Bio-Inspired Representation Learning for Visual Attention Prediction

Yuan Yuan, Senior Member, IEEE, Hailong Ning, and Xiaoqiang Lu, Senior Member, IEEE

Abstract—Visual Attention Prediction (VAP) is a significant and imperative issue in the field of computer vision. Most of existing VAP methods are based on deep learning. However, they do not fully take advantage of the low-level contrast features while generating the visual attention map. In this paper, a novel VAP method is proposed to generate visual attention map via bio-inspired representation learning. The bio-inspired representation learning combines both low-level contrast and high-level semantic features simultaneously, which are developed by the fact that human eye is sensitive to the patches with high contrast and objects with high semantics. The proposed method is composed of three main steps: 1) feature extraction, 2) bio-inspired representation learning and 3) visual attention map generation. Firstly, the high-level semantic feature is extracted from the refined VGG16, while the low-level contrast feature is extracted by the proposed contrast feature extraction block in a deep network. Secondly, during bio-inspired representation learning, both the extracted low-level contrast and high-level semantic features are combined by the designed densely connected block, which is proposed to concatenate various features scale by scale. Finally, the weighted-fully-connected layer is exploited to generate the ultimate visual attention map based on the obtained representations after bio-inspired representation learning. Extensive experiments are performed to demonstrate the effectiveness of the proposed method.

Index Terms—Visual Attention Prediction, Bio-Inspired, Contrast features, Semantic features, Densely Connected, Reduction-Attention, Centre-bias Prior

I. INTRODUCTION

With the rapid development of mobile Internet, data are soaring, and the era of big data has come. We have to deal with a lot of information every day [1]–[3], especially in the form of pictures and videos. Saliency detection provides us an effective idea for acquiring the most valuable information from massive data. The study of saliency methods can be divided into two research directions [4]: 1) the prediction of human eye fixation and 2) the salient object detection. The former is to predict the gaze positions that human focus on at first glance, attaching to a regression problem. And the latter is to detect the salient objects in an observed scene, which is similar to hyperspectral image classification [5], belonging to a classification problem. In this paper, we focus on the eye fixation prediction task to get a continuous-valued density map, named visual attention map. The visual attention map has been used in scene understanding [6]–[9], object recognition [10], target detection [11], target tracking [12], image quality assessment [13] and video compression [14].

In the field of the neuroscience, humans are more interested in the patch which stands out from its surrounding area. Fig. 1 shows four different images in MIT1003 dataset: (a) and (b) are images containing high-contrast patches and corresponding visual attention maps, (c) and (d) are images containing high-semantics objects and corresponding visual attention maps.

![Fig. 1. Various visual attention maps for different images in MIT1003 dataset: (a) and (b) are images containing high-contrast patches and corresponding visual attention maps, (c) and (d) are images containing high-semantics objects and corresponding visual attention maps.](image-url)
ought to take the two attention mechanisms into consideration simultaneously. In other words, both low-level contrast and high-level semantic features are supposed to be fully leveraged for VAP.

In the existing related works [16]–[20], the human attention in an image is described by two kinds of features: low-level contrast features and high-level semantic features. The low-level contrast features are developed to describe the discriminative regions in the image, because the contrast plays an important role in visual attention cognitive. In other words, high contrast regions are more easily noticed by people. On the contrary, the high-level semantic features are driven by internal semantic aspects (face, people, text, etc.). They can provide better locations about the semantic regions. As a result, researchers in the field of computer vision have adopted these two kinds of features respectively for Visual Attention Prediction (VAP). The majority of earlier VAP methods [16]–[18] employ low-level contrast features. Specifically, color, intensity, orientation and other visual features are extracted manually to predict the visual attention map by calculating the contrast between the central area and the surrounding area. However, these low-level contrast features methods do not adequately take high-level semantic aspects in an observed scene into account. In recent years, the emerging deep Convolutional Neural Networks (CNNs) [21] and the availability of large datasets have achieved impressive results in VAP. CNNs can extract powerful features to predict visual attention in an end-to-end manner. However, these CNNs methods focus on building the high-level semantic features from the last convolutional layers or fully-connected layers, making it hard to capture the low-level contrast features.

Based on the above facts, an intuitive method is proposed to predict the visual attention map. The proposed method generates the visual attention map by bio-inspired representation learning which can combine both low-level contrast and high-level semantic features. The generated representation coincides with the human visual attention mechanism, since human is sensitive to the patches with high contrast and objects with high semantics. As shown in Fig. 2, the proposed method can be divided into three main steps: 1) feature extraction, 2) bio-inspired representation learning and 3) visual attention map generation. Firstly, five different features are extracted from each convolutional block of the refined VGG16 (see III-A1). The features from the first two convolutional blocks may contain the high-frequency contour information, and the features from the last three convolutional blocks may contain information of the internal semantic aspects [22]. To obtain the low-level contrast features, a contrast feature extraction block is proposed to process the features from the first two convolutional blocks (see III-A2) in a deep network. The features from the last three convolutional blocks are usually treated as high-level semantic features. Secondly, during bio-inspired representation learning, both the extracted low-level contrast and high-level semantic features are combined by the designed densely connected block (see III-B3), which is proposed to concatenate various features scale by scale. Specifically, the designed densely connected block introduces some short connections [23] from higher-level features (deriving from the higher layers of the refined VGG16) to the lower-level features (deriving from the lower layers of the refined VGG16). In this top-down view, the high-level features can provide better locations about the semantic regions, while the low-level features can capture the discriminative regions in the image. Finally, to generate the ultimate visual attention map, a weighted-fusion layer is utilized to fuse the learned centre-bias prior map and five rough visual attention maps [24]. The centre-bias prior map is produced using a two-dimensional Gaussian function, and the five rough visual attention maps are produced by the readout network based on the obtained representations after bio-inspired representation learning (see Fig. 2). To sum up, the main contributions of this paper are threefold:

- A novel network is proposed to learn the bio-inspired representation for visual attention prediction, which can combine both low-level contrast and high-level semantic features.
- A contrast feature extraction block is proposed to extract the low-level contrast features in the network. This is the first method applied to extracting low-level contrast features automatically in a deep network.
- A densely connected block is proposed by introducing short connections from the higher-level features to the lower-level features. In this top-down view, the high-level features can provide better locations about the semantic regions, while the low-level features can capture rich spatial information.

The remaining parts of this paper are organized as follows: Section II reviews the related works. Section III gives a detailed description of the proposed method. The experiments are shown in Section IV. Finally, a conclusion is presented in Section V.

II. RELATED WORKS

In recent years, a large number of visual attention prediction methods have been proposed. According to the kind of the adopted feature, these VAP methods can be mainly divided: 1) methods based on low-level contrast features and 2) methods based on high-level semantic features.

Most of VAP computational methods are based on low-level contrast features. Beginning with the seminal work of Itti et al. [16], many methods have been proposed for VAP from various mechanisms. On the basis of the mechanism for VAP, existing traditional methods can be classified as: cognitive method [25], information theoretic method [26], graphical method [27], spectral analysis method [28], decision theoretic method [29], pattern classification method [30], and some other methods [31]. On the one hand, these methods only employ low-level contrast features and do not take into account the semantic aspects (face, people, text, etc.) which can provide better locations about the semantic regions. On the other hand, the extracted low-level contrast features are insufficient to handle large-scale data with complex distributions because they are hand-crafted.

With the popularity of deep learning, high-level semantic features can be automatically extracted [31]–[33], and these
high-level semantic features have been exploited for VAP [19], [20], [22], [34]–[42]. Compared with these methods based on low-level contrast features, methods based on high-level semantic features achieved better results. The first attempt for VAP with deep neural networks was the Ensemble of Deep Networks (eDN) [34]. However, this method cannot outperform the state-of-the-arts at that time because of its limited training data and poor semantic information for only three convolutional layers. To address the problem in eDN, Kümmnerer et al. proposed two methods named Deep Gaze I [19] and Deep Gaze II [19], and achieved a goodish result. Subsequently, many researchers began to notice the impact of the centre-bias prior on prediction results and a large number of related works emerged. [19], [20], [36] introduced centre-bias prior into the deep computational methods for VAP and improved the predicted results to some extent. For the past two years, some researchers on object detection have found that multi-scale features contribute to achieving a good performance for detecting objects of different scales [22]. Drawing on this success, many researchers adopted multi-scale architecture for VAP, and the predicting performance is further improved by a substantial margin [22], [37]. To sum up, most of existing deep learning methods leveraged multi-scale features, introduced centre-bias prior and acquired great success in VAP. However, the utilized features in these methods were not enough to reflect the human visual attention mechanism. Specifically, almost all existing methods based on high-level semantic features overlooked the contribution of low-level contrast features in VAP, which play an important role in visual attention cognition.

