Knowledge-Based Systems 284 (2024) 111279

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Global semantic-guided network for saliency prediction

Jiawei Xie [a], Zhi Liu [a,b,*], Gongyang Li [a,b], Xiaofeng Lu [a,b], Tao Chen [c,d]

[a] Shanghai Institute for Advanced Communication and Data Science, School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
[b] Wenzhou Institute of Shanghai University, Wenzhou 325000, China
[c] Big Data Research Lab, University of Waterloo, Ontario, Canada
[d] Niacin (Shanghai) Technology Co. Ltd., Shanghai, China

ARTICLE INFO

Keywords:
Saliency prediction
Channel interaction
Semantic information
Feature fusion
Transformer

ABSTRACT

The human visual system effectively analyzes scenes based on local, global and semantic properties. Deep learning-based saliency prediction models adopted two-stream networks, leveraged prior knowledge of global semantics, or added long-range dependency modeling structures like transformers to incorporate global saliency information. However, they either brought high complexity to learning local and global features or neglected the design for enhancing local features. In this paper, we propose a Global Semantic-Guided Network (GSGNet), which first enriches global semantics through a modified transformer block and then incorporates semantic information into visual features from local and global perspectives in an efficient way. Multi-head self-attention in transformers captures global features, but lacks information communication within and between feature subspaces (heads) when computing the similarity matrix. To learn global representations and enhance interactions of the subspaces, we propose a Channel-Squeeze Spatial Attention (CSSA) module to emphasize channel-relevant information in a compression manner and learn global spatial relationships. To better fuse local and global contextual information, we propose a hybrid CNN-Transformer block called local–global fusion block (LGFB) for aggregating semantic features simply and efficiently. Experimental results on four public datasets demonstrate that our model achieves compelling performance compared with the state-of-the-art saliency prediction models on various evaluation metrics.

1. Introduction

Human visual attention mechanism guides humans to observe scenes selectively and ignore less informative regions, enabling humans to analyze complex and diverse scenes quickly. Accordingly, in the field of computer vision, researchers attempt to understand human attention mechanisms and construct models to imitate this behavior, proposing a saliency prediction (SP) task (*i.e.*, fixation prediction task). This task usually refers to predicting human eye fixation locations for modeling this mechanism. Many applications including object segmentation [1–3], object detection [4], video coding [5] and driver focus of attention [6] benefit from this task.

When humans freely view images, their attention is influenced by local, global and semantic information within the visual stimulus. Early SP models [7–10] leverage neurobiological and psychological knowledge to extract low-level features, including local contrast, color, luminance and texture, to predict fixations in natural images. Some researchers [11–13] have discovered that visual saliency is context-sensitive and salient regions are highlighted globally over the entire scene, such as faces, cars and texts. Combining both local and global saliency information enhances SP task performance [14]. However, traditional models have limited generalization capabilities and are not able to handle complex scenes.

With the prevalence of deep learning and the availability of saliency datasets [15,16], many SP models [17–19] have adopted deep Convolutional Neural Network (CNN)-based architectures to capture local saliency information, thereby boosting the performance of the SP task. Rich high-level semantic features extracted by the deep networks help saliency models to better mimic human viewing patterns [20]. As for global contextual information, it can be acquired through various methods. One approach is to incorporate Convolutional Long Short-Term Memory (ConvLSTM) into the model structure. Additionally, global information acquisition can be facilitated through the use of a global saliency model [21], but overall, the method is not an end-to-end training paradigm. Another method involves capturing global semantic information in a two-stream manner. For ConvLSTM-based models

[*] Corresponding author at: Shanghai Institute for Advanced Communication and Data Science, School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China.
E-mail address: liuzhisjtu@163.com (Z. Liu).

https://doi.org/10.1016/j.knosys.2023.111279
Received 30 May 2023; Received in revised form 10 September 2023; Accepted 4 December 2023
Available online 5 December 2023

[22,23], global information is captured by the long-range modeling architecture, but local saliency features are not enhanced comparatively. In two-stream models [24,25], images are commonly downsampled to a lower resolution, resulting in a partial loss of visual information, in order to learn global saliency features. Recently, transformers [26] have achieved significant success in the field of computer vision [27–29]. Due to their strong ability to model long-range dependency, they also have been introduced into the SP model [30]. Multi-head self-attention (MHSA) in transformer blocks calculates the global spatial similarity matrix of feature vectors in multiple subspaces (heads). However, during the matrix computation, the lack of inter-channel interactions within each head and among heads may limit the representation of relevant relationships among channels. Attention modules like spatial attention [31–33] learn the spatial global information by compressing channels while maintaining the spatial dimension.

Based on the above observations on the drawbacks of existing traditional and deep learning-based SP models, we propose a global semantic-guided network (GSGNet) that refines both local and global saliency features in multi-level visual features, with the aid of global semantic information. We aim to integrate semantic information into features from local and global perspectives using hybrid blocks of convolutional modules and transformers. Due to the inherent characteristics of those two structures, both local and global features are easily and efficiently learned. To add relevant information regarding feature channels in the MHSA, our key idea is to compress the information conveyed by channels to represent overall characteristic in each subspace through effective channel communication and then the global spatial features can be further refined. In this way, the global information in the channel domain is taken into account in the similarity matrix to help highlight global saliency information. Therefore, we propose a Channel-Squeezed Spatial Attention (CSSA) module to better exchange information among channels and effectively capture global features in the spatial domain. A parallel CSSA-based block called spatial attention inception block (SAIB) is employed to enrich the global semantic features that are from the deepest stage of a CNN backbone, which provides semantic information that serves as a guide for the subsequent learning of local and global saliency features. To facilitate both local and global feature learning, a local–global fusion block (LGFB) is designed to effectively and straightforwardly integrate features from other backbone stages with the refined semantic features. We leverage the intrinsic properties of CNN and transformer to build a hybrid block by creating two branches, in which the CNN branch emphasizes local informative details, while the other employs the proposed CSSA module for global feature integration. Furthermore, we evaluate our GSGNet on four widely used eye-tracking datasets. The experimental results demonstrate that our model achieves compelling performance compared with other existing SP models. Our code is available at https://github.com/oraclefina/GSGNet. In summary, the main contributions of our work are detailed as follows:

- We propose a global semantic-guided network (GSGNet) to incorporate global semantic information as a guide to refine multi-level features and model human fixations based on both local and global semantic features.
- We propose a channel-squeeze spatial attention (CSSA) module, which facilitates the exchange of channel information and enables the successive global feature representation through sequential channel and spatial interactions.
- We propose a local–global fusion block (LGFB) that combines the merits of CNN and transformer to enrich local and global information for visual features at different levels.

2. Related work

In this section, we briefly review the related work on saliency prediction and vision transformer.

2.1. Saliency prediction

Traditional models [7,34,35] for saliency prediction primarily depend on hand-crafted features. Early models extracted low-level features such as color, contrast and orientation, to generate saliency maps and integrated them based on feature integration theory (FIT) [36]. Subsequent models attempted to incorporate more sophisticated features, like texts, human faces and gaze direction, to predict salient regions in natural images. While these traditional models have achieved reasonable performance on the SP task, their reliance on hand-crafted features limits their ability to effectively handle complex and diverse scenes, which reflects the weak generalization capability of these models.

Recently, the development of saliency prediction has been advanced by deep learning models. Vig et al. [37] proposed a deep convolutional neural network called eDN, which is capable of automatically extracting visual features and fusing them to generate saliency maps. Later, Kümmerer et al. [17] made the initial attempt to apply transfer learning for saliency prediction. The final prediction was computed by selecting and combining multi-scale features from various layers of AlexNet [38]. The model leveraged the knowledge of image classification tasks and achieved a significant improvement. Since then, many works have employed CNN backbones pre-trained on ImageNet [39] as feature extractors and proposed various methods to utilize and enhance the semantic information extracted from these backbones.

Due to the hierarchical architecture, CNN extractors can automatically capture multi-level features from the shallower layers to the deeper layers. The integration of features at multiple levels is a crucial strategy in designing networks for various vision tasks [40–42]. Hu et al. [43] constructed a feature pyramid that includes a spatial attenuation module to effectively combine and enhance multi-level features, capturing local and global context within, around and beyond the salient objects. Multi-level features are also essential for SP models. Liu et al. [44] proposed a novel multi-resolution CNN model to learn bottom-up visual saliency and top-down visual features. Jia et al. [45] employed a two-stage training approach and leveraged full-level features from various CNN backbones (*i.e.*, DenseNet [46] and NasNet [47]). Reddy et al. [48] have decomposed the design of SP models into four key components, namely input features, multi-level integration, readout architectures and loss functions. For multi-level integration, a UNet-like structure was proposed to incorporate multi-level features, refining low-level and high-level features hierarchically. Che et al. [25] designed a modified U-Net structure with a novel cross-scale short connection module to learn multi-scale features. Cornia et al. [18] extracted multi-level features from the last three stages of VGGNet-16 [49] and fed them into an encoder network to obtain the prediction. Their experiments demonstrate that the high-level features make a significant contribution to the final results. Wang et al. [50] built three decoders with varying receptive field sizes to transfer multi-level saliency information from the last three stages of the backbone into three saliency maps which were then fused to obtain the fixation map. Ning et al. [51] enhanced spatial and temporal information od multi-level features to generate motion-aware maps to better predict dynamic saliency in an audio–visual scene. Yang et al. [19] employed a dilated inception network and high-level features to capture contextual information at multiple scales. Lai et al. [52] leveraged knowledge from biological vision and utilized multi-level features from spatial visual semantics, object-level semantics and a conditional center prior, to generate saliency maps.

The performance of the above models indicates that rich semantic information encoded in high-level features is beneficial for the SP task, which makes the model understand the scene information of natural images. The integration of contextual information at multiple levels also plays a crucial role in the task. In this paper, we aim to enhance the utilization of the high-level semantic features and leverage them to guide the fusion of multi-level features. We adjust the spatial or channel dimension of the semantic features to participate in local and global fusion in the proposed LGFB.

