Neurocomputing 144 (2014) 569–580

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Exploiting global rarity, local contrast and central bias for salient region learning

Jinxia Zhang [a,b], Jundi Ding [a,*], Jingyu Yang [a]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolingwei St., Nanjing, Jiangsu 210094, China
[b] Visual Attention Lab, Brigham and Women's Hospital, 64 Sidney St. Suite. 170, Cambridge, MA 02139, United States

ARTICLE INFO

Article history:
Received 17 April 2013
Received in revised form
30 November 2013
Accepted 20 April 2014
Communicated by T. Heskes
Available online 9 May 2014

Keywords:
Global rarity
Local contrast
Central bias
Salient region
Learning

ABSTRACT

In this paper, we are to present a model that integrates and benefits from the global rarity, local contrast and central bias for saliency detection. Previous saliency works only consider one or two of them. Further, to avoid some inherent drawbacks of existing three factors, we first over-segment the image into many small coherent regions. And then, we exploit the self-information and regional de-noising, regional contrast and consistency, Gaussian function and regional averaging to get three new factors of global rarity, local contrast and central bias. Finally, we embed them into a nonlinear neural network to figure out their own contributions in saliency detection. Extensive experiments and comparisons illustrate the effectiveness of our saliency model with three new built factors.

1. Introduction

Visual saliency detection has become very popular in decades. Modeling visual saliency is very helpful for many tasks in computer vision, including image segmentation [1], image retrieval [2] and image mosaics [3]. Recently, a lot of models have been proposed to detect visual saliency. Among them, some models emphasize the global rarity to compute the saliency. The idea is that human eyes are not attracted by frequent features in an image but by the rare features. Mancas et al. [4] propose a visual attention model based on rarity and measure visual saliency using information theory. Hou and Zhang [5] extract the spectral residual of an image in frequency domain and convert the result in frequency domain to the saliency map in spatial domain using inverse Fourier transformation. The global rarity can well detect the salient objects of small sizes. However, some parts of the background which have rare features will be wrongly highlighted (see Fig. 1(B)). Furthermore, when the salient object is large, the detection result may be not acceptable.

Many other models consider the local contrast. The idea is that eyes are attracted by the features distinct from the surrounding features. For example, inspired by the cognitive discovery of the primate visual system, Itti et al. [6] propose the first computational model of visual saliency. This model computes the central-surround contrast using different multi-scale features and sums different feature maps with equal weights to get the final single saliency map. Harel et al. [7] propose a graph based model using the same feature maps as Itti. They compute the contrast between a point and other points to get the saliency map. Ma and Zhang et al. [8] estimate visual saliency based on local contrast analysis using a fuzzy growing method. Cheng et al. [9] propose a saliency extraction method based on regional contrast. The benefit of the local contrast is that it can detect the salient object even when the object is large. However, this method may wrongly highlight the background part close to the object (see Fig. 1(C)).

Besides, there are many models that state the central bias often exists in free scene viewing [10–12]. The possible reasons are as follows: First, the screen center may be the optimal location for early information processing. Second, the screen center may be the convenient location to start oculomotor exploration of the scene. Third, it may be that the central bias reflects a tendency to re-center the eye in its orbit. Fourth, when humans take a picture, they often frame the object of interest near or in the central part of the image. Hence, all the works [12–14] attempt to take a central bias for saliency detection. The central bias is helpful when the salient object is in or near the center of one image. However, it cannot well detect visual saliency when the object is off the center of the image.

To be sure, global rarity, local contrast and central bias each are with great advantages and important roles in visual saliency detection. At the same time, as pointed out above, they each have some disadvantages for saliency detection. So, only considering

* Corresponding author. Fax: +86 025 84315565.
E-mail address: dingjundi2010@njust.edu.cn (J. Ding).

http://dx.doi.org/10.1016/j.neucom.2014.04.003
0925-2312/

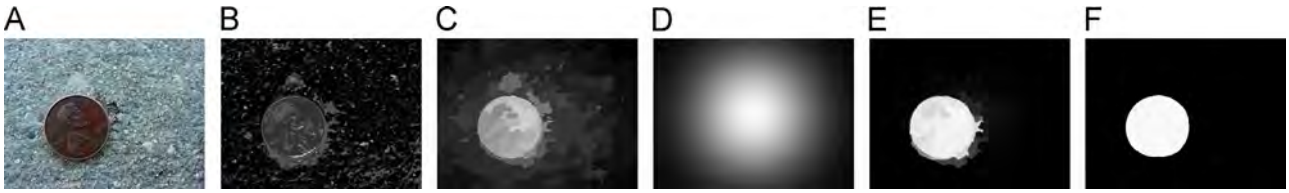

Fig. 1. Different saliency detection results. (A) The original image. (B), (C) and (D) The saliency results only considering global rarity, local contrast and central bias proposed by previous researchers respectively. (E) The result by combining our three new built factors. (F) The ground truth.

one factor is usually not ideal. Several models suggest integrating two factors. An example is in [15] where the features both globally rare and locally contrasted are considered to be salient. Here, we are to integrate all of such three factors. This is our first contribution in this paper. That is, we are to build a model that unifies global rarity, local contrast and central bias factors and benefits from the advantages of all of them for saliency detection.

At the same time, as pointed out above, they each have some disadvantages for saliency detection. So, we are not simply to put existing three factors together. To overcome such drawbacks, we first over-segment the image into many homogeneous regions. After that, we exploit the self-information of each pixel and the region de-noising operation to get a rarity map. For the local contrast factor, we first compute the regional contrast. Then we consider the regional consistency according to the idea that nearby regions with similar features are more likely to have the close saliency value to get a local contrast map. At last, as shown in Fig. 1(D), most existing central bias factors only consider the centrifugal distance. It brings a lot of evident false alarms. To avoid false alarms, we first use the Gaussian kernel function to model an offcenter map for each pixel, and then average the offcenter values of all pixels in each region to get the central bias. As our second contribution, they will be detailed below.

After getting these factors, how to combine them is another issue. Specifically, we think each factor has its own contribution. For an image, different factors may play roles with different powers for saliency detection. So, we do not simply use an average strategy to linearize their roles by assigning equal weights to them, just like most previous works on feature combination [6,7]. In a recent work [13], a linear, least square regression strategy is used to learn the weights of different features. However, it seems the performance is not very good for our salient region detection. One possible reason is that different factors should not be simply linearly combined. Thus, we here suggest a nonlinear learning strategy to combine these three factors. In particular, we take a neural network [16,17] to nonlinearly combine them, which is our third contribution in this paper. In [18], Zhao et al. use the nonlinear AdaBoost algorithm to combine different features. They are required to iteratively put up some weak classifiers to eventually get a strong classifier. Each weak classifier deals with a single feature. The results rely on the choice of these weak classifiers. In addition, the performance of AdaBoost may be defective when there are only a small number of features. In contrast, the neural network as a computational model just aiming to replicate the neural structure is relatively more flexible and adaptive. Even if there are a small number of features, a neural network can adjust with different stimuli and get a good performance.

One typical result of our model is shown in Fig. 1(E). Clearly, the salient object is more uniformly highlighted while the background is largely suppressed. It is greatly consistent with the ground truth (Fig. 1(F)). To further illustrate the effectiveness of our model, we first conduct the experiments on a commonly tested database of MSRA-1000 and compare it with 11 popular models. The results show that our model can better detect the salient objects in the image. We also conduct the eye fixation experiments on two well-known databases. And the experimental comparison with existing eight models shows that our model can well predict eye fixations.

The rest of this paper is organized as follows: Our proposed saliency region detection model is introduced in Section 2. Section 3 demonstrates that our model has a better performance to detect salient objects than other representative models and also can well predict eye fixations, and Section 4 concludes the paper.

