propose a novel Pixel-wise Contextual Attention network, which is referred as PiCANet, to learn these informative contextual regions for each image pixel. It significantly improves the soft attention model [1] by generating contextual attention for every pixel , which is a genuine novel idea for the whole neural network community. Specifically, as shown in Figure 1, the proposed PiCANet learns to generate soft attention over the context regions for each pixel, where the attention weights indicate how relevant each context location is w.r.t. the referred pixel. The features from the context regions are then weighted and aggregated to obtain an attended contextual feature, which only considers informative context locations while ignores detrimental ones for each pixel. As a result, the proposed PiCANets can facilitate the saliency detection task significantly.

To incorporate contexts with different scopes, we formulate the PiCANet in two forms: global and local PiCANets, to selectively integrate global context and local context, respectively. Furthermore, our implementations of the PiCANets are fully differentiable. Thus they can be flexibly embedded into ConvNets and enable joint training.

We hierarchically embed global and local PiCANets into a U-Net architecture [30], which is an encoder-decoder convolutional network with skip connections, to detect salient objects. In the decoder, we progressively employ several global and local PiCANets on multiscale feature maps. Thus, we construct the attended contextual features from the global view to local contexts, from coarse scale to fine scales, and use them to enhance the convolutional features to facilitate saliency inference at each pixel. Figure 1 shows some examples of the learned attention maps. For each pixel (the red and the blue dots), the learned global attention shown in Figure 1(b) can attend to backgrounds for foreground objects and vice verse, which exactly matches the global contrast mechanism. While the learned local attention shown in Figure 1(c) can attend to regions that have the similar appearance with the referred pixel in its local context to make the saliency map more homogeneous.

Our contributions can be summarized as follows:

1. We propose the novel PiCANet to generate attention over the context regions for each pixel. Consequently, informative contextual features can be obtained to facilitate the final decision. Furthermore, we formulate PiCANet in both global and local forms to attend to global and local contexts, respectively, and with full differentiability to enable joint training with ConvNets.

2. We propose a novel saliency detection model by embedding PiCANets into a U-Net architecture. PiCANets are used to hierarchically incorporate the attended global context and multiscale local contexts, which can effectively improve saliency detection performance.

3. Extensive experimental results on six benchmark datasets demonstrate the effectiveness of the proposed PiCANets and the saliency model when compared with other state-of-the-art models. We also present in-depth analyses and explain why the proposed PiCANets perform well.

2. Related Work

Attention networks. Recently, attention models are introduced into neural networks to mimic the visual attention mechanism of focusing on informative regions in visual scenes. Mnih *et al*. [28] propose a recurrent attention model with hard alignment. However, it is difficult to train such hard attention models. Subsequently, Bahdanau *et al*. [1] develop an attention model with differentiable soft alignments for machine translation. In recent years, attention models have been applied to several vision tasks. Xu *et al*. [41] use an recurrent attention model for image caption to align words with image regions. In [32], Sermanet *et al*. adopt a recurrent attention model for fine-grained classification via attending to discriminative regions. In addition, attention models are introduced for visual question answering to attend to question-related image regions [40, 44]. Li *et al*. [20] utilize attention to attend to the global context to guide object detection. These works demonstrate that attention models can be significantly helpful for computer vision tasks via attending to informative contexts. However, existing approaches only consider generating one global contextual attention map at one time, which we refer as the *image-wise contextual attention*. These models limit the application of attention networks in convolutional nets, especially for pixel-wise tasks, since different pixels have different informative context regions. In [3], Chen *et al*. generate attention weights for each pixel for semantic segmentation. Nevertheless, this method uses attention to select adaptive scales on multiscale features for each pixel, which we refer as the *pixel-wise scale attention*. In contrast, our proposed PiCANet generates attention for context regions of each pixel.

Saliency detection. Traditional saliency models mainly rely on various saliency cues to detect salient objects, including local contrast [15], global contrast [4], and background prior [43]. Lately, with the utilization of CNNs, many work have achieved promising results on saliency detection. Next, we briefly review these models.

Liu *et al*. [24] and Li and Yu [18] adopt CNNs to extract multiscale contextual features on multiscale image regions to infer saliency for each pixel and each superpixel, respectively. Similarly, Zhao *et al*. [47] use CNNs on both global and local contexts. In [19], an FCN based saliency model and a multiscale image region based saliency model are combined. Wang *et al*. [37] recurrently adopt an FCN to refine saliency maps progressively. Liu and Han [23] use a U-Net based network to hierarchically predict and refine saliency maps from the global view to finer local views. Similarly, Luo *et al*. [26] and Zhang *et al*. [45] also utilize U-Net based models to incorporate multi-level contexts to detect salient objects. Wang *et al*. [38] also use several stages to progressively refine saliency maps by combining local and global context information. In [9], short connections are introduced into the multi-scale side outputs within the HED network [39] to improve saliency detection performance. Hu *et al*. [10] propose to adopt a level sets based loss to train their saliency detection network and use guided super-pixel filtering to refine saliency maps.