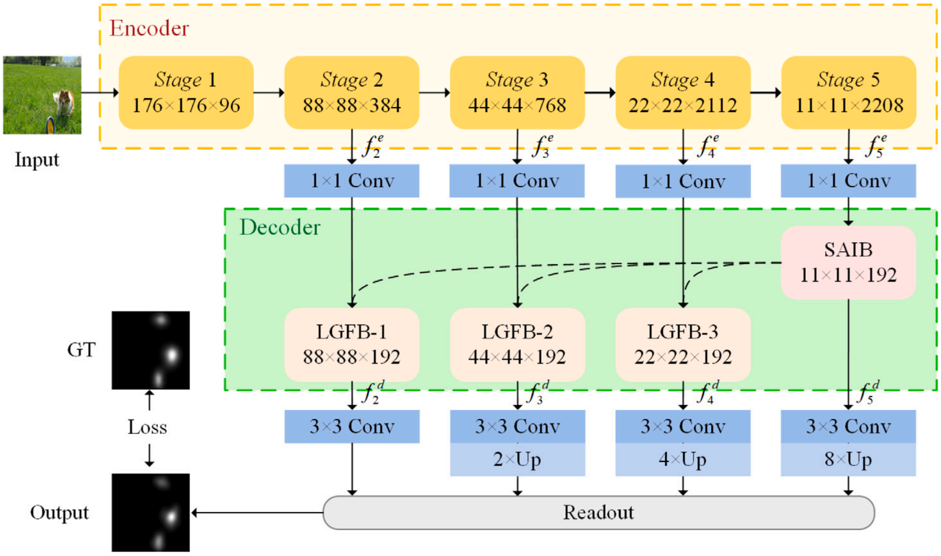

Fig. 1. The architecture of our proposed GSGNet. Our model takes an RGB image as the input. The encoder captures visual features and learns multi-level representations f_2^e, f_3^e, f_4^e and f_5^e. Then the channel dimension of learned features is reduced by convolutional layers. High-level features f_5^e are enhanced by a spatial attention inception block (SAIB) and then local–global fusion blocks integrate features under the guidance of enhanced high-level features. f_3^d, f_4^d and f_5^d are upsampled to the same spatial size as f_2^d. Finally, the saliency map is obtained by a readout module.

2.2. Vision transformer

Transformer has achieved great success in the domains of natural language processing (NLP) [53] and computer vision, such as image classification [27] and object tracking [54], *etc*. CNNs exhibit an inductive bias towards acquiring local visual representations, whereas transformers possess a remarkable capacity to capture long-range dependencies due to their self-attention mechanism. In addition, many studies have attempted to improve the effectiveness and efficiency of vision transformers through various operations on query and key–value features to learn multi-scale features [55–57]. Wang et al. [58] designed spatial-reduction attention to decrease the spatial dimension of the keys and values. This approach reduces resource consumption and makes the model flexible to learn multi-scale features. Fan et al. [59] applied spatial pooling to adjust the feature map size, reducing computational expenses. For the visual saliency task, some works attempt to capture global semantic features. Cornia et al. [22] and Liu et al. [23] incorporated LSTM to capture long-range information. These studies have indicated that modeling the relevant dependence in the spatial domain can enhance saliency predictions. Dodge et al. [24] and Che et al. [25] constructed two-stream networks, in which features from the coarse-scale inputs provide global information. Recently, Lou et al. [30] have employed transformer blocks in the SP task to improve mid- and high-level visual features, resulting in competitive performance when compared to the state-of-the-art models. Liu et al. [60] leveraged the knowledge of a vision transformer with a token-based multi-task decoder to propagate global contexts and fuse multi-level tokens for salient object detection. Moreover, Li et al. [61] proposed a pure transformer-based network to predict human fixations in dynamic scenes, achieving exceptional performance on video saliency prediction. However, when computing attention maps in vision transformers, the dot-product operation typically focuses on the relationships between tokens or pixels. And the multi-head self-attention operation does not take into account the channel information and feature interaction between heads. Woo et al. [31] proposed a spatial module that utilizes average-pooling and max-pooling along the channel dimension to emphasize informative regions. Inspired by their research, we propose a channel-squeeze spatial attention (CSSA) module to capture the global channel information in a compression manner and then further refine the global spatial information by the dot-product operation. To enhance global semantic features in one block, the spatial attention inception block (SAIB) applies different spatial reduction methods onto parallel CSSA modules, capturing multi-scale features. Additionally, the proposed CSSA module takes low-level features with larger spatial dimensions as queries and high-level features with smaller resolutions as keys and values to incorporate semantic features from global perspective in the proposed local–global fusion block.

3. The proposed method

In this section, we describe the details of our method. In Section 3.1, we present the overall architecture of our proposed model, which is shown in Fig. 1. In Section 3.2, we describe the details of Channel-Squeeze Spatial Attention (CSSA) module and corresponding spatial attention inception blocks (SAIB). In Section 3.3, we show the design of the Local–Global Fusion Block (LGFB). Finally, we briefly discuss the readout design and the loss function in Section 3.5.

3.1. Architecture overview

Learned from structures like FPN [62] and U-Net [63] widely used in dense prediction tasks like semantic segmentation, we design a general encoder–decoder structure for the SP task, as illustrated in Fig. 1. The encoder part is DenseNet-161 [46] pre-trained on ImageNet, which generates low-level features at the early stages and high-level semantic features at the later stages. The backbone is divided into five stages and each stage is denoted as *Stage i* ($i \in \{1,2,3,4,5\}$). Here, we employ four stages ranging from *Stage 2* to *Stage 5*. The multi-level features, *i.e.*, f_2^e, f_3^e, f_4^e and f_5^e, are first fed to the corresponding 1×1 convolutional layers to reduce the channel number to 192. The decoder part consists of a spatial attention inception block (SAIB) that enhances semantic features f_5^e via a parallel CSSA structure, as well as three LGFBs that leverage these refined global semantic features to guide both local and global fusion of features from various levels using a CSSA module and CNN-based architectures. Then we utilize 3×3 convolutional layers following the decoder outputs, namely *i.e.*, f_2^d, f_3^d, f_4^d and f_5^d, to reduce the channel dimension to 128. f_3^d, f_4^d and f_5^d are upsampled 2×, 4× and 8×, respectively. Finally, the readout module produces the final saliency map of the input image by utilizing these enhanced features.

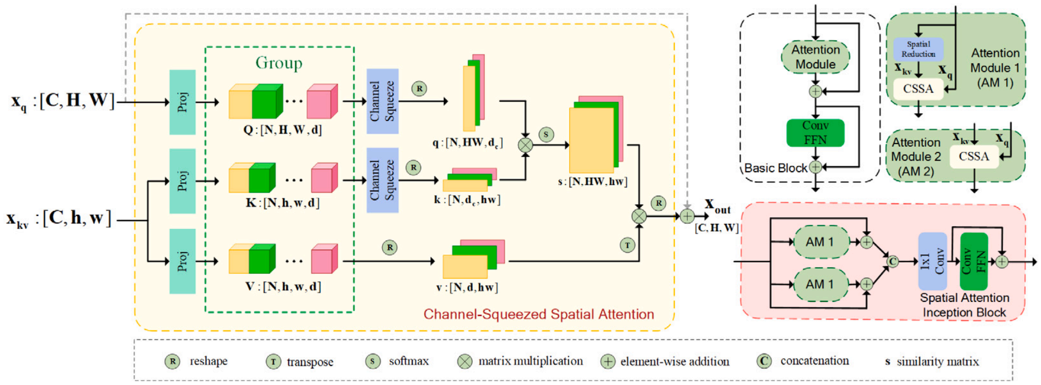

Fig. 2. The details of the Channel-Squeeze Spatial Attention (CSSA) module, Spatial Attention Inception Block (SAIB) and corresponding modules. The CSSA module can take features of different or same spatial sizes as inputs $\mathbf{x_q}$ and $\mathbf{x_{kv}}$. The Attention Module 1 takes $\mathbf{x_{kv}}$ that is processed through a spatial reduction method. SAIB takes features that are the source of $\mathbf{x_q}$ and $\mathbf{x_{kv}}$ and enhances features with spatial reduced $\mathbf{x_{kv}}$.

3.2. Channel-squeeze spatial attention

Due to the inherent characteristic of self-attention [26], transformers possess the ability to capture long-range relationships. Transformer blocks typically learn diverse feature representations through MHSA, which first splits features along the channel dimension and then performs a matrix multiplication to obtain spatial attention maps for non-local learning. However, MHSA does not take into account channel relationships. Spatial attention modules [31] utilize average-pooling and max-pooling techniques to aggregate channel information of features, refining them to effectively focus on the informative parts in the spatial dimension. This indicates that by using an appropriate method for channel interaction, we can enhance the global features extracted from spatial attention maps, which aligns with the objective of self-attention to capture long-range dependencies. To explore spatial semantic information and fully leverage inter-channel relationships, we propose a Channel-Squeeze Spatial Attention (CSSA) module. The details of CSSA are illustrated on the left of Fig. 2. Basically, the modules first learn the overall characteristic by channel compression and then refine the global spatial information through the dot-product operation.

Given the inputs $\mathbf{x_q} \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{x_{kv}} \in \mathbb{R}^{C\times h\times w}$ ($H \geq h$ and $W \geq w$), we first inject positional information to the features by depth-wise convolution as in [64]. And it is proven that larger padding causes convolutions to encode more absolute positional information [65]. The process is defined as follows:

$$\begin{aligned} \mathbf{x_q} &= \mathbf{x_q} + dwconv(\mathbf{x_q}), \\ \mathbf{x_{kv}} &= \mathbf{x_{kv}} + dwconv(\mathbf{x_{kv}}), \end{aligned} \tag{1}$$

where $dwconv(\cdot)$ indicates depth-wise convolution operation with padding 3 and kernel size 7. Then we apply linear projections through the 1 × 1 convolutional layer to get the feature embeddings which are queries $\mathbf{q}$, keys $\mathbf{k}$ and values $\mathbf{v}$. Following the procedure of MHSA, the features are simply divided along the channel dimension in order into groups. After the grouping, we obtain the query vector $\mathbf{Q}$, the key vector $\mathbf{K}$ and the value vector $\mathbf{V}$. The size of $\mathbf{Q}$ is $\mathbb{R}^{N\times H\times W\times d}$ and the size of $\mathbf{K}$ and $\mathbf{V}$ is $\mathbb{R}^{N\times h\times w\times d}$, where N is the number of groups and d is the channel dimension within a group. In the experiment, we set N to 4 to ensure enough features in each group, since the dimension of features is reduced to 192. The process is as follows:

$$\begin{aligned} \mathbf{Q} &= \mathrm{Group}(conv(\mathbf{x_q})), \\ \mathbf{K} &= \mathrm{Group}(conv(\mathbf{x_{kv}})), \\ \mathbf{V} &= \mathrm{Group}(conv(\mathbf{x_{kv}})), \end{aligned} \tag{2}$$

where $conv(\cdot)$ means 1 × 1 convolutional layer and $\mathrm{Group}(\cdot)$ is the grouping operation.