2. Proposed salient region learning model

In the following, we introduce the Bayesian framework which we use to detect visual saliency. Let x denote a point in the image, which can be a pixel, a region or an object. The binary random variable S_x indicates visual saliency of the point x at location l_x with features f_x. Using Bayes' theorem, we can calculate the saliency value of a point:

$$SV_x = p(S_x = 1|F = f_x, L = l_x) \tag{1}$$

$$SV_x = \frac{p(F = f_x, L = l_x|S_x = 1)p(S_x = 1)}{p(F = f_x, L = l_x)} \tag{2}$$

For simplicity, we assume the features and the location of a point are independent and conditionally independent giving $S_x=1$ [19]. So $p(F = f_x, L = l_x) = p(F = f_x)p(L = l_x)$ and $p(F = f_x, L = l_x|S_x = 1) = p(F = f_x|S_x = 1)p(L = l_x|S_x = 1)$. Then the formula (2) can be written as

$$SV_x = \frac{p(F = f_x|S_x = 1)p(L = l_x|S_x = 1)p(S_x = 1)}{p(F = f_x)p(L = l_x)} \tag{3}$$

$$SV_x = \frac{1}{p(S_x = 1)}\frac{p(F = f_x|S_x = 1)p(S_x = 1)}{p(F = f_x)}\frac{p(L = l_x|S_x = 1)p(S_x = 1)}{p(L = l_x)} \tag{4}$$

$$SV_x = \frac{1}{p(S_x = 1)}p(S_x = 1|F = f_x)p(S_x = 1|L = l_x) \tag{5}$$

We assume that the prior probabilities of different points being salient are equal without the location and features. So the saliency value of a point is in direct proportion to the following formula:

$$SV_x \propto p(S_x = 1|F = f_x)p(S_x = 1|L = l_x) \tag{6}$$

The first term $p(S_x = 1|F = f_x)$ detects saliency due to features and the second term $p(S_x = 1|L = l_x)$ measures saliency based on the spatial location of a point. In this paper, we use the global rarity (marked as rarity) and local contrast (marked as distinctiveness) features of salient points to estimate $p(S_x = 1|F = f_x)$. And we model the central bias for $p(S_x = 1|L = l_x)$. In sum, three important factors, i.e., the rarity, distinctiveness and central bias, are used for our visual saliency detection. We call our model RDC in short. In contrast to averaging different factors, we follow a learning approach by training a neural network so that a more important factor will have a higher power. The details of three factors and the nonlinear combination method for salient region learning are elaborated in the following.

2.1. Rarity

We compute the rarity factor based on the basic features: CIELAB color. The color image is decomposed into three components: L for lightness and a and b for the color-opponent dimensions. These three components constitute three feature maps. The feature maps generated from the feature component L can be seen in Fig. 2(B). First, we compute the self-information for each pixel of the feature map to get the self-information map (Fig. 2(C)). To get rid of the false alarms in the background, we further de-noise the self-information map based on regions to get the rarity map, i.e., the conspicuity map in the rarity factor.

2.1.1. Getting the self-information map

In each feature map, we calculate the self-information for each pixel using the definition of self-information [12,20,21,4]. First, we use a histogram of 100 bins to represent the distribution of feature values in the feature map. Let $N(i)$ denote the number of the pixels in the i-th bin. Then we use the following formula to compute the frequency for the feature value of each bin:

$$p(i) = \frac{N(i)+1}{\sum_{i=1}^{100}(N(i)+1)} \tag{7}$$

We add 1 to the number of the feature value of each bin to avoid the multiplicative inverse of the frequency being infinite. Then we compute the self-information for the feature value of each bin:

$$H(i) = \log\frac{1}{p(i)} \tag{8}$$

After that, we use the nearest neighbor interpolation method to compute the self-information for the feature value of each pixel in the feature map.

Different self-information maps got from corresponding feature maps can be seen in Fig. 2(C). We observe that in the self-information map the values of the object are generally larger than those of the background. But there are also a lot of noise in the background which should not be highlighted. To solve this problem, we further de-noise the self-information map to get the final rarity map.

2.1.2. De-noising the self-information map

We over-segment the image into a number of regions using the mean-shift method [22] (see Fig. 2(A)) and de-noise the self-information map based on the regions to get the rarity map.

To reduce the noise, we first smooth the self-information map using Gaussian convolution. Because pixels in the same region have the similar or even the same feature values, we think these pixels are likely to have the same rarity value. So we compute the average self-information of all pixels in one region as the rarity value of this region. And the pixels in one region share the same rarity value.

The rarity maps are shown in Fig. 2(D). Compared with the self-information maps in Fig. 2(C), it can be seen that the object becomes brighter and the background becomes darker, which demonstrate that the de-noising operation is effective.

Note, to facilitate the visualization of different maps and the weighting of different factors in the neural network, we normalize the value of each point in different maps generated in our global

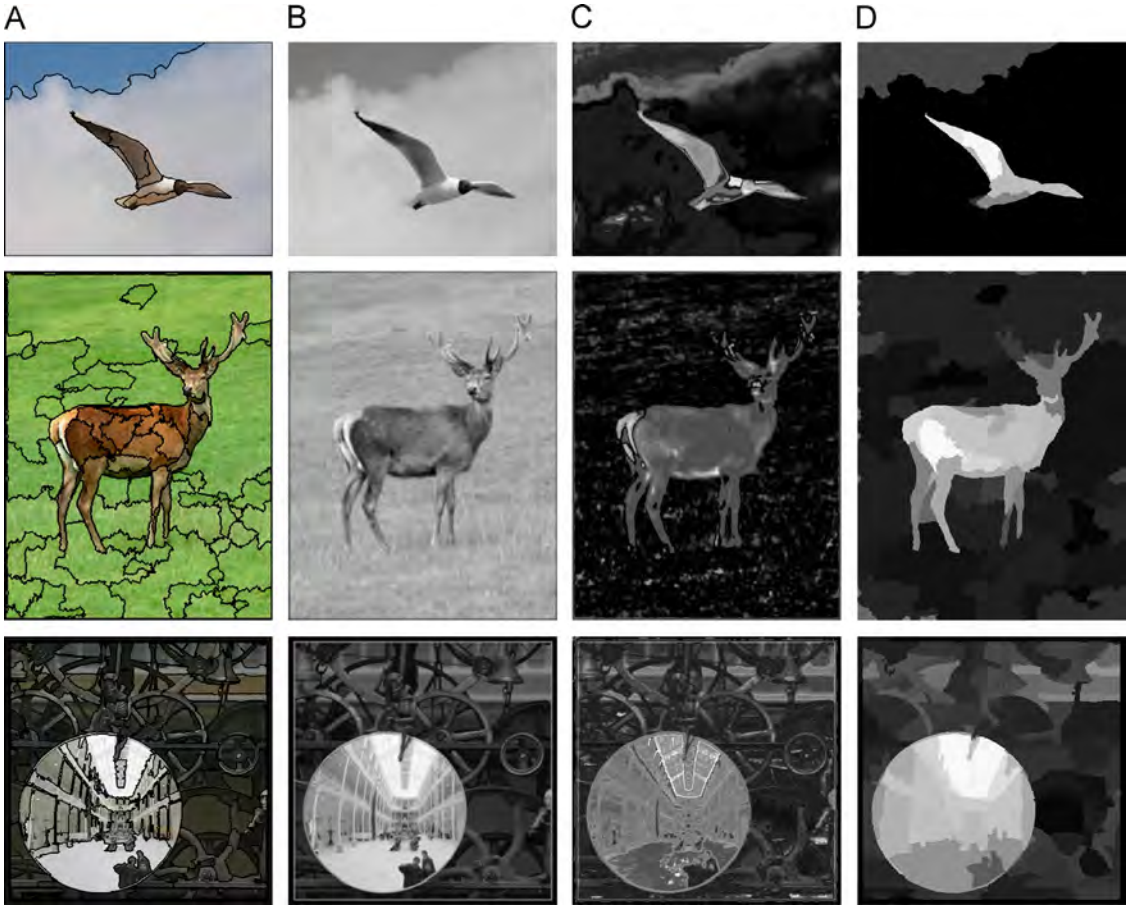

Fig. 2. Different maps generated in the rarity factor: (A) the over-segmented images, (B) the feature maps generated from the feature channel L, (C) the self-information maps and (D) the rarity maps.

rarity, local contrast and central bias factors to be in the range [0, 1].

2.2. Distinctiveness

The methods in [9,7] also compute local contrast for saliency detection. However, they only consider the local contrast factor while our model integrates and benefits from global rarity, local contrast and central bias factors. Secondly, the methods in [9,7] only compute the regional contrast based on the feature dissimilarity and distance coherence. The drawback of these methods is that a part of the background may be wrongly highlighted, seen in Fig. 4(D). So in our paper, to overcome this drawback, we propose a refinement operation in the local contrast factor according to the regional consistency that if the regions are more similar in feature and closer in space, these regions will have the close saliency values. The visual comparison in Fig. 4 and quantitative comparison in Fig. 12 show that the refinement operation is useful. Furthermore, unlike the work [9], the method to compute feature dissimilarity in our paper is easy, efficient and has the good extension ability. In the work [9], the authors first quantize colors in the RGB color space and then measure color differences in the Lab color space to compute feature dissimilarity, which is hard to generalize to other feature spaces. While in our paper, we first define the feature value of a region as the average feature value of all pixels in this region and then simply define feature dissimilarity of two regions as the difference between the feature values of two regions. This method is easy to compute, highly improves the computation efficiency and easily generalizes to other features.