Therefore, in order to fully leverage the information of both low-level contrast and high-level semantic features simultaneously, an intuitive VAP method is proposed for generating visual attention map via bio-inspired representation learning. Similar to [23], the proposed method introduces some short connections from higher-level features to the lower-level features. But unlike [23], we adopt a new processing strategy for the obtained lower-level features because of the task difference. Specifically, VAP aims to locate where the salient objects are, which does not need to highlight the boundaries of the salient objects [23]. Hence, instead of using the lower-level features directly, we input them into the contrast feature extraction block for automatically obtaining the low-level contrast features in the network. The details of the proposed method will be described in the following sections.

### III. THE PROPOSED METHOD

In this work, an intuitive method is proposed to generate visual attention map via bio-inspired representation learning. The proposed method is dwelled on in this section. As is depicted in Fig. 2, the proposed method takes a batch of images with spatial size of $224 \times 224$ as input. Firstly, the high-level semantic features $\{F_3, F_4, F_5\}$ are extracted from the refined VGG16 (see III-A1), while the low-level contrast features $\{F_1, F_2\}$ are extracted by the proposed contrast feature extraction block (see III-A2) in a deep network. Secondly, during bio-inspired representation learning, both the extracted low-level contrast and high-level semantic features are combined by the designed densely connected block (see III-B2), which is proposed to concatenate various features scale by scale. Finally, the weighted-fusion layer is exploited to generate the ultimate visual attention map $\hat{M}_{final}$ based on the obtained representations $\{G_1, G_2, G_3, G_4, G_5\}$ after bio-inspired representation learning. The details are introduced as follows.
A. Feature Extraction

This step can be divided into two parts: the refined VGG16 and contrast feature extraction block. Thereinto, the first part is used for extracting high-level semantic features (the outputs of conv3-3, conv4-3 and conv5-3 in the refined VGG16) and raw low-level features (conv1-2 and conv2-2 in the refined VGG16). The second part is used for obtaining low-level contrast features from the raw low-level features, which can extract low-level contrast features in a deep network. The refined VGG16 and contrast feature extraction block are elaborated as follows.

1) The Refined VGG16: The CNNs require the size of the input image to be fixed. For example, existing pre-trained VGG models \cite{43} require the input size to be 224×224. This requirement is determined by the network structure, because CNNs generally include multiple fully connected layers, and the number of fully connected neurons is usually fixed. Generally, multi-level features (the output of different layer in the neural network) can effectively improve the results of image retrieval, image classification, and object detection tasks \cite{44}. The same is true for the VAP task. Differently, the VAP task only needs to locate the most salient region and has nothing to do with the object boundary. Therefore, we extract the low-level features for refining contrast features instead of improving the edge extraction result. To obtain multi-level features, we use a strategy of feature pyramid network to extract various features from different layers.

In this paper, we refine VGG16 for extracting multiple features on account of its elegance, simplicity and effectiveness on extracting various visual features. The origin VGG16 consists of five convolutional blocks, where each block is followed by a max-pooling layer, and three fully connected layers. Considering what we want to harness is the feature map, the origin VGG16 network is refined. We only adopt the convolutional blocks except for the last pooling layer and omit fully connected layers. In addition, we avoid the feature map being too small in the fifth convolutional block, we set the stride of the fourth pooling as 1, and meantime use a convolutional kernel with holes of size 2 in the fifth convolutional block to keep the same receptive filed as the origin architecture.

As is shown in Fig. 2, formulaically, let \( I \in \mathbb{R}^{224 \times 224 \times 3} \) represent one of the input images. For the \( m \)-th convolutional layer, we denote its input as \( I_m \), where \( m \in \{1, \cdots, 13\} \), and its output as \( X_m \). As for the input of each convolutional layer, a kernel \( W_m \) is adopted to perform 2D convolution operation. And then a bias term \( b_m \) is added to the convolution operation result. An activation function is used for non-linear mapping. The process of each convolutional layer can be formulated as:

\[
X_m = \delta (I_m \ast W_m + b_m),
\]

where \( \ast \) denotes convolution operation and \( \delta (\cdot) \) indicates activation function. The Rectified Linear Unit (ReLU) is adopted as activation function in this paper for its high efficiency and it can be formulated as:

\[
\delta (x) = \max (0, x),
\]

where \( x \) denotes an arbitrary real number. In addition, each convolutional blocks except for the last pooling layer and omit fully connected layers. In this work, we set the stride of the fourth pooling as 1, and meantime use a convolutional kernel with holes of size 2 in the fifth convolutional block to keep the same receptive filed as the origin architecture.

2) Contrast Feature Extraction Block: Considering that human eyes are sensitive to high contrast patches, a contrast feature extraction block (Fig. 3) is devised for low-level contrast features automatical extraction in a deep network. The outputs of conv1-2 and conv2-2 in the refined VGG16 may contain the high-frequency contour information, which is the edge of high contrast patches. As a result, we take the outputs of conv1-2 and conv2-2 in the refined VGG16 as the inputs of this contrast feature extraction block for automatically obtaining contrast features in deep network. Let \( O = [o_1, o_2, \cdots, o_C] \in \mathbb{R}^{W_1 \times H_1 \times C_1} \) denote the input of this block. Firstly, the input \( O \) is squeezed as one channel in an average manner across all channels. We call this one-channel feature map as intensity map, which is denoted as \( O \in \mathbb{R}^{W_1 \times H_1} \). Secondly, five different Gaussian kernels are employed for \( O \) to get a Gaussian Pyramid \( \overline{O}^G = \{\overline{o}_1, \overline{o}_2, \cdots, \overline{o}_L\} \in \mathbb{R}^{W_1 \times H_1 \times L} \) \cite{19}, where \( L=5 \).

The obtained Gaussian Pyramid by different Gaussian kernels contains different spatial information with various scale. Formulailcally,

\[
\overline{o}_l = O \ast G_{sl},
\]

where the standard deviations the Gaussian kernels \( G_{sl} \) are 5, 10, 20, 40 and 80 pixels, respectively. Thirdly, the residual features \( O' \in \mathbb{R}^{W_1 \times H_1 \times C_1'} \) are computed by subtracting each channel of the input features \( O \) from the Gaussian Pyramid \( \overline{O}^G \), where \( C_1'=5 \times C_1 \). Fourthly, we get squared residual features by squaring these residual features. Finally, eventu
contrast features are obtained via adaptively merging squared residual features and Gaussian Pyramid features instead of simply concatenating them. Let $O'' \in \mathbb{R}^{W_1 \times H_1 \times C''}$ represent the eventual contrast features, then the whole process can be expressed as the following formulation:

$$O'' = C_F(O) = AM(O', \overline{O''})$$

$$= \left( \sum_{s=1}^{C'} \sum_{k=1}^{L} (o_{s,k} - \overline{o_{s,k}})^2 \right) \ast W_{AM} + \overline{O''} \ast W_{AM},$$

(4)

where $C_F(\cdot)$ denotes the whole contrast feature extraction operation, $AM(\cdot, \cdot)$ indicates the adaptively merging operation, $W_{AM}$ and $W_{AM}$ are the adaptively merging weights.

To validate the effectiveness of the proposed contrast feature extraction block, the comparative experiment is performed and the experiment result is shown in Table [V].

B. Bio-Inspired Representation Learning

The insight of the proposed method is that both low-level contrast and high-level semantic features contribute to VAP. As a result, both low-level contrast and high-level semantic features should be taken into account. To this end, bio-inspired representation learning step is devised. This step can be divided into two main parts: 1) the reduction-attention block for alleviating the problem of dimensionality disaster and adaptively recalibrating channel-wise feature responses while combining various features during bio-inspired representation learning, and 2) the densely connected block for combining both low-level contrast and high-level semantic features by a concatenation operator scale by scale. We elaborate these two parts hereinafter.

1) Reduction-Attention Block: Traditional convolutional neural network is a kind of spatial attention network by learning different filters. Therefore, it only considers the local connectivity pattern on each input channel, and the input channels are treated with a uniform weight. Recent works [45]–[47] have demonstrated that different channels of the feature map contain different amount of information. In other words, different channels of the feature map should be allocated with suitable weights which match their amount of information. Accordingly, inspired by the SE block [45], a new reduction-attention block is devised. The devised reduction-attention block can process the original feature into a channel descriptor, which contains global information about each channel feature and can discard some interferences in the intermediate representation. In general, the devised reduction-attention block aims to avoid overfitting and learn a series of weights for each input channel. Specifically, to ease the problem of dimensionality disaster while performing bio-inspired representation learning [8], a dimension reduction architecture is used for obtaining the reductive feature in the first place. Then, a retrofitted SE block is followed by the adaptive dimension reduction architecture. Different from the origin SE block, the second fully connected layer is omitted and the input is reductive in the retrofitted SE block to alleviate overfitting.

