Aggregated Deep Saliency Prediction by Self-attention Network

Ge Cao, Qing Tang, and Kang-hyun Jo(✉)

University of Ulsan, Ulsan 44610, Republic of Korea
acejo@ulsan.ac.kr

Abstract. The data-driven method has recently obtained great success on saliency prediction thanks to convolutional neural networks. In this paper, a novel end-to-end deep saliency prediction method named VGG-SSM is proposed. This model identifies three key components: feature extraction, self-attention module, and multi-level integration. An encoder-decoder architecture is used to extract the feature as a baseline. The multi-level integration constructs a symmetric expanding path that enables precise localization. Global information of deep layers is refined by a self-attention module which carefully coordinated with fine details in distant portions of a feature map. Each component surely has its contribution, and its efficiency is validated in the experiments. Additionally, In order to capture several quality factors, the loss function is given by a linear combination of some saliency evaluation metrics. Through comparison with other works, VGG-SSM gains a competitive performance on the public benchmarks, SALICON 2017 version. The PyTorch implementation is available at https://github.com/caoge5844/Saliency.

Keywords: Saliency prediction · Self-attention · Multi-level integration

1 Introduction

Capturing the salient area in a scene is an instinctive ability of human beings. For visual saliency, it describes the spatial location which attracts the observer most. When observing a graph without any special tasks, as an elusive process, humans can't pay attention to every portion with the same intensity. Many works show that computational saliency can be found usages in a wide range of applications like object recognition [1], tracking regions of interest [2], and image retargeting [3] and so on.

With the advent of the deep neural network, saliency prediction also achieved great success thanks to generous data-driven methods and large annotated datasets [4]. Generally, computational saliency models predict the probability distribution of the location of eye attention over the images. Visual saliency data are traditionally colected by eye-trackers [5], more recently with mouse clicks [4]. No matter which kind of method is used to collect the saliency data, where human observers look in the images is regarded as the ground truth to estimate the accuracy of the predicted saliency maps. Through the computation of the proposed model, the predictions use various evaluation metrics to evaluate how best of a saliency model. The work by [6] broadly classified the various metrics as location-based or distribution-based. Though a large variety of metrics to evaluate saliency prediction maps exist, the main difference between them

D.-S. Huang and P. Premaratne (Eds.): ICIC 2020, LNAI 12465, pp. 87–97, 2020.
https://doi.org/10.1007/978-3-030-60796-8_8

concerns the ground-truth representation. In this paper, seven different evaluation metrics are used to analyze and evaluate the proposed model.

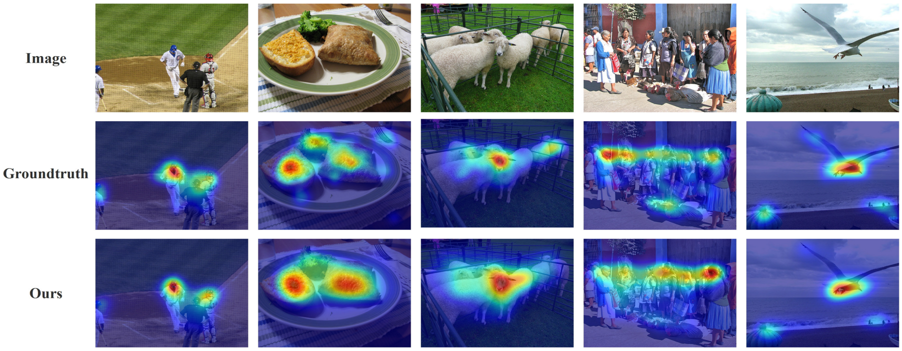

Fig. 1. Example results of the proposed method on images from SALICON dataset.