The summary of our method to compute the distinctiveness factor is as follows. Given a color image I, we first over-segment the image into a number of regions using the mean-shift method (see Fig. 4(A)). Let K denote the number of the regions. We use three image components: L, a and b as feature maps. The feature maps generated from the component L can be seen in Fig. 4(B). In each feature map, we define the activation value of a region as the sum of its contrasts to all the other regions. To avoid some part of the background near the salient object being wrongly highlighted, we further refine each activation map based on the similarity and distance of regions to get the distinctiveness map, i.e., the conspicuity map in the distinctiveness factor.

2.2.1. Activating the feature map

In this step, given a feature map $F: m*n \to \mathcal{R}$, our goal is to compute an activation map $A: m*n \to \mathcal{R}$. The region which is different from the surrounding regions will have high value in the activation map A.

In each feature map, we compute the local contrast of regions to get the activation map. The contrast of a region r_i compared to the region r_j can be defined as follows:

$$W(r_i \leftarrow r_j) = D_F(r_i, r_j) * D_S(r_i, r_j) * f(r_j) \tag{9}$$

The first term $D_F(r_i, r_j)$ indicates the feature dissimilarity of two regions. If region r_i is different from other regions, it is more likely to be distinct.

Because pixels of the same region have the similar or even the same feature values in each feature map, we define the feature value of a region as the average feature value of all pixels in this region. The generated average maps can be seen in Fig. 4(C).

Then $D_F(r_i, r_j)$ is simply defined as the difference between the feature values of two regions. This can highly improve the computation efficiency and be easily generalized to other features:

$$D_F(r_i, r_j) = |F(r_i) - F(r_j)| \tag{10}$$

The second term $D_S(r_i, r_j)$ indicates the spatial relationship of two regions and has the following computational formula. This term increases the effects of closer regions:

$$D_S(r_i, r_j) = \exp\left(-\frac{\left(\frac{m(r_i) - m(r_j)}{M}\right)^2 + \left(\frac{n(r_i) - n(r_j)}{N}\right)^2}{2\sigma^2} \right) \tag{11}$$

Among the formulas, $(m(r_i), n(r_i))$ denotes the position of the region r_i and is defined as the centriod of the region. M and N are respectively the number of rows and columns of the image separately. And σ is the scale parameter which controls the power of spatial weighting. σ with smaller values would weaken the effect of spatial weighting so that far regions would make little contribution to the saliency detection. But if σ is too small, some part of the salient object may be wrongly suppressed (see Fig. 3(B)). σ with larger values would strengthen the influence of spatial weighting. So far regions would make a greater contribution. However, if σ is too large, some part of the background may be wrongly highlighted (see Fig. 3(D)). The results are satisfactory when σ is set from 0.15 to 0.25. In our implementation, we use $\sigma = 0.2$ for the salient object detection and the eye fixation prediction.

The third term $f(r_j)$ represents the relative size of the compared region. The bigger regions have greater impact on the region r_i. Let

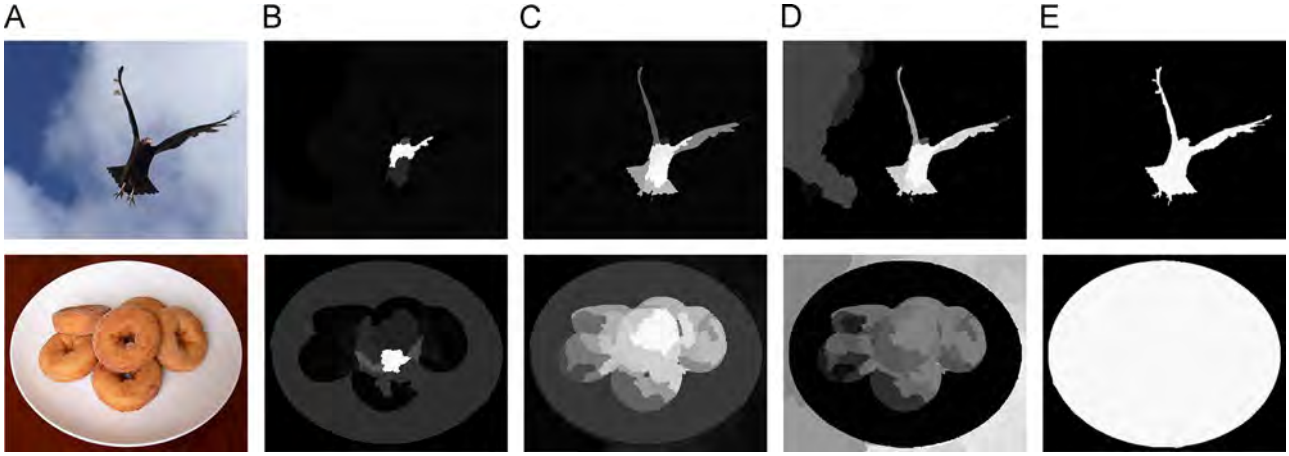

Fig. 3. Distinctiveness maps of feature channel L using different σ: (A) the original images, (B) the distinctiveness maps generated using $\sigma = 0.05$, (C) using $\sigma = 0.2$, (D) using $\sigma = 0.5$ and (E) the ground truths.

$S(\cdot)$ denote the size of a region. So the formula of $f(r_j)$ is

$$f(r_j)=\frac{S(r_j)}{S(I)} \tag{12}$$

As is shown in the formula (9), the contrast of a region r_i compared to the region r_j is in direct proportion to their dissimilarity, their closeness and the relative size of the compared region. That is, a nearby and dissimilar compared region of big size will have a greater influence on the region r_i than a far and similar compared region of small size.

The activation value of a region is defined as the sum of the contrasts compared to all the other regions:

$$A(r_i)=\sum_{j=1}^{K} W(r_i \leftarrow r_j) \tag{13}$$

Then the region distinct from the surrounding regions will have high value in the activation map (see Fig. 4(D)). Through the observation, we find that some regions of the background near the object may be wrongly highlighted. To solve this problem, we refine the activation map according to the regional consistency that nearby regions with similar features are more likely to have the close saliency values to get a local contrast map.

2.2.2. Refining the activation map

First, we introduce the definition of similar regions. If the feature dissimilarity of two regions is less than the average feature dissimilarity of all the regions, we define these two regions as the similar region (SR) of each other:

If $D_F(r_i,r_j)<D_{ave}$, then $r_i \in SR_j$ and $r_j \in SR_i$, where

$$D_{ave}=\frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K} D_F(r_i,r_j) \tag{14}$$

Then the distinctiveness value of a region r_i is defined as the weighted sum of activation values of its similar regions:

$$D(r_i)=\frac{1}{|SR_i|}\sum_{r_s \in SR_i}\frac{D_{ave}-D_F(r_i,r_s)}{D_{ave}}*A(r_s) \tag{15}$$

Because the feature dissimilarity between a region and itself is zero, the weight $(D_{ave}-D_F(r_i,r_s))/D_{ave}$ equals 1 for the region r_i itself. The more dissimilar the feature value of a region r_s and the feature value of the region r_i is, the smaller the weight is for the region r_s. When the feature dissimilarity between a region r_s and the region r_i equals the average feature dissimilarity of all the regions D_{ave}, the weight for the region r_s equals zero. $|SR_i|$ represents the number of similar regions of the region r_i.

The drawback of the refinement operation in formula (15) is the following: if the region r_i in foreground and a distant region r_s in background have similar feature values, the weight for the region r_s is wrongly set to be large. We further incorporate spatial information into the refinement operation so that the closer regions have greater impact on the region r_i than the farther regions:

$$D(r_i)=\frac{1}{|SR_i|}\sum_{r_s \in SR_i} D_S(r_i,r_j)*\frac{D_{ave}-D_F(r_i,r_s)}{D_{ave}}*A(r_s) \tag{16}$$

The spatial information $D_S(r_i,r_j)$ has been introduced in formula (11). The weight in formula (16) indicates that close and similar regions have close saliency values. The distinctiveness maps can be seen in Fig. 4(E). Compared with the activation maps in Fig. 4(D), we find that the regions near the object which are wrongly highlighted in activation maps are blackened. Besides, the object becomes brighter and the background becomes darker in the distinctiveness maps, which show that the refinement operation is effective.