As is shown in Fig. 4, firstly, the feature dimension of $F_A \in \mathbb{R}^{W \times H \times C}$ is reduced by a operation of $1 \times 1$ convolutional layer to get a more elegant feature $F''_A \in \mathbb{R}^{W \times H \times C''} = [\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_{C''}]$. Meanwhile, the nonlinear interaction between channels is increased after the $1 \times 1$ convolutional layer. Secondly, the global spatial information of each channel in $F''_A$ is squeezed so as to generate a channel descriptor. Thirdly, in order to convert the generated channel descriptor into the channel-weighted matrix, the global spatial information of each channel is extended to a channel-weighted matrix. And finally, the channel-weighted feature is obtained by element-wise multiplication between the input feature and the channel-weighted matrix.

$$F''_A = h(F_A) = \Psi(a) \circ F'_A = \overline{W} \circ F'_A,$$  (7)

where $h(\cdot)$ indicates the reduction-attention operation, $\circ$ represents element-wise multiplication, and $\Psi(\cdot)$ denotes the operation of extending a vector to a three-dimensional matrix, which is defined by us. Specifically for $\Psi(\cdot)$, a channel weight matrix $\overline{W} = [w_1, w_2, \ldots, w_{C''}] \in \mathbb{R}^{W \times H \times C''}$ is generated, in which all values of $w_{c'} \in \mathbb{R}^{W \times H}$ are equal to the value of $a_c$.

To validate the effectiveness of the proposed reduction-attention block, the comparative experiment is performed and the experiment result is shown in Table [V].

2) Densely Connected Block: As is shown in Fig. 2, the densely connected block is based on DenseNet [48] and the work of Cheng et al. [23]. We choose five-level features $\{F_1, F_2, F_3, F_4, F_5\}$. Thereinto, $F_3, F_4$ and $F_5$ are treated as high-level semantic features, which directly derived from the refined VGG16, while $F_1$ and $F_2$ are considered as low-level contrast features, which come from the contrast feature extraction block. In this densely connected block, five-
level ultimate feature representations \{G_1, G_2, G_3, G_4, G_5\} are obtained by lateral connecting scale by scale and cross-level dense connecting. For instance, in order to get \(G_1\), the channel numbers of features \(F_1, F_2, F_3, F_4\) and \(F_5\) are reduced using the designed reduction-attention block. Then, the resize-convolution operation \(\mathbb{R}\) is introduced for upsampling all the obtained multiple features except for \(F_1\). The reason for choosing the resize-convolution operation is its advantage in averting the checkerboard effect. Finally, \(F_1\) and other upsampling features are combined to get the ultimate representation \(G_1\). After the iteration above, the other four ultimate representations \(G_2, G_3, G_4,\) and \(G_5\) are acquired. Here, we suppose that there are total \(J\) ultimate representations or \(J\) branches. Formulically,

\[
G_j = \begin{cases} 
A^{[j-2]}(F_j), & \text{if } j = J \\
\mathbb{R} \left( \sum_{i=j}^{J-1} w_i^j R_i^{j-i} + R_j^{J-1-j} \right), & \text{else if } j = 1 \\
A^{[j-1]} \left( \sum_{i=j}^{J-1} w_i^j R_i^{j-i} + R_j^{J-1-j} \right), & \text{otherwise }
\end{cases}
\]

where \(A^{[j-2]}(\cdot)\) and \(A^{[j-1]}(\cdot)\) denote the tandem operation of resize-convolution operation and reduction-attention operation, in which the superscript \(j-2\) and \(j-1\) signify the count of this kind of operation. \(R_i^{j-i}\) denotes the activation of \(F_i\) after \(A^{[i-2]}(\cdot)\), and similarly, \(R_i^{J-1-j}\) denotes the activation of \(F_i\) after \(A^{[J-1-i]}(\cdot)\). \(w_i^j\) denotes the weight of the short connection from the \(i\)-th branch to the \(j\)-th branch. Note that \(A^{[0]}(\cdot) = 1\) and \(R_0^j = F_j\).

To validate the effectiveness of the devised densely connected block, the comparative experiment is performed and the experiment result is shown in Table \(V\).

C. Visual Attention Map Generation

In this step, we aim to predict the final visual attention map based on the obtained features after bio-inspired representation learning. Firstly, five rough visual attention maps are generated by the readout network, which is stacked by three successive reduction-attention blocks. And subsequently, a weighted-fusion layer is adopted for fusing these five rough visual attention maps with the learned centre-bias prior map to generate the final visual attention map. The learned centre-bias prior map is added to compensate for the secondary effect by the photographer’s bias. Hereinafter, we dwell on this step in two main parts: centre-bias prior block and loss function.

1) Centre-bias Prior Block: Physiological studies have shown that when observing an image, human attention tends to concentrate on the centre. This is a common phenomenon named centre-bias, which is because the photographer is accustomed to placing the object of interest at the centre of the image. As a result, when people watch such images repeatedly, their attention is attracted by the centre of the image naturally. Note that when there are no highly salient objects, human attention is also attracted to the centre of the image. Therefore, this phenomenon has great influence on the final evaluation metrics, such as Area Under Curve (AUC), Normalized Scanpath Saliency (NSS), Linear Correlation Coefficient (CC) and so on, even leading to unfair evaluation.

In order to address this problem, a learnable centre-bias prior block is designed using a two-dimensional Gaussian function for modeling the centre-bias. Compared to most of existing works, the proposed centre-bias prior is learnable and derives from data purely instead of the assumptions from indistinct biological studies. The insight of our centre-bias prior block comes from the work of Conia et al. \(\{36\}\), which also adopts a learnable manner modeling the centre-bias prior. Differently, our centre-bias prior block is more effective with less parameter than theirs. Specifically, a specifical Gaussian map is generated as Eq. \(9\) depending on the learnable specific horizontal variance \(\sigma_x^2\) and vertical variance \(\sigma_y^2\). Nevertheless, the method of Conia et al. \(\{36\}\) relies on learnable means and variances, in which there is no need to learn means since the center of the generated Gaussian probability distribution is fixed. The equation of generating centre-bias prior map is as follows:

\[
M_{\text{prior}} = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left( -\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2} \right).
\]

Here, each Gaussian map represents a spatial pattern. We can model the centre-bias using Gaussian map purely from data rather than the assumptions from indistinct biological studies.

To validate the effectiveness of the proposed centre-bias prior block, the comparative experiment is performed and the experiment result is shown in Table \(V\).

2) Loss Function: To get five rough visual attention maps, the obtained five features \(G_1, G_2, G_3, G_4,\) and \(G_5\) after the inspired representation learning are fed into the readout network. The readout network is stacked by three reduction-attention blocks to learn interactions across channels instead of learning new spatial features. Let \(M_j\) represent the obtained \(j\)-th rough visual attention map, and \(W_j\) denotes all the parameters before generating the \(j\)-th rough visual attention map \(M_j\). For simplicity, the bias term is absorbed into \(W_j\). Therefore, the objective function for generating the \(j\)-th rough visual attention map can be given with Kullback-Leibler divergence and weight decay term as follows:

\[
L_j (M_j, Z_{\text{den}}) = \text{KL} (M_j, Z_{\text{den}}) + \alpha \sum_t (W_j^t)^2 \\
= \frac{1}{T} \sum_t Z_{\text{den}}^t \log \left( \frac{Z_{\text{den}}^t}{M_j^t + \epsilon} \right) + \alpha \sum_t (W_j^t)^2,
\]

where \(\text{KL}(\cdot, \cdot)\) means the Kullback-Leibler divergence of two probability distributions, \(Z_{\text{den}}\) represents the groundtruth density map, \(t\) indexes the \(t\)-th pixel, \(\epsilon\) denotes the regularization constant to avoid the Nan value of the loss during training phase, \(\alpha\) is the coefficient of weight penalty, and \(W_j^t\) indicates the value of the \(t\)-th pixel in matrix \(W_j\).