Although existing DNN based models incorporate various contexts for saliency detection, these methods all use context regions holistically. Typically, the work in [23, 26, 45], which have similar U-Net architectures with the one we use in this paper, incorporate multiscale contexts via diverse network architectures which indiscriminately integrate the information from their receptive field. In contrast, we use the proposed PiCANets to only selectively attend to informative context locations. In [17], authors use a recurrent attention model to select local regions to refine their saliency maps. However, they adopt the spatial transformer attention network [13] to select one refining region at each time step, where their model still falls into the *image-wise attention* category. In contrast, our PiCANets can generate soft contextual attention for each pixel.

3. Pixel-wise Contextual Attention Network

The proposed PiCANet aims at generating an attention map at each pixel over its context region and constructing an attended contextual feature to enhance the feature representability of Convnets. Given a convolutional (Conv) feature map $\boldsymbol{F} \in \mathbb{R}^{W \times H \times C}$, where W, H, C denote its width, height and number of channels, respectively, we propose two pixel-wise attention modes: global attention and local attention. For each location (w, h) in $\boldsymbol{F}$, the former generates attention over the whole feature map $\boldsymbol{F}$, while the latter works on a local region centered at (w, h).

3.1. Global PiCANet

For the global attention, we show the network architecture in Figure 2(a). Since we tend to generate attention over the global context for each pixel, we need to make each pixel be able to "see" the overall feature map $\boldsymbol{F}$ first. To this end, one can use various network architectures whose receptive field is the whole image, *e.g*., a fully connected layer. Here we employ a more effective and efficient ReNet model [35], which uses four recurrent neural networks to sweep an image both horizontally and vertically along both directions, to incorporate the global context. Specifically, as shown in the orange dashed box in Figure 2(a), a bidirectional LSTM (biLSTM) [6] is first deployed along each row of $\boldsymbol{F}$, then the two hidden states of each pixel are concatenated, making each pixel memorize both its left and right contexts. Next, another biLSTM is deployed along each column of the obtained feature map, so that each pixel can memorize both its top and bottom contexts. By alternately scanning horizontally and vertically, the contexts from four directions can be blended, which propagate the information of each pixel to all other pixels. Thus, global context is efficiently incorporated at each pixel.

Next, we use a vanilla Conv layer to transform the ReNet feature map to D channels, where $D = W \times H$. Then, at each pixel (w, h), the obtained feature vector, which is denoted as $\boldsymbol{x}^{w,h}$, is normalized via a softmax function to generate the global attention weights $\boldsymbol{\alpha}^{w,h}$:

$$\alpha_i^{w,h} = \frac{\exp\left(x_i^{w,h}\right)}{\sum_{j=1}^{D} \exp\left(x_j^{w,h}\right)}, \tag{1}$$

where $i \in \{1, \cdots, D\}$, $\boldsymbol{x}^{w,h}, \boldsymbol{\alpha}^{w,h} \in \mathbb{R}^D$, and $\alpha_i^{w,h}$ corresponds to the contextual relevance at the i^{th} context location (W_i, H_i) w.r.t. the referred pixel (w, h).

Finally, as shown in Figure 2(b), for the pixel (w, h), the features at all locations in $\boldsymbol{F}$ are weighted summed by $\boldsymbol{\alpha}^{w,h}$ to construct the attended contextual feature $\boldsymbol{F}_{att}$:

$$\boldsymbol{F}_{att}^{w,h} = \sum_{i=1}^{D} \alpha_i^{w,h} \boldsymbol{f}_i, \tag{2}$$

where $\boldsymbol{f}_i \in \mathbb{R}^C$ is the Conv feature at (W_i, H_i) in $\boldsymbol{F}$ and $\boldsymbol{F}_{att}$ has the same size with $\boldsymbol{F}$.

3.2. Local PiCANet

As for the local attention, at each pixel (w, h), we only perform the attending operation on a local neighboring context region centered at (w, h), which forms a local feature cube $\bar{\boldsymbol{F}}^{w,h} \in \mathbb{R}^{\bar{W} \times \bar{H} \times C}$, with the width $\bar{W}$ and the height $\bar{H}$. The network architecture is shown in Figure 2(c). Again, we first need each pixel to "see" the $\bar{W} \times \bar{H}$ context region. We simply use Conv layers to achieve this purpose. Specifically, we deploy several Conv layers on $\boldsymbol{F}$ to make their receptive field achieve the size of $\bar{W} \times \bar{H}$. Then, as the same as global PiCANet, a Conv layer is used to transform the resultant feature map to $\bar{D} = \bar{W} \times \bar{H}$ channels. Next, the local attention weights $\bar{\boldsymbol{\alpha}}^{w,h}$ are also generated by the softmax normalization (similar to (1)). Finally, as