2.3. Central bias

Judd et al. [12] use the Euclidean distance between each pixel and the center of the image to model the central bias factor. Zhao et al. [13] use a 2D Gaussian distribution to model the time-dependent and time-independent central biases. They all model the central bias as a circle, which cannot provide any information of salient objects and bring a lot of evident false alarms.

In this paper, we first use the Gaussian kernel function to model offcenter map as an eclipse which is adjusted to the length–width ratio of the image. To get rid of evident false alarms, we further average the offcenter map based on regions to get the final central bias map, i.e., the conspicuity map in the central bias factor.

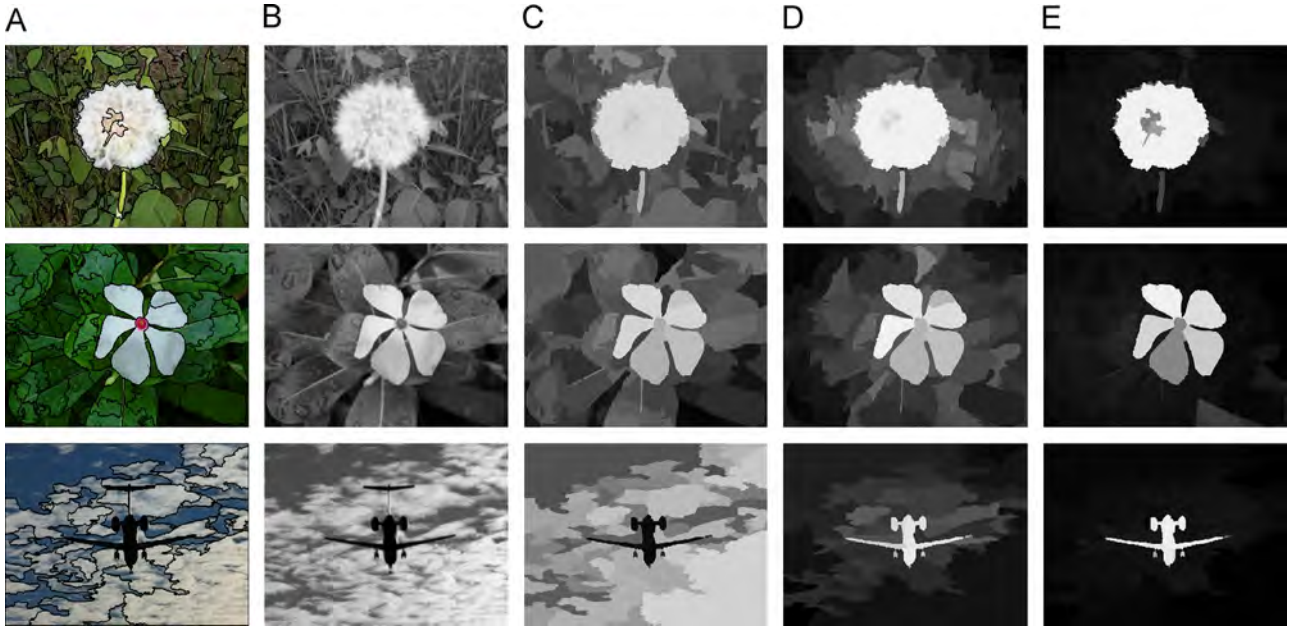

Fig. 4. Different maps in the distinctiveness factor: (A) the over-segmented images, (B) the feature maps generated from the feature channel L, (C) the average maps, (D) the activation maps and (E) the distinctiveness maps.

2.3.1. Computing offcenter map

In this paper, we use the Gaussian kernel function to model the offcenter map as an eclipse. Then the offcenter map is a real-valued function whose value depends only on the distance between a point and the center point. The central point has the largest central bias value which is 1. And the farther the point is away from the central point, the smaller the value is. Let M and N denote respectively the number of rows and columns in the image separately. (m,n) denotes the position of a random point and (m_c, n_c) denotes the position of the center point:

$$O(m,n) = \exp\left(-\frac{\left(\frac{m-m_c}{M}\right)^2 + \left(\frac{n-n_c}{N}\right)^2}{2\sigma^2}\right) \tag{17}$$

The resulting offcenter map is adjusted to the length–width ratio of the image. We also compare the offcenter map modeled as an eclipse with the offcenter map modeled as a circle, which is not adjusted to the length–width ratio of the image:

$$O(m,n) = \exp\left(-\frac{\left(\frac{m-m_c}{\frac{1}{2}(M+N)}\right)^2 + \left(\frac{n-n_c}{\frac{1}{2}(M+N)}\right)^2}{2\sigma^2}\right) \tag{18}$$

These two types of offcenter map model both the spatial similarity between a point and the center point. σ is the scale parameter and controls the influence of central bias. Smaller values of σ would reduce the effect of central bias so that far points would have a smaller central bias value. However, if σ is too small, salient objects would be wrongly suppressed (see Fig. 5B). Larger values of σ would strengthen the influence of central bias. But if σ is too large, background is tend to be highlighted (see Fig. 5D). In our implementation, we use $\sigma = 0.25$ for both types of offcenter maps on the salient object detection database [23] and the eye fixation prediction database [12,24]. By quantitatively comparing these two types of offcenter maps, it indicates that modeling the central bias as an ellipse is a little better than modeling central bias as a circle, shown in Fig. 12. However, both of these two offcenter maps only provide centrifugal distance, do not include any information of salient objects and bring evident false alarms. So we further average each offcenter map based on regions to get the final central bias map.

2.3.2. Average offcenter map

We first over-segment the image into a number of regions using the mean-shift method, as the segmentation operation in the rarity and distinctiveness factors (see Fig. 6(A)). Then we average the offcenter values of all pixels in one region as the center bias value of this region. And the pixels in one region share the same center bias value.

Compared with the offcenter maps (see Fig. 6(B)), it can be seen that the center bias maps (see Fig. 6(C)) can give more information of salient objects and avoid a lot of false alarms. The quantitative comparison of the offcenter map and the final central bias map can be seen in Fig. 12. From this figure, we can see that central bias map is better than the offcenter map which shows the effectiveness of the average operation.

2.4. Learning nonlinear combination

In this paper, we use a neural network of three layers: one input layer, one hidden layer and one output layer. The illustration of the nonlinear combination using the neural network is shown in Fig. 7.

The number of the nodes in the input layer corresponds to the number of features of the training sample. These features $(x_1, \ldots, x_a, \ldots, x_A)$ are the conspicuity values of the training samples in different conspicuity maps got from three factors. In this paper, we get three conspicuity maps in both the rarity and the distinctiveness factors. Also, we get one conspicuity map in central bias factor. So, A equals 7 in this paper.

Each node h_b in the hidden layer is a combination of all nodes in the input layer:

$$h_b = g(\Theta_{0b} + \Theta_{1b}x_1 + \cdots + \Theta_{ab}x_a + \cdots + \Theta_{Ab}x_A) \tag{19}$$

In this formula, Θ_{ab} is the weight used to map from the input layer to the hidden layer and the sigmoid function $g(z) = 1/(1+\exp^{(-z)})$ [16] is used as the mapping function.

The number of the nodes $(h_1, \ldots, h_b, \ldots, h_B)$ in the hidden layer indicates the complexity of the neural network. If the number of the nodes in the hidden layer is small, the neural network has fewer parameters and more prone to under-fitting. Meanwhile if the number of the nodes in the hidden layer is large, the neural network has more parameters and more prone to over-fitting. In this paper, we set this number to 30. To avoid the possible problem of over-fitting, we choose a large number of training samples. Actually, the number of training samples can be very large because in the database there are hundreds of images and each image contains thousands of points.