For channel interaction, average pooling captures the overall spatial features, but it assigns equal importance to all channels. Max pooling only focuses on the highest values. Meanwhile, when these operations are applied to the grouped features, there is no exchange of information among groups. To tackle this issue, we apply 3D convolutions [66], which were initially introduced to extract visual features from video frames and acquire temporal feature representations. Here, the time axis represents the group dimension. Therefore, with the sliding window of 3-dimensional kernels and channel compression, 3D convolutions can effectively enhance the spatial features within and among groups. To compute the spatial similarity matrix $\mathbf{s}$, we perform channel squeeze operations on $\mathbf{Q}$ and $\mathbf{K}$, respectively. After the squeeze operations and reshape operation, d is compressed to d_c ($d \geq d_c \geq 1$) and we obtain $\mathbf{q} \in \mathbb{R}^{N\times HW\times d_c}$, $\mathbf{k} \in \mathbb{R}^{N\times d_c\times hw}$ and $\mathbf{v} \in \mathbb{R}^{N\times d\times hw}$. This process can be defined as:

$$\begin{aligned} \mathbf{q} &= \mathrm{Reshape}(\mathrm{CS}(\mathbf{Q})), \\ \mathbf{k} &= \mathrm{Reshape}(\mathrm{CS}(\mathbf{K})), \\ \mathbf{v} &= \mathrm{Reshape}(\mathbf{V}), \end{aligned} \tag{3}$$

where $\mathrm{CS}(\cdot)$ is the channel squeeze operation, *i.e.*, average-pooling, max-pooling or 3D convolution. Then we perform the matrix multiplication between $\mathbf{q}$ and $\mathbf{k}$ to obtain spatial similarity matrix $\mathbf{s} \in \mathbb{R}^{HW\times hw}$ and apply softmax function on it as follows:

$$\mathbf{s} = Softmax(\mathbf{q} \otimes \mathbf{k}), \tag{4}$$

where $\otimes$ denotes the matrix multiplication. The transposed $\mathbf{v}$ is weighted by the similarity matrix $\mathbf{s}$ and then is reshaped to $C \times H \times W$. Lastly, the reconstructed features are added with the residual connection of $\mathbf{x_q}$, generating the output features $\mathbf{x_{out}} \in \mathbb{R}^{C\times H\times W}$ as followed:

$$\mathbf{x_{out}} = \mathrm{Reshape}(\mathbf{s} \otimes \mathbf{v}^{\mathrm{T}}) + \mathbf{x_q}, \tag{5}$$

Based on the input features operations, CSSA is applied in two basic modules, *i.e.*, attention module 1 and attention module 2 shown in Fig. 2. The attention module 1 employs a spatial reduction operation to decrease the spatial resolution of $\mathbf{x_{kv}}$. This approach reduces computational costs and facilitates effective learning of feature representations [58]. In addition to the attention part, transformer blocks [27,67] follow a standard design structure. The features are first improved by a residual attention module and then processed by a residual feed-forward neural network (FFN) module, as shown in Fig. 2. Inspired by the inception module [68] which intends to capture multi-scale contextual features in one block, we build a Spatial Attention Inception Block (SAIB) based on CSSA. The original inception module employs several convolutional layers with varying kernel sizes, enabling the learning of multi-scale features with diverse receptive sizes. By using different spatial reduction operations, the CSSA module can effectively

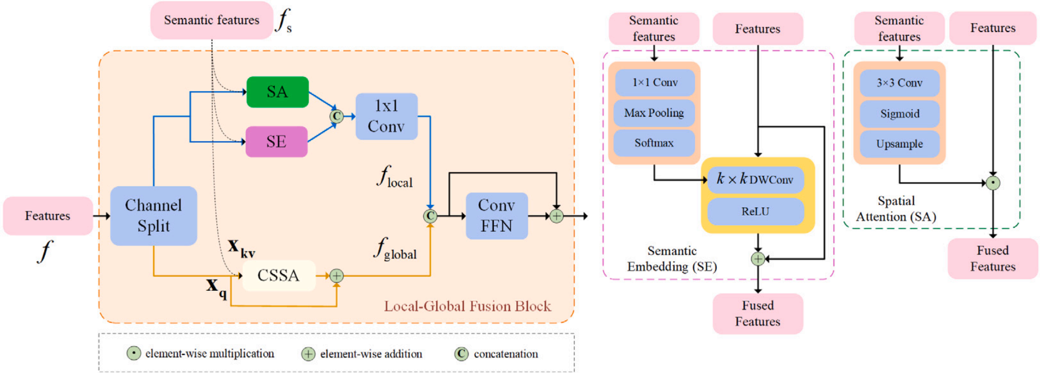

Fig. 3. Illustration of the proposed local–global fusion block. f_s is the enhanced global semantic features and f is from low- and mid-level features. The blue branch is to fuse features locally with semantic embedding (SE) and spatial attention (SA), while the yellow branch is to integrate two-level features from a global perspective. Finally, f_{local} and f_{global} are mixed together through a residual FFN module. In the experiment, k in semantic embedding branch is set to 7.

extract features from multiple perspectives and enhance the feature representations. As illustrated in Fig. 2, SAIB employs two attention modules to obtain multiple feature representations. Following these modules, a 1×1 convolutional layer is utilized to reduce the dimension of the concatenated features to C, which are then fed into the FFN module to obtain the enhanced features. In Fig. 1, the SAIB enhances the features f_5^e and uses spatial reduction operations like average-pooling and bilinear downsampling to downsample the features to a fixed spatial size 3×3.

3.3. Local–global fusion block

There exist semantic gaps between high-level semantic features and other-level features in the CNN backbone. Simple fusion operations, such as addition and concatenation, are insufficient and ineffective in integrating features of multi-levels. As is widely recognized, CNNs possess an inductive bias that enables them to capture local details, whereas transformers are adept at modeling long-range dependencies. Thus, the integration of multi-level features at both local and global levels can be achieved by combining CNNs and transformers. Moreover, high-level features contain abundant contextual information that can guide the integration of low- and mid-level features from two perspectives. We propose a Local–Global Fusion Block (LGFB) that incoporates semantic information into features both locally and globally.

Transformers exhibit great performance, but at the expense of significant computational resources, which are closely tied to the resolution and dimensions of the features. Given the low resolution of the high-level semantic features, it is already low in computation costs. Since the semantic features have a small resolution, which reduces the computation cost in the spatial domain, we intend to reduce the channel dimension of the features. Therefore, we aim to decrease the channel dimension of the features. Inspired by ShuffleNet [69], we employ channel split operation before two branches to reduce computational requirements. As depicted in Fig. 3, f is split into two parts based on the channel dimension for local and global fusion. The global branch (yellow lines in Fig. 3) utilizes the proposed CSSA to integrate features globally to learn relative importance in the spatial domain, while the local branch (blue lines in Fig. 3) uses convolution-based modules to fuse features locally. For the global branch, CSSA is exploited to enhance global information, with low- and mid-level features f as the query $\mathbf{x}_{\mathbf{q}}$ and global semantic features f_s as the key–values $\mathbf{x}_{\mathbf{kv}}$. As for the local branch, since the encoder has already owned a great ability to capture local features, we apply two simple CNN-based feature embedding structures in our decoder, *i.e.*, semantic embedding and spatial attention.

Semantic Embedding (SE): Semantic features usually have a small resolution and are full of contextual information. Therefore, features f_s are suitable to be processed as the kernels of convolutions. Inspired by [70], we modify the f_s with 1×1 convolutional layer and max-pooling operation to $k \times k$ kernels and use softmax to normalize the sum of values to 1 in the spatial domain. Then we employ a depth-wise convolution with the semantic kernels to enhance the other-level features. The kernels are dynamically generated from semantic features f_s, which adjust weights with the input. The whole process is depicted in the middle of Fig. 3. When k is set to 1, the semantic embedding becomes similar to the channel attention in SENet [71]. The channel attention generates attention weights by compressing the spatial information, which results in a loss of spatial information. In contrast, our semantic embedding is exploited to learn the spatial relationship for each channel of features individually by adopting depth-wise convolution with dynamic semantic kernels for local spatial feature fusion.

Spatial Attention (SA): As shown in Fig. 3, we first apply a 3×3 convolution to squeeze the dimension of global semantic features f_s to 1. Then a sigmoid function scales the values of features between 0 and 1 to obtain a spatial attention map. Finally, the attention map is upsampled and multiplied by the low-level features to highlight the informative regions.

The fused maps after SA and SE are concatenated, and a 1×1 convolutional layer is used to restore the channel dimension to half. After passing through two branches, the concatenated features of f_{local} and f_{global} are fed into a residual FFN module to interact with channel information. Due to the limitations of graphics memory, the LGFB-1 shown in Fig. 1 takes the down-sampled features of f_2^e that are generated after a 1×1 convolutional layer as input. A resize-convolution operation is used to restore the features to their original spatial size (88×88).

3.4. Visualizations of learned features on CSSA and LGFB

We visualize the attention maps for similarity matrix $\mathbf{s}$ of CSSA in various parts of the GSGNet, as shown in Fig. 4. The final two images in the top row show that the attention maps, which were learned from high-level features f_5^e in two groups, assigned great importance to the corresponding position as the ground truth. However, since the semantic features have a small resolution, these attention maps lack detailed spatial information. As illustrated in the middle row of Fig. 4, it can be observed that the average attention maps from LGFBs focus on the salient regions that closely resemble the ground truth in the spatial domain. This suggests that CSSA has effectively learned the global relevant importance in the spatial domain. In the bottom row, attention maps of the four groups in LGFB-1 are illustrated. It is evident that each group acquires its distinct features. Specifically, Group 1 and Group 2 exhibit a corresponding relationship with the averaged attention map and the ground truth, whereas Group 3 and Group 4 represent redundant attention maps. Overall, each group learns complementary

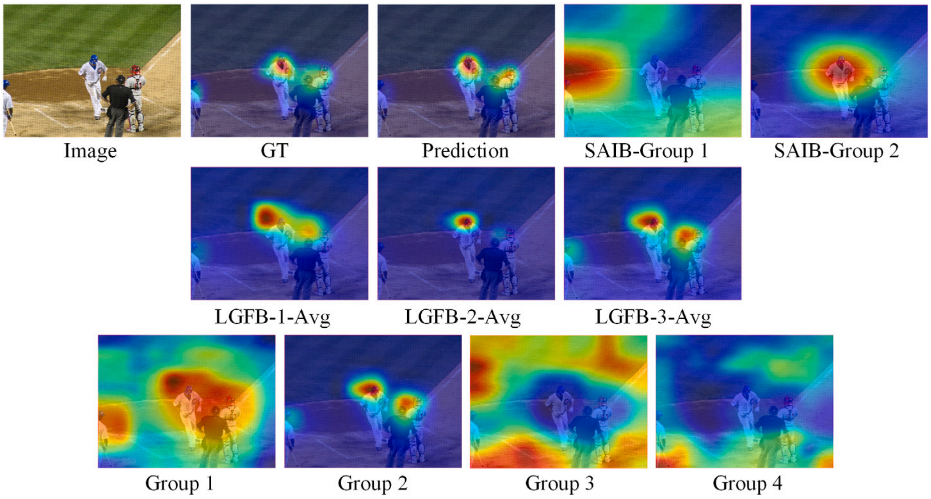

Fig. 4. Visualizations of CSSA attention maps from the similarity matrix **s**. The top row has two groups of SAIB. The middle row is the average attention map of LGFB-1, LGFB-2 and LGFB-3. And the bottom row is the attention map from four groups of LGFB-1.

