International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

SuperCNN: A Superpixelwise Convolutional Neural Network for Salient Object Detection

Shengfeng He · Rynson W.H. Lau · Wenxi Liu · Zhe Huang · Qingxiong Yang

Received: date / Accepted: date

Abstract Existing computational models for salient object detection primarily rely on hand-crafted features, which are only able to capture low-level contrast information. In this paper, we learn the hierarchical contrast features by formulating salient object detection as a binary labeling problem using deep learning techniques. A novel superpixelwise convolutional neural network approach, called SuperCNN, is proposed to learn the internal representations of saliency in an efficient manner. In contrast to the classical convolutional networks, SuperCNN has four main properties. First, the proposed method is able to learn the hierarchical contrast features, as it is fed by two meaningful superpixel sequences, which is much more effective for detecting salient regions than feeding raw image pixels. Second, as SuperCNN recovers the contextual information among superpixels, it enables large context to be involved in the analysis efficiently. Third, benefiting from the superpixelwise mechanism, the required number of predictions for a densely labeled map is hugely reduced. Fourth, saliency can be detected independent of region size by utilizing a multiscale network structure. Experiments show that SuperCNN can robustly detect salient objects and outperforms the state-of-the-art methods on three benchmark datasets.

Keywords Convolutional Neural Networks · Deep Learning · Feature Learning · Saliency Detection

Shengfeng He · Rynson W.H. Lau · Wenxi Liu · Zhe Huang · Qingxiong Yang
City University of Hong Kong
E-mail: shengfeng_he@yahoo.com, rynson.lau@cityu.edu.hk, magicpiratex@gmail.com, igamenovoer@gmail.com, qiyang@cityu.edu.hk

1 Introduction

The human brain and visual system are able to quickly localize the regions in a scene that stand out from their neighbors. Saliency detection aims at simulating the human visual system for detecting pixels or regions that most attract human's visual attention. Although earlier saliency detection work focused on predicting eye fixations on images [22,17], recent research has shown that extracting salient objects or regions [9,31,41] is more useful and beneficial to a wide range of computer vision, graphics and multimedia applications. For example, predicting eye fixations may not be the best way to determine region of interest for image cropping [35] and content-aware image/video resizing [4], as eye fixation prediction only determines parts of the object, leading to object distortion.

Perceptual research [21,40] has shown that *contrast* is a major factor to visual attention in the human visual system. Various saliency detection algorithms based on different contrast cues [9,18] have been designed with success. However, as they typically combine individual hand-crafted image features (e.g., color, histogram and orientation) with different fusion schemes [31,36] to form the final saliency map in a local or global manner, they are not suitable for all cases. For example, local methods cannot detect homogenous regions, while global methods suffer from background distractions. Although learning techniques are adapted to detect salient objects [31,24], they focus on learning the fusion scheme, i.e., saliency integration by combining saliency maps obtained from different types of features.

To alleviate the need for hand-crafted features, feature learning using convolutional neural networks (CNNs) [28] has been successfully applied to different vision tasks, such as image classification [27] and

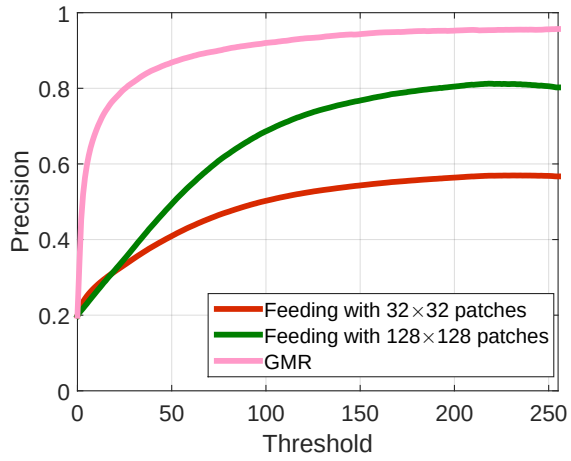


Fig. 1: Detection precision w.r.t using different saliency thresholds, when feeding the CNN (using a single scale structure of [14]) with raw pixels and tested it on the MSRA-1000 dataset. Feeding the network with small patches (red curve) is not suitable for detecting salient objects, as saliency should be determined from a large context. Although enlarging the patch size improves the performance (green curve), it leads to a large network and takes around 2 mins to obtain a dense saliency map. In addition, as saliency is independent of object appearance, feeding with raw pixels cannot successfully detect objects under different contrasts, as compared with the state-of-the-art, GMR [50].

scene parsing [14,42]. Nonetheless, there are two problems when applying CNNs to saliency detection. First, contrast-based saliency should be determined from a large context. However, for applications requiring a densely labeled map, classical CNNs are not suitable as they are typically fed with small image patterns in order to maintain efficiency. Second, whether an object is salient or not is independent of its appearance but dependent on its contrast to its surrounding. For example, a red object that is salient in a grass field may not necessarily be salient in others. As a result, training a CNN in a way like image classification or image parsing is not appropriated in our case, as demonstrated in Figure 1.

In this paper, we propose a novel superpixelwise convolutional neural network (SuperCNN) approach to address the above problems. Instead of a 2D image pattern, the input to SuperCNN is a sequence of superpixels, which can easily take into account long range context while limiting the capacity of the network. To recover the contextual information among the superpixels, a spatial kernel and a range kernel (inspired by bilateral filtering) are derived to produce two meaningful sequences as inputs to describe color uniqueness and color distribution, respectively. Ideally, a salient object exhibits distinct colors from its surrounding, and these colors are compactly distributed in the image plane. Once the networks are properly trained, SuperCNN is able to produce the two internal representations of salient objects, which capture different saliency properties. The learned hierarchical features encode information analogous to what the human visual system does, which is to perform selection in a hierarchical manner [34,29]. The softmax function is used after the feature extractor to convert the two feature vectors into saliency scores. To robustly detect salient regions regardless of their sizes, we use a multiscale network structure with shared weights. The architecture of the proposed salient object detection framework is illustrated in Figure 2.

Although the computational time of individual predictions of SuperCNN is similar to the classic CNNs fed with a 2D image pattern, SuperCNN requires two orders of magnitude fewer predictions than the classical CNNs for computing features of an image. It takes only $0.45s$ to produce a saliency map for a 400×300 image. In addition, it encodes the information from a large context and is generalized for different types of input data once it is properly trained. We have extensively evaluated it on three different benchmark datasets. The hierarchical features that we have learned generalize well to all three datasets. While SuperCNN produces comparable results to the state-of-the-art methods on the simple MSRA-1000 dataset, it achieves much better performances than all other methods on the other two datasets that contain complex scenarios.

2 Related Work

Visual attention can be driven by either low-level stimuli (bottom-up fashion) or high-level objectives (top-down fashion). Most saliency detection methods are based on bottom-up computational models using low-level features such as color, motion, or orientation of edges. Depending on the extent of the context where saliency is computed, these methods can be roughly categorized into local methods and global methods. Comprehensive literature review on these saliency detection methods can be found in [7,47].

Local saliency methods compute saliency of an image region with respect to a small neighborhood. An earlier local saliency detection method [22] is based on a biologically-plausible architecture [26]. It uses an image pyramid to compute color and orientation contrasts. Ma and Zhang [33] combine local contrast analysis with a fuzzy growth model. Harel *et al.* [17] propose a graph-based random walk method using multiple features. As

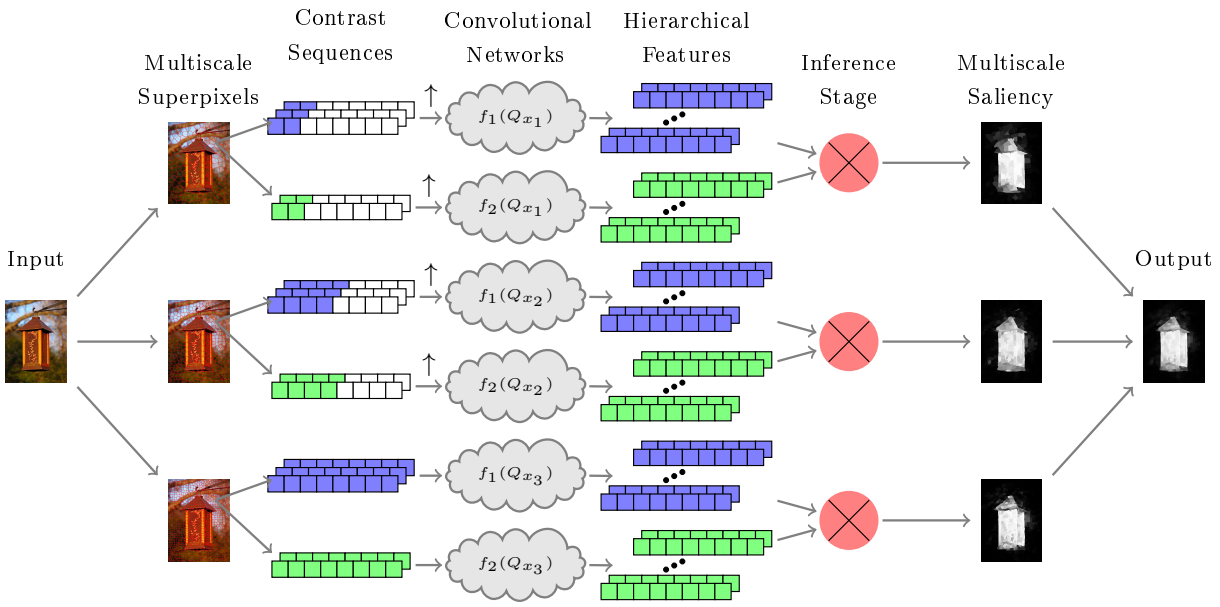


Fig. 2: Architecture of the proposed SuperCNN for salient object detection. The input image is first segmented into different numbers of superpixels (i.e., regions). Two meaningful sequences, color uniqueness sequence and color distribution sequence, are extracted from each superpixel and fed to the convolutional networks. (Upsampling is needed for coarser scales to keep the same input size.) Due to the superpixelwise strategy, each feature vector takes into account the information from the entire image in an efficient manner. The inferred results are integrated into the saliency scores. Finally, all the scales are combined to form a smooth saliency map.

these methods are based on computing local contrast, they are sensitive to high frequency content like image edges or noise only, and they attenuate any homogenous interior regions.