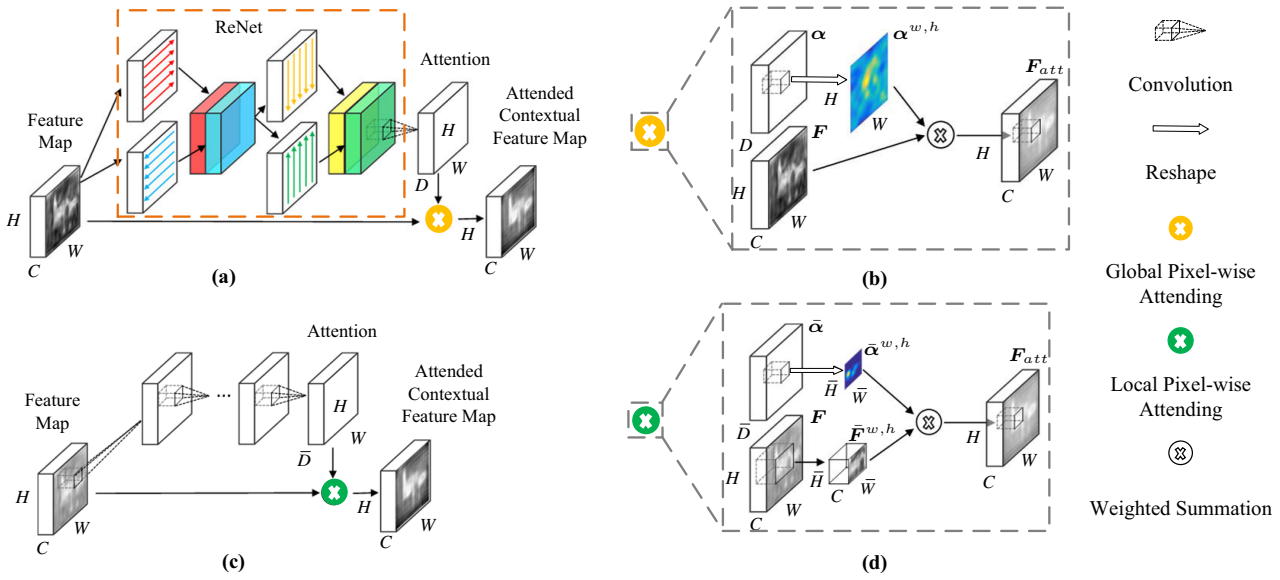


Figure 2. (a) Architecture of the proposed global PiCANet. (b) Illustration of the detailed global attending operation. (c) Architecture of the proposed local PiCANet. (d) Illustration of the detailed local attending operation.

shown in Figure 2(d), for pixel (w, h), the features in $\bar{\boldsymbol{F}}^{w,h}$ are weighted summed by $\bar{\boldsymbol{\alpha}}^{w,h}$ to obtain $\boldsymbol{F}_{att}$:

$$\boldsymbol{F}_{att}^{w,h} = \sum_{i=1}^{\bar{D}} \bar{\alpha}_i^{w,h} \bar{\boldsymbol{f}}_i^{w,h}. \tag{3}$$

3.3. Effective and Efficient Implementation

For computational efficiency, the attending operation for all pixels can be implemented simultaneously by a convolution-like way. We can also adopt the hole algorithm [2] in the attending operation, which supports sparsely sampling feature maps by using dilated convolution. Thus, we can use a small D or $\bar{D}$ with dilation to attend to large context regions to make PiCANets more efficient. The gradients of the PiCANets can be easily calculated, making end-to-end training feasible via the back-propagation algorithm [31]. We can also use a batch normalization (BN) [11] layer before softmax normalization to make the network training more effective.

4. Salient Object Detection using PiCANets

In this section, we elaborate our network architecture which adopts PiCANets hierarchically for salient object detection. The whole network is based on a U-Net [30] architecture as shown in Figure 3(a). However, different from [30], the encoder of our U-Net is an FCN with the hole algorithm [2] to keep the resolutions of feature maps. The decoder follows the idea of U-Net to use skip connections and with our proposed global and local PiCANets embedded.

Considering the global PiCANet requires the input feature map to have a fixed size, we set input images to have a fixed size of 224×224. The encoder part is an FCN with a pretrained backbone network, *e.g.*, the VGG [33] network or a ResNet [8]. We take the VGG 16-layer network as an example, which contains 13 Conv layers, 5 max-pooling layers, and 2 fully connected layers. As shown in Figure 3(a), in order to preserve relative large spatial sizes in higher layers for accurate saliency detection, we modify the pooling strides of the pool4 and pool5 layers to be 1 and adopt the hole algorithm [2] to introduce dilation of 2 for the conv5 layers. We also follow [2] to transform the last 2 fully connected layers to Conv layers. Specifically, we use 1024 3×3 kernels with dilation of 12 for the fc6 layer and 1024 1×1 kernels for the fc7 layer. Thus, the stride of the whole encoder network is reduced to 8, and the spatial size of the final feature map is 28×28.