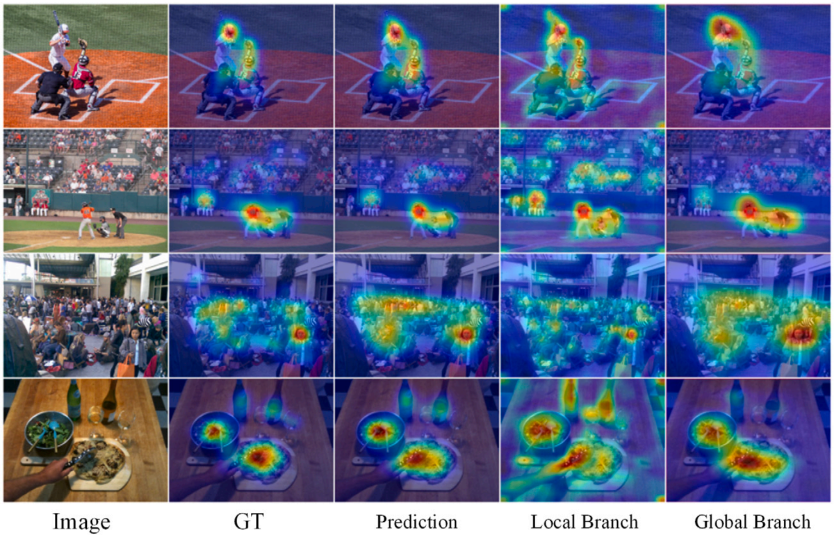

Fig. 5. Visualization of attention maps of the local and global branches in LGFB-1.

features that spread throughout the entire image. And based on the average result (LGFB-1-Avg), the weight of less important features constitutes a minor proportion. In model pruning [72–74], it is proven that attention heads in MHSA learn diverse feature representations, many of which are found to be redundant and can be removed without causing a significant impact on the overall performance of the model.

We also visualize where the two branches of the LGFB focus on in Fig. 5. The local branch appears to prioritize local characteristics and is capable of capturing intricate details. It effectively highlights human faces and identifies informative regions such as bottles and food, which are overlooked by the global branch. And the global branch compares the global relationships and suppresses less informative regions, highlighting the most salient parts. Like the example in the second row, the global branch inhibits the attention weights of individuals in the audience. The visualization demonstrates that both global and local fine-grained features are crucial for the SP task.

3.5. Readout module and loss function

Readout Module: The readout module is usually exploited to aggregate the refined features and convert the feature maps into a saliency map with the same resolution as the input. As illustrated in Fig. 1, the enhanced features, f_2^d, f_3^d, f_4^d and f_5^d, from four stages are concatenated together and then fed into the readout module. Inside the module, we employ two stacks of 1 × 1 convolutional layer and 7 × 7 depth-wise convolution to reduce the dimension (512 → 128 → 64) and fuse features. The depth-wise convolution has a large effective receptive field which is helpful to aggregate semantic information. Then we utilize two 4 × 4 depth-wise transposed convolutions which have few parameters to upscale the map by 4 times to the original resolution. Finally, we use two 3 × 3 convolutions to generate the saliency map.

Loss function: Metrics for SP are classified into location-based and distribution-based ones. Following [30,48,75,76], we use the combination of Kullback–Leibler Divergence (KL) and Pearson’s Correlation Coefficient (CC) as a loss function. KL measures the difference between two probability distributions and CC tells how correlated two variables are:

$$KL(\mathbf{P},\mathbf{Q}) = \sum_i \mathbf{Q}_i \log(\epsilon + \frac{\mathbf{Q}_i}{\mathbf{P}_i + \epsilon}), \tag{6}$$

$$CC(\mathbf{P},\mathbf{Q}) = \frac{\sigma(\mathbf{P},\mathbf{Q})}{\sigma(\mathbf{P}) \times \sigma(\mathbf{Q})}, \tag{7}$$

where **P** and **Q** are the predicted maps and GT maps, respectively, i is the pixel index, ϵ is a regularization term and $\sigma(\mathbf{P},\mathbf{Q})$ is the covariance of **P** and **Q**. Since KL and CC are both distribution-based metrics, in Section 4.5.4, we also conducted experiments on Normalized Scanpath Saliency (NSS), which is a location-based metric. NSS is computed as the average normalized saliency at fixated locations and defined as follows:

$$NSS(\mathbf{P}, \mathbf{Q}^B) = \frac{1}{N}\sum_i \overline{\mathbf{P}}_i \times \mathbf{Q}_i^B, \\ \text{where } N = \sum_i \mathbf{Q}_i^B \text{ and } \overline{\mathbf{P}} = \frac{\mathbf{P} - \mu(\mathbf{P})}{\sigma(\mathbf{P})} \tag{8}$$

where $\mathbf{Q}^B$ is the binary fixation map.

During the training phase, we observed that features from *Stage-4* and *Stage-5* tend to dominate and overwrite low-level features from *Stage-2* and *Stage-3*. This suggests that the shallow-stage features are not learned adequately. Experiments in [77] indicate that saliency models, such as Deepgaze II [78], rely on higher-level features in the deeper stages to obtain saliency maps. Conversely, the low-level features in the lower stages have a minimal impact on the final prediction. To address this issue, we employ deep supervision during the training phase and apply binary cross entropy (BCE) loss to supervise f_2^d and f_3^d at the early training stage. Two side outputs, each comprising two 3 × 3 convolutional layers, are applied to f_2^d and f_3^d. These auxiliary outputs are only used for initializing the weights of the shallow stages, in order to achieve a proficient starting point for learning the weights in the encoder and the corresponding components in the decoder. After the first training epoch, we discontinue supervision on f_2^d and f_3^d, and only use the KL and CC for the rest of the training. The total loss function is defined as follows:

$$\begin{aligned} Loss(\mathbf{P}, \mathbf{G}) =& KL(\mathbf{P}, \mathbf{G}) - CC(\mathbf{P}, \mathbf{G}) \\ &+ \alpha[BCE(\mathbf{P}_2, \mathbf{G}_2) + BCE(\mathbf{P}_3, \mathbf{G}_3)] \end{aligned} \tag{9}$$

where $\mathbf{P}_i$ is the saliency map generated from f_i^d, **G** is the ground-truth map and $\mathbf{G}_i$ is the ground-truth map which is resized to the same resolution as f_i^d ($i \in \{2, 3\}$). To balance the gradient and maintain the optimization direction controlled by saliency metrics, α is empirically set to 0.001 during the first epoch and later set to 0.

4. Experiment and results

4.1. Datasets

In the experiments, we use the following five widely used saliency prediction datasets.

SALICON [15] is currently the largest saliency prediction dataset which offers 10,000 training images, 5000 validation images and 5000 test images. Jiang et al. designed a mouse-tracking paradigm to simulate the natural viewing behavior of humans and the aggregation of the mouse trajectories represents the probability distribution of visual attention. Ground truths of the test set are held out and predictions can be submitted to the SALICON Saliency Prediction Challenge[1] for further evaluation.

MIT1003 [79] is an eye-tracking dataset containing 1003 natural indoor and outdoor scenes which are 779 landscape images and 228 portrait images. Groud-truths are generated by fixation data from 15 observers aged 18–35.

MIT300 [80] contains 300 natural images and collects eye movement data from 39 observers. It is held out and used as a benchmark test set in the MIT/Tübingen Saliency Benchmark.[2]

TORONTO [8] contains 120 natural images and the eye movement data is collected from 20 observers.

PASCAL-S [81] consists of 850 natural images from the PASCAL VOC 2010 dataset with eye-tracking data of 8 observers.

4.2. Evaluation metrics

There are a variety of metrics [82] that are proposed to evaluate the similarity and dissimilarity between the ground-truth labels and the predicted saliency maps. In [83], these metrics are classified into two categories as location-based and distribution-based metrics based on the form of the ground truth. For the former, metrics like NSS (Normalized Scanpath Saliency), IG (Information Gain) [84], AUC (Area under ROC Curve) and sAUC (Shuffled AUC) use the original fixation locations as the ground truth. Metrics like CC (Pearson's Correlation Coefficient), SIM (Similarity) and KL (Kullback–Leibler Divergence) require continuous distribution maps which are obtained by blurring each fixation location through a Gaussian filter. For KL, the lower value represents better performance, while the others opposite. In the experiment, for convenience, we use KL, CC, SIM, AUC and NSS metrics to evaluate the results of ablation studies. For the evaluation on the TORONTO and PASCAL-S datasets, the seven metrics are used. We use a center bias as the baseline model in IG and set the number of fixation maps to 10 in sAUC.

4.3. Implementation details

We follow a similar procedure in the state-of-the-art SP models [78, 85]. The parameters of our encoder DenseNet-161 are first initialized by the weights pre-trained on ImageNet, and then trained on the SALICON training set and monitored by the SALICON validation set. Then we fine-tune the model on the MIT1003 dataset. The dataset is randomly divided into two subsets which contain 903 and 100 images, respectively, while the former is used for training and the latter is for evaluation. We apply Adam optimizer [86] with an initial learning rate of 1×10^{-4} and set the batch size to 10 and training epochs to 20. The learning rate is decreased by a factor of 0.01 after two epochs and then decreased by a factor of 0.01 every five epochs. The minimal learning rate is 1×10^{-8}. All input images are resized to 352 × 352 pixels.

4.4. Comparison with the state-of-the-art models

We compare our method with state-of-the-art models, including DVA [50], GazeGAN [25], SimpleNet [48], MSI-Net [85], CEDNS [85], DINet [19], DeepGaze I [17], DeepGaze II [78], SAM-ResNet [22], EML-NET [45], SalED [76], TranSalNet [30], SalFBNet [75], ACSalNet [87], FSM [88], TempSAL [89] and UNISAL [90], and traditional models, including ITTI [7], GBVS [91], SUN [35], AIM [8] and CAS [34]. For SALICON and MIT300, we obtain their corresponding scores from the benchmarks by submitting the predicted saliency maps to the evaluation system.