Global saliency methods estimate saliency by considering contrast relations over the entire image. Achanta *et al.* [1] detect salient regions by computing color deviation from the mean image color on a per-pixel basis. Cheng *et al.* [9] propose a fast color histogram based method, and compute saliency based on dissimilarity among the histogram bins. To take into account spatial relationships, Perazzi *et al.* [41] apply two contrast measures based on the uniqueness and spatial distribution of elements. Yan *et al.* [49] propose a hierarchical model to reduce the effect of small-scale structures on saliency detection. Recently, Jiang *et al.* [25] use two additional cues, focusness and objectness, together with color contrast to detect salient objects. Despite the demonstrated improvements, these methods measure saliency by fusing saliency maps computed from hand-crafted features, which are only able to capture low-level contrast information. While high-level knowledge has also been applied for detecting saliency [6,32], it is limited to specified objects or assumptions. Furthermore, incrementally adding more input features leads to a more complicated and time-consuming algorithm.

Deep learning techniques aim to learn hierarchical feature representations from natural images, where the higher-level features are defined from lower-level ones. CNNs and other deep learning techniques have been shown to be effective in many vision applications, such as face detection and pose estimation [39], facial point detection [44], scene parsing [14,42], and image classification [27]. The works on scene parsing using CNNs [14,42] have a similar spirit to our work, as both works can be treated as a labeling problem. However, unlike CNN-based scene parsing, SuperCNN does not require a complex network architecture or post-processing to handle large image context. To the best of our knowledge, our method is the first to explore the contrast information using CNNs.

Deep learning techniques have also been used in eye fixation prediction. Shen *et al.* [43] sample salient regions from the dataset, and learn the high-level features in an unsupervised manner using sparse coding. However, the models used to learn high-level information are usually limited to specific objects (e.g., faces or texts), due to the constrained categories of the dataset. On the other hand, predicting eye fixations is less useful than identifying the complete salient object in vision appli-

cations [31]. As a contrary, the proposed method is a general purpose salient object detector.

3 Salient Object Detection with SuperCNN

In general, human organizes information hierarchically. CNNs are driven by this observation. They aim at learning hierarchical internal representations. We formulate salient object detection as a binary labeling problem to predict whether an input region is salient. Generally, a CNN consists of alternating filter bank modules and spatial pooling modules. The hierarchical feature representations are learned via end-to-end training. The resulting output of each module is called feature maps (or feature vectors in 1D). The CNNs seek to find a highly nonlinear transformation function f to map the input into a space where the input can be linearly classified. To obtain a densely labeled map, classical CNNs divide the input image into grid patterns to be fed to the networks. The output feature maps are then classified as scores for each class of interest. However, these classical methods cannot be easily employed for salient object detection due to three problems:

- Saliency should be determined from a large context. To do this, the CNN typically needs to be fed with a sufficiently large image patch, leading to an unmanageable large network.
- Pixelwise prediction by the CNN can be noisy. Predicting an image with mega pixels also takes time, especially for large networks.
- Feeding raw image pixels to the networks is difficult to detect saliency, as saliency does not depend on particular object appearance (e.g., red color or human face).

In this section, we present SuperCNN to address the above problems and implement the transformation function f. Section 3.1 discusses how we extract hierarchial contrast features. Section 3.1.3 describes the network structure, and Section 3.2 presents the saliency score inference. Sections 3.3 and 3.4 describe the multiscale structure and the schemes to reduce overfitting, respectively.

3.1 Hierarchical Contrast Features Extraction

Regions that contrast strongly with their surroundings typically catch viewers' attention [13]. Our goal is to learn contextual contrast information, and thus the input of the CNNs must be meaningful to contrast and as raw as possible (since raw data is more flexible to learn good internal representations [14,42]). However, pixel-level contrast is computationally expensive when taking spatial information into account, not to mention the overhead of considering a large context with CNNs. Superpixel-based saliency detection methods [9,41,49,36] have shown to be accurate and efficient. These advantages motivate us to design superpixelwise convolutional networks.

Feeding superpixels to the CNNs faces a major issue – the structural information of the image is destroyed. Although some methods like superpixel lattices [38] are proposed to address this problem, the imposed lattice, however, sacrifices the segmentation accuracy. As shown in a psychology study [20], visual attention of the human visual system is mainly affected by the spatial distances of the surrounding objects. As a result, we aim at re-injecting the spatial information into the regions, rather than recovering the original image structure. We treat the segmented image as a 1D array, and recover the contextual information by introducing a spatial kernel to the color uniqueness. Other than spatial information, salient objects can typically be distinguished by color distribution. A range kernel is further applied to describe the distributional property of the salient objects. Hence, two meaningful input sequences are produced, and we feed them to a two-column CNN. As demonstrated in [11], multi-column CNNs fed with various inputs lead to complementary and superior predictions. In our implementation, images are segmented into regions using the SLIC superpixel method [2].

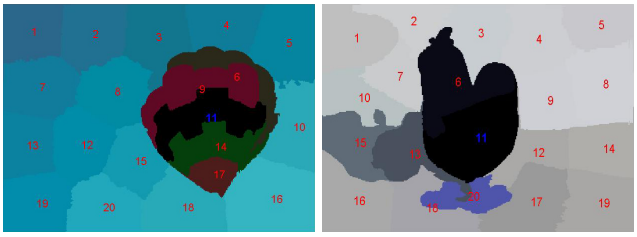
3.1.1 Color Uniqueness Sequence

Color uniqueness sequence is used to describe the color contrast of a region. Given an image I and the segmented regions $R = \{r_1, \ldots, r_x, \ldots, r_N\}$, each region r_x contains a color uniqueness sequence $Q_x^C = \{q_1^c, \ldots, q_j^c, \ldots, q_M^c\}$ with size M, where $M \leq N$. Each element, q_j^c is defined as:

$$q_j^c = t(r_j) \cdot |C(r_x) - C(r_j)| \cdot w(P(r_x), P(r_j)), \tag{1}$$

where $t(r_j)$ counts the total number of pixels in region r_j. Regions with more pixels are considered to have higher contributions to the contrast than those with fewer pixels. $C(r_x)$ is the mean color vector of region r_x (Figure 3a), $|C(r_x) - C(r_j)|$ is a 3D vector storing the absolute differences of each color channel (Figure 3b), $P(r_x)$ is the mean position of region r_x, $w(P(r_x), P(r_j)) = exp(-\frac{1}{2\sigma_s^2}||P(r_x) - P(r_j)||^2)$ is a Gaussian weight to describe the distance between r_x and r_j. The sequence Q_x^C is then sorted by the spatial distance to region r_x in order to maintain the spatially local correlation of the convolution operation (see Section 3.1.3). In addition, sorting the sequence can be

(a) Mean Color Images



(b) Absolute Difference Images

(c) Uniqueness Sequences

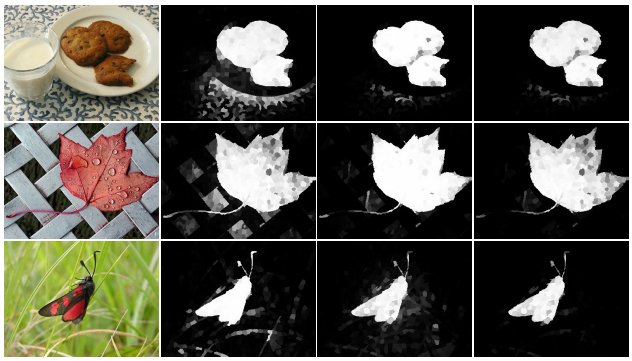
(d) Sorted Uniqueness Sequences

Fig. 3: Two example color uniqueness sequences. Both (c) and (d) refer to superpixel #11 in the two images shown in (a) and (b) (in blue color).

treated as adding extra information to the input sequence (Figure 3d) to help learn the internal representations. According to Eq. (1), each uniqueness sequence describes the relationship between region r_x and all the other $M-1$ regions by computing color differences. Figure 3 shows the intermediate results from the color uniqueness sequences.