The number of the nodes in the output layer is 1. This node outputs a value between 0 and 1, indicating the saliency value of a point. And this node is a combination of all nodes in the hidden layer shown in the following formula. In this formula, Θ_b is the weight used to map from the hidden layer to

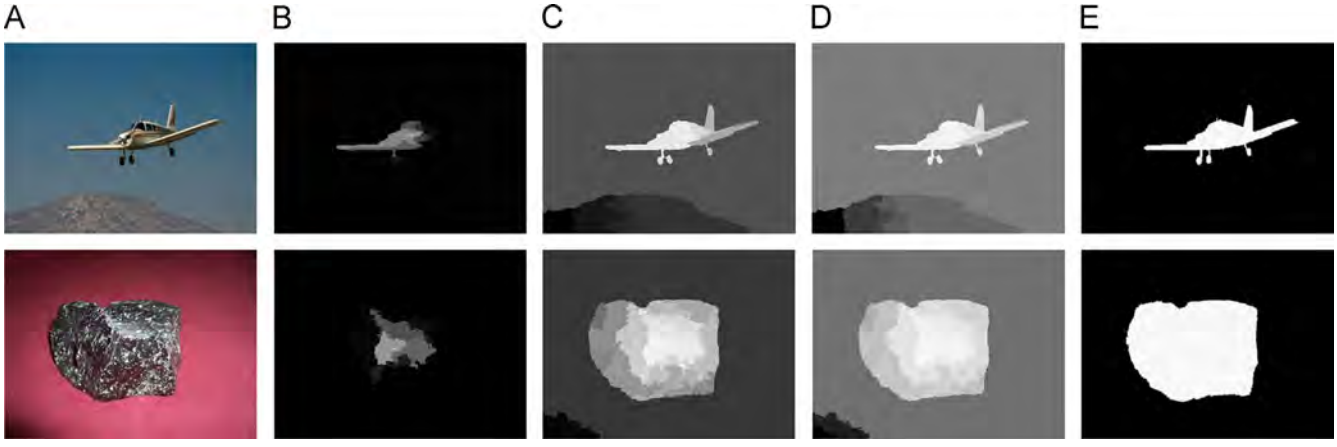

Fig. 5. Offcenter maps of feature channel L using different σ: (A) the original images, (B) the offcenter maps generated using $\sigma = 0.05$, (C) using $\sigma = 0.25$, (D) using $\sigma = 0.5$ and (E) the ground truths.

Fig. 6. Different maps in the central bias factor: (A) the over-segmented images, (B) the offcenter maps and (C) the central bias maps.

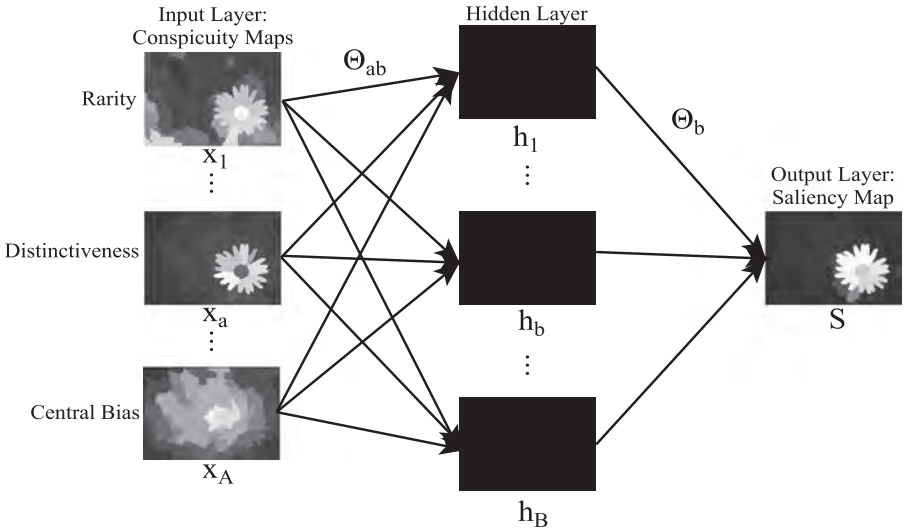

Fig. 7. A neural network used to learning a nonlinear combination.

the output layer:

$$s = g(\Theta_0 + \Theta_1 h_1 + \cdots + \Theta_b h_b + \cdots + \Theta_B h_B) \tag{20}$$

Let $\{(x^{(j)}, y^{(j)})\}$ $(j = 1 : J)$ denote the training samples. $x^{(j)}$ records the features of training samples. $y^{(j)}$ equals 1 for positive samples and $y^{(j)}$ equals 0 for negative samples. And let $s_\Theta(x^{(j)})$ denote the output of saliency value giving the specific features $x^{(j)}$ and the specific weights Θ. The algorithm details to learn the optimal weights $\Theta = [\Theta_{ab}; \Theta_b]$ of the neural network which are given in Algorithm 1.

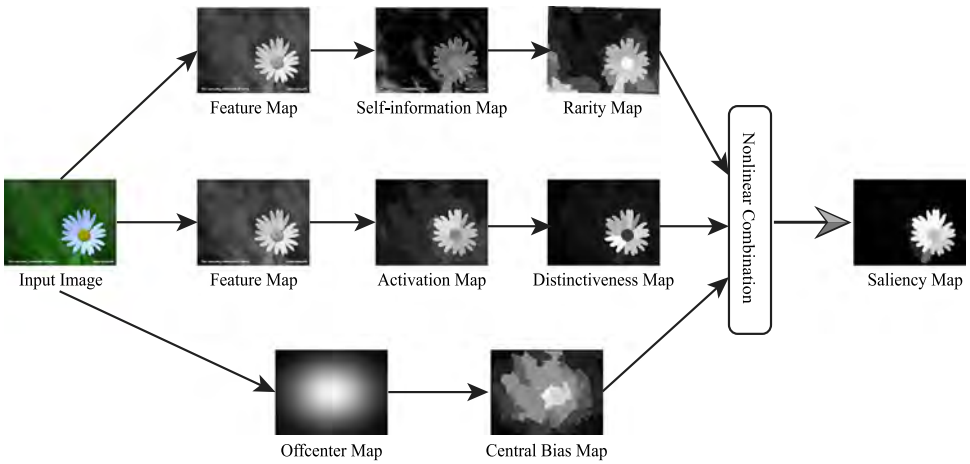

Fig. 8. Process of our visual salient region learning model.

Our model used for salient region learning is illustrated in Fig. 8. The input image is analyzed in three parallel pathways according to three factors: rarity, distinctiveness and central bias. Finally, the conspicuity maps got in these factors are nonlinearly combined into the saliency map using a neural network.

Algorithm 1. Learning a nonlinear combination using a neural network.

Input:
Training images and corresponding ground truths. See Section 3 for the introduction of ground truths on different databases.
A testing image I_t.
Output:
Saliency map of I_t.
Training stage:
1. From training images and corresponding ground truths, get training samples $\{(x^{(j)}, y^{(j)})\}$ $(j = 1 : J)$.
2. Initialize each weight in $\Theta = [\Theta_{ab}; \Theta_b]$ to a random small value in $[-\epsilon, \epsilon]$.
3. Use the backpropagation algorithm to learn the optimal weights which minimizes the cost function $\Theta = \arg\min_\Theta O(\Theta)$, where
$$O(\Theta) = -\frac{1}{J}\sum_{j=1}^{J}[y^{(j)}\log(s_\Theta(x^{(j)})) + (1-y^{(j)})\log(1-(s_\Theta(x^{(j)})))].$$
4. After getting the optimal weights, the final saliency value of a point is computed using the formulas (19) and (20).
Testing stage (for a new image I_t):
For each point in I_t, compute the conspicuity values in different modules, then apply the neural network to obtain the saliency value of the point.

3. Experimental comparisons

We have evaluated our visual salient region learning model in two practical applications, i.e., salient object detection and eye fixation prediction. For salient object detection, we compare different representative models about their performance on the MSRA-1000 database provided by Achanta et al. [23]. Then we compare different models about the performance of eye fixation prediction on the MIT database provided by Judd et al. [12] and the TORONTO database provided by Bruce et al. [24].

To evaluate our model, we use a 2-fold cross-validation approach. The whole database is randomly divided into two parts. We first use the first part as the training set and test on the second part, followed by using the second part as the training set and testing on the first part. Results are then averaged over these two partitions. The advantage of 2-fold cross-validation approach is that each image in the database is used for both training and testing.

3.1. Salient object detection

The MSRA-1000 database contains 1000 images and includes the accurate human-marked object-contour ground truths. To train and test our model, the database is randomly divided into two parts and each has 500 images. In the training stage, positive samples and negative samples are respectively taken from the object and the background.

We compare our model with 11 representative saliency detection models: AC [23], CA[25], FT [26], GB [7], HC [9], IT [6], LC [27], MZ [8], RARE [15], RC [9] and SR [5].

The visual comparison of different models is shown in Fig. 9. It can be seen that the salient points of the models AC, CA, FT, GB, IT, LC, MZ, RARE and SR often locate near or on the edges of the object, namely the salient object is not uniformly highlighted. The saliency maps generated by HC and RC tend to highlight the whole object, but the problem is that a portion of the background is wrongly highlighted. Saliency maps generated by our RDC are more consistent with the ground truths: they highlight the object, commendably define the object borders and well suppress the background.

