Text-Guided Saliency Prediction Model

Yixin Huang, Shanghai Jiao Tong University

***Abstract*—Text-guided saliency prediction is a new field of interest that lacks exploration. In this paper, a simple text-guided saliency model is propsed and tested on the SJTU-TIS dataset. Text information is incorporated into the pure saliency prediction model through a cross-attention layer or a tailored loss function. In avoidance of overfitting, the model is firstly trained on SALICON and then on SJTU-TIS. The influence of salient text and non-salient text are further compared to bring about further insight about the saliency prediction task.**

***Index Terms*—saliency model, text guided, SJTU-TIS.**

I. Introduction

SALIENCY prediction, which is the task of predicting human vision's attention on images, is of key importance in computer vision. Traditional saliency prediction methods have followed biological evidence and fail to take varying reasons that may lead to completely distinct saliency maps into consideration. With the advent of deep neural networks, the performance of saliency prediction models skyrocket. However, previous papers focuses only on pure image input. In real world situation, viewers' attention will be influenced more or less by non-image information. Recently, [1] opens up a new discussion on the influence of text-guidance on visual attention, which is a subset of the environmental influence on visual attention. In this paper, we are going to follow [1]'s guidance and explores further on this issue.

II. Base Model Introduction

A. Saliency Prediction Model

The SAM model proposed by [2] is one of the founding model in deep saliency prediction. After extracting image features by ResNet/VGG, SAM utilizes its Attentive Convolutional LSTM to do saliency prediction. The LSTM works by sequentially updates its internal state. Followed by the Attentive Convolutional LSTM is a simple decoder that outputs the saliency maps.

A more recent model is the TranSalNet introduced by [3]. It follows a similar scheme as SAM. ResNet50 is used as an image encoder and the feature vector is a concatenation of the output of the last 3 layers. Then, instead of Attentive Convolutional LSTM, a transformer based architecture is adopted when generating saliency map from the extracted features, leading to remarkable improvements on model performance. In this paper, the model is based on the SOTA model TranSalNet.

B. Text Feature Extraction Model

The incorporation of text features into the model is a major difference from previous tasks. In homework 1, BLIP[4] is utilized to do text embedding. Besides encoding text with pretrained BERT, BLIP also proposes a method to encode text in an image-aware fashion. However, due to the limitation of GPU storage, pure text embedding is adopted in the paper.

III. Dataset introduction

A. SALICON

SALICON[5] is a classic dataset on saliency prediction tasks. The dataset contains 10000 images and their corresponding fixation and saliency maps. SALICON will be used to pretrain the model to avoid overfitting on the newly introduced SJTU-TIS dataset. Due to the lack of text attached, when being inputted into the model tailored for text-related model, an empty string is appended. Ideally when training for SALICON, the model should act as without the text-related parts and the text-related weights are to be finetuned on SJTU-TIS.

B. SJTU-TIS

SJTU-TIS[2] contains pictures with texts attached. According to the content of texts, the dataset could be split into 4 categories. Firstly, the image type 'salient' is described by texts that focus on the general object of the images. Secondly, 'non-salient' describes trivial detail of the images. Thirdly, 'all' attaches text description that could either be salient of non-salient. Lastly, 'pure' image type encompasses images without text description. Experiments are conducted on these four categories of images to explore the influence of the content of texts on human visual attention. Plus, different from the images in SALICON, images in SJTU-TIS has varying sizes. Hence, the images should all be resized before entering the encoder model.

IV. Model Design

A. Model structure modification

Inspired by BLIP, which encode image data into text embedding by doing cross attention, the text attached to the images is incorporated into the model by performing cross attention with images. The formula for cross-attention is shown as below, where V denotes the sentence vector and X denotes the image feature matrix.

$$Q = W_Q V \tag{1}$$

$$K = W_K X \tag{2}$$

$$V = W_V X \tag{3}$$

$$A = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

The main obstacle is the size mismatch. I choose to simply map the sentence embedding to an image-like vector when performing matrix multiplication with the weight matrix, which corresponds to the first line in the cross attention formula. Since the output embedding of BLIP encoder is of a fixed size 256, this could easily be achieved using a Linear layer.