A novel end-to-end saliency prediction architecture is proposed to predict the saliency maps in this paper. Three key components in this architecture are identified respectively. First is the encoder-decoder architecture which directly extracts feature information. The second component is the self-attention module. The proposed model incorporates a Self-attention module that focuses on global, long-range dependencies to refine the details at every location. Each pixel in the feature maps can carefully coordinate with distant portions in the feature map, not limit to convolutional computation. In the third aspect, multi-level integration is constructed to reuse input feature maps for more local semantic information. Except for structural modify, the combination loss function outperform other loss function used single metric. The paper makes the following contributions:

1. This paper proposes a novel end-to-end saliency prediction method called VGG-SSM. The whole architecture is divided into separate components and analysis their efficiency respectively.
2. Self-attention module is incorporated with encoder-decoder based architecture to enhance global saliency information. The multi-level integration also improves the ability in local feature extraction.
3. The loss function used is formulated by some existing saliency metrics. The combined loss function makes multiple competing metrics be satisfied in concert.

Figure 1 shows examples of saliency maps predicted by the proposed method, which called the Saliency Self-attention Model (SSM), compared with ground truth saliency maps obtained from eye fixation. The proposed method is validated on publicly available datasets: SALICON. Experiments and evaluations results show that the proposed method improves the predictions.

The remaining content is organized as follows. Section 2 summarized the related work. The details of each component in the whole architecture and the loss functions used are introduced in Sect. 3. Section 4 provides the experiments details and results. Finally, Sect. 5 concludes the paper.

2 Related Work

Previous work on saliency prediction focused on low-level features. Far-reaching work by Itti [7] construct the first model to predict the saliency on images, which relied on color, intensity, orientation maps, and integrated them to get a global saliency map. After this seminal work, generous complementary methods about combining the low-level features were put forward. Judd [5] collected eye-tracking data to learn a model of saliency-based on low, middle, high-level features. Borji [8] combined low-level feature of previous best bottom-up models with top-down cognitive visual features and learn a direct mapping from those features to human eye fixations.

Same to other related fields of computer vision, deep learning solution achieved a far superior performance once it was proposed on saliency detection. And with the continuous progress of deep learning techniques, especially the success of Convolutional architectures, the performance of saliency detection is still steadily improving. *Ensemble of Deep Networks* (eDN) model by Vig et al. [9], one of the first proposals using a data-driving approach and richly-parameterized model, successfully predict image saliency map and outperform the previous work. After this proposal, many works based on convolutional neural networks emerged. Cornia et al. [10] explored combining CNN with recurrent architectures that focus on the most salient regions of the input image to iteratively refine the predicted saliency map. Pan et al. [11] introduced the Generative Adversarial Network into saliency detection. Their work used the generator to predict saliency maps that resemble the ground truth, and the discriminator to judge the authenticity of the saliency map. Recently, Reddy et al. [12] identified input features, multi-level integration, readout architecture, and loss function and proposed neater, minimal, more interpretable architecture, and achieved state-of-the-art performance on the SALICON [4], the largest eye-fixation dataset. This dataset contributed the availability of sufficient data and designed a mouse-contingent multi-resolutional paradigm to enable large-scale data collection.

This paper proposes a network architecture combining with attention mechanisms, which captures global dependencies. In particular, self-attention [13], also called intra-attention, applies in the natural language process, calculates the response at a position in a sequence by attending to all positions within the same sequence. Zhang et al. [14] introduced the self-attention module for image generation tasks. The proposed architecture also combines the self-attention module to efficiently find global and large-range dependencies within saliency maps.

3 Proposed Architecture

In this section, we introduce the proposed architectures, called SSM (Saliency Self-attention Model).

In general, the whole architecture adopts the convolutional encoder-decoder architecture. Section 3-A shows the detail of the network. The main innovation is the self-attention module, which is described in Sect. 3-B. Section 3-C shows the details of multi-level integration. The combination of evaluation metrics is used to evaluate the proposed network, and it is indicated in Sect. 3-D. Figure 2 shows the architecture of the proposal.