By introducing $w(P(r_x), P(r_j))$, the spatial information is embedded into color uniqueness. Thus, the internal relationship between color uniqueness and spatial distance can be learned using CNNs. A salient region implies large differences over the entire color uniqueness sequence, while most of the elements of a non-salient region exhibits low values due to the low contrast with its neighboring regions. Since CNNs have the ability to collectively encode information, i.e., the network is able to learn the influential range of an input sequence and to determine the influential elements of the sequence, we set $M = N$ to cover the entire image.

Consequently, for each region, input Q_x^C is a 1D array of size M and contains three channels of absolute differences. The uniqueness sequence captures the dominant structure information by integrating color-spatial information. (Figure 4b shows the uniqueness saliency maps produced by SuperCNN.) However, considering global color rarity alone is not enough to handle all situations. For example, if a girl stands in front of a green field with several yellow flowers distributed in the field, the colors from the girl and the yellow flowers are rare colors in this scenario. However, only the girl should be considered as salient. Hence, we explore another input sequence as a complement.



(a) Input (b)Uniqueness (c) Distribution (d) Combined

Fig. 4: Saliency maps from the color uniqueness and distribution sequences, which focus on different color properties, producing complementary results.

3.1.2 Color Distribution Sequence

Although having contrast with the surrounding may indicate saliency, a high contrast region might not necessarily be salient (e.g., a background object). Detecting saliency with only color uniqueness often highlights background regions. Color distribution is complementary to color uniqueness. It is able to differentiate foreground objects from the background. Colors belonging to the foreground object are more compact, while colors belonging to the background are usually widely distributed over the whole image [31]. The second row of Figure 4 shows such an example – the colors of the grid are widely distributed, while the colors of the leaf are more compact.

We formulate a new input sequence to describe color distribution. Similar to the color uniqueness sequence, each region can be represented by a color distribution sequence $Q_x^D = \{q_1^d, \ldots, q_j^d, \ldots, q_M^d\}$. Each element q_j^d is defined as:

$$q_j^d = t(r_j) \cdot |P(r_x) - P(r_j)| \cdot w(C(r_x), C(r_j)), \qquad (2)$$

where $w(C(r_x), C(r_j)) = exp(-\frac{1}{2\sigma_r^2}||C(r_x) - C(r_j)||^2)$ is the range kernel to describe color similarity. The position value is normalized to adapt to different image sizes. By integrating the range kernel into the position difference, the distribution of color $C(r_x)$ can be easily identified. Similar to the uniqueness sequence, Q_x^D

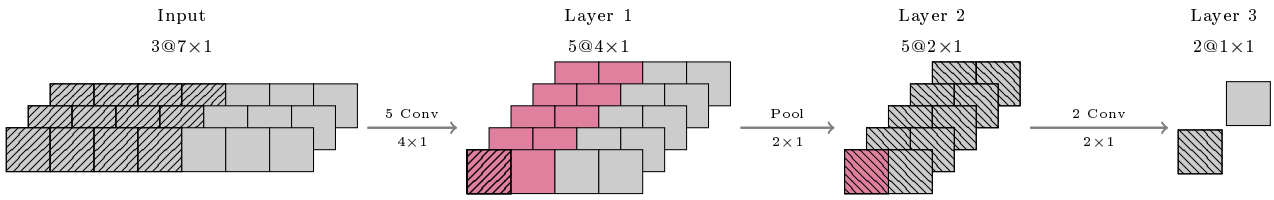


Fig. 5: A simple example of our color uniqueness network. For each superpixel, there is a 1D array (the number of superpixels is 7 here) containing three channels of absolute differences. There are three layers in this example. We first apply five 4×1 convolution operations to the input array, followed by one 2×1 max pooling, and then two 2×1 convolutions. The final output can be represented as class distributions.

is sorted by the spatial distance to region r_x. Typically for a salient object, the regions within this object should exhibit small position differences but high color similarity. These combined values encode the distribution relationship within the sequence, and thus can be learned by CNNs. In addition, the distribution sequence describes objectness information rather than color contrast, which is a complement to the color uniqueness sequence, as shown in Figure 4c. On the other hand, a psychophysical study [46] shows that human usually pays more attention to the center region of the image. This preferred location of salient objects can also be learned by the distribution sequence, as it is constructed by spatial differences (i.e., the value of an element will be different if it is located at the center or at the boundary). Like the color uniqueness sequence, we set $M = N$ to include the whole image.

3.1.3 Network Structure

The proposed SuperCNN has a multi-column trainable architecture. Each of the columns is fed with 1D sequences. It is a feature extractor and consists of sequential layers. Figure 5 illustrates a simple example of our color uniqueness column. The two operators included in the example form two key properties of the CNN. The convolutional operator exploits spatial correlation among the local regions, and the max pooling operator reduces the computational complexity and provides invariance to slight translations. Our network architecture extends the network shown in Figure 5 to three sequential stages, each of which contains multiple layers. This three-stage architecture is inspired by [27,14, 15], which obtain state-of-the-art performances with efficiency using a similar architecture on different applications. There are two layers involved in the first two stages: a filter bank layer and a spatial pooling layer. The pooling layer is always followed by a nonlinearity function. The last stage only contains a filter bank layer. Finally, each column is followed by a classification module.

For a network f_u at column $u \in \{1, \ldots, U\}$ with L layers, given an input sequence Q_x, the output of f_u can be represented as:

$$f_u(Q_x) = W_{u,L} H_{u,L-1}, \tag{3}$$

where $H_{u,l}$ at layer l can be computed as:

$$H_{u,l} = tanh(pool(W_{u,l} H_{u,l-1} + b_{u,l})), \tag{4}$$

where $l \in \{1, \ldots, L\}$ and $H_0 = Q_x$. $W_{u,l}$ is the Toeplitz matrix of connection between layers l and $l-1$. $b_{u,l}$ is the bias vector. Filters $W_{u,l}$ and bias vectors $b_{u,l}$ are the trainable parameters of the network. The filter banks perform a 1D convolution operation on the input to produce multiple feature maps, each of which describes local information of the input. Spatial pooling operator *pool* is able to inject spatial invariance while passing the features to the next layer. Max-pooling is used in our implementation and pooling regions do not overlap. The nonlinearity is brought by the point-wise hyperbolic tangent function *tanh*.

Finally, there are U output feature maps F_u produced. For each of these feature maps, the regions within it are classified as either belonging to the salient object or not, once the networks are properly trained. As our goal is to compute a saliency value instead of a binary value, we apply a softmax activation function to transform the network scores into conditional probabilities of whether each region is salient. For each region, $a \in \{0, 1\}$ indicates the saliency binary label. The class distributions $d_{u,a}$ of region r_x are predicted from F_u by a two-layer neural network:

$$y_{u,x} = W_{u,c2} \tanh(W_{u,c1} F_u(r_x) + b_{u,c1}), \tag{5}$$

$$d_{u,a}(r_x) = \frac{e^{y^a_{u,x}}}{e^{y^0_{u,x}} + e^{y^1_{u,x}}}, \tag{6}$$

where $W_{u,c1}$, $W_{u,c2}$ and $b_{u,c1}$ are the trainable parameters of the classifier at the u^{th} column. The network trainable parameters $W_{u,l}$ and $b_{u,l}$ are trained

in a supervised manner, by minimizing the negative log-likelihood (NLL) between the prediction and the groundtruth over the training set:

$$L(W_{u,l}, b_{u,l}) = -\sum_{x \in R} \sum_{a \in \{0,1\}} \hat{d}_{u,a}(r_x) \ln(d_{u,a}(r_x)), \quad (7)$$

where $\hat{d}_{u,a}(r_x)$ is the groundtruth class distribution. The minimization is implemented through stochastic gradient descent (SGD).

3.2 Saliency Inference

To determine the saliency of a region, each network column predicts a two-class distribution d_u, which typically takes the argmax for classification (i.e., salient object segmentation). The class distribution of a region being salient, i.e., $a = 1$, in Eq. (6) is a positive normalized value and therefore can be considered as saliency confidence. The saliency value of region r_x is defined as: $S_u(r_x) = d_{u,1}(r_x)$. Saliency map S_u is then normalized to $(0, 1)$.