Next, to take advantage of the merit of different rough

\[\text{https://distill.pub/2016/deconv-checkerboard/}\]
visual attention maps and tackle the secondary effect by the photographers bias, a weighted-fusion layer is added for image-level fusion. Note that the architecture of the weighted-fusion layer is similar to the readout network. The weighted-fusion layer is also stacked by three reduction-attention blocks to learn interactions across different rough visual attention maps and the learned centre-bias prior map. Finally, after the weighted-fusion layer, a Gaussian convolutional layer is adopted to produce the final visual attention map. As a result, when we suppose that the parameter in the weighted-fusion layer is $K$, the final visual attention map can be given as follows:

$$M_{final} = \left( \sum_{j=1}^{J} M_{j} + M_{prior} \right) \ast K \ast G_{\sigma},$$

(11)

where $M_{final}$ is the final visual attention map, and $G_{\sigma}$ represents the Gaussian convolution kernel. Therefore, the final fusion loss function can be formulated as:

$$L_{fuse}(M_{final}, Z_{den}) = KL(M_{final}, Z_{den}).$$

(12)

Ultimately, the final loss function can be calculated by:

$$L_{final}(M_{j}, M_{final}, Z_{den}) = \sum_{j=1}^{J} L_{j}(M_{j}, Z_{den}) + L_{fuse}(M_{final}, Z_{den}).$$

(13)

Then all parameters $W_{j}$ and $K$ can be learned by minimizing the final loss $L_{final}(M_{j}, M_{final}, Z_{den})$ over all the training images via RMSProp. After training, given a test image $I_{test}$, a corresponding visual attention map $M_{test}$ can be predicted by the proposed method. The main procedure of the proposed method is shown in Algorithm 1.

### Algorithm 1: The proposed method

**Input:**
- Training images $I$ and their corresponding labels $Z_{den}$.
- Testing image $I_{test}$.

**Output:**
- Testing visual attention map $M_{test}$.
- All the parameters $W_{j}$ of generating the $j$-th visual attention map.
- All the parameters $K$ in the weighted-fusion layer.

**Initialization:**
- The weights of the first four convolutional blocks in the refined VGG16 are initialized from the origin VGG16, and the remaining weights are randomly sampled by truncated normal distribution.

**Repeat:**
1. Calculate the feature map $F_{j}$;
2. Learn the bio-inspired feature representations $G_{j}$;
3. Infer the $j$-th rough visual attention map $M_{j}$ and calculate the loss $L_{j}$ according to Eq. (10);
4. Generate the centre-bias prior map $M_{prior}$ according to Eq. (9);
5. Infer the final visual attention map $M_{final}$ and calculate the fusion loss $L_{fuse}$ according to Eq. (12);
6. Calculate the final loss $L_{final}$ according to Eq. (13);
7. Update the parameters $W_{j}$ and $K$ by utilizing RMSProp.

**Until:** A fixed number of iterations

**Return:** $M_{test}$, $W_{j}$, $K$

from Microsoft COCO dataset [53]. In this work, 10000 training images and 5000 validating images are leveraged.

2) **OSIE** [50]: The OSIE dataset contains 700 natural indoor and outdoor scenes, aesthetic photographs from Flickr and Google. Each image corresponds to an eye tracking annotation from 15 viewers.

3) **MIT1003** [15]: The MIT1003 dataset includes 1003 natural indoor and outdoor scenes, which are randomly selected from Flickr creative commons and LabelMe. The groundtruth map is generated by recording eye tracking data from 15 viewers when they observe these images freely.

4) **TORONTO** [51]: The TORONTO dataset involves 120 color images of outdoor and indoor scenes. Corresponding fixation maps are generated by the eye tracking data from 20 subjects. This dataset is widely used to evaluate the method of VAP.

5) **PASCAL-S** [52]: The PASCAL-S dataset is a collection of 850 natural images from the validation set of PASCAL VOC 2010. The groundtruth maps are generated by the eye tracking data during 2 seconds from 8 observers.

### B. Evaluation Metrics

There are several evaluation metrics to measure the consistency between the predicted result and the groundtruth [54]. These metrics can be roughly divided into three categories: 1) the distribution-based metric, 2) the value-based metric and 3) the location-based metric. The distribution-based metrics include Earth Movers Distance (EMD) and Linear Correlation.
Coefficient (CC). The value-based metric involves Normalized Scanpath Saliency (NSS). And the location-based metric is mainly Area Under Curve (AUC), which consists of several variations: AUC-Judd, AUC-Borji, shuffled-AUC. Here, for clarity, the predicted visual attention map, the groundtruth density map and the groundtruth fixation map are denoted as $M_{\text{final}}, Z_{\text{den}},$ and $Z_{\text{fix}},$ respectively. Next, these metrics are elaborated in the following.

1) **Earth Movers Distance (EMD):** EMD can be used to measure the normalized minimum cost of changing from one distribution to another. Therefore, the performance of the proposed method can be evaluated by calculating the EMD between the predicted visual attention map $M_{\text{final}}$ and the groundtruth density map $Z_{\text{den}}.$ The smaller EMD is, the smaller the distance between $M_{\text{final}}$ and $Z_{\text{den}}$ is, and the better performance of the proposed method is.

2) **Linear Correlation Coefficient (CC):** CC is also called the Pearson linear correlation coefficient. It can be used to measure the linear correlation between the predicted visual attention map $M_{\text{final}}$ and the groundtruth density map $Z_{\text{den}}.$ The closer to 1 the score of CC is, the higher the linear correlation between $M_{\text{final}}$ and $Z_{\text{den}}$ is, and the better performance of the proposed method is. The formula is as follows:

$$CC = \frac{\text{cov}(M_{\text{final}}, Z_{\text{den}})}{\sigma'(M_{\text{final}}) \times \sigma'(Z_{\text{den}})},$$

where $\sigma'(M_{\text{final}})$ and $\sigma'(Z_{\text{den}})$ are the standard deviation corresponding to the predicted visual attention map $M_{\text{final}}$ and the groundtruth density map $Z_{\text{den}},$ and $\text{cov}(M_{\text{final}}, Z_{\text{den}})$ denotes the covariance between $M_{\text{final}}$ and $Z_{\text{den}}.$

3) **Normalized Scanpath Saliency (NSS):** NSS can be used to measure the saliency value at human fixations. Let $M_{\text{final}}$ represent the normalized predicted visual attention map. Then NSS can express the average value of $M_{\text{final}}$ at all human fixations. Given the predicted visual attention map $M_{\text{final}}$ and the groundtruth fixation map $Z_{\text{fix}},$ NSS can be calculated by the following formula:

$$NSS = \frac{1}{T'} \sum_{t=1}^{T'} M_{\text{final}}(t) \times Z_{\text{fix}}(t)$$

where $T' = \sum_{t} Z_{\text{fix}}(t),$ and

$$M_{\text{final}}' = \frac{M_{\text{final}} - \mu(M_{\text{final}})}{\sigma(M_{\text{final}})}.$$

Here, $T'$ represents the number of all fixated pixels. NSS is sensitive to the false positive. Hence, the larger NSS is, the higher the accuracy of the predicted visual attention map is.

4) **Area Under Curve (AUC):** AUC is a location-based metric. The groundtruth fixation map $Z_{\text{fix}}$ contains human fixation information for the image. According to $M_{\text{final}}$ and $Z_{\text{fix}},$ the true positive rate and the false positive rate can be calculated, and then the receiver operating characteristic (ROC) curve is plotted. By calculating the area under the ROC curve, the AUC score is obtained. Depending on the choice of non-fixation distribution, three different variations of AUC are adopted to evaluate the performance of the proposed method: AUC-Judd, AUC-Borji, and shuffled-AUC (s-AUC). The first two versions choose the non-fixation in the image as a uniform distribution, while the s-AUC uses the human fixation in other images under the dataset as non-fixation distribution. In general, the s-AUC is more reasonable because it gives a penalization to the methods which consider the centre bias in eye fixations [56]. The score of AUC should range 0.5 to 1. And the closer to 1 the score is, the better performance of the proposed method is.

C. Implementation Details

The details in network architecture and the details in training and testing phase are elaborated as follows.

1) **The Details in Network Architecture:** As for the network, the input images are resized to $224 \times 224.$ The weights of the first fourth convolutional blocks in the refined VGG16 are initialized from the origin VGG16, which is trained on the ImageNet. Other weights are randomly sampled by truncated normal distribution. Taking into account the specificity of VAP task, namely including images with high-contrast objects and images with high-semantics objects, both low-level contrast and high-level semantic features should be harnessed. To this end, the outputs of $\text{conv}1-2,$ $\text{conv}2-2,$ $\text{conv}3-3,$ $\text{conv}4-3$ and $\text{conv}5-3$ are gathered from the refined VGG16. The spatial dimensions of the obtained features are $224 \times 224,$ $112 \times 112,$ $56 \times 56,$ $28 \times 28,$ and $28 \times 28,$ respectively. Then, the outputs of $\text{conv}1-2$ and $\text{conv}2-2$ are fed into the contrast feature extraction block for automatically obtaining contrast features in a deep network. These features have different receptive field and contain various complementary information. Subsequently, these obtained multiple features are input into the next steps for inferring the final visual attention map.