Next, we elaborate our decoder part. As shown in Figure 3(a), the decoder network has 6 decoding modules, named $\mathcal{D}^7, \mathcal{D}^5, \mathcal{D}^4, \cdots, \mathcal{D}^1$. As shown in Figure 3(b), in $\mathcal{D}^i$, where $i \in \{7, 5, 4, \cdots, 1\}$, we usually generate a decoding feature map $\boldsymbol{Dec}^i$ by fusing an intermediate encoder feature map $\boldsymbol{En}^i$ with the size of $W \times H \times C^i$ and the preceding decoding feature map $\boldsymbol{Dec}^{i-1}$ with the size of $W/2 \times H/2 \times C^i$. $\boldsymbol{En}^i$ is the Conv feature map before the ReLU activation of the i^{th} Conv block in the VGG encoder part, and they are marked in Figure 3(a). We first use a BN layer and the ReLU activation on $\boldsymbol{En}^i$. At the same time, we upsample $\boldsymbol{Dec}^{i-1}$ to have the spatial size of $W \times H$ by using a deconvolutional layer with bilinear interpolation. Next, we concatenate these two feature maps and fuse them into a feature map $\boldsymbol{F}^i$ with C^i channels by using a Conv and a ReLU layer. Then we utilize a global or a local PiCANet on $\boldsymbol{F}^i$ to obtain its attended contextual

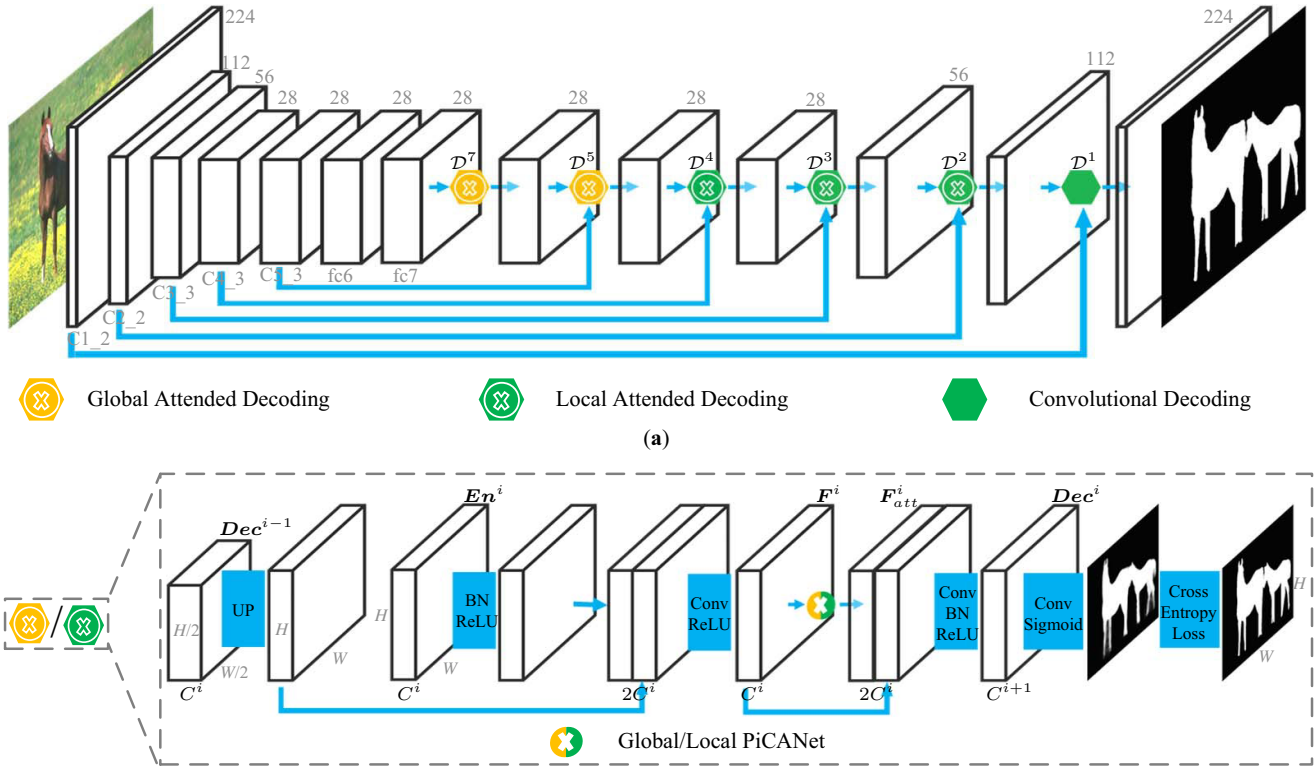


Figure 3. (a) Architecture of our saliency network with the VGG 16-layer backbone. We only show the skip-connected encoder layers of the VGG network. "C" means "convolution" while $\mathcal{D}^*$ indicates a decoding module. The spatial sizes are marked over the cuboids which represent the feature maps. (b) Illustration of an attended decoding module. $\boldsymbol{En}^i$ denotes a convolutional feature map from the encoder network. $\boldsymbol{Dec}^*$ denotes a decoding feature map. $\boldsymbol{F}^i$ denotes a fusion feature map and $\boldsymbol{F}^i_{att}$ denotes its attended contextual feature map. "UP" denotes upsampling. Some important spatial sizes and channel numbers are also marked.

feature map $\boldsymbol{F}^i_{att}$. Finally we fuse $\boldsymbol{F}^i$ and $\boldsymbol{F}^i_{att}$ into $\boldsymbol{Dec}^i$ with the size $W \times H \times C^{i+1}$, via a Conv layer, a BN layer, and a ReLU layer. We also adopt deep supervision to facilitate the network training. Specifically, in each $\mathcal{D}^i$, we use a Conv layer with sigmoid activation on $\boldsymbol{Dec}^i$ to generate a saliency map with size $W \times H$, then the resized ground truth saliency map is used to supervise the network training based on the average cross-entropy loss.