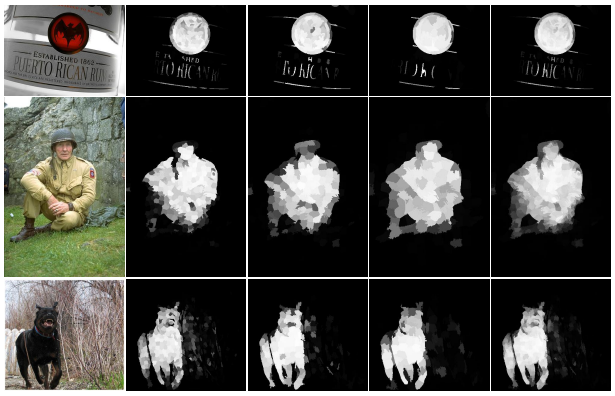
In our framework, U is set to 2, as we have two input sequences. Two saliency maps are obtained, each of which is complementary to the other. A common approach to integrate multiple cues is by linear summation or multiplication. As we seek to obtain objects that are salient in all cues, we employ multiplication to integrate the saliency maps. The final saliency map can be obtained by:

$$S(r_x) = \prod_{u \in U} v_c^u \cdot S_u(r_x), \quad (8)$$

where v_c^u is the learned weight by linear regression according to the mean absolute difference between the saliency map and the groundtruth. Saliency map S is again normalized to $(0, 1)$. Figure 4 shows the predictions of the color uniqueness sequences, the color distribution sequences and the final saliency map. The superpixelwise strategy easily embeds the global relationships in the predictions. Thus, it is able to avoid post-processing (e.g., conditional random field) to maintain the consistency of the labeling results.

3.3 Multiscale SuperCNN

Although Eq. (8) considers different aspects of saliency, its performance may still be affected by the size of the pattern. In addition, the resulting saliency map may not be smooth enough (see Figure 6b – 6d). We handle this problem with a multiscale structure. The number of superpixels is set differently depending on the scale.



(a) Input (b) Scale 1 (c) Scale 2 (d) Scale 3 (e) Final

Fig. 6: Saliency maps produced from three scales.

A small number of superpixels is able to diminish the color contrast effect from small scale patterns. Segmentation errors can be mitigated by considering multiple scales. However, different numbers of superpixels (i.e., different input sequences) may lead to extra training efforts and more parameters to be handled. Similar to [14], we share the parameters across different scales.

Given the number of superpixels N of the finest scale, the other scales are set to $N/2^g$, where $g \in \{1, \ldots, G\}$ is the scale number. The input sequences from the other scales are then upsampled to have the same size, i.e., N. Finally the upsampled sequences are fed to the networks. The networks for all the other scales are copies of the finest scale networks, sharing all parameter values. The final multiscale saliency map is the weighted sum of all the scales:

$$S_f = \sum_{g \in G} v_s^g \cdot S^g. \quad (9)$$

Similar to Eq. (8), v_s^g is the learned weight using linear regression. S_f is then normalized. Results of the multiscale saliency are shown in Figure 6. The resulting saliency maps may recover small scale patterns, while producing smooth predictions, which are important for energy-based image editing [4].

3.4 Reducing Overfitting

While the proposed superpixewise framework hugely reduces the required number of predictions during testing, the overfitting problem may occur as the number of examples within the training data is reduced. Since a training image has only N superpixels (e.g., $N = 1000$), only N training examples are available in an image. As

Approach	Test error
No data augmentation and Dropout	24.3%
Data augmentation	20.1%
Data augmentation + Dropout	17.4%

Table 1: The test errors of the color uniqueness network with respect to different approaches to reduce overfitting. The reported results are the binary classification error rates on the ECSSD dataset.

a typical saliency dataset contains only hundreds of labeled images, the total number of available training examples is insufficient to train the proposed SuperCNN with a large number of parameters.

Introducing jitter to the training data has been shown very effective to augment the dataset and prevent overfitting [10]. We apply bounded random distortions to each training example as pre-processing during training. The distortion level is randomly determined from a specified range, including horizontal reflection, rescaling ($\pm 5\%$), translation ($\pm 10\%$ of the image size), rotation ($\pm 5^\circ$) and superpixel number (-5%). These distortions greatly increase the size of the training set, allowing us to learn a large number of parameters without considerable overfitting.

We further combat overfitting by regularizing the neural networks. While numerous regularizers [5,8] have been proposed to prevent overfitting, the recently proposed "dropout" [19] has shown to be effective and efficient. The main idea of dropout is to set the activations to zero with probability 0.5 to prevent co-adaptation of neurons. By randomly dropping out neurons, the procedure forces each neuron to rely on the population behavior of its inputs rather than relying excessively on the outputs of other neurons. We use dropout in the first two convolutional layers. Dropout plays a significant role in SuperCNN, as it is able to avoid overfitting with relatively small training size and improve generalization performance. A limitation of this approach is that it typically doubles the converging time in training. Table 1 shows the binary classification test errors (i.e., using the class with higher distribution as the label) of the color uniqueness network with respect to different approaches for combating overfitting.

4 Experiments

To study the performance of the proposed salient object detection framework, we have quantitatively and qualitatively evaluated it on three fully labeled datasets: MSRA-1000 [1] (1000 images), the Berkeley image set BSDS-300 [3] (300 images), and a newly proposed dataset ECSSD [49] (1000 images). All three datasets have manually segmented groundtruth. While the MSRA-1000 dataset is widely used by most methods for evaluation, the other two datasets are more challenging. The ECSSD dataset contains images with more complex backgrounds, and most images in the BSDS-300 dataset contain multiple objects.

We train the proposed SuperCNN on the ECSSD dataset as it consists of more complex scenes, allowing SuperCNN to learn more robust features for real world scenes. The training follows a 5-fold cross validation – 800 images are randomly selected for training and the remaining 200 images are used for testing. It involves training a two-column, three-stage neural network system. The first two stages both include a bank of filters of kernel size 50, a max-pooling operator of kernel size 2 and a *tanh* unit. The last stage contains only a bank of filters. The other two datasets are evaluated using the trained network.

Each input image is transformed to the CIE LAB space, and the input sequences are normalized to have zero mean and unit variance. We use a three-scale structure, 1000, 500 and 250. We set the finest scale to 1000 superpixels in order to learn more representative hierarchical features. As it may not be possible to obtain an exact number of superpixels, we force the SLIC algorithm to produce no more than the specified number of segments and the remaining elements are then filled with zeros. The standard deviations σ of the spatial and range kernels of the input sequences are set to 0.4, both in training and testing.

For the network column handling the color uniqueness sequence, the first stage transforms each 3-channel input feature vector to 16 dimensions. The second stage further transforms it to 64 dimensions. Finally, the third stage transforms it to an output 256D feature vector. The first two convolutional layers are connected to all the kernel maps in the previous layers, and the last layer is a combination of 32 randomly connected feature vectors from the previous layer. For the network column handling the color distribution sequences, the input is a 2D feature vector. The architecture is the same as the column for handling color uniqueness, except for the numbers of bank of filters; instead of being 16, 64 and 256, they are now 12, 48 and 192, respectively. The last layer is a combination of 24 randomly connected feature vectors from the previous layer. The size of the input sequence is the same as the number of superpixels (i.e., 1000) to involve the whole image in making a global decision. The proposed SuperCNN typically needs four to six days for the training to converge.

The proposed framework is implemented in Lua, using *Torch7* toolbox [12], and tested on a PC with a 4-core i7 CPU and 18GB RAM. Due to the special

network structure, [14] shows that CNNs can be parallelized on either CPUs or GPUs. Taking advantage of the parallel implementation of convolutions on CPU, our algorithm takes on average $0.45s$ to process one image of resolution 400×300. Segmentation and preprocessing take a total of $0.32s$, while the detection time takes $0.13s$. Despite using a large number of superpixels (vs. GMR [50] using only 200 superpixels), the execution time is comparable to the state-of-the-art methods (e.g., GMR [50] takes $0.38s$ to process one image).

4.1 Quantitative Evaluation

We have compared the proposed method with 7 state-of-the-art methods, the top three algorithms according to [7] (RC [9], CA [16], CBS [23]), plus four latest algorithms (GMR [50], HS [49], PCA [36], GC [37]). The implementations provided by the authors were used for fair comparison. Like the previous work [41,37], we quantitatively evaluate the performances of these methods by measuring their *precision-recall* values, F-measure numbers, and mean absolute errors.

4.1.1 Precision and Recall

Precision indicates the percentage of the output salient pixels that are correct, while recall indicates the percentage of the ground truth pixels detected. The F-measure is defined as:

$$F_\beta = \frac{(1+\beta^2) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}, \tag{10}$$

where β^2 is set to 0.3 to emphasize on precision [1].

Two different threshold criteria are used to conduct the evaluation. In the first experiment, we binarize the saliency map for every threshold in the range of $[0, 255]$. In the second experiment, we use an image dependent adaptive threshold [1], which is computed as twice the mean value of the saliency map.

Results of the evaluations on the three datasets are shown in Figures 8, 10 and 12. It is interesting to note the extremities of the precision and recall curves in the first experiment (i.e. fixed threshold, Figure 8a, 10a and 12a): at maximum recall where the threshold is equal to zero and thus all the pixels are considered as salient, i.e., the precision value at maximum recall indicate the average percentage of salient pixels within an image of the dataset (that is why all the methods have the same precision at this recall). As can be seen, the average sizes of salient objects are relative small for all the three datasets, which implies that to obtain high precision is more important and difficult than high recall. On the other hand, the minimum recall value indicate the robustness of the saliency detection algorithm, high precision at this recall represent most of the high confidence saliency values (close to 255) are correctly located to the salient object.