We also quantitatively evaluate the detection performance of different models. First we convert the type of saliency values into uint8. So the saliency values will be in the range [0,255]. Then we binarize the saliency map using the fixed threshold varying from 0 to 255 and compute the precision and recall at each threshold by comparing the result with the ground truth. With the precision and recall at each threshold, we can get a curve recording different precisions and recalls at different thresholds for each model (see Fig. 10). It provides quantitative comparison of different saliency detection models and indicates how well they highlight salient regions in an image. From the figure, it can be seen that our RDC has outperformed all other models. When the fixed threshold is 0, each model gets its own maximum recall. At that time, all points in the image are labeled to be foreground and all models have equal values for precision and recall. When the fixed threshold is 255, only the points with a value of 255 are labeled to be foreground and each model gets its own maximum precision. We can see that the maximum precision value of RDC is the largest. And for each value of recall, the precision of RDC is also higher than that of

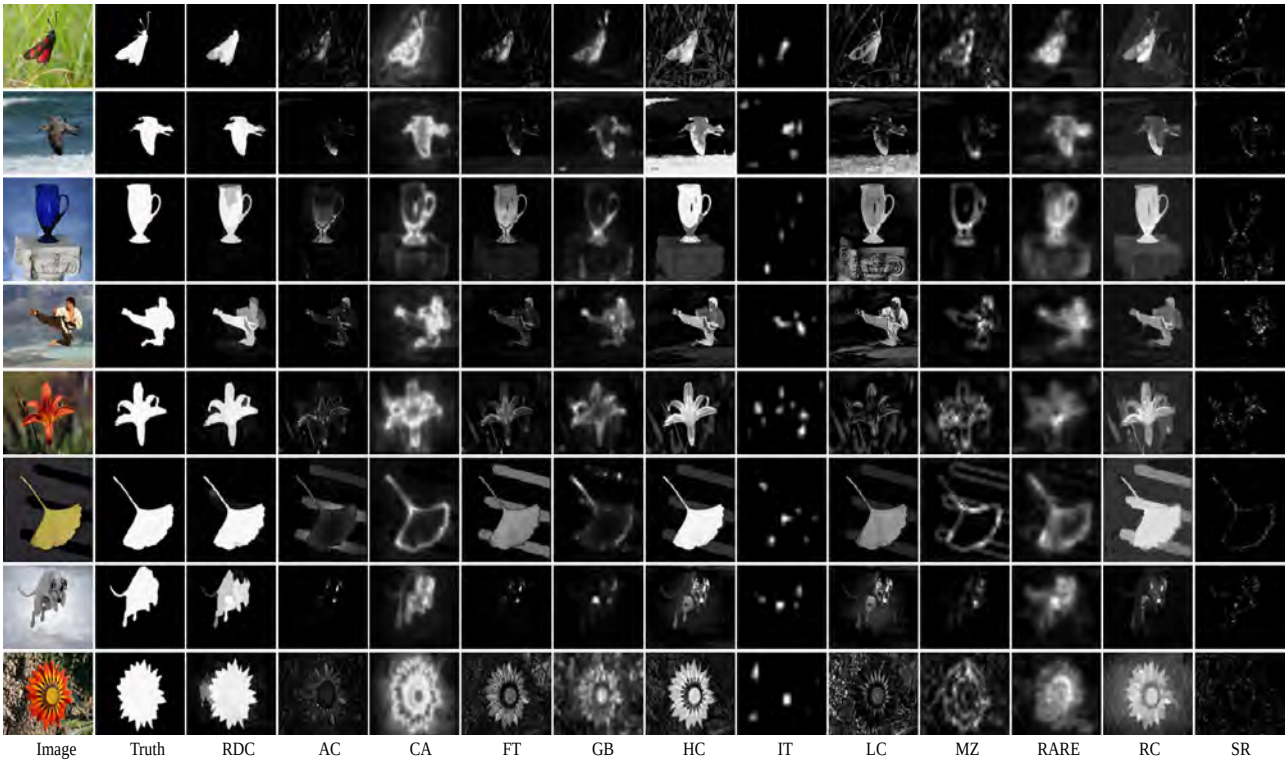

Fig. 9. Visual comparison of different saliency detection models.

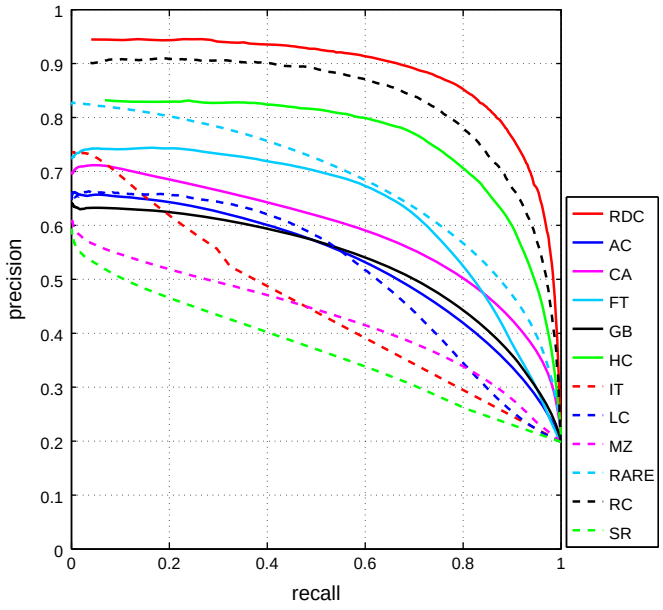

Fig. 10. PR curves of different models on MSRA-1000 database.

other models which indicates that RDC highlights the salient object more and well suppresses the background.

Because our RDC includes central bias factor, it is interesting to see how our model works when the salient objects are not in the center of the images. The related images and our detection results are shown in Fig. 11. In row 1, we list the original images. It can be seen that the salient object is on the left bottom in the first image and salient objects are very close to the edges in other images. In row 2 and row 3, we list the detection results of our RDC and the ground truth respectively. We can see that our model can well highlight the whole salient objects and suppress background. The reason is that even if the salient object is not in the center of the image, the global rarity and local contrast factors would make contributions for the saliency detection. This, to a certain extent, shows that our built factors complement each other nicely to help our unified model to detect visual saliency.

To understand our model and the roles of three factors, rarity, distinctiveness and central bias, we also quantitatively compare the performance of different maps generated in these factors, shown in Fig. 12. From this figure, it can be seen that the final saliency map has the best performance. The maps generated in the distinctiveness factor have better performance than those in rarity and central bias factors. The rarity map and the central bias map have similar performance. Moreover, we can see that the rarity map is much better than the self-information map which shows that the de-noising operation in the rarity factor is useful. The distinctiveness map outperforms the activation map which indicates that the refinement operation in the distinctiveness factor is effective. Offcenter map modeled as an ellipse is a little better than that modeled as a circle. And central bias map is much better than both types of offcenter maps, which proves that average operation in the central bias factor is beneficial.

To further understand our model, we also experiment and compare saliency detection results using only one factor, two factors and all three factors, shown in Fig. 13. For the sake of simplicity, *R* denotes rarity factor, *D* denotes distinctiveness factor and *C* denotes central bias factor. We can see that RDC has the best performance compared with all others, which indicates that every factor makes contribution in our model. More specifically, comparing RDC with DC, we can see that the precision of RDC is higher than DC when the recall is large. When comparing RDC with RD, it can be seen that at each recall the precision of RDC is higher than RD. Also when comparing RDC with RC, we can see that the precision of RDC is higher than RC at each recall. So *R*, *D* and *C* all help the model to highlight the salient object and suppress background. The same conclusions can be found when comparing two factors with one factor. In addition, if comparing between two

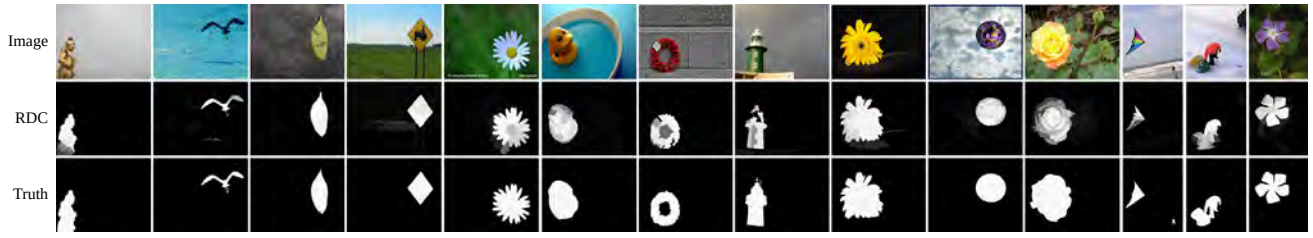

Fig. 11. Our detection results when the salient objects are not in the center of the images.