Quantitative Comparison: The quantitative results of SALICON test set are presented in Table 1. It can be observed that our model is competitive with the state-of-the-art and ranks among the top three in terms of seven metrics. CC (0.912) and AUC (0.870) scores of our model are superior to other models. Our model is comparable to SalED on KL (0.190) metric. And our model ranks third on sAUC and IG. Compared to other metrics, NSS falls short of the highest score (2.050) attained by EML-NET and CEDNS. The main reason is that those two models use NSS as a component of their loss functions, generating more concentrated prediction points. Higher NSS means the predicted saliency maps are more discrete, while distribution-based metrics like KL and CC prefer more continuous maps. Our NSS-version model achieves the best performances on NSS (2.060), sAUC (0.748) and IG (0.914). However, the performance on KL, CC and SIM drops, which reflects that to some degree the location-based and distribution-based metrics may have an adversarial relationship. The results on MIT300 are shown in Table 2, in which our model achieves competitive performance with other SP models. Our model outperforms all compared models on AUC, sAUC and KL. Our scores on CC and SIM are slightly lower than SalFBNet, and the performance on NSS is consistent with the SALICON benchmark.

[1] http://salicon.net/challenge-2017/.

[2] https://saliency.tuebingen.ai/.

Table 1
Quantitative performance comparison on SALICON benchmark. The best results are marked in **red**.

Models	sAUC↑	IG↑	NSS↑	CC↑	AUC↑	SIM↑	KL↓
GazeGAN [25]	0.736	0.720	1.899	0.879	0.864	0.773	0.376
SimpleNet [48]	0.743	0.880	1.960	0.907	0.869	0.793	0.201
MSI-Net [85]	0.736	0.793	1.931	0.889	0.865	0.784	0.307
SAM-ResNet [22]	0.741	0.538	1.990	0.899	0.865	0.793	0.610
CEDNS [92]	0.744	0.845	2.050	0.840	0.863	0.732	–
DINet [19]	0.740	0.436	1.981	0.905	0.864	0.798	0.700
EML-NET [45]	0.746	0.736	2.050	0.886	0.866	0.780	0.520
SalFBNet [75]	0.740	0.839	1.952	0.892	0.868	0.772	0.236
SalED [76]	0.745	0.909	1.984	0.910	0.869	0.801	**0.190**
TranSalNet [30]	0.747	–	2.014	0.907	0.868	**0.803**	0.373
FSM [88]	0.732	0.716	1.863	0.875	0.862	0.772	0.365
ACSalNet [87]	0.744	0.890	1.981	0.905	0.868	0.798	0.232
TempSAL [89]	0.745	0.896	1.967	0.911	0.869	0.800	0.195
Ours (w NSS)	**0.748**	**0.914**	**2.060**	0.900	0.869	0.787	0.208
Ours	0.746	0.907	1.988	**0.912**	**0.870**	0.800	**0.190**

Table 2
Quantitative performance comparison on MIT300. The best results are marked in **red**.

Models	AUC↑	sAUC↑	NSS↑	CC↑	KL↓	SIM↑
ITTI [7]	0.543	0.536	0.408	0.131	1.496	0.338
GBVS [91]	0.806	0.630	1.246	0.479	0.888	0.484
AIM [8]	0.762	0.665	0.882	0.342	1.248	0.410
CAS [34]	0.758	0.640	1.019	0.385	1.072	0.432
SUN [35]	0.694	0.626	0.762	0.277	1.282	0.393
DVA [50]	0.843	0.726	1.931	0.663	0.629	0.585
GazeGAN [25]	0.861	0.732	2.212	0.758	1.339	0.649
EML-NET [45]	0.876	0.747	**2.488**	0.789	0.844	0.676
CASNet II [93]	0.855	0.740	1.986	0.705	0.586	0.581
TranSalNet [30]	0.873	0.747	2.413	0.807	1.014	0.690
DeepGaze I [17]	0.843	0.723	1.723	0.614	0.668	0.572
DeepGaze II [78]	0.873	0.776	2.337	0.770	0.424	0.664
MSI-Net [85]	0.874	0.779	2.305	0.779	0.423	0.670
SalFBNet [75]	0.877	0.786	2.470	**0.814**	0.415	**0.693**
UNISAL [90]	0.877	0.784	2.369	0.785	0.415	0.676
Ours	**0.878**	**0.788**	2.423	0.811	**0.410**	0.690

In order to verify the generalization ability of our model and make a comprehensive comparison, we also evaluate models on the TORONTO and PASCAL-S datasets, as shown in Table 3. For traditional models, ITTI is implemented based on the code provided by [91]. Other models are implemented with SMILER [94]. As for the deep learning-based methods, we use their public codes and to make a fair comparison, the used model weights are trained on SALICON. From the comparison, we see that our model achieves a better performance on most of the seven metrics than other SP models. This indicates that our model has a better generalization ability to be applied to other datasets. In addition, our model has a moderate amount of calculation costs and model parameters compared with other deep learning-based SP models. On the TORONTO dataset, our model achieves the best performance on the seven evaluation metrics. Also, on the PASCAL-S dataset, our model outperforms other models on five metrics apart from SIM and AUC.

Qualitative Comparison: We visualize the predicted saliency maps of our model and six deep learning-based SP models (DVA, MSI-Net, DINet, GazeGAN, SAM-ResNet and TranSalNet) in Fig. 6. As shown in Fig. 6, our model is able to capture salient regions with low-level features such as contrast as well as high-level attributes such as humans, texts and animals. Compared with other deep learning-based models, our model can effectively weigh the relative importance relationship among salient regions in complex scenes. For instance, as shown in the fourth column of Fig. 6, some compared models ignore the human in red and some make an incorrect estimate of the importance of salient regions. In contrast, our model captures every salient region and correctly judges the relative importance of these areas.

4.5. Ablation study

In this section, we conduct ablation studies on the proposed architecture to understand the influence of each component.

4.5.1. The contribution of main components

We quantitatively evaluate the contributions of the main components of our models and test the four variants five times for each and report the average and standard deviation scores over random seeds for more accurate comparison. The results are displayed in Table 4. We build a strong baseline model based on the FPN structures. Features from high-level to low-level are progressively added together and the features of the largest size are fed to the readout module to obtain the predicted saliency map with the same resolution as the input image. And we can observe from Table 4 that the baseline model provides a strong comparison standard. Next, Spatial Attention Inception Block (SAIB) is added over the baseline to enhance global semantic features. The performance gains large improvement on SIM, CC, KL and NSS, which reflects the necessity of semantic information enrichment. Local–Global Fusion Blocks (LGFB) aim to integrate multi-level features sufficiently from local–global perspectives and the model obtains the appealing performance gain as the SAIB does. This indicates that the local and global fusion method is effective and essential for the SP task. From Table 4, it is obvious that the SAIB and LGFB bring significant improvements on KL metric by 5% compared with the baseline model.

4.5.2. Ablation study of CSSA

To fully assess the effectiveness of CSSA module, we conduct ablation studies on SALICON validation set. The results are presented in Table 5, in which the order is the number of groups, channel-squeeze method and compression ratio, listed in parentheses. β represents compression ratio and g is the number of groups. For 3D convolution, we initialize the kernels by adopting truncated normal distribution.

Importance of CSSA module: In Table 5 (A), we remove CSSA modules from SAIB and LGFB and keep semantic embedding and spatial

Table 3
Quantitative performance comparison on TORONTO and PASCAL-S. We also report the implementation backend (Platform), model parameters (Params) and multiply–accumulate operations (MACs). Note that all the deep learning-based models are trained on SALICON for a fair comparison. The best results are marked in **red**.

Models	Platform	Params (M)	MACs (G)	TORONTO							PASCAL-S						
				AUC↑	NSS↑	CC↑	SIM↑	KL↓	sAUC↑	IG↑	AUC↑	NSS↑	CC↑	SIM↑	KL↓	sAUC↑	IG↑
ITTI [7]	–	–	–	0.8014	1.2969	0.4779	0.4779	0.9688	0.6495	0.4114	0.8176	1.3034	0.4248	0.3587	1.2868	0.6572	0.4206
SUN [35]	–	–	–	0.6846	0.7095	0.2376	0.3569	1.4530	0.6178	0.0048	0.6555	0.5928	0.1852	0.2681	1.7914	0.5919	−0.0822
AIM [8]	–	–	–	0.7677	0.8370	0.3115	0.3684	1.4448	0.6743	0.1035	0.7804	0.8279	0.2776	0.2820	1.6810	0.6560	0.0159
CAS [34]	–	–	–	0.7827	1.2695	0.4481	0.4372	1.0208	0.6854	0.3715	0.7763	1.1229	0.3561	0.3372	1.4048	0.6691	0.3121
GBVS [91]	–	–	–	0.8315	1.5191	0.5691	0.4864	0.8478	0.6342	0.5311	0.8565	1.5212	0.4984	0.3957	1.1382	0.6652	0.5719
GazeGAN [25]	PyTorch	230.48	81.93	0.8504	1.8535	0.6668	0.5814	0.6955	0.6955	0.7584	0.8813	1.9719	0.6130	0.5004	0.8771	0.7358	0.8985
DVA [50]	Caffe	25.07	–	0.8622	2.1237	0.7152	0.5842	0.6538	0.6874	0.8119	0.8853	2.2614	0.6559	0.5163	0.8808	0.7221	0.9256
SAM-ResNet [22]	Theano	70.09	–	0.8632	2.1440	0.7391	0.6232	1.8749	0.7138	−0.2905	0.8959	2.3381	0.7023	**0.5605**	1.2365	0.7544	0.5958
TranSalNet [30]	PyTorch	76.56	54.33	0.8598	2.0097	0.7043	0.5867	0.6138	0.7135	0.8255	0.8927	2.1947	0.6673	0.5043	0.8023	0.7567	0.9636
DINet [19]	Tensorflow	27.04	–	0.8630	2.1373	0.7477	0.6250	0.5757	0.7215	0.8893	0.8935	2.3240	0.6986	0.5535	0.7478	0.7546	1.0479
MSI-Net [85]	Tensorflow	24.93	–	0.8694	2.1192	0.7419	0.6203	0.5412	0.7188	0.9175	0.8924	2.2646	0.6850	0.5363	0.7481	0.7516	1.0323
ACSalNet [87]	PyTorch	47.35	43.48	0.8649	2.0118	0.7093	0.6006	0.5833	0.7146	0.8374	**0.8969**	2.2611	0.6886	0.5237	0.7574	0.7505	1.0139
FSM [88]	Tensorflow	2.13	–	0.8625	2.0070	0.7155	0.6099	0.5982	0.7160	0.8486	0.8924	2.2221	0.6798	0.5398	0.7585	0.7523	1.0144
TempSAL [89]	PyTorch	242.52	79.97	0.8686	2.1102	0.7451	0.6241	0.5415	0.7183	0.9109	0.8959	2.3243	0.6991	0.5481	0.7142	**0.7604**	1.0709
Ours	PyTorch	31.38	29.78	**0.8717**	**2.1556**	**0.7564**	**0.6255**	**0.5321**	**0.7242**	**0.9236**	0.8966	**2.3549**	**0.7074**	0.5519	**0.7126**	**0.7604**	**1.0762**