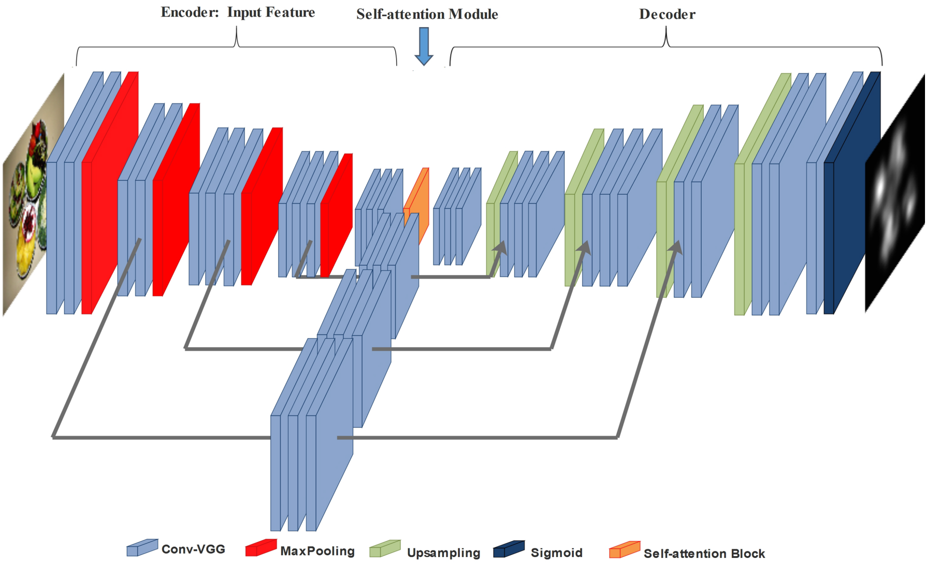

Fig. 2. The overview of the proposed Saliency Self-attention Model. After computing multi-scale feature maps on the inputs image through the encoder, a self-attention module based on attention mechanism is used to improve the global feature. Through the decoder, the model output the saliency prediction maps.

3.1 Overall Structure

The overall structure of the proposed network is introduced in this part. For saliency prediction, the fully convolutional framework achieves a great performance. As illustrated in Fig. 2, the whole network could be divided into three parts. The first is the feature maps extraction part, which can encode the input image and generate multi-scale feature maps. The second i the self-attention module we show in the next part. The third is the decoder, which upsamples the feature map to the same size with input image. The input size is initially resized to 256 ×256 and the initial channel is 3. In the encoder part, the network is identical in architecture to VGG16 [15] except the final max-pooling layer and three fully connected layers. Through the 13 convolutional layers and 4 max-pooling layers, the last layer of encoder have a small feature map with 16 ×16. And then the feature maps are fed into the self-attention module. For the decoder part, its layers' order is reversed with the encoder, with the max-pooling layers replaced by upsampling to successively restore feature maps' size. At the final of the network is a 1 ×1 convolutional layer with sigmoid non-linearity which ultimately produces the predicted saliency maps. There also have three U-Net like architecture that concatenates the same scale feature maps in encoder and decoder. Except for the weights of the encoder which are initialized with VGG-16 models pre-trained on ImageNet [16], other components' weights are randomly initialized. Hence VGG-SSM is used as the name for the proposed model.

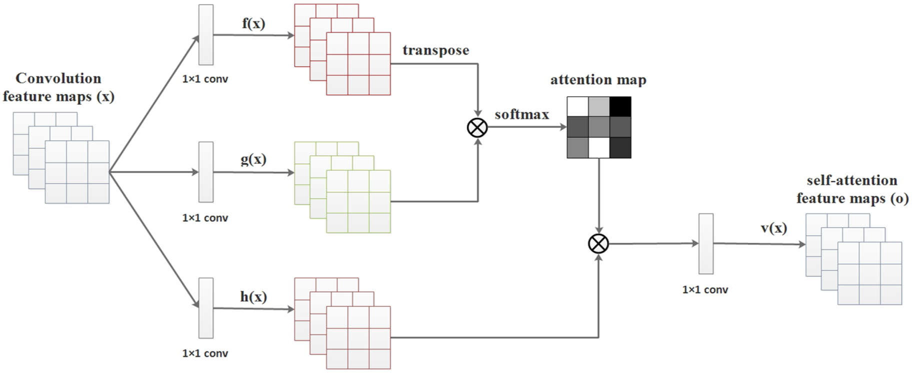

Fig. 3. The proposed self-attention module for VGG-SSM. The ⊗ denotes matrix multiplication.