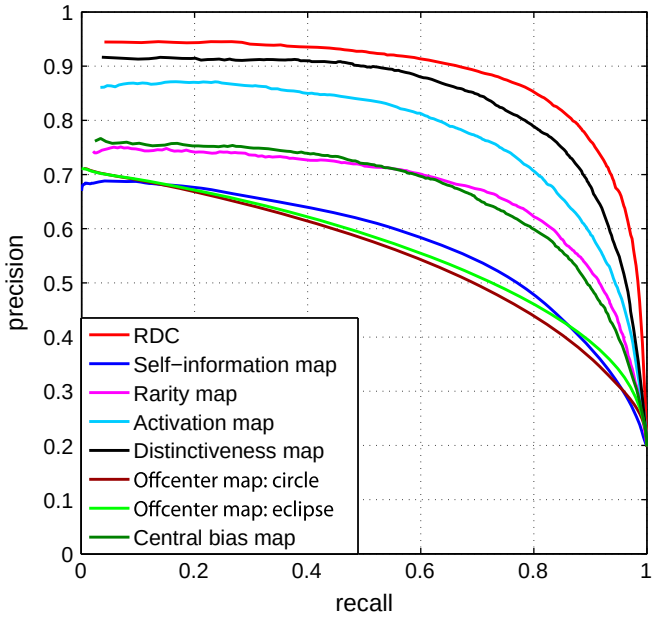

Fig. 12. The quantitative comparison of different maps generated in three factors of our model.

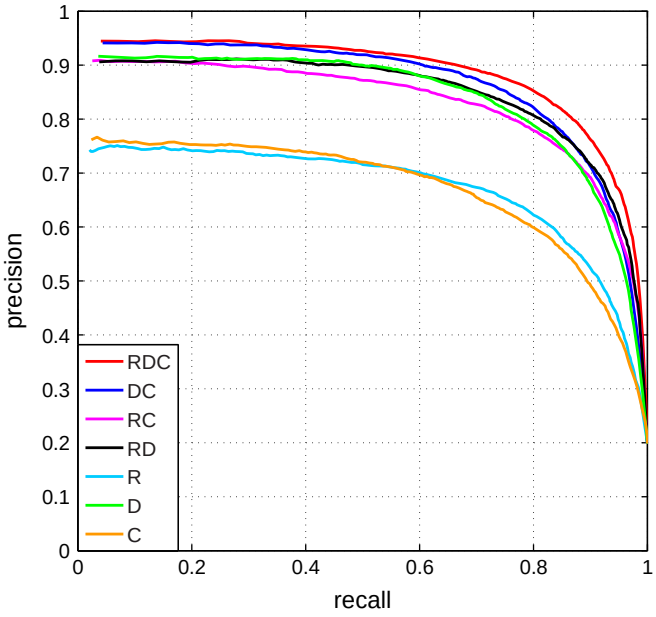

Fig. 13. The quantitative comparison of results using one factor, two factors and all three factors.

factors, we can see that DC has the best performance and RC has the worst performance. If only considering one factor, it can be seen that *D* factor outperforms *R* and *C* factors. *R* and *C* factors have similar performances.

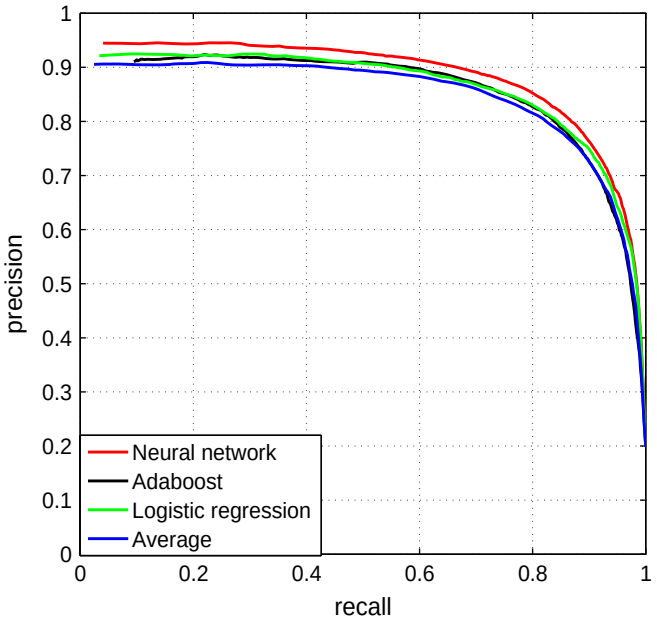

Fig. 14. The quantitative comparison of different combination strategies.

We also compare our neural network combination strategy with other strategies: average strategy [6,7], least square regression strategy [13] and AdaBoost strategy [18], shown in Fig. 14. Average and least square regression are linear combination strategies. AdaBoost and our neural network are nonlinear combination strategies. To be fair, different strategies share the same features which are rarity, distinctiveness and central bias. Average strategy computes the final saliency result by assigning the same weight to different features. Logistic regression strategy minimizes the squared error function to get corresponding weight for each feature. AdaBoost strategy learns a final strong classifier which is a weighted combination of weak classifiers. Fig. 14 shows that our neural network combination strategy outperforms other strategies. The average strategy has the worst performance. The least square regression and AdaBoost strategies have similar performances.

3.2. *Eye fixation prediction*

The MIT and TORONTO databases both include natural outdoor and indoor images. And the ground truths are eye fixation maps which record fixations from different observers who freely viewed corresponding images. To train our model, positive samples are taken from the top 5/100 salient pixels of the eye fixation map and negative samples are taken from the bottom 80/100. The MIT database has 1003 color images. For 2-fold cross-validation, the database is randomly divided into two parts. One part has 501 images and another part has 502 images. The TORONTO database has 120 color images. For 2-fold cross-validation, the database is randomly divided into two parts and each has 60 images.

Fig. 15. The examples of eye fixation maps. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

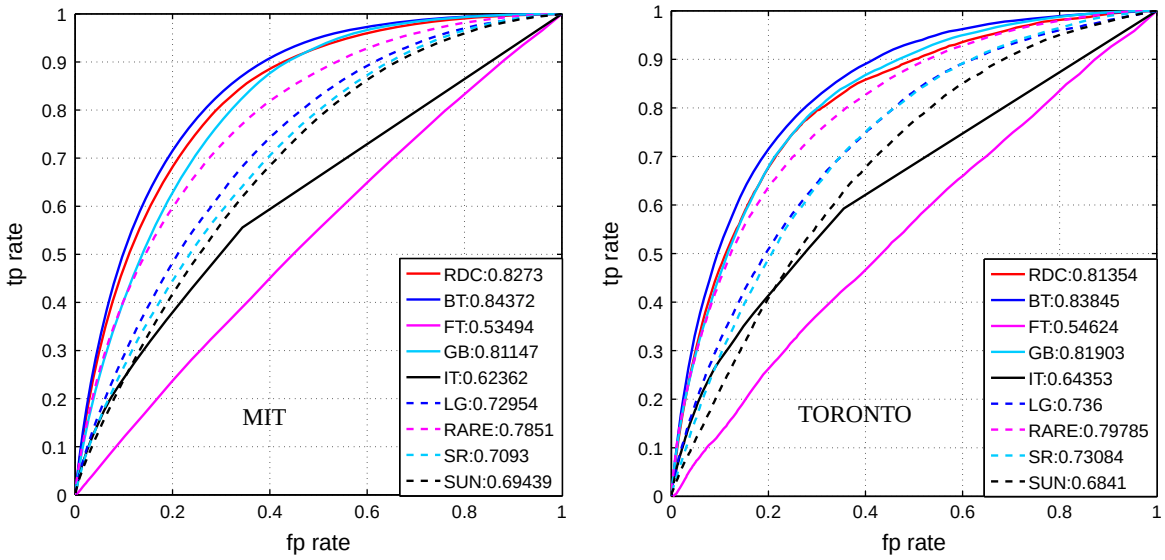

Fig. 16. ROC curves of different models on MIT database and TORONTO database.

Fig. 15 shows the examples of eye fixation maps. The red points in the images record the eye fixations of humans. From the maps, it can be seen that most of the eye fixations locate inside the object and do not cover the whole object. So we use the most salient part of a map to better predict the eye fixations. We first compute the new value of each point in the rarity, distinctiveness and central bias maps as the original value to the power of one point five. So the most salient locations stand out and the sub-salient locations are suppressed. Then we smooth these maps using Gaussian convolution to get the new rarity, distinctiveness and central bias maps. Finally we nonlinearly combine the new rarity, distinctiveness and central bias maps to get the saliency map to predict the eye fixations. As a result, the final saliency map highlights the most salient parts instead of highlighting the whole object. Meanwhile, the final saliency map is in low-resolution and the saliency values in the map change smoothly which are more consistent with fixation density because the dense of the eye fixations also changes smoothly.