As regards the place to insert the cross attention layer, a possible idea is to place it at the output of the encoder. Since the intuition of doing cross attention is to let the image features be aware of the text features and the output of the ResNet is the extracted features of images, it is natural to place a cross-attention here. Since the decoder of TranSalNet makes use of the features of the last 3 layers, the cross-attention could be applied to all layers to make all the image features aware of the texts attached. The influence will be discussed later. Plus, the cross attention layer could be placed inside the trans-encoder. At this place the image features are transformed to a 3D tensor and doing the cross attention on a 3D instead of 4D tensor will be more natural and easier to implement. Thirdly, the text feature could simply be used to introduce new losses and hence guide the learning, as is in [6].

As of the third place to insert text information, I will utilize the contrastive loss proposed in [7]. [7] mainly tackles the problem of image and text matching and our saliency map problem could be converted to this problem to some extent. If we use the saliency map as a mask on the original image, then the object left should mainly be the object described in the text provided. To use the loss proposed in this task, the image and text vector should be of the same hidden size. Hence, the text feature should go through another linear layer to be projected to the embedded space of the images.

More specifically, the contrastive loss requires both the image and the text to be able to reproduce one another and the images and text to be matched. The loss could be separated into 2 parts, the matching part and the generation part. For the matching part, a hinge-based triplet ranking loss is adopted with emphasis on hard negatives, i.e., the negatives closest to each training query. For the generation part, the learned visual representation should also has the ability to generate sentences that are close to the ground-truth captions. the loss is designed to maximize the log-likelihood of the predicted output sentence. The final loss is an addition of these two parts.

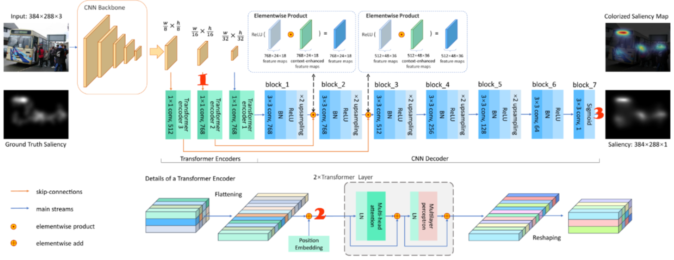

Fig. 1. Network structure of TranSalNet and possible places to insert text features. The schemes to insert text features are numbered according to the introduction sequence

B. Loss function design

The loss function in this model is generally based on what is used in SAM. From experimental experience, MSE loss is added to the loss function to let the model better learn the saliency map. The final loss function is as follows:

$$\begin{aligned}\mathcal{L} = w_{KL}\mathcal{L}_{KL} + w_{MSE}\mathcal{L}_{MSE} - w_{CC}\mathcal{L}_{CC} - w_{NSS}\mathcal{L}_{NSS} \\ -w_{Sim}\mathcal{L}_{Sim} + w_{Contra}\mathcal{L}_{Contra} + b\end{aligned} \tag{5}$$

where $\mathcal{L}_{KL}$ refers to KL-divergence between the outputs and the ground truth; $\mathcal{L}_{MSE}$ is the MSE loss; $\mathcal{L}_{CC}$ is the correlation coefficient; $\mathcal{L}_{NSS}$ is the normalized scanpath saliency; $\mathcal{L}_{Sim}$ is the tensor similarity loss and $\mathcal{L}_{Contra}$ is the contrastive loss mentioned before. There is also a bias term to let the overall loss near to 0. During training we could observe that if not with this bias, the loss stays stable at around -3 in the salient image task and couldn't lead to convergence in the following training epoches.The weights of these losses and the bias are hyperparameters to be tuned for each individual task.