3.2 Self-attention Module

Most saliency prediction models are built using CNN (Convolutional Neural Network) or RNN (Recurrent Neural Network). Unlike convolutional and recurrent operations, which both focus on building blocks that process local feature at a time, a non-local model [17] is adapted to combine self-attention with the previous part's network. Non-local means computing a weighted mean of all pixels in an image or a feature map. It allows distant pixels to contribute to the filtered response at a location based on patch appearance similarity. The self-attention module makes pixels in the feature map connect with all other pixels, no matter how distant. The approaches of the self-attention module are shown in Fig. 3. The input feature maps $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ from the last layers of the encoder is firstly transformed into two feature spaces with 1 ×1 convolution.

$$f(\boldsymbol{x}) = \mathbf{W_f} * \boldsymbol{x},\ g(\boldsymbol{x}) = \mathbf{W_g} * \boldsymbol{x} \tag{1}$$

where $*$ denotes convolutional opration, $\mathbf{W_f}$ and $\mathbf{W_g}$ are the 1 ×1 convolution kernels with C_1 channels. So $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ could be represented as $f(\boldsymbol{x}), g(\boldsymbol{x}) \in \mathbb{R}^{H \times W \times C_1}$. Then the attention map could be computed as Eq. 2.

$$\beta = \exp(\mathbf{s})/(\sum_{\mathrm{i=1}}^{\mathrm{N}} \exp(\mathbf{s})) \tag{2}$$

where $\boldsymbol{s} = f(\boldsymbol{x})^T g(\boldsymbol{x})$, in which $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ have been reshaped to $\{H \times W \times C_1\}$, $N = H \times W$. So after computing the softmax operation, the shape of β and $\boldsymbol{s}$ is the same with $\{H \times W, H \times W, C_1\}$. For memory efficiency, the method reduce the channel to $C_1 = C/k$ when computing 1 ×1 convolution, and choose $k = 8$ (i.e., $C_1 = C/8$) following [14] as the default value.

$$\boldsymbol{o} = \beta \otimes h(\boldsymbol{x})^T * \mathbf{W_v} \tag{3}$$

where $h(\boldsymbol{x}) = \mathbf{W_h} * \boldsymbol{x}$. In the above formulation, $\mathbf{W_f} \in \mathbb{R}^{C_1 \times C}$, $\mathbf{W_g} \in \mathbb{R}^{C_1 \times C}$, $\mathbf{W_h} \in \mathbb{R}^{C_1 \times C}$, $\mathbf{W_v} \in \mathbb{R}^{C_1 \times C}$. Additionally, the output is multiplied by a learnable scale parameter and added with the input feature map to avoid the information-vanishing in the computed process of the network. Hence the final output is given by Eq. 4.

$$\boldsymbol{y} = \gamma \boldsymbol{o} + \boldsymbol{x} \tag{4}$$

Where γ is initialized to 0. The learnable γ is introduced to make the network learn the optimal weights for non-local evidence instead of accepting it directly.

3.3 Multi-level Integration

VGG-SSM employs a U-Net [18] like architecture that symmetrically expands the input feature maps after the first upsampling layer decoder. Feature maps in encoder and decoder with the same scale are concatenated to avoid information-vanishing. As shown in Fig. 2, there are three integrations in the whole architecture. Every step of expansion is composed of an upsampling of the feature map and concatenation with the same scale feature map from the encoder. Additionally, three 3 ×3 convolutional layers followed by ReLU are used to gradually extract deeper features at the original scale. The channels and scales are the same as the parameters of the convolutional layer before max-pooling.

3.4 Loss Function

The loss function evaluates the performance of the predicted saliency map compare with the ground truth. This paper uses a linear combination of three different saliency evaluation metrics: Kullback-Leibler Divergence (KLdiv), Pearson Cross-Correlation (CC), and Similarity (SIM). The new loss function is defined as follows:

$$L\left(\widehat{\boldsymbol{I}},\boldsymbol{I}\right) = \alpha \mathrm{KLdiv}\left(\widehat{\boldsymbol{I}},\boldsymbol{I}\right) + \beta CC\left(\widehat{\boldsymbol{I}},\boldsymbol{I}\right) + \gamma SIM\left(\widehat{\boldsymbol{I}},\boldsymbol{I}\right) \tag{5}$$

where $\widehat{\boldsymbol{I}}$ and $\boldsymbol{I}$ are predicted saliency maps and the ground truth.