In each $\mathcal{D}^i$, we set C^i to be the same as the channel number of the i^{th} Conv block in the encoder network. We adopt global PiCANets in $\mathcal{D}^7$ and $\mathcal{D}^5$ and local PiCANets in the next three decoding modules. For $\mathcal{D}^1$, we simply fuse $\boldsymbol{En}^1$ and $\boldsymbol{Dec}^2$ into $\boldsymbol{Dec}^1$ with simple Conv layers for computational efficiency. The influence of different embedding choices of global and local PiCANets is shown in Section 5.4.

5. Experiments

5.1. Datasets

We use six widely used saliency benchmark datasets to evaluate our method. **SOD** [29] contains 300 images with complex backgrounds and multiple foreground objects. **ECSSD** [42] has 1,000 semantically meaningful and complex images. The **PASCAL-S** [21] dataset consists of 850 images selected from the PASCAL VOC 2010 segmentation dataset. **DUT-O** [43] includes 5,168 challenging images, each of which usually has complicated background and one or two foreground objects. **HKU-IS** [18] contains 4,447 images with low color contrast and multiple foreground objects in each image. The last one is the **DUTS** [36] dataset, which is currently the largest salient object detection benchmark dataset. It contains 10,553 images in the training set, i.e., DUTS-TR, and 5,019 images in the test set, i.e., DUTS-TE. Most of the images have challenging scenarios for saliency detection.

5.2. Evaluation Metrics

We adopt four evaluation metrics to evaluate our model. The first one is the precision-recall (PR) curve. Specifically, saliency maps are first binarized and then compared with the ground truth under varying thresholds, thus obtaining a series of precision-recall value pairs to draw the PR curve.

The second metric is the F-measure score which com-

prehensively considers both precision and recall:

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

where we set β^2 to 0.3 as suggested in previous work.

However, as demonstrated in [27], traditional evaluation metrics easily suffer from the interpolation flaw, dependency flaw, and equal-importance flaw, leading to unfair comparison. Thus the authors propose the *weighted* F-measure score F_β^ω to address these drawbacks. We also follow [5, 34, 10] to adopt it as one of our metrics with the default settings in [27].

The fourth metric we use is the Mean Absolute Error (MAE). It computes the average absolute per-pixel difference between predicted saliency maps and corresponding ground truth saliency maps.

5.3. Implementation Details

Network structure. In the decoding modules, all of the convolutional kernels in Figure 3(b) are set to 1×1. In each global PiCANet, we use 256 hidden neurons for the ReNet, then we use a 1×1 Conv layer to generate $D = 100$ dimensional attention weights, which can be reshaped to 10×10 attention maps. In its attending operation, we use $dilation = 3$ to attend to the 28×28 global context. In each local PiCANet, we first use a 7×7 Conv layer with $dilation = 2$, zero padding, and ReLU activation to generate an intermediate feature map with 128 channels. Then we adopt a 1×1 Conv layer to generate $\bar{D} = 49$ dimensional attention weights, from which 7×7 attention maps can be obtained. Then we utilize these local attention maps to attend to 13×13 local context regions with $dilation = 2$ and zero padding.

Training and testing. We follow [38] and the suggestion in [36] to use the DUTS-TR set as our training set. For data augmentation, we simply resize each image to 256×256 with random mirror-flipping and randomly crop 224×224 image regions for training. The whole network is trained end-to-end using stochastic gradient descent (SGD) with momentum. Since deep supervision is adopted in each decoding module, we empirically weight the losses in $\mathcal{D}^7, \mathcal{D}^5, \mathcal{D}^4, \cdots, \mathcal{D}^1$ by 0.5, 0.5, 0.5, 0.8, 0.8, and 1, respectively, without further tuning. We train the decoder part from scratch with a learning rate of 0.01 and finetune the encoder with a 0.1 times smaller learning rate. We set the batchsize to 10, the maximum iteration step to 20,000, and decay the learning rates by a factor of 0.1 every 7,000 steps. The momentum and the weight decay are set to 0.9 and 0.0005, respectively.

We implement our model based on the Caffe [14] library. A GTX Titan X GPU is used for acceleration. When testing, each image is simply resized to 224×224 and then fed into the network to obtain its saliency map. The testing process only costs 0.178s for each image when using the VGG 16-layer backbone. Our code will be released.

Table 1. Quantitative results of different settings of our model and baseline models. "MP" and "AP" mean max-pooling and average pooling, respectively. "+75G432LP" means using **G**lobal PiCANets in $\mathcal{D}^7$ and $\mathcal{D}^5$, and **L**ocal **P**iCANets in $\mathcal{D}^4$, $\mathcal{D}^3$, $\mathcal{D}^2$. Other settings can be inferred similarly. **Blue** indicates the best performance.