Although the proposed method performs similarly to GMR [50] on the simple MSRA-1000 dataset (Figure 8), it performs much better than all other methods on the other two challenging datasets (Figure 10 and 12). Furthermore, the saliency maps produced by our method are smoother and contain more correctly assigned high confidence salient regions than all the other methods (i.e. high precisions at low recall values). In addition, we achieve the best F-measures either using fixed or adaptive thresholds on all the three datasets. The main reason is that the hand-crafted features of the other methods are somewhat limited to particular scenarios. (We will be visually verified this in Section 4.2.) For example, the performance of GMR [50] is guaranteed while the background prior (i.e., pixels close to the image boundary likely belong to the background) is valid (this assumption can be met easily in the simple MSRA-1000 dataset but not for the others); PCA [36] uses patch-based features that tend to highlight object boundaries. On the contrary, our method learns the hierarchical contrast features in a global manner, which are more robust than the hand-crafted features especially for complex scenes like cluttered background or similar colors between foreground and background.

4.1.2 Mean Absolute Error

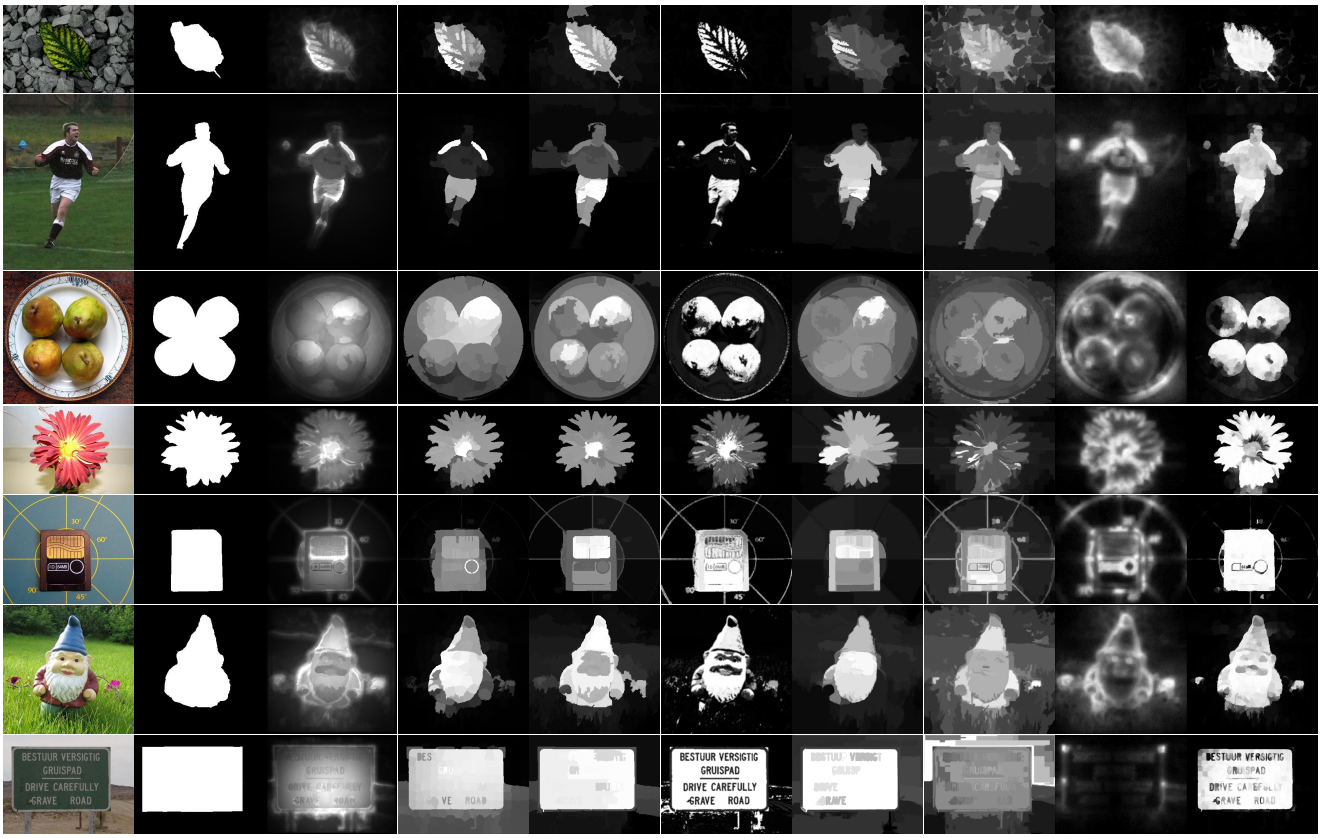
For a faithful comparison, *mean absolute error* (MAE) [41] is introduced to reflect the negative saliency assignments. It is defined between a saliency map S and the binary groundtruth GT as:

$$MAE = \frac{1}{|I|} \sum_x |S(I_x) - GT(I_x)|, \tag{11}$$

where $|I|$ is the total number of pixels. The MAE results on three datasets are shown in Figure 13. The proposed method achieves the lowest MAE values on all three datasets. This means that its predicted saliency pixels are very close to those of the groundtruth. This is partly due to the way that we aggregate two saliency measures – we aim to obtain objects that are salient in both measures.

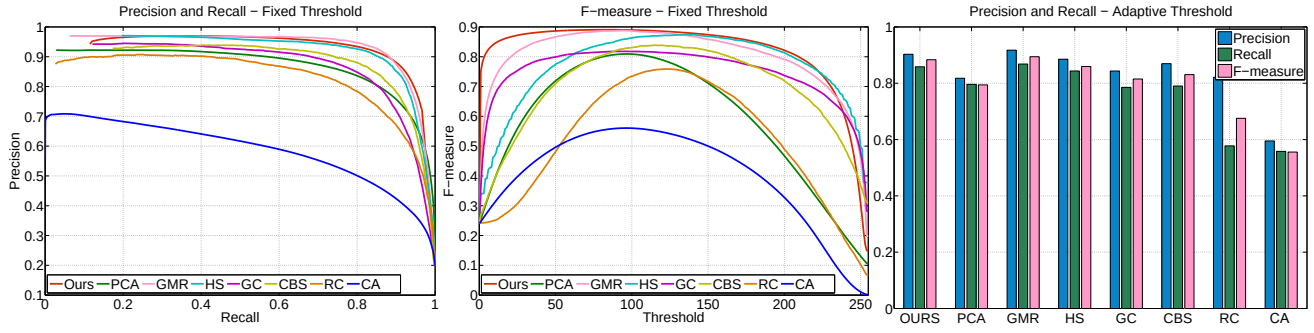
4.1.3 Component Analysis

We further evaluate the effectiveness of different components: color uniqueness sequences, color distribution



(a) Input (b) GT (c) PCA [36] (d) GMR [50] (e) HS [49] (f) GC [37] (g) CBS [23] (h) RC [9] (i) CA [16] (j) Ours

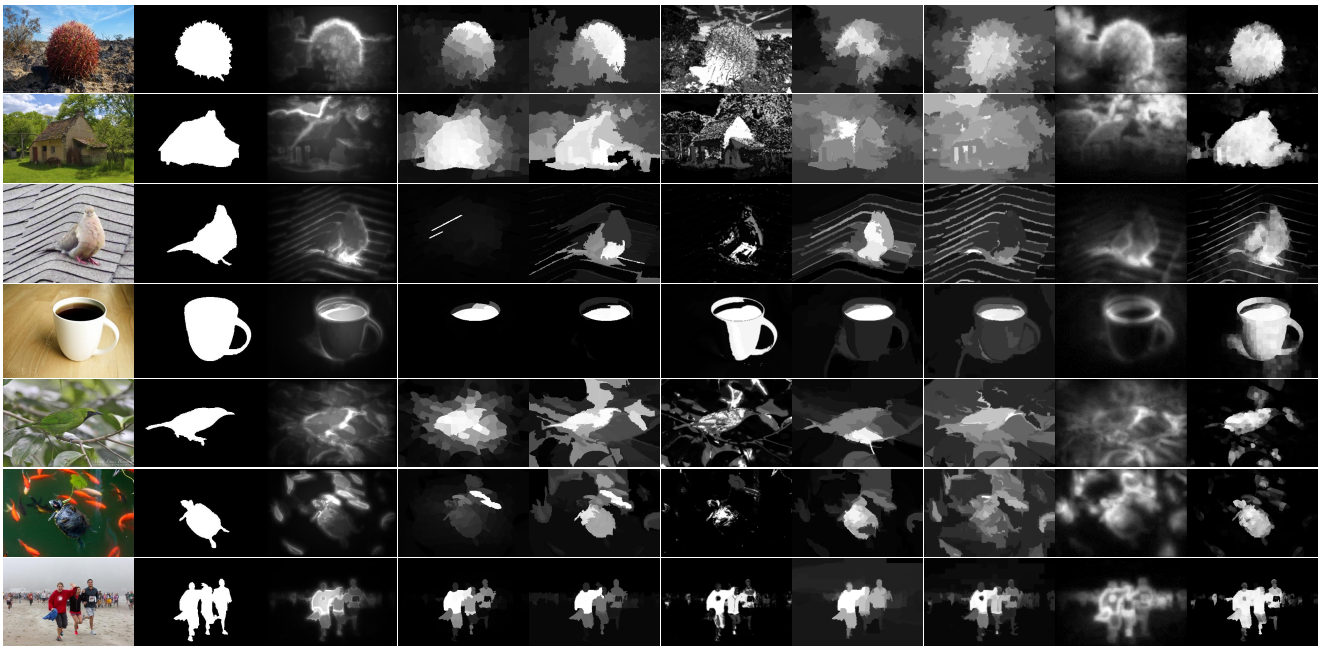
Fig. 7: Qualitative comparison of the state-of-the-art methods on the MSRA-1000 dataset. Our learned hierarchical features are able to render the entire objects as salient, yielding continuous saliency maps that are closest to the groundtruth. The quantitative evaluation of the MSRA-1000 dataset is presented in Figure 8.