V. Experiemnt

A. Experiment Setup

The optimizer chosen is the Adam Optimizer. Due to GPU memory restriction, the batch size is set to 8. Further, though the pretrained model performs well and steadily enough on SALICON dataset, when it is trained on SJTU-TIS, it's AUC has a strong tendency to converge to 0.5, which is a sign of training failure. Thus, most of the layers of the TranSalNet is frozen to ensure the general performance. The number of layers frozen will influence the final results.

Firsly, the model is trained on SALICON for 40 epoches to let the model be familar with this saliency prediction task. Secondly, the model is finetuned on the 4 categories of data separately since the reaction when given different kinds of description on the same picture should be different based on the level of saliency of the text description.

B. Experiment Result

In Table 1 is the pretrained model performance on SALICON dataset. The model may not be fully tuned due to time limitation since to train on SALICON, the batch size could only be set to 2 on my GPU. However, compared to directly training on SJTU-TIS, a pretrained model enhances the performance significantly.

TABLE I
Performance of the model Pretrained on SALICON

AUC	sAUC	CC	NSS
0.8263	0.6723	0.8534	1.5732

It is worth noting that during training, it is not always true that the four metrics are improved. Chances are that CC and NSS decrease when AUC, sAUC increase. Maybe a better loss function could be proposed to solve this problem.

Secondly, the model is finetuned on SJTU-TIS. In this part we adopt the first version of text feature insertion scheme mentioned in previous part. When direcly applying the model on SJTU-TIS, the result is as follows:

TABLE II
PERFORMANCE OF THE PRETRAINED MODEL ON SJTU-TIS

	AUC	sAUC	CC	NSS
pure	0.7692	0.6385	0.6906	1.3295
non salient	0.7026	0.6022	0.4546	1.2204
salient	0.7328	0.6065	0.6354	1.6565
all	0.7791	0.6270	0.6439	1.3094

Since this dataset has a severe tendency to lead to overfit, all models are just finetuned for 3 to 10 epoches on SJTU-TIS. However, the performance improvement is remarkable. After the finetune, the result is as follows:

TABLE III
PERFORMANCE OF THE FINETUNED MODEL ON SJTU-TIS

	AUC	sAUC	CC	NSS
pure	0.7942	0.6471	0.7249	1.5962
non salient	0.7289	0.6144	0.4880	1.3961
salient	0.7765	0.6332	0.6292	1.6857
all	0.7841	0.6421	0.6658	1.5777

To better illustrate the influence of the text features, one sample map comparison is illustrated below. Apparently, the level of saliency of the text provided makes a great difference to the saliency map prediction and that our model could react to the text features change appropriately. Also, since our model only uses pure text embedding instead of word embedding in a text-aware fashion, the performance may just be inferior. If the resources could be more abundant, the performance may be better. Still, the dataset has a strong tendency to lead to overfitting in the model. The predicted saliency map could be strongly identical to the ground truth. The provided sample pictures are on the test set and it is obvious that the predicted saliency map still is not as accurate as expected.

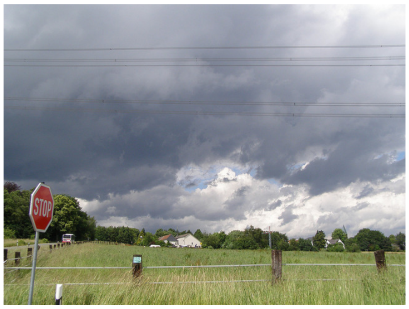

Fig. 2. The original picture for the following saliency map. The salient text is ’Dark clouds spread across the sky’ while the non-salient text is ’A stop sign near the highway’