Settings	DUT-O [43]			DUTS-TE [36]		
	F_β	F_β^ω	MAE	F_β	F_β^ω	MAE
U-Net [30]	0.761	0.651	0.073	0.819	0.715	0.060
+75GP	0.778	0.662	0.071	0.834	0.724	0.057
+75G432LP	**0.794**	**0.691**	0.068	**0.851**	**0.748**	0.054
+MP	0.780	0.671	0.070	0.833	0.727	0.057
+AP	0.778	0.670	0.069	0.831	0.724	0.056
+75432LP	0.787	0.680	0.069	0.842	0.738	0.055
+7G5432LP	0.792	0.690	0.069	0.849	0.744	0.054
+754G32LP	0.794	0.688	**0.065**	0.850	0.747	**0.053**

5.4. Ablation Study

Effectiveness of the proposed PiCANets. To demonstrate the effectiveness of the proposed PiCANets, we show quantitative comparison results of our model against baseline models on two challenging datasets in Table 1. "U-Net" is the baseline network without PiCANets. "+75GP" means we only embed two global PiCANets into $\mathcal{D}^7$ and $\mathcal{D}^5$, while "+75G432LP" means we embed global PiCANets into $\mathcal{D}^7$ and $\mathcal{D}^5$, and local PiCANets into $\mathcal{D}^4$, $\mathcal{D}^3$, $\mathcal{D}^2$. The comparison results show that when we gradually use PiCANets to incorporate global and multiscale local contexts selectively, the model performance can be progressively boosted. A more detailed ablation study of progressively embedding PiCANets in each decoding module is given in the supplementary material.

For a fair comparison, we also adopt max-pooling (MP) and average-pooling (AP) to incorporate these contexts. Table 1 shows that although using these non-parametric pooling schemes to incorporate global and local contexts can bring performance gains, using our proposed PiCANets to select informative contexts is a much better way.

We also show visual comparison results to demonstrate the effectiveness of the proposed PiCANets. In Figure 5(a) we show an image and its ground truth saliency map while (b) shows the predicted saliency maps of the baseline U-Net (top) and our model (bottom). We can see that our saliency model can obtain more uniformly highlighted saliency map with the help of PiCANets. In Figure 5(c), we show a comparison of the Conv feature map $\boldsymbol{F}^5$ (top) against the attended contextual feature map $\boldsymbol{F}^5_{att}$ (bottom) with the global PiCANet. While (d) shows $\boldsymbol{F}^2$ (top) and $\boldsymbol{F}^2_{att}$ (bottom) with the local PiCANet. We can see that the global PiCANet in

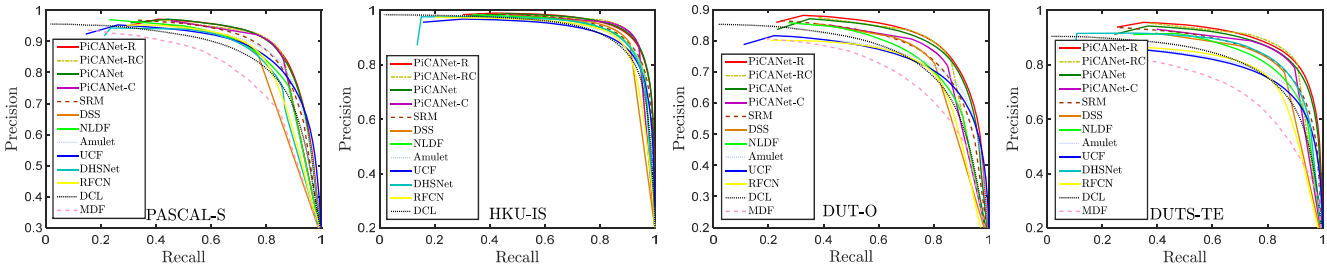
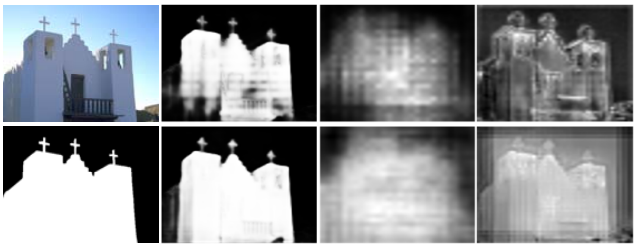


Figure 4. Comparison on four large datasets in terms of the PR curve.



(a) (b) (c) (d)

Figure 5. Visual comparison of our model against the baseline U-Net. (a) An image and its ground truth. (b) Saliency maps of the baseline U-Net (top) and our model (bottom). (c) $\boldsymbol{F}^5$ (top) and $\boldsymbol{F}^5_{att}$ (bottom). (d) $\boldsymbol{F}^2$ (top) and $\boldsymbol{F}^2_{att}$ (bottom).