(a) (b) (c)

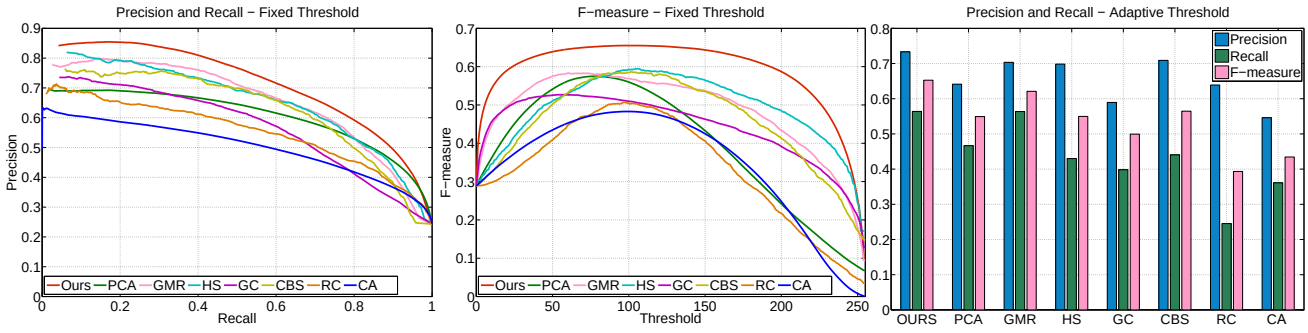
Fig. 8: Quantitative evaluation on the MSRA-1000 dataset. (a) Precision-recall curves. (b) F-measure curves. (c) Precision, recall, and F-measure for adaptive thresholds. The proposed approach consistently produces better results than the state-of-the-art methods.

sequences and three scales. Results in Figure 14 show the importance of constructing two complementary sequences to form a two-column neural network. As the color uniqueness sequence and the color distribution sequence (i.e., CU and CD in Figure 14, both computed from the finest scale, Scale 1) predict saliency in different aspects, the combined results achieve superior performance. The multi-scale structure also achieves a better recall than a single scale structure, as small scale patterns are recovered (i.e., Scale 1 – 3 vs. Final in Figure 14c).



(a) Input (b) GT (c) PCA [36] (d) GMR [50] (e) HS [49] (f) GC [37] (g) CBS [23] (h) RC [9] (i) CA [16] (j) Ours

Fig. 9: Qualitative comparison of the state-of-the-art methods on the ECSSD dataset. Our learned hierarchical features are able to render the entire objects as salient in clustered backgrounds, yielding continuous saliency maps that are closest to the groundtruth. The quantitative evaluation of the ECSSD dataset is presented in Figure 10.

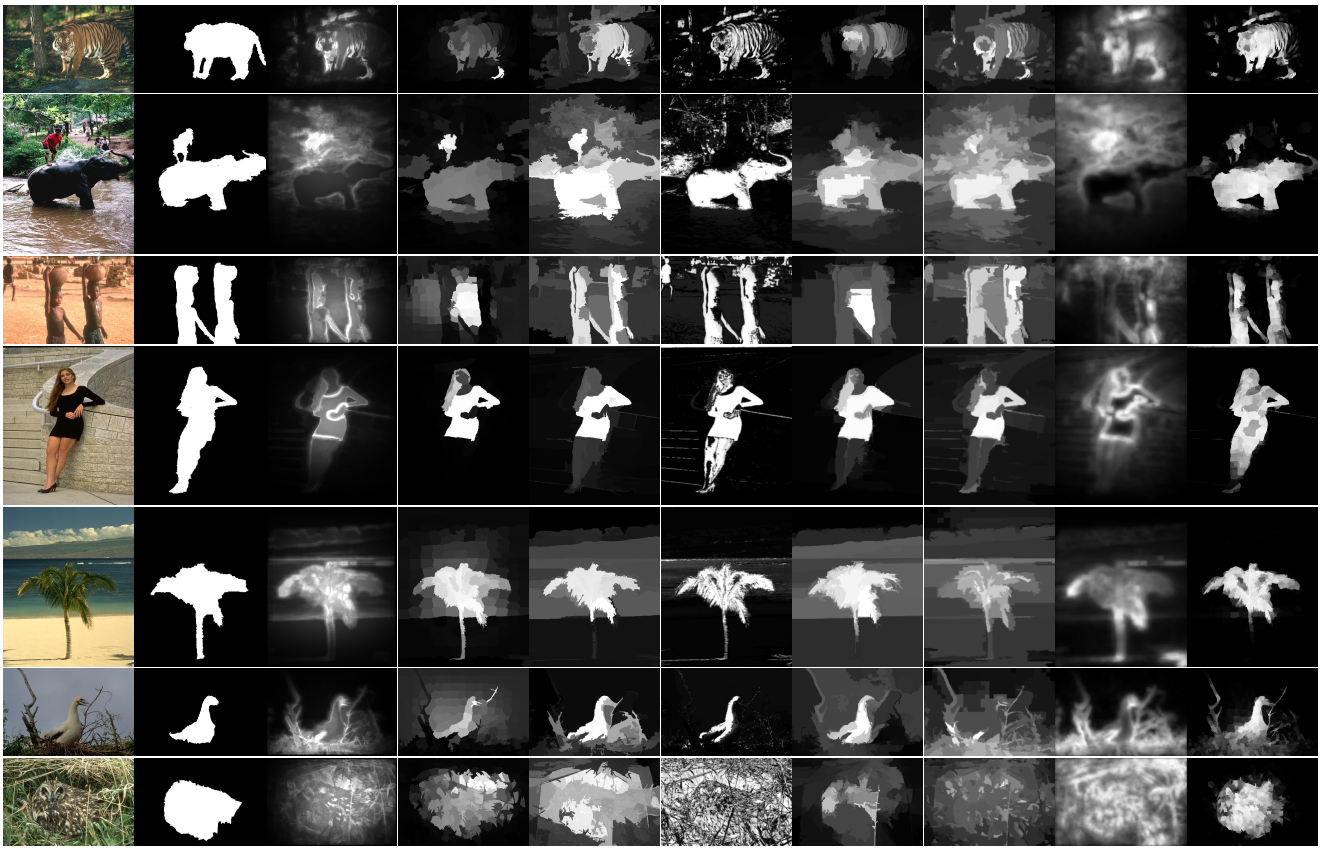


(a) (b) (c)

Fig. 10: Quantitative evaluation on the ECSSD dataset. (a) Precision-recall curves. (b) F-measure curves. (c) Precision, recall, and F-measure for adaptive thresholds. The proposed approach consistently produces better results than the state-of-the-art methods.