The acquired basic features $F_2,$ $F_3,$ $F_4$ and $F_5$ are upsampled scale by scale to enlarge the spatial resolution. Then, each upsamplied feature is concatenated with higher spatial resolution feature. It is noteworthy that the upsampling operation is in a resize-convolution approach instead of a simply deconvolution approach due to the checkerboard artifacts of uneven overlap during the deconvolution operation. Specifically, the feature map with low spatial resolution is resized to a higher resolution using a nearest-neighbor interpolation method. Subsequently, a $3 \times 3$ convolutional layer is followed to further learn the sampling pattern. In addition, when various features are used, the devised reduction-attention block is employed to automatically learn the combination weight due to the different contribution of various features. To verify the effectiveness of this combination approach, an ablation analysis is performed. Firstly, multiple features are concatenated directly after upsampling to $224 \times 224.$ Secondly, multiple features are concatenated scale by scale in a densely connected manner. Thirdly, the reduction-attention block is added to automatically learn the combination weight for different features. Expectedly, the best prediction result is obtained when the reduction-attention block is added to learn the combination weight. Therefore, the adopted combination strategy is effective.
As for the readout network in inferring five rough visual attention maps, it consists of three stacked reduction-attention blocks to learn interactions across channels instead of learn new spatial features. The output channel numbers of the three stacked reduction-attention block are 32, 16 and 1, respectively. After the readout network, five rough visual attention maps are obtained. Next, to address the centre-bias problem in the human eye fixations, the learned centre-bias prior map is fused with the obtained rough visual attention maps. Then, we perform an image-level fusion by a weighted-fusion layer to get the fused visual attention map, which adopts similar architecture with the readout network. Afterwards, the fused visual attention map is convolved with a $7 \times 7$ Gaussian kernel so that we obtain the visual attention map with spatial size of $224 \times 224$. Finally, the visual attention map with spatial size of $224 \times 224$ is resized to get the same size as the original image.

2) Training and Testing: The proposed method is trained on SALICON dataset and tested on OSIE, MIT1003, TORONTO and PASCAL-S datasets with the tensorflow library. For the training phase, all input images are resized to a uniform size with $224 \times 224$ to satisfy the input size of the VGG network. The minibatch input is fed into the network, and the final loss (Eq. 15) is minimized using a RMSprop optimizer. The initial learning rate is set as $10^{-4}$. The weight decay is set as 0.0005. And the momentum is set as 0.9. We find that when the batch size is set to 10, the convergence speed is quickest and the performance is best. 5000 validation images of SALICON dataset are exploited to decide when to stop the training process for avoiding overfitting furtherly. Specifically, after the training of each epoch, the performance on these 5000 validation images are evaluated and the training is terminated when the performance begin to decrease or the training epoch achieves 100 to reduce unnecessary training cost. For the testing phase, we use OSIE, MIT1003, TORONTO and PASCAL-S datasets for evaluating the prediction result. Given a query image, it will generate a final visual attention map automatically with the same size as the query image. The experiment is performed on our PC with a TITAN X (Pascal) GPU and 64G RAM.

D. Comparison with State-of-the-arts

To demonstrate the effectiveness of the proposed method, we compare the proposed method with several state-of-the-art methods. The compared methods are comprehensive, including 6 classical methods, 4 deep learning methods trained on SALICON dataset, 2 deep learning methods trained on MIT1003 dataset and 2 deep learning methods trained on two datasets.

1) Classical Methods: These methods are widely used to compare the performance of the VAP model. The comparison classical methods include ITTI [16], JUDD [15], BMS [57], GBVS [27], CAS [30] and AWS [56] methods. ITTI is the seminal work for VAP, in which multi-scale image features are combined into a single topographical for generating visual attention map. JUDD predicts the visual attention map based on low, middle and high-level image features. BMS finds the surrounding area by Boolean topology analysis to predict the visual attention map. GBVS is a bottom-up visual VAP model, which is based on graph. CAS proposes a detection algorithm based on the four principles observed in the psychology literature. AWS utilizes a contextual adaptation mechanism to ensure that the contribution of image points to optical changes is invariant in behavior caused in the visual system.

2) Methods Trained on SALICON Dataset: These methods are similar to the proposed method, which also adopts SALICON dataset for training. The comparison methods trained on SALICON dataset include SAM-VGG [36], SAM-ResNet [59], DVA [22], and SU [53] methods. SAM-VGG and SAM-ResNet come from one of the latest work for VAP, which highlight the most significant areas of the input image to refine the predictive saliency map by convolution LSTM iteration. DVA is also a recent state-of-the-art work, which can capture hierarchically saliency information from deep layers with global saliency information to shallow layers with local saliency responses. SU capitalizes on a deep convolutional neural network to predict the visual attention map and segment salient objects in a unified framework.

3) Methods Trained on MIT1003 Dataset: These methods are used to compare the performance when using different training dataset. The comparison methods trained on MIT1003 dataset include Mr-CNN [55] and eDN [44] methods. Mr-CNN adopts a multi-resolution convolutional neural network to infer local contrast, global contrast, and top-down visual factors from raw image data simultaneously so as to generate the visual attention map. eDN is the first VAP method based on deep convolutional neural networks.

4) Methods Trained on Two Datasets: These methods are trained in two stages, including a larger dataset-SALICON for first stage training and a smaller dataset for second training. These methods are used to compare the performance when using the augmented training dataset. The comparison methods trained on MIT1003 dataset include JN [59] and DeepFix [20] methods. JN proposed two different VAP designs, including a shallow convent trained from scratch, and another deeper solution with the first three layers adapted from another network of trained classifications. DeepFix achieves the best score with respect to most evaluation metrics on several challenging benchmark datasets.

On one hand, the qualitative visual comparisons on MIT1003 dataset are shown in Fig. 5. Considering that some codes are not public, only parts of visual attention maps are generated using their available codes with recommended parameters settings. Three kind of stimuli images are selected in Fig. 5 including images with high-contrast object (color, intensity, texture, etc.): Fig. 5(a)-(c), images with high-semantic object (face, head, eye, etc.): Fig. 5(d)-(f), and images in complex scenes (both objects are included): Fig. 5(g)-(j). As can be seen, the eDN method achieves the worst result because of its poor semantic features for limited training data and scarce contrast features. For images with high-contrast objects, it can be seen that all methods except for eDN can highlight the regions which attract human attention roughly. This is because these methods utilize low-level features, which contains local contrast information. Note that the results based on the low-level contrast features from the proposed contrast feature extraction block are quite satisfied. For images with
high-semantic objects, the traditional methods based on local contrast are not applicable any more. The methods based learning show better results, which is because these methods can learn abundant semantic information layer by layer. For images in complex scenes, we find that the results of the proposed method are excellent. This is because the proposed method can effectively take advantage of low-level contrast features and high-level semantic features based on the unique physiological structure of humans eyes and human prior knowledge. Overall, the proposed method can surpass other listed methods in any case, especially for the images in complex scenes (Fig. 5 (j)-(g)). In addition, some visual attention maps on the other three testing datasets using the proposed method are given in Fig. 6. As it can be seen, the generated visual attention maps using the proposed method are also close to the groundtruth density maps on OSIE, TORONTO and PASCAL-S datasets. Therefore, the results in Fig. 7 can further illustrate the good performance of the proposed method.

On the other hand, we also compare our results with several state-of-the-art methods qualitatively on OSIE, TORONTO, PASCAL-S and MIT1003 datasets. Since most VAP methods based on deep learning are not publicly available and the reported results are not based on all considered datasets, the comparison methods are different depending on the dataset. Tables II-IV show the results on OSIE, TORONTO, PASCAL-S and MIT1003 datasets, respectively. As can be seen evidently in these tables, the proposed method surpasses other methods by a substantial margin with respect to most evaluation metrics, especially in s-AUC. The high s-AUC value shows that
the proposed method can accurately locate gaze positions of human eyes for the query image. Specifically, on the TORONTO dataset (see Table II), the proposed method achieves state-of-the-art results. On OSIE and PASCAL-S datasets, the proposed method can also struggle to compete with the most popular DeepFix method. A little imperfection is that the CC of the proposed method is slightly lower than the most popular DeepFix method on OSIE dataset (see Table I). This is because the DeepFix method adopts a more complicated two-stage training manner. In general, the proposed method can achieve promising results, and we mainly attribute the good performance of the proposed method to the simultaneous use of both high-level semantic and low-level contrast features.