$\mathcal{D}^5$ helps to better discriminate the foreground object from backgrounds, while the local PiCANet in $\mathcal{D}^2$ enhances the feature map to be more homogenous, which makes the whole foreground object highlighted more uniformly.

To further understand why PiCANets can achieve such improvements, we visualize the learned attention maps of two pixels in one image in Figure 6. In column (b), the top image shows that the global attention of the background pixel mainly attends to the foreground object while the bottom image shows that for the foreground pixel, it mainly attends to the background regions. This observation greatly matches the global contrast mechanism. Thus our global PiCANet can help the network to effectively tell the salient objects from the backgrounds. As for the local attention, since we used fixed attention size (13×13) for different decoding modules, we can incorporate multiscale attention from coarse to fine, with large contexts to small ones, as shown by red rectangles in Figure 6. The (c) and (d) columns show that local attention mainly attends to homogeneous regions with the referred pixel, thus enhancing the saliency map to be uniform, just as shown in the bottom image in column (a). More visualization can be found in the supplementary material.

Influence of the embedding choice. We also show comparison results of different embedding choices of our global and local PiCANets in Table 1. It shows that only embedding local PiCANets ("+75432LP") is inferior. While

(a) (b) (c) (d)

Figure 6. Illustration of the learned attention maps of the proposed PiCANet. (a) shows an image and its predicted saliency map of our model. We show the attention maps of two pixels (denoted as red dots. The top row shows a background pixel and the bottom row shows a foreground pixel.) in $\mathcal{D}^5$ (b), $\mathcal{D}^4$ (c), and $\mathcal{D}^3$ (d), respectively. The attended context regions are marked by red rectangles.

the results of "+7G5432LP" and "+754G32LP" are slightly worse than our final choice, i.e., "+75G432LP". We do not consider to use global PiCANets in other decoding modules since the ReNet is time-consuming for large feature maps.

5.5. Comparison with State-of-the-arts

We compare our saliency model against other 9 state-of-the-art models, namely, SRM [38], DSS [9], NLDF [26], Amulet [45], UCF [46], DHS [23], RFCN [37], DCL [19], and MDF [18].

In Table 2, we show the quantitative comparison results. Since [19] and [9] adopt the fully connected conditional random field (CRF) [16] as a post-processing technique, while [38] use the ResNet50 [8] network as their backbone, for a fair comparison we also adopt them in our model and compare it with other models under different settings. The PR curves on four large datasets are also given in Figure 4. We observe that our model consistently performs better than all other models under all settings, especially in terms of the *weighted* F-measure. It is also worth noting that even only use the VGG 16-layer backbone and without any post-processing method, our vanilla PiCANet still performs favorably against all other models. When using both

Table 2. Comparison of different methods on 6 datasets under different settings. **Blue** indicates the best performance under each setting while **red** indicates the best performance under all settings. "-C", "-R", and "-RC" means using the CRF postprocessing, the ResNet50 backbone, and both of them, respectively.

Dataset	SOD [43]			ECSSD [43]			PASCAL-S [21]			HKU-IS [18]			DUT-O [43]			DUTS-TE [36]		
Metric	F_β	F_β^ω	MAE	F_β	F_β^ω	MAE	F_β	F_β^ω	MAE	F_β	F_β^ω	MAE	F_β	F_β^ω	MAE	F_β	F_β^ω	MAE
VGG-16 [33] backbone																		
MDF [18]	0.760	0.501	0.192	0.832	0.705	0.105	0.782	0.579	0.165	-	-	-	0.694	0.565	0.092	0.711	0.509	0.114
RFCN [37]	0.807	0.592	0.166	0.898	0.727	0.095	0.850	0.671	0.132	0.898	0.718	0.080	0.738	0.562	0.095	0.783	0.587	0.090
DHS [23]	0.827	0.686	0.133	0.907	0.841	0.060	0.841	0.732	0.111	0.902	0.806	0.054	-	-	-	0.829	0.698	0.065
UCF [46]	0.803	0.644	0.169	0.911	0.789	0.078	0.846	0.709	0.128	0.886	0.751	0.074	0.735	0.565	0.132	0.771	0.588	0.117
Amulet [45]	0.808	0.686	0.145	0.915	0.841	0.059	0.858	0.762	0.103	0.896	0.813	0.052	0.743	0.626	0.098	0.778	0.657	0.085
NLDF [26]	0.842	0.708	0.130	0.905	0.839	0.063	0.845	0.743	0.112	0.902	0.838	0.048	0.753	0.634	0.080	0.812	0.710	0.066
PiCANet	**0.855**	**0.721**	**0.108**	**0.931**	**0.865**	**0.047**	**0.880**	**0.781**	**0.088**	**0.921**	**0.847**	**0.042**	**0.794**	**0.691**	**0.068**	**0.851**	**0.748**	**0.054**
VGG-16 [33] backbone + CRF [16]																		
DCL [19]	0.825	0.641	0.198	0.901	0.820	0.075	0.823	0.678	0.189	0.885	0.736	0.137	0.739	0.575	0.157	0.782	0.606	0.150
DSS [9]	**0.846**	0.718	0.126	0.916	0.871	0.053	0.846	0.751	0.112	0.911	0.866	0.040	0.771	0.691	0.066	0.825	0.754	0.057
PiCANet-C	0.836	**0.727**	**0.102**	**0.933**	**0.898**	**0.036**	**0.881**	**0.809**	**0.079**	**0.925**	**0.889**	**0.031**	**0.784**	**0.722**	**0.059**	**0.850**	**0.791**	**0.046**
ResNet50 [8] backbone																		
SRM [38]	0.845	0.671	0.132	0.917	0.853	0.054	0.862	0.760	0.098	0.906	0.836	0.046	0.769	0.658	0.069	0.827	0.722	0.059
PiCANet-R	**0.858**	**0.723**	**0.109**	**0.935**	**0.867**	**0.047**	**0.881**	**0.780**	**0.087**	**0.919**	**0.840**	**0.043**	**0.803**	**0.695**	**0.065**	**0.860**	**0.756**	**0.051**
ResNet50 [8] backbone + CRF [16]																		
PiCANet-RC	0.856	**0.742**	**0.100**	**0.940**	**0.908**	**0.035**	**0.883**	**0.812**	**0.077**	**0.927**	**0.890**	**0.031**	**0.804**	**0.743**	**0.054**	**0.866**	**0.811**	**0.041**