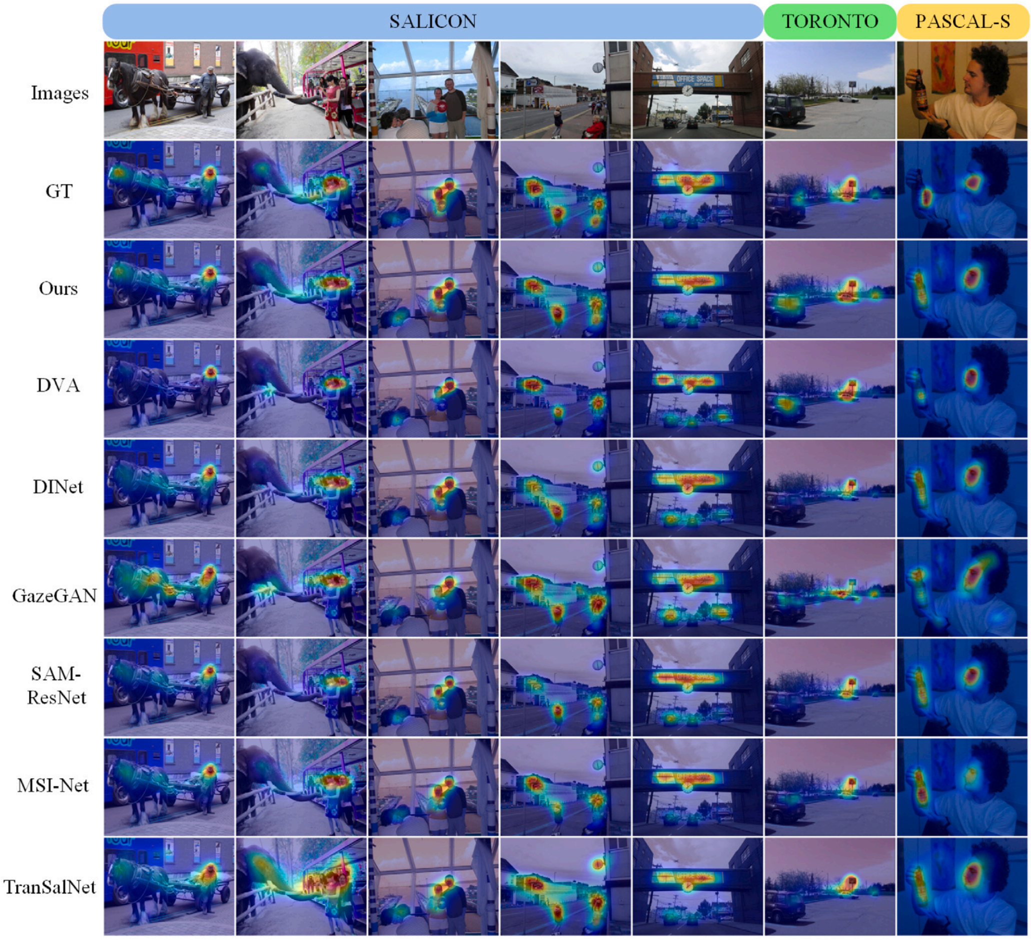

Fig. 6. Visual comparisons with start-of-the-art SP models. The first five columns are from SALICON validation set. The last two columns are from TORONTO and PASCAL-S, respectively.

Table 4
Ablation study on main components of the proposed model on SALICON validation set. We test the variants in five times and the average and standard deviation scores of the five metrics are reported as mean(±std) in the table. The best results are **highlighted**.

Baseline	SAIB	LGFB	KL↓	CC↑	SIM↑	NSS↑	AUC↑
✓			0.1933(±0.00120)	0.9087(±0.00034)	0.8007(±0.00018)	1.9388(±0.00220)	0.8721(±0.00009)
	✓		0.1850(±0.00119)	0.9118(±0.00014)	0.8041(±0.00038)	1.9420(±0.00147)	0.8727(±0.00003)
		✓	0.1850(±0.00124)	0.9118(±0.00016)	0.8040(±0.00019)	1.9421(±0.00147)	0.8727(±0.00005)
	✓	✓	**0.1822**(±0.00061)	**0.9128**(±0.00012)	**0.8051**(±0.00036)	**1.9434**(±0.00112)	**0.8729**(±0.00008)

Table 5
Ablation study on CSSA module on SALICON validation set. As listed in parentheses, the order is the number of groups, channel-squeeze method and compression ratio. The best results are **highlighted**.

	Type	KL↓	CC↑	SIM↑	NSS↑	AUC↑
Ours	Full model(4, *3D Conv*, 0.25)	0.1822	**0.9129**	**0.8053**	1.9419	**0.8730**
(A)	*w/o CSSA*	0.1875	0.9109	0.8024	1.9397	0.8723
(B)	*Vanilla*	0.1831	0.9124	0.8048	1.9429	0.8728
	Max	0.1834	0.9125	0.8048	1.9417	0.8728
	Avg	0.1839	0.9127	0.8050	1.9419	**0.8730**
(C)	$\beta = 1$	0.1826	0.9123	0.8042	**1.9439**	0.8729
	$\beta = 0.75$	0.1836	0.9124	0.8047	1.9434	0.8729
	$\beta = 0.5$	0.1835	0.9124	0.8048	1.9430	0.8729
	$\beta = 0$	0.1829	0.9124	0.8043	1.9420	0.8729
(D)	g = 1	0.1848	0.9120	0.8044	1.9435	0.8727
	g = 8	**0.1818**	**0.9129**	0.8050	1.9436	0.8729
	g = 12	0.1832	0.9126	0.8050	1.9431	0.8729
	g = 16	0.1835	0.9125	0.8047	1.9435	0.8729

Table 6
Ablation study on LGFB on SALICON validation set. *loc* is the local branch, while *glo* is the global branch. The best results are **highlighted**.

	Type	KL↓	CC↑	SIM↑	NSS↑	AUC↑
Ours	Full model	**0.1822**	**0.9129**	0.8053	1.9419	**0.8730**
(A)	*w/o global semantics*	0.1846	0.9119	0.8044	1.9416	0.8728
(B)	*w/o loc*	0.1837	0.9120	0.8042	1.9405	0.8728
	w/o glo	0.1866	0.9122	0.8049	1.9455	0.8728
(C)	*w channel attention* [71]	0.1840	0.9121	0.8043	1.9432	0.8728
(D)	*Sequential* (*loc-glo*)	0.1829	0.9125	0.8052	1.9453	0.8729
	Sequential (*glo-loc*)	0.1835	0.9125	**0.8054**	**1.9461**	0.8729

attention blocks to solely enhance local features. Compared with the baseline model in Table 4, this variant gains improvements on the five metrics. And in comparison with our full model, the performance drops off as expected, reflecting that introducing long-range modeling ability is beneficial for our model.

Different Channel-Squeeze Methods: Here, we compare three different channel-squeeze ways: average and max along the channel dimension, and 3D convolution with the vanilla transformer. The vanilla transformer is implemented as in [30]. Experimental results with various channel-squeeze methods are shown in Table 5 (B). Max and average operations reduce the channel dimension of queries and keys to 1 and still achieve considerable or even better performance compared with the vanilla transformer, which reflects the redundancy of features. We can observe that 3D convolution achieves the best performance than others on four metrics, *i.e.*, KL, CC, SIM and AUC, and its performance is slightly weaker on NSS.

Different Compression Ratios: We investigate the influence of different compression ratios of 3D convolution on performance. The compression of the channel dimension can force the model to learn practical information from redundant features. Unlike average and max operations, 3D convolution brings extra parameters required for optimization and the compression ratio is hard to determine. We test the ratio on empirical settings ranging from 0 to 1. The ratio in Table 5 (C) represents the compression degree of the channel dimension. If the ratio is 0.25, the channel dimension is reduced to 25% of the original one. When the ratio is 0, the channel dimension of feature maps is squeezed to 1. Compared with the results in Table 5 (C), it is observed that our model with 25% achieves the best performance on KL (0.1822), CC (0.9129), SIM (0.8053) and AUC (0.8730) and obtains a similar performance on NSS as discussed before.

Different Numbers of Groups: In Table 5 (D), we conducted experiments with different numbers of groups. Since the features' channel dimension in the decoder is either 192 or 96, the group embedding dimension is set to 32 for larger groups (8,12,16). From Table 5 (D), we can see that the performance drops off with more groups. When the group number is 4, the performance has reached a saturation state. The limitation of the channel dimension might result in performance degradation with more groups.

4.5.3. Ablation study of LGFB

To verify the effectiveness of the local–global feature fusion block, we conduct ablation studies about local and global structures.

Importance of the global semantics: In Table 6 (A), we remove the guidance of the global semantic features and enhance multi-level features by themselves. This leads to a performance degradation, reflecting that the global semantic features help to refine the fusion of local and global features.

Importance of the local and global branches: We test the model performance with only one branch used (*w/o loc* and *w/o glo*). In the experiment *w/o glo*, we keep the semantic embedding (SE) and spatial attention (SA) block by removing the CSSA module and the FFN module. In Table 6 (B), with only one type of fusion, the performance of *loc* and *glo* degrades on four metrics compared with our model. This indicates that feature fusion from both global and local perspectives is necessary to improve our model's performance. The CNN-Transformer hybrid structures are skilled in extracting both local and global features based on the visualization in Fig. 5.

Table 7
Ablation study on loss function on SALICON validation set. The *BCE-a* indicates that BCE loss is used during the training phase. And the *BCE-1* means that BCE loss is only utilized for epoch one. The best results are **highlighted**.

Type	KL↓	CC↑	SIM↑	NSS↑	AUC↑
base	0.1839	0.9122	0.8037	1.9405	0.8727
w BCE-a	0.1841	0.9111	0.8019	1.9356	0.8725
w BCE-1	**0.1822**	**0.9129**	**0.8053**	1.9419	0.8730
*w BCE-1 + 0.5*NSS*	0.1887	0.9030	0.7982	**2.0088**	**0.8737**

Semantic embedding and channel attention: To evaluate the performance between the proposed semantic embedding and the channel attention in SENet [71], we replace the semantic embedding parts with the channel attention in our model. In Table 6 (C), the variant *w channel attention* shows a performance degradation on KL, CC, SIM and AUC. The reason is that the channel attention aims to highlight the features based on channels and cannot capture local spatial features, while our semantic embedding learns the spatial relationship and enhances the semantic features in the spatial domain for each channel of features. Additionally, the FFN part in LGFBs already plays a role in communicating channel information. Thus, the channel attention does not bring much improvement for our model.

Structure of the local and global branches: In the LGFB, we establish a parallel structure to fuse local and global features. We examine the proposed block in a sequential mode. From Table 6 (D), we can observe that the *Sequential* (*glo-loc*) achieves better performance on SIM and NSS but worse performance on KL, CC and AUC compared with our model. Overall, both the parallel structure and the sequential structures can obtain good performance.