KLdiv is an information-theoretic measure of the difference between two probability distributions:

$$\mathrm{KLdiv}\left(\widehat{\boldsymbol{I}},\boldsymbol{I}\right) = \sum\nolimits_i \boldsymbol{I} log\left(\epsilon + \frac{\boldsymbol{I}}{\widehat{\boldsymbol{I}} + \epsilon}\right) \tag{6}$$

where i indexes the i^{th} pixel and ϵ is a regularization constant. So KLdiv is computed on pixel-level.

CC is a statistical method used generally in the sciences for measuring how corrected or dependent two variables are.

$$CC\left(\hat{\boldsymbol{I}},\boldsymbol{I}\right) = \sigma\left(\hat{\boldsymbol{I}},\boldsymbol{I}\right)/\left(\sigma\left(\hat{\boldsymbol{I}}\right) \times \sigma(\boldsymbol{I})\right) \tag{7}$$

where $\sigma\left(\widehat{\boldsymbol{I}},\boldsymbol{I}\right)$ denotes the covariance of $\widehat{\boldsymbol{I}}$ and $\boldsymbol{I}$.

SIM, also referred to as histogram intersection, measures the similarity between two distributions. SIM is computed as the sum of the minimum values at each pixel, after normalizing the input maps. Given a saliency map $\widehat{\boldsymbol{I}}$ and its ground truth $\boldsymbol{I}$:

$$SIM\left(\widehat{\boldsymbol{I}},\boldsymbol{I}\right)=\sum\nolimits_{i} min\left(\widehat{\boldsymbol{I}},\boldsymbol{I}\right), where \sum\nolimits_{i}\widehat{\boldsymbol{I}}=\sum\nolimits_{i}\boldsymbol{I}=1 \tag{8}$$

iterating over discrete pixel location *i*.

The results of experiments using the proposed combined loss function are shown in Sect. 4-C.

4 Experiments and Results

The experiments' details and comparison results are shown in this section. Section 3-A shows the detail of the training process and other implementation details. Section 3-B describes the contributions of each component. The comparison between different loss functions is shown in Sect. 3-C. Finally, Sect. 3-D compares the proposed method with other state of the art. Here describe each part in detail.

4.1 Experimental Setup

Datasets: For training the proposed model and verify the results, we use the largest available dataset, SALICON [5] for saliency prediction. The dataset consists of 10,000 images for training, 5,000 images for validating, and 5,000 images for testing, taken from Microsoft COCO dataset [19]. We train the proposed model on SALICON datasets with 10,000 training images and use 5,000 images for validating. The ground truth maps are recorded by eye-tracker. It also provides the eye fixation simulated by mouse-click, but this part of the data is not used in the proposed method. The ground truth maps of test dataset are not available publicly, so the prediction only could be tested on the newest release, SALICON 2017, from the LSUN challenge.

Loss parameters: The parameters in the proposed loss function, α, β, γ are set to 10, -1 and -1 to balance the contribution of each components of loss function individually. Differently from the KLdiv loss which value should be minimized, the *CC* and the *SIM* loss is maximized to obtain the higher performance in saliency prediction. The values of the balancing weights are chosen by the target of obtaining good results on all evaluation metrics and by the numerical variation range single metrics have at convergence.

Evaluation metrics: This paper uses seven different evaluation metrics [6] adopted by SALICON to evaluate the proposed model. Among them, KLdiv, *CC* and *SIM* have been demonstrated in Sect. 3-D. AUC is the area under the ROC curve, the most widely used metric for evaluating saliency maps. The shuffled AUC metric (sAUC) samples negatives from other images, instead of uniformly at random. The Normalized Scanpath Saliency (NSS) is introduced to the saliency community as a simple correspondence measure between saliency maps and ground truth. Information Gain (IG) measures saliency model performance beyond systematic bias as an information theoretic metric.