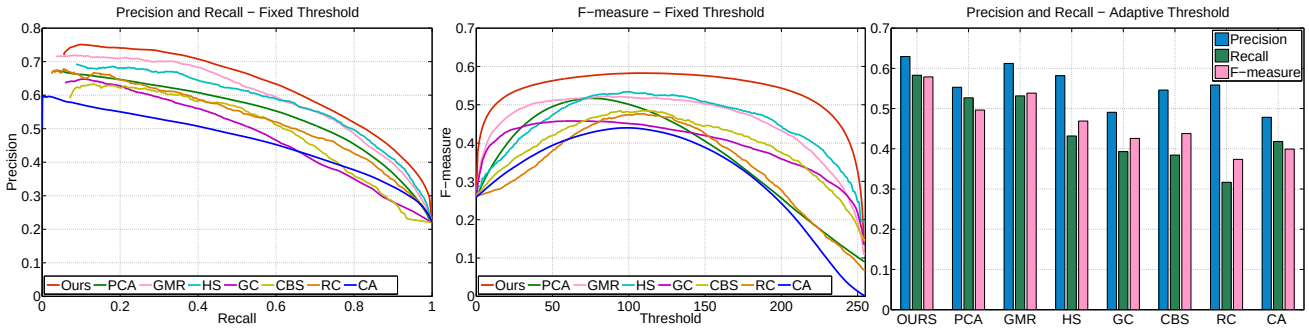
4.2 Qualitative Evaluation

Figures 7, 9 and 11 show visual comparisons of the state-of-the-art methods on the three datasets. All the existing methods perform poorly in the scenarios with cluttered background (e.g., the sixth row of Figure 9 and the last row of Figure 11). Methods that rely on background priors, such as GMR [50], cannot successfully render salient objects when these assumptions are invalid (e.g., the colors of the image boundary are similar to the salient object in the fifth row of Figure 9). The proposed method, on the contrary, is able to distinguish salient objects from these complex distractions. On the other hand, as the learned features are hierarchical and capture salient information in a global manner, the proposed method is able to extract the whole object properly. For example, in the second row of Figure 7 and the fourth row of Figure 11, the proposed method assigns smooth saliency values to the whole persons, while the other methods only capture parts of them. In contrast to patch-based or central-surround features like PCA [36] and CA [16], which attenuate homogeneous regions to certain extent, the proposed method renders highly salient regions over the entire homogenous regions. In addition, the assigned saliency values are very confidence (close to the groundtruth),



(a) Input (b) GT (c) PCA [36] (d) GMR [50] (e) HS [49] (f) GC [37] (g) CBS [23] (h) RC [9] (i) CA [16] (j) Ours

Fig. 11: Qualitative comparison of the state-of-the-art methods on the BSDS-300 dataset. Our learned hierarchical features are able to render the entire objects as salient in complex scenarios, yielding continuous saliency maps that are closest to the groundtruth. The quantitative evaluation of the BSDS-300 dataset is presented in Figure 12.



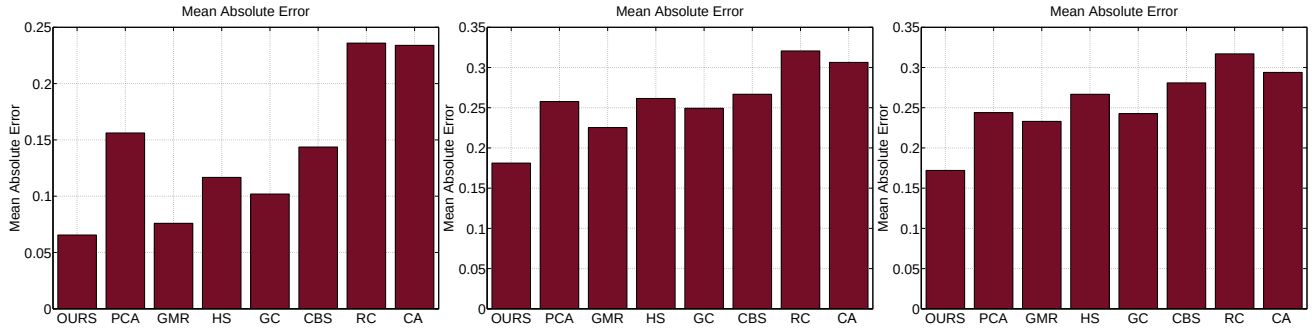
(a) (b) (c)

Fig. 12: Quantitative evaluation on the BSDS-300 dataset. (a) Precision-recall curves. (b) F-measure curves. (c) Precision, recall, and F-measure for adaptive thresholds. The proposed approach consistently produces better results than the state-of-the-art methods.

which is beneficial to applications like segmentation. We further discuss two saliency applications using the proposed method, image resizing and stylization.

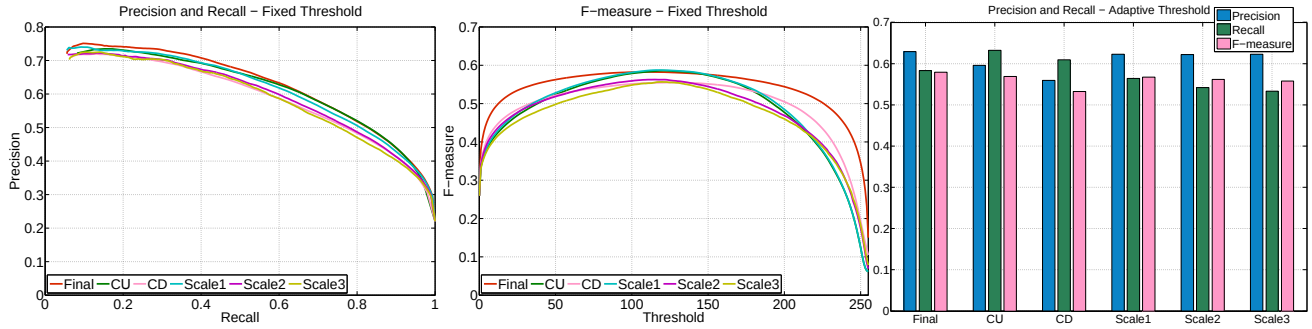
4.2.1 Image Resizing

While image resizing is a popular operation, it often affects the image content or the aspect ratio. Effective image resizing should be content-aware. Salient objects typically represent important image content to be preserved. Here, we examine the importance of obtaining



(a) MSRA-1000 dataset. (b) ECSSD dataset. (c) BSDS-300 dataset.

Fig. 13: Mean absolute errors of the state-of-the-art methods on the three datasets. The proposed approach consistently achieves the lowest error rates on all three datasets.



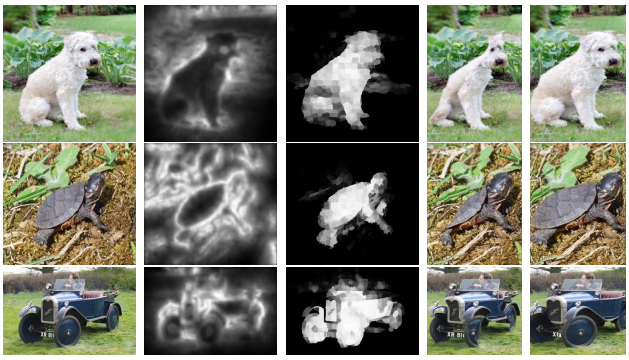
(a) (b) (c)

Fig. 14: Evaluation of different components in the proposed framework. The experiments are performed on the BSDS-300 dataset. (a) Precision-recall curves. (b) F-measure curves. (c) Precision, recall, and F-measure for adaptive thresholds. We can observe the advantages of aggregating two complementary network columns and the multi-scale structure. The aggregation of color uniqueness (CU) and color distribution (CD) leads to a better precision as it removes background distractions (i.e., CU and CD vs. Scale 1). The multi-scale structure leads to a better recall as it recover the small scale patterns (i.e., Scale 1 – 3 vs. Final).

continuous saliency maps to the energy-based resizing techniques. The comparison is conducted between the proposed method and the context-aware saliency detection method CA [16], using the non-homogeneous resizing technique proposed by Avidan et.al [4]. We can see from Figure 15 that the resizing results produced by our continuous saliency maps preserve the important objects very well. On the contrary, since CA (and other methods that emphasize on edges) cannot detect interior regions, the resulting saliency maps may mislead the resizing algorithm. We note that the proposed method consistently produces smooth saliency maps, which is important for energy-based applications.

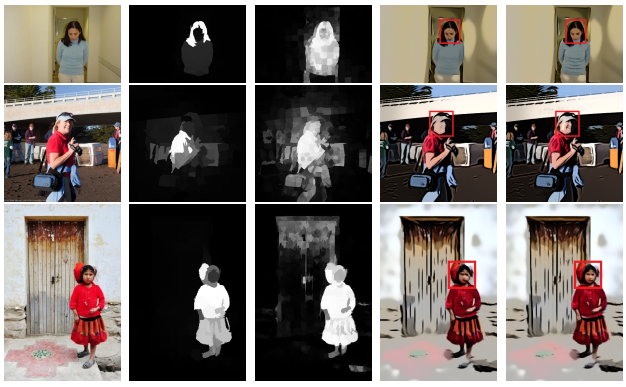
4.2.2 Image Stylization

Similar to photographers, artists have tendency to emphasize the important objects in the scene when they paint. The emphasized objects are usually drawn with far more details than the background. This observation has been adopted by non-photorealistic rendering techniques to generate interesting effects. Here, we compare the proposed method with the state-of-the-art method GMR [50], using XDoG [48] for portrait stylization. For the portrait images shown in Figure 16(a), the persons are the salient objects. Artists would tend to capture more details from the faces and the bodies. However, Figure 16(b) shows that GMR fails to recover the entire human bodies by using hand-crafted features, due to the distraction from the high contrast outliers (e.g., clothes or backgrounds). On the contrary, Figure 16(c) shows that the proposed method can recover the whole bodies as salient. Hence, the details of the persons' faces are well preserved after stylization.



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

Fig. 15: Importance of obtaining continuous saliency maps to content aware image resizing [4]. (b) The saliency maps produced by CA [16] tend to emphasize edges. (c) The saliency maps produced by the proposed method are able to recover homogeneous regions. Resizing results of CA [16] (d) and the proposed method (e) show the importance of extracting continuous saliency.