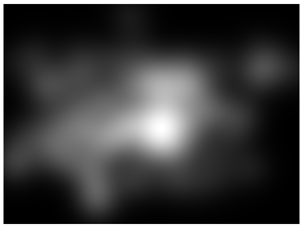

Fig. 3. salient ground truth

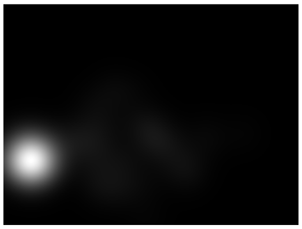

Fig. 4. non-salient ground truth

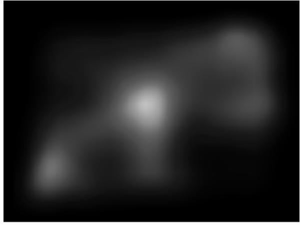

Fig. 5. salient prediction

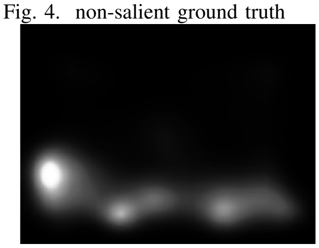

Fig. 6. non-salient prediction

C. Further discussion

Since non-salient text description is of more importance, the model performance on non-salient tasks will be further discussed. In Table IV, different cross attention layer implementation configuration at position in Fig.1 is investigated. The number of attention means the places where attention information is inserted and the layer information after the addition notation means the layers being unfreezed.

TABLE IV
PERFORMANCE OF THE FINETUNED MODEL ON SJTU-TIS NON-SALIENT TASKS WITH DIFFERENT STRUCTURE

	AUC	sAUC	CC	NSS
1 attention + the last 2 conv	0.7289	0.6144	0.4880	1.3961
3 attention + none	0.7237	0.6118	0.4873	1.3619
3 attention + the last 2 conv	0.7248	0.6124	0.4860	1.3793
3 attention + the last 4 conv	0.7222	0.6110	0.4824	1.3736
3 attention + all decoder	0.7206	0.6102	0.4852	1.3398

It is clear from the experiment result that based on the current model design, too many unfreezed layers lead to inferior performance.

Secondly we shall dive deeper into performance of cross attention layers inserted at position 2. Basically the performance is the same so no further discussion and exploration is needed.

TABLE V
PERFORMANCE OF THE MODEL ON SJTU-TIS NON-SALIENT TASKS WITH CROSS ATTENTION LAYER PLACED IN TRANSENCODER

AUC	sAUC	CC	NSS
0.7262	0.6131	0.4864	1.3833

Thirdly, we investigate the third text feature incorporation scheme. The result is provided in Table VI. Surprisingly this simple scheme leads to a non-trivial improvement in all evaluation metrics, which illustrates the success of the proposed task conversion. It is worth noting that the image-matching method used is proposed in 2019 and there are many pretrained model based method to tackle this issue. Due to GPU memory limitation, the experiment isn't conducted and is left for further study.

TABLE VI
PERFORMANCE OF THE MODEL ON SJTU-TIS NON-SALIENT TASKS WITH CONTRASTIVE LOSS INTRODUCED.

AUC	sAUC	CC	NSS
0.7571	0.6236	: 0.5105	1.5623

VI. CONCLUSION

In this paper, a simple text-guided saliency prediction model is introduced and tested on SJTU-TIS. Generally, pure image-based models fail to address non-salient tasks well enough while introducing a text information enhances performance remarkably. Further invesigation on this intriguing field is required to ensure better performance on this issue.

REFERENCES

[1] Y. Sun, X. Min, H. Duan, and G. Zhai, "The influence of text-guidance on visual attention," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2023, pp. 1–5.

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

[3] J. Lou, H. Lin, D. Marshall, D. Saupe, and H. Liu, "Transalnet: Towards perceptually relevant visual saliency prediction," *Neurocomputing*, vol. 494, p. 455–467, Jul. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2022.04.080

[4] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," 2022.

[5] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[6] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," 2018.

[7] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *ICCV*, 2019.