Implementation Details: The training process resizes the input images into 256 × 256 resolution and trains VGG-SSM 30 epochs with the learning rate starting from 1e−4 and reducing after 3 epochs. The ADAM optimization algorithm is employed to train the whole network with the default batch size is set to 24. All the training and testing are conducted on one NVIDIA GeForce GTX 1080 Ti GPU with 11 GB memory.

4.2 Contribution of Each Component

The contributions of the self-attention module and the multi-level integration on SALICON test sets are described in this part. And the proposed combined loss function is used in the evaluation. To this end, this paper constructs three different components: the plain encoder-decoder architecture can be regarded as a baseline (This paper use VGGM to represent it), the self-attention module, and the multi-level integration. Table 1 illustrates the results of VGGM, VGGM plus self-attention module (Here use VGGSAM to represent), and the final version of the proposed model with all its components. As Table 1 shown, the results show that the overall architecture obtains the best grades on every evaluation metric and each component gives a great contribution to the final performance. It's obvious that the overall architecture makes a constant improvement on all metrics. For instance, the baseline achieved a result of 0.279 in terms of KLdiv, while it achieves a relative improvement of 5.0% with a self-attention module, and the result is improved by 1.5% when adding multi-level integration.

Table 1. Performance comparison of different version on test set of SALICON-2017.

Model	KLdiv ↓	CC ↑	AUC ↑	NSS ↑	SIM ↑	IG ↑	sAUC ↑
VGGM	0.279	0.854	0.858	1.839	0.745	0.750	0.727
VGGSAM	0.265	0.869	0.860	1.891	0.759	0.795	0.732
VGGSSM	**0.261**	**0.875**	**0.861**	**1.909**	**0.764**	**0.802**	**0.733**

4.3 Comparison Between Different Loss Functions

In this part, this paper verifies the effects of using different combinations of the loss function on SALICON validation set.

In Table 2, we compare the proposed loss function with its components individually as loss functions (KLdiv, CC, SIM). The results on SALICON validation set show the superiority of the proposed loss function. Although each single metric gain the best performance on its own evaluation term, the other evaluation terms obtain unsatisfactory results. Apparently, the combined loss function proposed to obtain an excellent trade-off among all the evaluation terms.

Table 2. Comparison between proposed loss function and its components using individually as loss function on Validation set of SALICON-2017.

Loss Function	KLdiv ↓	CC ↑	SIM ↑
KLdiv	**0.249**	0.872	0.764
CC	1.145	**0.881**	0.760
SIM	1.133	0.878	**0.773**
KLdiv+CC+SIM	0.251	0.876	0.769

Table 3 illustrates the result by adding CC and SIM to the KLdiv loss. Though we obtain better results when adding CC loss to KLdiv loss on CC evaluation metric, it brings reductions in other evaluation metrics. Higher performance can be achieved by adding CC and SIM terms to the loss. KLdiv+CC+SIM loss get all the results to value bold, which represent the best result upon different loss function.

Table 3. Comparison results between various loss functions on validation set of SALICON-2017.

Loss Function	KLdiv ↓	CC ↑	SIM ↑
KLdiv	0.249	0.872	0.764
KLdiv+CC	**0.247**	0.875	0.767
KLdiv+CC+SIM	0.251	**0.876**	**0.769**

4.4 Comparison with State-of-the-Art

The proposed models are compared with state of the art on SALICON test sets quantitatively. Table 4 shows the results in terms of KLdiv, CC, AUC, NSS, SIM, IG, and sAUC. VGG-SSM achieves great performance on two different metrics and outperforms other works by a large margin on KLdiv and IG. The proposed model also obtains competitive performance on other metrics.

Table 4. Performance comparison with state-of-the-art on test set of SALICON-2017.