4.5.4. Ablation study of loss function

In the experiment, we apply a deep supervision method with supervision on the *Stage 2* and *Stage 3* to obtain a better learning start status. From Table 7, initializing the weights of shallow layers in the encoder and the corresponding part in the decoder by deep supervision (*w BCE-1*) achieves a better performance. And if we train our model with the BCE loss during the whole training phase as *w BCE-a*, the performance of the model has a drop. The reason might be the different numerical ranges among BCE, KL and CC, causing the unbalanced gradient backward. Since the metrics related to SP are of real concern, BCE is only used as an auxiliary function and can be removed after one epoch (*w BCE-1*). When the loss function is applied at the early stage of training, it helps to ease the situation that low-level features are ignored or overwritten and improves the model performance. In addition, we also conduct an ablation study on NSS metric and the result is in the last row of Table 7. We add NSS to our loss function, which brings improvements on the location-based metrics, *i.e.*, NSS (1.9446 → 2.0088) and AUC (0.8030 → 0.8037), and a decline on the

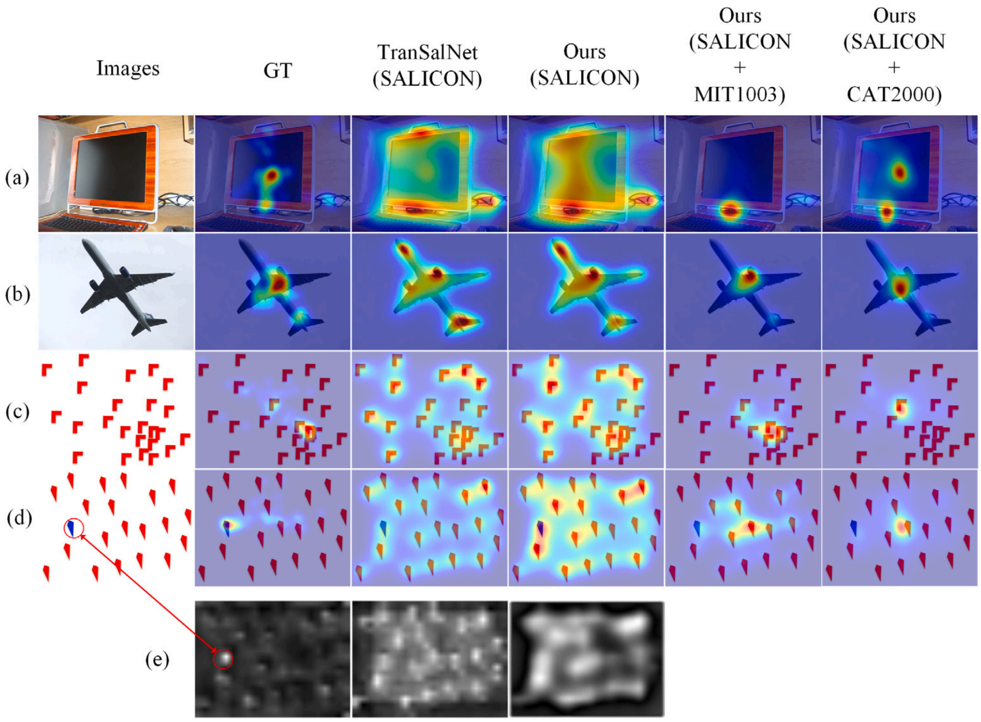

Fig. 7. Some failure cases. The first two images are from PASCAL-S and the last two are from MIT1003. Both are excluded from the training set. In the last two columns, our model is finetuned on MIT1003 training set and on CAT2000, respectively. In (e), we visualize the features in the shallow stage of our model, the output of the FFN, the FFN sum of the skip-connection in LGFB-1 and the 3 × 3 convolution, from left to right, respectively.

distribution-based metrics, *i.e.*, KL (0.1830 → 0.1887), CC (0.9129 → 0.9030) and SIM (0.8053 → 0.7982), which is the same phenomenon as the SALICON test set. As suggested in [83], the choice of these location-based and distribution-based metrics is application-related. NSS is suitable for tasks, like image-retargeting, that require the relative importance of different image regions. KL is appropriate for detection applications, as it can penalize failures and enable models to ignore areas of low probability.

4.6. Failure cases and analysis

With the aid of global semantic features, the GSGNet achieves promising performance on the various eye-tracking datasets. But there are still some situations that our model cannot handle well. In Fig. 7, we visualize some instances of failure using our model, indicating potential directions for future research. The example in Fig. 7(a) depicts a common failure scenario, in which there is no meaningful semantic content (such as the black screen), causing humans to pay attention to the image center. Models trained on the datasets full of semantic scenes cannot well deal with this. Incorporating a center bias as a prior like [17] is a possible solution. The model finetuned on CAT2000 appears to address the problem, but the results of other images show that the model has a center-biased behavior. Secondly, in Fig. 7(b), our model, trained on the SALICON dataset, predicts the shape of the airplane, which makes it suitable for auxiliary use in applications such as object segmentation [95]. However, the situation is tricky for modeling human eye fixations. In some cases, humans may observe the shape of an object, whereas, in other cases, they may not. Finetuning our model on MIT1003 alleviates this issue, reflecting that the dataset bias impacts the model's behavior. Moreover, in Fig. 7(c) and (d), for deep learning-based models, the challenges lie in the unnatural images, since in this situation human attention is significantly influenced by low-level properties such as texture and color. Finetuning the model on MIT1003 enables our model to focus on the clustered objects. However, it still falls short in the color scenario. In Fig. 7(e), we provide a visualization of the process of generating the features in shallow layers and observe that the model has learned the color-related features, as indicated by the red circle. However, as the features continue to be processed (from left to right in Fig. 7(e)), the color information gradually diminishes and is overwhelmed by the semantic features. One possible solution is to build a separate module to process the features in the shallow layers and design a method to adaptively adjust the importance of the low-level features for the final prediction. Overall, the current eye-tracking datasets have inherent characteristics that can influence the behaviors of saliency prediction models. Saliency models like UNISAL [90] have proposed a domain-shift modeling method with domain-adaptive batch normalization, prior, fusion and smoothing to learn dataset-related knowledge from multiple eye-tracking datasets. Additionally, a high-quality dataset containing diverse scenarios can be further studied for saliency prediction using methods such as semi-supervised [75] and self-supervised learning.

5. Conclusions

In this paper, we propose a semantic-guided network for predicting saliency maps. Features in the encoder's deepest layer contain rich semantic and contextual information, which are further refined by the channel-squeeze spatial attention (CSSA)-based blocks, which capture global representations from the processed spatial attention maps. Additionally, multi-level features are integrated by the local–global fusion block (LGFB) combining the merits of CNNs and transformers, fusing local and long-range spatial information at multiple perceptual levels. We have conducted relevant ablation studies and evaluated our model on four saliency datasets. Quantitative and qualitative results have demonstrated the effectiveness of the proposed model on the SP task.

CRediT authorship contribution statement

Jiawei Xie: Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Zhi Liu:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Gongyang**

Li: Writing – review & editing, Visualization, Funding acquisition. **Xiaofeng Lu:** Validation, Resources, Formal analysis. **Tao Chen:** Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62171269, and in part by the China Postdoctoral Science Foundation under Grant 2022M722037.

References

[1] W. Wang, J. Shen, R. Yang, F. Porikli, Saliency-aware video object segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 40 (1) (2017) 20–33.

[2] G. Li, Z. Liu, R. Shi, W. Wei, Constrained fixation point based segmentation via deep neural network, Neurocomputing 368 (2019) 180–187.

[3] G. Li, Z. Liu, R. Shi, Z. Hu, W. Wei, Y. Wu, M. Huang, H. Ling, Personal fixations-based object segmentation with object localization and boundary preservation, IEEE Trans. Image Process. 30 (2021) 1461–1475.

[4] X. Fang, J. Zhu, X. Shao, H. Wang, LC3net: Ladder context correlation complementary network for salient object detection, Knowl.-Based Syst. 242 (2022) 108372, http://dx.doi.org/10.1016/j.knosys.2022.108372, URL https://www.sciencedirect.com/science/article/pii/S0950705122001411.

[5] Z. Luo, L. Song, S. Zheng, N. Ling, H.264/advanced video control perceptual optimization coding based on JND-directed coefficient suppression, IEEE Trans. Circuits Syst. Video Technol. 23 (6) (2013) 935–948, http://dx.doi.org/10.1109/TCSVT.2013.2240919.

[6] T. Huang, R. Fu, Prediction of the driver's focus of attention based on feature visualization of a deep autonomous driving model, Knowl.-Based Syst. 251 (2022) 109006, http://dx.doi.org/10.1016/j.knosys.2022.109006, URL https://www.sciencedirect.com/science/article/pii/S0950705122004865.

[7] 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.

[8] N. Bruce, J. Tsotsos, Saliency based on information maximization, in: Proceedings of Advances in Neural Information Processing Systems, vol. 18, 2005.

[9] D. Walther, C. Koch, Modeling attention to salient proto-objects, Neural Netw. 19 (9) (2006) 1395–1407.

[10] E. Erdem, A. Erdem, Visual saliency estimation by nonlinearly integrating features using region covariances, J. Vis. 13 (4) (2013) 11.

[11] A. Torralba, A. Oliva, M.S. Castelhano, J.M. Henderson, Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search., Psychol. Rev. 113 (4) (2006) 766.

[12] W. Einhäuser, M. Spain, P. Perona, Objects predict fixations better than early saliency, J. Vis. 8 (14) (2008) 18.

[13] M. Cerf, E.P. Frady, C. Koch, Faces and text attract gaze independent of the task: Experimental data and computer model, J. Vis. 9 (12) (2009) 10.

[14] A. Borji, L. Itti, Exploiting local and global patch rarities for saliency detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 478–485, http://dx.doi.org/10.1109/CVPR.2012.6247711.

[15] X. Huang, C. Shen, X. Boix, Q. Zhao, Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 262–270.

[16] A. Borji, L. Itti, Cat2000: A large scale fixation dataset for boosting saliency research, 2015, arXiv preprint arXiv:1505.03581.

[17] M. Kümmerer, L. Theis, M. Bethge, Deep gaze I: Boosting saliency prediction with feature maps trained on ImageNet, 2014, arXiv preprint arXiv:1411.1045.

[18] M. Cornia, L. Baraldi, G. Serra, R. Cucchiara, A deep multi-level network for saliency prediction, in: Proceedings of the IEEE International Conference on Pattern Recognition (ICPR), 2016, pp. 3488–3493.

[19] S. Yang, G. Lin, Q. Jiang, W. Lin, A dilated inception network for visual saliency prediction, IEEE Trans. Multimed. 22 (8) (2019) 2163–2176.

[20] M. Kümmerer, T.S. Wallis, L.A. Gatys, M. Bethge, Understanding low-and high-level contributions to fixation prediction, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4789–4798.