Because the number of eye fixations is very small compared with the total number of the pixels in the image, there exists the problem of skewed class distribution. ROC curves are insensitive to class skews [28], so we use ROC curves to quantitatively evaluate the performance of different models to predict eye fixations. The process to get the ROC curve of each model is as follows: we first convert the type of the saliency values into uint8. Then we choose the fixed threshold varying from 0 to 255 to binarize the saliency map and compute the fp (false positive) rate and tp (true positive) rate at each threshold by comparing the result with the eye fixation map. After that, we get a curve recording different fp rates and tp rates at different thresholds for each model. To compare the performance of different ROC curves, a common method is to compute the area under each ROC curve (AUC) [29,30]. In general, the higher the value of AUC is, the better the corresponding ROC curve is.

We compare our model with eight other state-of-art models: BT [19], FT [26], GB [7], IT [6], LG [31], RARE [15], SR [5] and SUN [32].

Fig. 16 shows ROC curves of different models and the respective AUCs in MIT and TORONTO databases. From this figure, it can be seen that our RDC outperforms most of the models which indicate that RDC can well predict human fixations. When the fixed threshold is 0, the tp rate and the fp rate of all models are 1. When the fixed threshold is 255, the tp rate and the fp rate of all models are around 0. And for each value of fp rate, the tp rate of RDC is higher than most models. According to the AUC, the performance of RDC is better than most of the models except BT in both MIT and TORONTO databases. The reason why BT has the best performance is that it includes high-level features, e.g. face, person and car, besides low-level features while other models only compute low-level features. In this paper, we only consider the global rarity and local contrast features for the feature term of formula (6) in the Bayesian framework introduced in Section 2. However, more useful features including high-level features can be included into this framework, which we are planning to work on.

4. Conclusion

We have proposed a salient region learning model in this paper. The model includes three important factors for visual saliency detection: global rarity, local contrast and central bias. To improve

the performance, we propose new methods to compute these factors. Finally we learn a neural network to nonlinearly combine different factors to get the saliency map. The experimental comparisons demonstrate that our model outperforms other representative models to detect salient objects and also well predicts eye fixations. In the future, we plan to do research on effective models which incorporate high-level concepts and contextual features for visual saliency in target searching situation.

Acknowledgment

The research was supported by the National Natural Science Fund of China (Grant nos. 61103058, 90820306).

References

[1] J. Li, R. Ma, J. Ding, Saliency-seeded region merging: automatic object segmentation, in: Asian Conference on Pattern Recognition, 2011.
[2] S. Wan, P. Jin, Y. Lihua, An approach for image retrieval based on visual saliency, in: International Conference on Image Analysis and Signal Processing, 2009.
[3] D. Guo, J. Tang, J. Ding, C. Zhao, Saliency-based content-aware image mosaics, Adv. Multimedia Modeling 7732 (2013) 436–444.
[4] M. Mancas, C. Mancas-Thillou, B. Gosselin, B. Macq, A rarity-based visual attention map -application to texture description, in: International Conference on Image Processing, 2006.
[5] X. Hou, L. Zhang, Saliency detection: a spectral residual approach, in: International Conference on Computer Vision and Pattern Recognition, 2007.
[6] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.
[7] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, Neural Inf. Process. Syst. (2006) 545–552.
[8] Y.-F. Ma, H.-J. Zhang, Contrast-based image attention analysis by using fuzzy growing, ACM Multimedia (2003) 374–381.
[9] M.-M. Cheng, G.-X. Zhang, N.J. Mitra, X. Huang, S.-M. Hu, Global contrast based salient region detection, in: International Conference on Computer Vision and Pattern Recognition, 2011.
[10] B.W. Tatler, The central fixation bias in scene viewing: selecting an optimal viewing position independently of motor biases and image feature distributions, J. Vis. 7 (2007) 1–17.
[11] M. Bindemann, Scene and screen center bias early eye movements in scene viewing, Vis. Res. 50 (2010) 2577–2587.
[12] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, in: International Conference on Computer Vision, 2009.
[13] Q. Zhao, C. Koch, Learning a saliency map using fixated locations in natural scenes, J. Vis. 11 (2011) 1–15.
[14] D. Parkhurst, K. Law, E. Niebur, Modeling the role of salience in the allocation of overt visual attention, Vis. Res. 42 (2002) 107–123.
[15] N. Riche, M. Mancas, B. Gosselin, T. Dutoit, Rare: a new bottom-up saliency model, in: International Conference on Image Processing, 2012.
[16] C. Bishop, Pattern Recognition and Machine Learning, Springer, NY, 2006.
[17] J. Zhang, J. Ding, C. Liu, J. Yang, Bayesian learning based visual saliency detection, in: International Conference of Automatic Control and Artificial Intelligence, 2012, pp. 1844–1847.
[18] Q. Zhao, C. Koch, Learning visual saliency by combining feature maps in a nonlinear manner using adaboost, J. Vis. 12 (2012) 1–15.
[19] A. Borji, Boosting bottom-up and top-down visual features for saliency estimation, in: International Conference on Computer Vision and Pattern Recognition, 2012.
[20] A. Torralba, Modeling global scene factors in attention, J. Opt. Soc. Am. 20 (7) (2003) 1407–1418.
[21] R. Rosenholtz, A simple saliency model predicts a number of motion popout phenomena, Vis. Res. 39 (19) (1999) 3157–3163.
[22] P.M.D. Comanicu, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 603–619.
[23] R. Achanta, F. Estrada, P. Wils, S. Ssstrunk, Salient region detection and segmentation, in: International Conference on Computer Vision Systems, 2008.
[24] N. Bruce, J. Tsotsos, Saliency based on information maximization, in: Neural Information Processing Systems.
[25] S. Goferman, L. Zelnik-Manor, A. Tal, Context-aware saliency detection, in: International Conference on Computer Vision and Pattern Recognition, 2010.
[26] R. Achanta, S. Hemami, F. Estrada, S. Ssstrunk, Frequency-tuned salient region detection, in: International Conference on Computer Vision and Pattern Recognition, 2009, pp. 1597–1604.
[27] Y. Zhai, M. Shah, Visual attention detection in video sequences using spatiotemporal cues, in: ACM Multimedia, 2006.
[28] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (2006) 861–874.
[29] A. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognit. 30 (1997) 1145–1159.
[30] J. Hanley, B. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1982) 29–36.
[31] A. Borji, L. Itti, Exploiting local and global patch rarities for saliency detection, in: International Conference on Computer Vision and Pattern Recognition, 2012.
[32] L. Zhang, M.H. Tong, T.K. Marks, H. Shan, G.W. Cottrell, Sun: a Bayesian framework for saliency using natural statistics, J. Vis. 0 (0) (2008) 1–20.

Jinxia Zhang currently is a Ph.D. candidate in the Department of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. She works under the supervision of Dr. Jundi Ding and Prof. Jingyu Yang. She is also a visiting Ph.D. student in Visual Attention Lab, Brigham and Women's Hospital. She is working under the supervision of Prof. Jeremy M. Wolfe. She received her bachelor's degree in the Department of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China in 2009. Her research interests include visual attention, visual saliency detection, computer vision and machine learning.

Jundi Ding currently works in Nanjing University of Science and Technology (NUST), Nanjing, China. From 2008 to 2010, she was a Postdoctoral Fellow of Pattern Recognition Group at School of Computer Science and Technology in NUST, Nanjing, China. She received her Ph.D. degree in the Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2008. Her research interests include data mining, clustering, image segmentation, visual saliency detection, etc.

Jingyu Yang received his B.S. degree in computer science from Nanjing University of Science and Technology (NUST), Nanjing, China. From 1982 to 1984, he was a visiting scientist at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 1993 to 1994, he was a visiting professor in the Department of Computer Science, Missuria University. And in 1998 he acted as a visiting professor at Concordia University in Canada. He is currently a professor and a chairman in the Department of Computer Science and Engineering at NUST. He is the author of more than 300 scientific papers in computer vision, pattern recognition and artificial intelligence. He has won more than 20 provincial and national awards. His current research interests are in the areas of pattern recognition, robot vision, image processing, data fusion and artificial intelligence.