Model	KLdiv ↓	CC ↑	AUC ↑	NSS ↑	SIM ↑	IG ↑	sAUC ↑
VGG-SSM (Ours)	**0.261**	0.875	0.861	1.909	0.764	**0.802**	0.733
EMLNET [20]	0.520	0.886	**0.866**	**2.050**	0.780	0.736	**0.746**
SAM-Resnet [10]	0.610	0.899	0.865	1.990	0.793	0.538	0.741
MSI-Net [21]	0.307	0.889	0.865	1.931	0.784	0.793	0.736
GazeNet [22]	0.376	0.879	0.864	1.899	0.773	0.720	0.736
ryanDINet [23]	0.777	**0.906**	0.864	1.979	**0.800**	0.347	0.742
Jinganu [23]	0.389	0.879	0.862	1.902	0.773	0.718	0.733
Lvjincheng [23]	0.376	0.856	0.855	1.829	0.705	0.613	0.726
Charleshuhy [23]	0.288	0.856	0.863	1.845	0.768	0.770	0.732

5 Conclusions

In this paper, a saliency self-attention Model VGG-SSM upon encoder-decoder architectures is proposed to predict saliency maps on natural images. This paper identifies three important components and does experiments to demonstrate the contribution of each part. The main novelty is the proposal of the self-attention module and its efficiency has been proved. Additionally, this paper compares the results of kinds of loss functions and validates the efficiency of combination loss function through an extensive evaluation. VGG-SSM achieves competitive results on SALICON test set. A similar method could be significant for other tasks that involve image refinement. Furthermore, the proposed model can be combined with a more recurrent network for potential further improvements.

References

1. Schauerte, B., Richarz, J., Fink, G.A.: Saliency-based identification and recognition of pointed-at objects. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4638–4643 (2010)
2. Frintrop, S., Kessel, M.: Most salient region tracking. In: 2009 IEEE International Conference on Robotics and Automation, pp. 1869–1874 (2009)
3. Takagi, S., Raskar, R., Gleicher, M.: Automatic image retargeting, vol. 154, no. 01, pp. 59–68 (2005)
4. Jiang, M., Huang, S., Duan, J., Zhao, Q.: SALICON: saliency in context, no. 06 (2015)
5. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 2106–2113 (2009)
6. Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., Durand, F.: What do different evaluation metrics tell us about saliency models? arXiv e-print, arXiv:1604.03605 (2016)
7. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
8. Borji, A.: Boosting bottom-up and top-down visual features for saliency estimation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 438–445 (2012)
9. Vig, E., Dorr, M., Cox, D.: Large-scale optimization of hierarchical features for saliency prediction in natural images. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2798–2805 (2014)
10. Cornia, M., Baraldi, L., Serra, G., Cucchiara, R.: Predicting human eye fixations via an LSTM-based saliency attentive model. IEEE Trans. Image Process. **27**(10), 5142–5154 (2018)
11. Pan, J., et al.: SalGAN: Visual Saliency Prediction with Generative Adversarial Networks. arXiv e-prints, arXiv:1701.01081 (2017)
12. Reddy, N., Jain, S., Yarlagadda, P., Gandhi, V.: Tidying Deep Saliency Pre diction Architectures. arXiv e-prints, arXiv:2003.04942 (2020)
13. Parikh, A.P., Täckström O., Das, D., Uszkoreit, J.: A Decomposable Attention Model for Natural Language Inference. arXiv e-prints, arXiv:1606.01933 (2016)
14. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-Attention Generative Adversarial Networks. arXiv e-prints, arXiv:1805.08318 (2018)
15. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv e-prints, arXiv:1409.1556 (2014)

16. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
17. Wang, X., Girshick, R., Gupta A., He, K.: Non-local Neural Networks. arXiv e-prints, arXiv:1711.07971 (2017)
18. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. arXiv e-prints, arXiv:1505.04597 (2015)
19. Lin, T.-Y., et al.: Microsoft COCO: Common Objects in Context. *arXiv e-print*, arXiv:1405.0312 (2014)
20. Jia, S., Bruce, N.D.B.: EML-NET: An Expandable Multi-layer NETwork for Saliency Prediction. arXiv e-prints, arXiv:1805.01047 (2018)
21. Kroner, A., Senden, M., Driessens, K., Goebel, R.: Contextual Encoder-Decoder Network for Visual Saliency Prediction. arXiv e-prints, arXiv:1902.06634 (2019)
22. Che, Z., Borji, A., Zhai, G., Min, X., Guo, G., Le Callet, P.: How is gaze influenced by image transformations? dataset and model. IEEE Trans. Image Process. **29**, 2287–2300 (2020)
23. LSUN 2017. https://competitions.codalab.org/competitions/17136#results