[21] F. Zhou, R. Yao, G. Liao, B. Liu, G. Qiu, Visual saliency via embedding hierarchical knowledge in a deep neural network, IEEE Trans. Image Process. 29 (2020) 8490–8505, http://dx.doi.org/10.1109/TIP.2020.3016464.

[22] M. Cornia, L. Baraldi, G. Serra, R. Cucchiara, Predicting human eye fixations via an lstm-based saliency attentive model, IEEE Trans. Image Process. 27 (10) (2018) 5142–5154.

[23] N. Liu, J. Han, A deep spatial contextual long-term recurrent convolutional network for saliency detection, IEEE Trans. Image Process. 27 (7) (2018) 3264–3274.

[24] S.F. Dodge, L.J. Karam, Visual saliency prediction using a mixture of deep neural networks, IEEE Trans. Image Process. 27 (8) (2018) 4080–4090, http://dx.doi.org/10.1109/TIP.2018.2834826.

[25] Z. Che, A. Borji, G. Zhai, X. Min, G. Guo, P. Le Callet, Gazegan: A generative adversarial saliency model based on invariance analysis of human gaze during scene free viewing, 2019, arXiv preprint arXiv:1905.06803.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of Advances in Neural Information Processing Systems, vol. 30, 2017.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: Proceedings of International Conference on Learning Representations, 2020.

[28] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[29] R. Wu, X. Wen, L. Yuan, H. Xu, DASFTOT: Dual attention spatiotemporal fused transformer for object tracking, Knowl.-Based Syst. 256 (2022) 109897, http://dx.doi.org/10.1016/j.knosys.2022.109897, URL https://www.sciencedirect.com/science/article/pii/S095070512200990X.

[30] J. Lou, H. Lin, D. Marshall, D. Saupe, H. Liu, TranSalNet: Towards perceptually relevant visual saliency prediction, Neurocomputing 494 (2022) 455–467.

[31] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 3–19.

[32] Y. Liu, Z. Shao, N. Hoffmann, Global attention mechanism: Retain information to enhance channel-spatial interactions, 2021, arXiv preprint arXiv:2112.05561.

[33] J. Park, S. Woo, J.-Y. Lee, I.S. Kweon, Bam: Bottleneck attention module, 2018, arXiv preprint arXiv:1807.06514.

[34] S. Goferman, L. Zelnik-Manor, A. Tal, Context-aware saliency detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (10) (2011) 1915–1926.

[35] L. Zhang, M.H. Tong, T.K. Marks, H. Shan, G.W. Cottrell, SUN: A Bayesian framework for saliency using natural statistics, J. Vis. 8 (7) (2008) 32.

[36] A.M. Treisman, G. Gelade, A feature-integration theory of attention, Cogn. Psychol. 12 (1) (1980) 97–136.

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

[38] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[40] G. Li, Z. Liu, D. Zeng, W. Lin, H. Ling, Adjacent context coordination network for salient object detection in optical remote sensing images, IEEE Trans. Cybern. (2022) 1–13, http://dx.doi.org/10.1109/TCYB.2022.3162945.

[41] B. Xu, Z. Chen, Multi-level fusion based 3d object detection from monocular images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2345–2353.

[42] G. Li, Z. Liu, M. Chen, Z. Bai, W. Lin, H. Ling, Hierarchical alternate interaction network for RGB-D salient object detection, IEEE Trans. Image Process. 30 (2021) 3528–3542.

[43] X. Hu, C.-W. Fu, L. Zhu, T. Wang, P.-A. Heng, SAC-net: Spatial attenuation context for salient object detection, IEEE Trans. Circuits Syst. Video Technol. 31 (3) (2021) 1079–1090, http://dx.doi.org/10.1109/TCSVT.2020.2995220.

[44] N. Liu, J. Han, D. Zhang, S. Wen, T. Liu, Predicting eye fixations using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[45] S. Jia, N.D. Bruce, Eml-net: An expandable multi-layer network for saliency prediction, Image Vis. Comput. 95 (2020) 103887.

[46] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[47] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8697–8710.

[48] N. Reddy, S. Jain, P. Yarlagadda, V. Gandhi, Tidying deep saliency prediction architectures, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 10241–10247.

[49] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[50] W. Wang, J. Shen, Deep visual attention prediction, IEEE Trans. Image Process. 27 (5) (2017) 2368–2378.

[51] H. Ning, B. Zhao, Z. Hu, L. He, E. Pei, Audio–visual collaborative representation learning for dynamic saliency prediction, Knowl.-Based Syst. 256 (2022) 109675, http://dx.doi.org/10.1016/j.knosys.2022.109675, URL https://www.sciencedirect.com/science/article/pii/S0950705122008486.

[52] Q. Lai, T. Zhou, S. Khan, H. Sun, J. Shen, L. Shao, Weakly supervised visual saliency prediction, IEEE Trans. Image Process. 31 (2022) 3111–3124, http://dx.doi.org/10.1109/TIP.2022.3158064.

[53] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[54] S. Zuo, Y. Xiao, X. Chang, X. Wang, Vision transformers for dense prediction: A survey, Knowl.-Based Syst. 253 (2022) 109552, http://dx.doi.org/10.1016/j.knosys.2022.109552, URL https://www.sciencedirect.com/science/article/pii/S0950705122007821.

[55] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, J. Gao, Multi-scale vision longformer: A new vision transformer for high-resolution image encoding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2998–3008.

[56] S. Ren, D. Zhou, S. He, J. Feng, X. Wang, Shunted self-attention via multi-scale token aggregation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10853–10862.

[57] J. Gu, H. Kwon, D. Wang, W. Ye, M. Li, Y.-H. Chen, L. Lai, V. Chandra, D.Z. Pan, Multi-scale high-resolution vision transformer for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12094–12103.

[58] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 568–578.

[59] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, C. Feichtenhofer, Multiscale vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6824–6835.

[60] N. Liu, N. Zhang, K. Wan, J. Han, L. Shao, Visual saliency transformer, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 4702–4712.

[61] C. Ma, H. Sun, Y. Rao, J. Zhou, J. Lu, Video saliency forecasting transformer, IEEE Trans. Circuits Syst. Video Technol. 32 (10) (2022) 6850–6862, http://dx.doi.org/10.1109/TCSVT.2022.3172971.

[62] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[63] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.

[64] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, SegFormer: Simple and efficient design for semantic segmentation with transformers, in: Proceedings of Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 12077–12090.

[65] X. Ding, X. Zhang, J. Han, G. Ding, Scaling up your kernels to 31x31: Revisiting large kernel design in cnns, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11963–11975.

[66] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2012) 221–231.

[67] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, S. Yan, Metaformer is actually what you need for vision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10819–10829.

[68] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[69] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.

[70] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, Z. Liu, Dynamic convolution: Attention over convolution kernels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11030–11039.

[71] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[72] P. Michel, O. Levy, G. Neubig, Are sixteen heads really better than one? in: Proceedings of Advances in Neural Information Processing Systems, vol. 32, 2019.

[73] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, 2019, arXiv preprint arXiv:1905.09418.

[74] H. Yang, H. Yin, P. Molchanov, H. Li, J. Kautz, Nvit: Vision transformer compression and parameter redistribution, 2021, arXiv preprint arXiv:2110.04869.

[75] G. Ding, N. İmamoğlu, A. Caglayan, M. Murakawa, R. Nakamura, SalFBNet: Learning pseudo-saliency distribution via feedback convolutional networks, Image Vis. Comput. 120 (2022) 104395.

[76] Z. Wang, Z. Liu, W. Wei, H. Duan, Saled: Saliency prediction with a pithy encoder-decoder architecture sensing local and global information, Image Vis. Comput. 109 (2021) 104149.

[77] T.R. Hayes, J.M. Henderson, Deep saliency models learn low-, mid-, and high-level features to predict scene attention, Sci. Rep. 11 (1) (2021) 1–13.

[78] M. Kümmerer, T. Wallis, M. Bethge, Deepgaze ii: Predicting fixations from deep features over time and tasks, J. Vis. 17 (10) (2017) 1147.

[79] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, in: Proceedings of IEEE International Conference on Computer Vision, 2009, pp. 2106–2113.

[80] T. Judd, F. Durand, A. Torralba, A benchmark of computational models of saliency to predict human fixations, MIT Technical Report, 2012.

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

[82] M. Kümmerer, T.S. Wallis, M. Bethge, Saliency benchmarking made easy: Separating models, maps and metrics, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 770–787.

[83] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, F. Durand, What do different evaluation metrics tell us about saliency models? IEEE Trans. Pattern Anal. Mach. Intell. 41 (3) (2018) 740–757.

[84] M. Kümmerer, T.S. Wallis, M. Bethge, Information-theoretic model comparison unifies saliency metrics, Proc. Natl. Acad. Sci. 112 (52) (2015) 16054–16059.

[85] A. Kroner, M. Senden, K. Driessens, R. Goebel, Contextual encoder–decoder network for visual saliency prediction, Neural Netw. 129 (2020) 261–270.

[86] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[87] C. Qing, H. Zhu, X. Xing, D. Chen, J. Jin, Attentive and context-aware deep network for saliency prediction on omni-directional images, Digit. Signal Process. 120 (2022) 103289, http://dx.doi.org/10.1016/j.dsp.2021.103289.

[88] S. Zabihi, H.R. Tavakoli, A. Borji, E. Mansoori, A compact deep architecture for real-time saliency prediction, Signal Process., Image Commun. 104 (2022) 116671.

[89] B. Aydemir, L. Hoffstetter, T. Zhang, M. Salzmann, S. Süsstrunk, Tempsal - uncovering temporal information for deep saliency prediction, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 6461–6470, http://dx.doi.org/10.1109/CVPR52729.2023.00625.

[90] R. Droste, J. Jiao, J.A. Noble, Unified image and video saliency modeling, in: Proceedings of European Conference on Computer Vision, 2020, pp. 419–435.

[91] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Proceedings of Advances in Neural Information Processing Systems, vol. 19, 2006.

[92] F. Qi, C. Lin, G. Shi, H. Li, A convolutional encoder-decoder network with skip connections for saliency prediction, IEEE Access 7 (2019) 60428–60438.

[93] S. Fan, Z. Shen, M. Jiang, B.L. Koenig, J. Xu, M.S. Kankanhalli, Q. Zhao, Emotional attention: A study of image sentiment and visual attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7521–7531.

[94] C. Wloka, T. Kunić, I. Kotseruba, R. Fahimi, N. Frosst, N.D. Bruce, J.K. Tsotsos, Smiler: Saliency model implementation library for experimental research, 2018, arXiv preprint arXiv:1812.08848.

[95] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S.C. Hoi, H. Ling, Learning unsupervised video object segmentation through visual attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3064–3074.