### Table I: The Quantitative Comparison on OSIE Dataset

| Method          | CC↑ | s-AUC↑ | AUC-Judd↑ | AUC-Borji↑ | NSS↑ | EMD↓ |
|-----------------|-----|--------|-----------|------------|------|------|
| AWS [56]        | 0.45 | 0.76 | 0.82 | 0.81 | 1.45 | 1.93 |
| GBVS [27]       | 0.44 | 0.68 | 0.80 | 0.80 | 1.55 | 1.67 |
| BMS [57]        | 0.46 | 0.76 | 0.83 | 0.82 | 1.47 | 1.89 |
| eDN [14]        | 0.40 | 0.68 | 0.82 | 0.82 | 1.16 | 2.02 |
| DeepFix [20]    | 0.80 | 0.79 | 0.91 | 0.93 | **3.04** | 1.04 |
| SAM-VGG [36]    | 0.78 | 0.70 | 0.91 | 0.80 | 2.74 | 1.12 |
| Proposed        | 0.77 | 0.84 | 0.92 | **0.87** | 2.87 | **0.67** |

| Method          | CC↑ | s-AUC↑ | AUC-Judd↑ | AUC-Borji↑ | NSS↑ | EMD↓ |
|-----------------|-----|--------|-----------|------------|------|------|
| ITTI [16]       | 0.48 | 0.65 | 0.80 | 0.80 | 1.30 |
| GBVS [27]       | 0.57 | 0.64 | 0.83 | 0.83 | 1.52 |
| JUDD [15]       | 0.41 | 0.67 | 0.78 | 0.77 | 1.15 |
| CAS [30]        | 0.45 | 0.69 | 0.78 | 0.78 | 1.27 |
| eDN [14]        | 0.50 | 0.62 | 0.85 | 0.84 | 1.25 |
| Mr-CNN [55]     | 0.49 | 0.71 | 0.80 | 0.79 | 1.41 |
| SAM-VGG [36]    | 0.72 | 0.68 | 0.89 | 0.82 | 1.97 |
| SAM-Resnet [56] | 0.73 | 0.69 | 0.89 | 0.82 | 2.11 |
| DVA [22]        | 0.72 | 0.76 | 0.86 | **0.86** | 2.12 |
| Proposed        | 0.74 | 0.77 | 0.90 | 0.84 | **2.39** |

### Table II: The Quantitative Comparison on Toronto Dataset

| Method          | CC↑ | s-AUC↑ | AUC-Judd↑ | AUC-Borji↑ | NSS↑ |
|-----------------|-----|--------|-----------|------------|------|
| Proposed        | 0.77 | 0.84 | 0.92 | 0.87 | 2.87 | 0.67 |

### Table III: The Quantitative Comparison on PASCAL-S Dataset

| Method          | CC↑ | s-AUC↑ | AUC-Judd↑ | AUC-Borji↑ | NSS↑ | EMD↓ |
|-----------------|-----|--------|-----------|------------|------|------|
| GBVS [27]       | 0.51 | 0.65 | 0.84 | 0.82 | 1.36 | 1.16 |
| BMS [57]        | 0.44 | 0.67 | 0.80 | 0.78 | 1.28 | 1.32 |
| eDN [14]        | 0.55 | 0.65 | 0.89 | **0.87** | 1.42 | 1.29 |
| JUDD [15]       | 0.68 | 0.69 | 0.88 | 0.86 | 1.90 | 1.04 |
| DeepFix [20]    | 0.78 | 0.73 | 0.91 | 0.82 | 2.60 | 0.54 |
| SU [58]         | 0.69 | 0.72 | 0.89 | 0.81 | 2.22 | 0.73 |
| SAM-VGG [36]    | 0.74 | 0.68 | 0.90 | 0.80 | 2.56 | 0.98 |
| SAM-Resnet [56] | 0.75 | 0.68 | 0.91 | 0.80 | 2.70 | 0.86 |
| DVA [22]        | 0.66 | 0.77 | 0.89 | 0.85 | 2.26 | – |
| Proposed        | 0.74 | 0.81 | 0.93 | 0.85 | **2.87** | **0.53** |

### Table IV: The Quantitative Comparison on MIT1003 Dataset

| Method          | CC↑ | s-AUC↑ | AUC-Judd↑ | AUC-Borji↑ | NSS↑ |
|-----------------|-----|--------|-----------|------------|------|
| ITTI [16]       | 0.33 | 0.66 | 0.77 | 0.76 | 1.10 |
| GBVS [27]       | 0.42 | 0.66 | 0.83 | 0.81 | 1.38 |
| JUDD [15]       | 0.30 | 0.68 | 0.76 | 0.74 | 1.02 |
| BMS [57]        | 0.36 | 0.69 | 0.79 | 0.76 | 1.25 |
| Mr-CNN [55]     | 0.38 | 0.73 | 0.80 | 0.77 | 1.36 |
| DeepFix [20]    | 0.72 | 0.74 | 0.90 | 0.87 | 2.58 |
| SAM-VGG [36]    | 0.76 | 0.61 | 0.91 | 0.85 | 2.85 |
| SAM-Resnet [56] | 0.77 | 0.62 | 0.91 | 0.86 | **2.89** |
| DVA [22]        | 0.64 | 0.77 | 0.87 | 0.85 | 2.38 |
| Proposed        | 0.73 | 0.79 | 0.92 | 0.88 | 2.54 |

### E. Ablation Analysis

In this subsection, we analyze the effectiveness of each component for the proposed method. We perform the experiment with different design options and give an ablation analysis for each component. More specifically, as is shown in Table V, seven different variations are constructed: the fusion feature of F1 and F2 without the contrast feature extraction block (NCF) method, the Contrast Feature (CF) method via combining the contrast features F1 and F2 with the Contrast Feature Extraction Block directly, the Semantic Feature (SF) method by merging the semantic features F3, F4 and F5 directly, the Directly Combining Feature (DCF) method by concatenating all of contrast features and semantic features directly, the Densely Combining Feature (DenCF) method using the designed densely connected connected block to combine all features, the Densely Combining Feature with the learnable Center-Bias Prior (DenCF+CBP) method, and the final version of the proposed method by adding the designed reduction-attention block to the DenCF method. Each component of the proposed method is analyzed in the following aspects.

1) **Low-level Contrast Features**: To validate the contribution of the extracted low-level contrast features, we compare the three methods CF, SF and DCF. As is shown in Table IV, the DCF method outperformed the SF method in all listed metrics. This is because the low-level contrast features are considered in the DCF method compared to the SF method. Therefore, we can conclude that the low-level contrast features do contribute to VAP, and the low-level contrast features are complementary to the high-level semantic features by a substantial margin.

2) **Combination Strategy**: To consider the advantages of multiple features, these features should be exploited simultaneously. Here, a densely connected manner is adopted to well combine these features. This combination strategy can further excavate the feature representation of each level and utilize the advantage of multiple features simultaneously. To further demonstrate the effectiveness of our combination strategy, we compared the performance of the proposed method with directly combined feature method. In Table IV, we can clearly see that the DenCF method is better than the DCF method, which manifests the effectiveness of the densely strategy.

3) **Designed Centre-bias Prior**: A learnable centre-bias block is designed for generating a data-dependent centre-bias prior map, which can compensate for the centre-bias when human observing images and improve the predicted...
Timing the final visual attention map. NSS metrics, which manifests the proposed contrast feature CF method is better than NCF method on the CC, AUC and this block (NCF method in Table V) to demonstrate its experiment with this block (CF in Table V) and without automatical extraction in a deep network. We performed the feature extraction block is proposed for low-level contrast features respectively, which proves the effectiveness of the reduction-Attention Block on TORONTO dataset in terms of s-AUC, NSS and CC.

\[ \Delta \text{s-AUC} = -0.027, \Delta \text{NSS} = 0.036, \Delta \text{CC} = 0.023 \]

Obviously, the proposed method adds an improvement of 0.027, 0.036 and 0.023 on TORONTO dataset in terms of s-AUC, NSS and CC, respectively. What's more, the performance of the proposed model is similar to the ITTI method, which may be because both the proposed method, which is entirely based on the deep semantic features.