Figure 7. Qualitative comparison. (GT: ground truth)

of the CRF post-processing and the ResNet50 backbone, our PiCANet-RC model achieves the best performance and shows significant performance gains over existing methods.

In Figure 7, we show qualitative comparison. We observe that our model can handle various challenging scenarios, including images with complex backgrounds and foregrounds (rows 1, 2, and 3), varying object scales, object touching image boundaries (row 5), object having the similar appearance with the background (row 4). Most importantly, even for the vanilla PiCANet and PiCANet-R which do not use any post-processing methods, they can highlight salient objects more uniformly than other models with the help of PiCANets. More visual comparison results can be found in the supplementary material.

6. Conclusion

In this paper, we propose novel PiCANets to selectively attend to global or local contexts and construct informative contextual features for each pixel. We apply PiCANets to detect salient objects in a hierarchical fashion. With the help of attended contexts, our model achieves the best performance on six benchmark datasets. We also provide in-depth analyses of the effectiveness of the PiCANets.

Acknowledgments

This work is supported in part by the National Science Foundation of China (No. 61473231 and 61522207) and NSF CAREER (No. 1149783).

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 2

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 4

[3] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 2

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

[5] C. Gong, D. Tao, W. Liu, S. J. Maybank, M. Fang, K. Fu, and J. Yang. Saliency propagation from simple to difficult. In *CVPR*, 2015. 6

[6] A. Graves, N. Jaitly, and A.-r. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013. 3

[7] B. Han, H. Zhu, and Y. Ding. Bottom-up saliency based on weighted sparse coding residual. In *ACM Multimedia*, 2011. 1

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 7, 8

[9] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. In *CVPR*, 2017. 1, 3, 7, 8

[10] P. Hu, B. Shuai, J. Liu, and G. Wang. Deep level sets for salient object detection. In *CVPR*, 2017. 3, 6

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[12] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *TPAMI*, 20(11):1254–1259, 1998. 1

[13] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015. 3

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014. 6

[15] D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *ICCV*, 2011. 1, 2

[16] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 7, 8

[17] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *CVPR*, 2016. 1, 3

[18] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, 2015. 1, 3, 5, 7, 8

[19] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016. 1, 3, 7, 8

[20] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan. Attentive contexts for object detection. *IEEE Transactions on Multimedia*, 19(5):944–954, 2017. 2

[21] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. 5, 8

[22] N. Liu and J. Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *arXiv preprint arXiv:1610.01708*, 2016. 1

[23] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, 2016. 1, 3, 7, 8

[24] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu. Predicting eye fixations using convolutional neural networks. In *CVPR*, 2015. 1, 3

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[26] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin. Non-local deep features for salient object detection. In *CVPR*, 2017. 1, 3, 7, 8

[27] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps? In *CVPR*, 2014. 6

[28] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *NIPS*, 2014. 2

[29] V. Movahedi and J. H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In *CVPR Workshops*, 2010. 5

[30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2, 4, 6

[31] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988. 4

[32] P. Sermanet, A. Frome, and E. Real. Attention for fine-grained categorization. *arXiv preprint arXiv:1412.7054*, 2014. 2

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4, 8

[34] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien. Real-time salient object detection with a minimum spanning tree. In *CVPR*, 2016. 6

[35] 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. 3

[36] 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*, 2017. 5, 6, 8

[37] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016. 1, 3, 7, 8

[38] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, 2017. 1, 3, 6, 7, 8

[39] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 3

[40] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 2

[41] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[42] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, 2013. 5

[43] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 2, 5, 6, 8

[44] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 2

[45] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, 2017. 1, 3, 7, 8

[46] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin. Learning uncertain convolutional features for accurate saliency detection. In *ICCV*, 2017. 1, 7, 8

[47] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, 2015. 1, 3