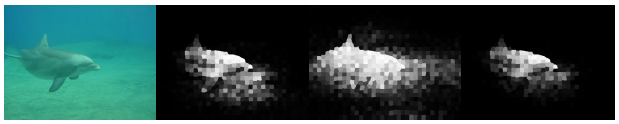


(a) (b) (c) (d) (e)

Fig. 16: Importance of detecting the entire salient objects to portrait stylization [48]. (b) The saliency maps produced by GMR [50] fail to detect faces of the salient objects. (c) The saliency maps produced by the proposed method include the whole salient objects. Hence, the stylization results of GMR [50] (d) cannot preserve face details, while the proposed method can (e).

4.3 Limitations

Although the proposed method is able to detect salient objects by learning hierarchical features in a global manner, the learned features still rely on contrast information. For a scene with similar foreground/background colors, the contrast information is usually invalid. Some image enhancement techniques like histogram equalization may not guarantee to highlight the salient objects. In fact, all existing approaches also suffer from this limitation, which can only be addressed by introducing extra information such as depth [18]. On the other hand, the input sequences of our networks include positional information, which may help recover the salient objects to a certain extent. In other words, the proposed method can predict the potential location of salient objects (most likely near to the center of the image) when the contrast information is not available, as shown in Figure 17.



(a) Input. (b) CU result. (c) CD result. (d) Combined.

Fig. 17: A failure case of the proposed method. Although the proposed method fails due to the low contrast between the salient object and the background, the learned positional information still helps recover the salient object to a certain extent.

Similar to other learning-based saliency detection methods [31], we require an extra training step, which takes a few days. On the other hand, once the networks are properly trained, the resulting detector can robustly extract salient objects in an efficient manner without parameter adjustment.

5 Conclusion and Future Work

In this paper, we propose a superpixelwise convolutional neural network approach for saliency detection, called SuperCNN. We overcome the barriers of classical CNNs that they are not suitable for contrast extraction and are only able to capture high-level information of specific categories. SuperCNN is a general purpose saliency detector. While it takes into account the whole image to make a global decision, it also significantly reduces the required number of predictions in runtime. In order to capture saliency information, two meaningful superpixel sequences, the color uniqueness and the color distribution sequences, are proposed to extract saliency properties. Due to the efficiency of the superpixelwise mechanism, the proposed SuperCNN can be applied to other CNN applications, such as image segmentation [15], image classification [27] and image parsing [14].

As a future work, we are currently considering to jointly train the two columns of SuperCNN. As shown in a recent work for pose estimation [30], jointly training two networks, one for joint point regression and

one for body part detection, is able to achieve superior performance compared with individually training each network. Another possible future work is to redesign SuperCNN into a deeper network. The top performer [45] in the latest ImageNet LSVRC-2014 contest shown that a carefully crafted deep architecture (22 layers) is able to achieve a surprisingly high performance in image classification, while maintaining efficiency.

Acknowledgements We would like to thank the anonymous reviewers for their insightful comments and constructive suggestions. The work described in this paper was partially supported by a GRF grant and an ECS grant from the RGC of Hong Kong (RGC Ref.: CityU 115112 and CityU 21201914).

References

1. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: CVPR, pp. 1597–1604 (2009)
2. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Ssstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE TPAMI pp. 2274–2282 (2012)
3. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE TPAMI **33**(5), 898–916 (2011)
4. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. ACM TOG **26**(3) (2007)
5. Bell, R., Koren, Y.: Lessons from the netflix prize challenge. SIGKDD Explorations Newsletter **9**(2), 75–79 (2007)
6. Borji, A.: Boosting bottom-up and top-down visual features for saliency estimation. In: CVPR, pp. 438–445 (2012)
7. Borji, A., Sihite, D., Itti, L.: Salient object detection: A benchmark. In: ECCV (2012)
8. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
9. Cheng, M., Zhang, G., Mitra, N., Huang, X., Hu, S.: Global contrast based salient region detection. In: CVPR, pp. 409–416 (2011)
10. Ciresan, D., Meier, U., Masci, J., Schmidhuber, J.: A committee of neural networks for traffic sign classification. In: IJCNN, pp. 1918–1921 (2011)
11. Ciresan, D.C., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: CVPR, pp. 3642–3649 (2012)
12. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: A Matlab-like Environment for Machine Learning. In: BigLearn NIPS Workshop (2011)
13. Einhauser, W., Konig, P.: Does luminance-contrast contribute to a saliency map for overt visualn attention? European Journal of Neuroscience **17**(5), 1089–1097 (2003)
14. Farabet, C., Couprie, C., Najman, L., Lecun, Y.: Learning hierarchical features for scene labeling. IEEE TPAMI **35**(8), 1915–1929 (2013)
15. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
16. Goferman, S., Zelnik-Manor, L., Tal, A.: Context-aware saliency detection. In: CVPR (2010)
17. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: NIPS, pp. 545–552 (2007)
18. He, S., Lau, R.: Saliency detection with flash and no-flash image pairs. In: ECCV, pp. 110–124 (2014)
19. Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. CoRR **abs/1207.0580** (2012)
20. Intriligator, J., Cavanagh, P.: The spatial resolution of visual attention. Cognitive Psychology **43**(3), 171 – 216 (2001)
21. Itti, L., Koch, C.: Computational modelling of visual attention. Nature Reviews Neuroscience **2**(3), 194–203 (2001)
22. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE TPAMI **20**(11), 1254–1259 (1998)
23. Jiang, H., Wang, J., Yuan, Z., Liu, T., Zheng, N.: Automatic salient object segmentation based on context and shape prior. In: BMVC (2011)
24. Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., Li, S.: Salient object detection: A discriminative regional feature integration approach. In: CVPR, pp. 2083–2090 (2013)
25. Jiang, P., Ling, H., Yu, J., Peng, J.: Salient region detection by ufo: Uniqueness, focusness and objectness (2013)
26. Koch, C., Ullman, S.: Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. Human Neurobiology **4**, 219–227 (1985)
27. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1106–1114 (2012)
28. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
29. Lee, T., Mumford, D.: Hierarchical Bayesian inference in the visual cortex. Journal of the Optical Society of America A **20**(7), 1434–1448 (2003)
30. Li, S., Liu, Z.Q., Chan, A.: Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. IJCV pp. 1–18 (2014)
31. Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., Shum, H.: Learning to detect a salient object. IEEE TPAMI **33**(2), 353–367 (2011)
32. Lu, Y., Zhang, W., Jin, C., Xue, X.: Learning attention map from images. In: CVPR, pp. 1067–1074 (2012)
33. Ma, Y., Zhang, H.: Contrast-based image attention analysis by using fuzzy growing. In: ACM Multimedia, pp. 374–381 (2003)
34. Macaluso, E., Frith, C., Driver, J.: Directing attention to locations and to sensory modalities: Multiple levels of selective processing revealed with pet. Cerebral Cortex **12**(4), 357–368 (2002)
35. Marchesotti, L., Cifarelli, C., Csurka, G.: A framework for visual saliency detection with applications to image thumbnailing. In: CVPR, pp. 2232–2239 (2009)
36. Margolin, R., Tal, A., Zelnik-Manor, L.: What makes a patch distinct? In: CVPR (2013)
37. Ming-Chng, Warrell, J., Lin, W., Zheng, S., Vineet, V., Crook, N.: Efficient salient region detection with soft image abstraction. In: ICCV (2013)
38. Moore, A., Prince, S., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In: CVPR, pp. 1–8 (2008)
39. Osadchy, M., LeCun, Y., Miller, M.: Synergistic face detection and pose estimation with energy-based models. Journal of Machine Learning Research **8**, 1197–1215 (2007)

40. Parkhurst, D., Law, K., Niebur, E.: Modeling the role of salience in the allocation of overt visual attention. Vision Research **42**(1), 107–123 (2002)
41. Perazzi, F., Krähenbühl, P., Pritch, Y., Hornung, A.: Saliency filters: Contrast based filtering for salient region detection. In: CVPR, pp. 733–740 (2012)
42. Pinheiro, P., Collobert, R.: Recurrent convolutional neural networks for scene parsing. In: ICML, pp. 82–90 (2014)
43. Shen, C., Mingli, S., Zhao, Q.: Learning high-level concepts by training a deep network on eye fixations. In: Deep Learning and Unsupervised Feature Learning NIPS Workshop (2012)
44. Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: CVPR, pp. 3476–3483 (2013)
45. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR **abs/1409.4842** (2014)
46. Tatler, B.: The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. Journal of Vision **7**(14) (2007)
47. Toet, A.: Computational versus psychophysical bottom-up image saliency: A comparative evaluation study. IEEE TPAMI **33**(11), 2131–2146 (2011)
48. Winnemoller, H., Kyprianidis, J., Olsen, S.: Xdog: An extended difference-of-gaussians compendium including advanced image stylization. Computers & Graphics **36**(6), 740–753 (2012)
49. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierachical saliency detection. In: CVPR (2013)
50. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.: Saliency detection via graph-based manifold ranking. In: CVPR (2013)