Reduction-Attention Block

The reduction-attention block is devised to adaptively recalibrate channel-wise feature responses during bio-inspired representation learning. We experiment with this block (Proposed in Table V) and without this block (DenCF+CGBP method in Table V) to demonstrate its effectiveness. Obviously, the proposed method adds an improvement of 0.027, 0.036 and 0.023.

On TORONTO dataset in terms of s-AUC, NSS and CC, respectively, which proves the effectiveness of the reduction-attention block.

5) Contrast Feature Extraction Block: The contrast feature extraction block is proposed for low-level contrast features automatical extraction in a deep network. We performed the experiment with this block (CF in Table V) and without this block (NCF method in Table V) to demonstrate its effectiveness. As it can be seen, the performance obtained by CF method is better than NCF method on the CC, AUC and NSS metrics, which manifests the proposed contrast feature extraction block has a profound effect on the generation of the final visual attention map.

F. Timing

According to [60], the time complexity of each convolutional and pooling layer can be computed as \( O(M^2 \cdot K^2 \cdot C_{in} \cdot C_{out}) \), where \( M \) denotes the size of the output feature map, \( K \) represents the size of the convolutional kernel, \( C_{in} \) is the channel number of the input feature map, and \( C_{out} \) indicates the channel number of the output feature map. As for the fully connected layer, it can be considered as a special convolutional layer, in which the size of the output feature map and the convolutional kernel are \( 1 \). As a result, the time complexity of fully connected layer can be computed as \( O(C_{in} \cdot C_{out}) \). With regard to the element-wise multiplication in reduction-attention block, its time complexity is \( O(M^2) \). Considering that other operation (bias operation, concatenation operation, etc.) can be ignored, the time complexity of proposed network is \( O(M^2 \cdot K^2 \cdot C_{in} \cdot C_{out}) + O(C_{in} \cdot C_{out}) + O(M^2) \).

In addition, according to [22], a summary of these methods is provided in Table VI. As visible, most of existing VAP methods are off-line training or based on deep learning framework. Since some codes of deep learning methods are not public, we only report the inferring speed performance of DVA, SalNet, MR-DNN, and eDN with other non-deep learning methods. Obviously, the inferring speed of the proposed method can struggle to compete with DVA and SalNet, although the proposed method adopts a slightly smaller input size than DVA and SalNet.

G. Failure Cases Analysis

Some failure predictions of the proposed method have been shown in Fig. 7. In general, these failure cases can be divided into two categories: 1) the images without the high-contrast and high-semantics information (Fig. 7 (a)-(b)); and 2) the synthetic images (Fig. 7(c)-(f)). As for the images without the high-contrast and high-semantics information, our method can not perform well maybe because these images have no obvious saliency aspects to be predicted. In addition, the proposed method is difficult to process synthetic images, which may be because the amount of synthetic image is small in the training dataset and the synthetic image contains little semantic information. Although the proposed method performs poorly for the synthetic image, it can still outperform the DVA method, which is entirely based on the deep semantic features. What’s more, the performance of the proposed model is similar to the ITTI method, which may be because both the proposed method and the ITTI method consider the low-level contrast features.

| Dataset     | Method          | s-AUC\( ^\Delta \) | \( \Delta \text{s-AUC} \) | NSS\( ^\Delta \) | \( \Delta \text{NSS} \) | CC\( ^\Delta \) | \( \Delta \text{CC} \) |
|-------------|-----------------|--------------------|------------------------|----------------|----------------|----------------|----------------|
| OSIE        | NCF             | 0.701              | -0.145                 | 1.212         | -1.881         | 0.492         | -0.280         |
|             | DF              | 0.734              | -0.110                 | 1.953         | -0.920         | 0.594         | -0.178         |
|             | SF              | 0.796              | -0.048                 | 2.494         | -0.379         | 0.712         | -0.060         |
|             | DenCF           | 0.812              | -0.032                 | 2.696         | -0.177         | 0.734         | -0.029         |
|             | DenCF+CGBP      | 0.834              | -0.010                 | 2.810         | -0.063         | 0.746         | 0.026          |
|             | Proposed        | 0.820              | -0.024                 | 2.851         | -0.022         | 0.761         | -0.011         |

| MIT1003     | NCF             | 0.623              | -0.144                 | 1.376         | -0.106         | 0.441         | -0.302         |
|             | DF              | 0.652              | -0.114                 | 1.905         | -0.487         | 0.566         | -0.177         |
|             | SF              | 0.716              | -0.050                 | 2.149         | -0.243         | 0.682         | -0.062         |
|             | DenCF           | 0.732              | -0.034                 | 2.301         | -0.091         | 0.717         | -0.026         |
|             | DenCF+CGBP      | 0.746              | -0.020                 | 2.371         | -0.021         | 0.729         | -0.014         |
|             | Proposed        | 0.739              | -0.027                 | 2.356         | -0.036         | 0.720         | -0.023         |

| PASCAL-S    | NCF             | 0.692              | -0.120                 | 1.386         | -1.481         | 0.432         | -0.310         |
|             | DF              | 0.710              | -0.102                 | 2.016         | -0.851         | 0.602         | -0.140         |
|             | SF              | 0.784              | -0.028                 | 2.595         | -0.272         | 0.689         | -0.053         |
|             | DenCF           | 0.791              | -0.021                 | 2.712         | -0.155         | 0.706         | -0.036         |
|             | DenCF+CGBP      | 0.801              | -0.011                 | 2.785         | -0.082         | 0.715         | -0.027         |
|             | Proposed        | 0.812              | -0.017                 | 2.813         | -0.054         | 0.719         | -0.023         |

| MIT1003     | NCF             | 0.685              | -0.159                 | 1.056         | -1.485         | 0.452         | 0.234         |
|             | DF              | 0.702              | -0.092                 | 1.752         | -0.789         | 0.594         | -0.139         |
|             | SF              | 0.717              | -0.077                 | 2.251         | -0.290         | 0.659         | -0.074         |
|             | DenCF           | 0.735              | -0.059                 | 2.453         | -0.088         | 0.681         | -0.052         |
|             | DenCF+CGBP      | 0.758              | -0.031                 | 2.510         | -0.031         | 0.706         | -0.027         |
|             | Proposed        | 0.7694             | -0.038                 | 2.532         | -0.009         | 0.712         | -0.021         |

| Method      | Input Size      | Training | Deep Learning | Runtime |
|-------------|-----------------|----------|---------------|---------|
| ITTI        | full size       | No       | No            | 4s      |
| GBVS        | full size       | No       | No            | 2s      |
| CAS         | \( \max \{w,h\} \times 250 \) | No | No | 16s |
| BMS         | \( w=600 \) | No | No | 0.3s |
| JUDD | \( 200 \times 200 \) | Yes | Yes | 10s |
| Mr-CNN      | \( 400 \times 400 \) | Yes | Yes | 14s* |
| SalNet      | \( 320 \times 240 \) | Yes | Yes | 0.1s* |
| DV A        | \( \max \{w,h\} \times 256 \) | Yes | Yes | 0.1s* |
| Proposed    | \( 224 \times 224 \) | Yes | Yes | 0.1s* |

* Runtime with GPU.
In this paper, an intuitive method has been proposed to generate visual attention map via bio-inspired representation learning. To perform the Visual Attention Prediction (VAP), the proposed method extracts the low-level contrast features automatically in a deep network, then combines them with the high-level semantic features via bio-inspired representation learning, and finally generates the ultimate visual attention map based on the obtained features after bio-inspired representation learning. Experiment results on MIT1003, OSIE, TORONTO and PASCAL-S datasets demonstrate that the designed contrast feature extraction block which is proposed to learn low-level contrast features in a deep network is effective. Moreover, other components in the proposed method are proved to be effective. In addition, the proposed method achieves the superior performance compared with other state-of-the-art methods.

REFERENCES

[1] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, “Simultaneous spectral-spatial feature selection and extraction for hyperspectral images,” IEEE Transactions on Cybernetics, vol. 48, no. 1, pp. 16–28, 2016.
[2] J. Yu, X. Yang, F. Gao, and D. Tao, “Deep multimodal distance metric learning using cliff constraints for image ranking,” IEEE Transactions on Cybernetics, vol. 47, no. 12, pp. 4014–4024, 2017.
[3] J. Han, X. Ji, X. Hu, D. Zhu, K. Li, X. Jiang, G. Cui, L. Guo, and T. Liu, “Representing and retrieving video shots in human-centric brain imaging space,” IEEE Transactions on Image Processing, vol. 22, no. 7, pp. 2723–2736, 2013.
[4] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, “Two-stage learning to predict human eye fixations via SDAEs,” IEEE Transactions on Cybernetics, vol. 2, no. 46, pp. 487–498, 2016.
[5] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, “Hyperspectral image unsupervised classification by robust manifold matrix factorization,” Information Sciences, vol. 485, pp. 154–169, 2019.
[6] D. Zhang, J. Han, L. Jiang, S. Ye, and X. Chang, “Revealing event saliency in unconstrained video collection,” IEEE Transactions on Image Processing, vol. 26, no. 4, pp. 1746–1758, 2017.
[7] D. Bo and L. Zhang, “A discriminative metric learning based anomaly detection method,” IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 11, pp. 6844–6857, 2014.
[8] X. Yao, J. Han, D. Zhang, and F. Nie, “Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3106–3209, 2017.
[9] X. Lu, X. Li, and L. Mou, “Semi-supervised multitask learning for scene recognition,” IEEE Transactions on Cybernetics, vol. 45, no. 9, pp. 1967–1976, 2017.
[10] L. Zhang and Y. Zhang, “Airport detection and aircraft recognition based on two-layer saliency model in high spatial resolution remote-sensing images,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 4, pp. 1511–1524, 2017.
[11] D. Bo and L. Zhang, “Target detection based on a dynamic subspace,” Pattern Recognition, vol. 47, no. 1, pp. 344–358, 2014.
[12] P. Zhang, T. Zhuo, W. Huang, K. Chen, and M. Kankanhalli, “Online object tracking based on CNN with spatial-temporal saliency guided sampling,” Neurocomputing, vol. 257, pp. 115–127, 2017.
[13] S. Jia and Y. Zhang, “Saliency-based deep convolutional neural network for no-reference image quality assessment,” Multimedia Tools and Applications, vol. 77, no. 12, pp. 14859–14872, 2018.
[14] H. Hadizadeh and I. V. Bajic, “Saliency-aware video compression,” IEEE Transactions on Image Processing, vol. 23, no. 1, pp. 19–33, 2013.
[15] T. Judd, K. Ehinger, F. Durand, and A. Torralba, “Learning to predict where humans look,” in IEEE International Conference on Computer Vision, 2010, pp. 2106–2113.
[16] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pp. 1254–1259, 2002.
[17] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu, “Global contrast based region detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 409–416.
[18] L. Itti and A. Borji, “Exploiting local and global patch rarities for saliency detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 478–485.
[19] M. Kummerer, T. A. Wallis, L. A. Gatys, and M. Bethge, “Understanding low- and high-level contributions to fixation prediction,” in IEEE International Conference on Computer Vision, 2017, pp. 4799–4808.
[20] K. Sss, K. Ayush, and R. V. Babu, “DeepFix: A fully convolutional neural network for predicting human eye fixations,” IEEE Transactions on Image Processing, vol. 26, no. 9, pp. 4446–4456, 2017.
[21] Y. Zhong, A. Ma, Y. S. Ong, Z. Zhu, and L. Zhang, “Computational intelligence in optical remote sensing image processing,” Applied Soft Computing, vol. 64, pp. 75–93, 2017.
[22] W. Wang and J. Shen, “Deep visual attention prediction,” IEEE Transactions on Image Processing, vol. 27, no. 5, pp. 2368–2378, 2018.
[23] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, “Deeply supervised salient object detection with short connections,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5300–5309.
[24] C. Zhao, X. Gao, W. J. Emery, Y. Wang, and J. Li, “An integrated framework for the spatio-temporal-spectral fusion of remote sensing images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 12, pp. 7135–7148, 2016.
[25] O. L. Meur, P. L. Callet, D. Barba, and D. Thoreau, “A coherent computational approach to model bottom-up visual attention,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 802–817, 2006.
[26] N. Bruce and J. Tsotsos, “Attention based on information maximization,” Journal of Vision, vol. 7, no. 9, pp. 950, 2010.
[27] B. Scholkopf, J. Platt, and T. Hofmann, “Graph-based visual saliency,” in International Conference on Neural Information Processing Systems, 2006, pp. 545–552.
[28] X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” in IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[29] D. Gao and N. Vasconcelos, “Decision-theoretic saliency: Computational principles, biological plausibility, and implications for neurophysiology and psychophysics,” Neural Computation, vol. 21, no. 1, pp. 239–271, 2009.
[30] S. Goferman, L. Zelnik-Manor, and A. Tal, “Context-aware saliency detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2376–2383.
[31] D. Hu, X. Lu, and X. Li, “Multimodal learning via exploring deep semantic similarity,” in Proceedings of the 24th ACM international conference on Multimedia, 2016, pp. 342–346.
[32] L. Zhang, L. Zhang, D. Tao, and X. Huang, “Tensor discriminative locality alignment for hyperspectral image spectral-spatial feature extraction,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 242–256, 2013.

[33] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, “Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding,” *Pattern Recognition*, vol. 48, no. 10, pp. 3102–3112, 2015.

[34] E. Vig, M. Dorr, and D. Cox, “Large-scale optimization of hierarchical features for saliency prediction in natural images,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2798–2805.

[35] M. Kummerer, L. Theis, and M. Bethge, “Deep gaze I: Boosting saliency prediction with feature maps trained on imagenet,” in *International Conference on Learning Representations Workshop*, 2015.

[36] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, “Predicting human eye fixations via an LSTM-based saliency attentive model,” *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, 2018.

[37] M. Liang and X. Hu, “Predicting eye fixations with higher-level visual features.” *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1178–1189, 2015.

[38] R. Cong, J. Lei, H. Fu, Q. Huang, X. Cao, and C. Hou, “Co-saliency detection for rgbd images based on multi-constraint feature matching and cross label propagation,” *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 568–579, 2017.

[39] W. Qiu, X. Gao, and B. Han, “Eye fixation assisted video saliency detection via total variation-based pairwise interaction,” *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4724–4739, 2018.

[40] R. Cong, J. Lei, H. Fu, W. Lin, Q. Huang, X. Cao, and C. Hou, “An iterative co-saliency framework for RGBD images,” *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 233–246, 2017.

[41] J. Han, C. Hao, N. Liu, C. Yan, and X. Li, “CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion,” *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3171–3183, 2018.

[42] X. Wang, Y. Zhong, X. Yao, L. Zhang, and Y. Xu, “Saliency-based endmember detection for hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 7, pp. 3667–3680, 2018.

[43] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.

[44] T. Zhao, B. Zhang, M. He, W. Zhang, N. Zhou, J. Yu, and J. Fan, “Embedding visual hierarchy with deep networks for large-scale visual recognition,” *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4740–4755, 2018.

[45] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[46] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T. S. Chua, “SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6298–6306.

[47] F. Wang, Z. Wang, J. Liang, and X. Liu, “Resilient state estimation for 2-D time-varying systems with redundant channels: A variance-constrained approach,” *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2479–2489, 2018.

[48] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2261–2269.

[49] M. Jiang, S. Huang, J. Duan, and Q. Zhao, “SALICON: Saliency in context,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1072–1080.

[50] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao, “Predicting human gaze beyond pixels,” *Journal of Vision*, vol. 14, no. 1, p. 28, 2014.

[51] N. D. B. Bruce and J. K. Tsotsos, “Saliency based on information maximization,” in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, 2005, pp. 155–162.

[52] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, “The secrets of salient object segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 280–287.

[53] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *European Conference on Computer Vision*, 2014, pp. 740–755.

[54] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit, “Saliency and human fixations: State-of-the-art and study of comparison metrics,” in *IEEE International Conference on Computer Vision*, 2014, pp. 1153–1160.

[55] N. Liu, J. Han, T. Liu, and X. Li, “Learning to predict eye fixations via multiresolution convolutional neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 392–404, 2018.

[56] A. Garcia-Diaz, V. Leboran, X. R. Fdez-Vidal, and X. M. Pardo, “On the relationship between optical variability, visual saliency, and eye fixations: A computational approach,” *Journal of Vision*, vol. 12, no. 7, p. 17, 2012.

[57] J. Zhang and S. Sclaroff, “Saliency detection: A boolean map approach,” in *IEEE International Conference on Computer Vision*, 2014, pp. 153–160.

[58] S. S. S. Kruthiventi, V. Gudisa, J. H. Dhulakia, and R. V. Babu, “Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5781–5790.

[59] J. Pan, E. Sayrol, X. Gironiato, K. Mcguinness, and N. E. Oconnor, “Shallow and deep convolutional networks for saliency prediction,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 598–606.

[60] K. He and S. Jian, “Convolutional neural networks at constrained time cost,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5353–5360.