CLIP-SalGan: A Text-Guided Saliency Model Combining GAN and CLIP Metrics

Shanghai Jiao Tong University, *Qi Siyuan, 521030910012*

***Abstract*—This document proposes a Saliency model: CLIP-SalGan to predict saliency map of a picture with its text description. Namely, that is a text-guided saliency model. The prediction result for the same picture can be changed by different text description. I combine Generative Adversarial Network(GAN [1]) and Contrastive Language-Image Pre-Training Network(CLIP [2]) to train the model.**

***Index Terms*—Saliency Model, text-guided, GAN, CLIP.**

I. Introduction

HUMAN'S vision can be affected by **textual descriptions**. For the same picture, inform different descriptions of salient objects in advance, then our attention will be focused on different places. We can apply this vision mechanism to computer vision, using text descriptions to build more powerful saliency models.

Based on **SalGan** [3] model, I used **CLIP** network to extract features or text descriptions and fused them into the **generator** of SalGan model for training. Therefore I call my model **CLIP-SalGan**.

II. Related Work

A. SalGan

In SalGan model, generator is used to **predict** the saliency map from the raw pixels of the input image, and discriminator is used to **distinguish** whether a saliency map is predicted or real. If the prediction image and the real image are indistinguishable, then the model is successfully trained.

B. CLIP

CLIP is a multimodal model based on **contrast learning** [4]. The training data of CLIP is a text-image pair: an image and its corresponding text description, and it is hoped that through contrast learning, the model can learn the matching relationship of text-image pairs. Therefore, this model is appropriate to extract features of text description.

III. Model Design

Firstly Fig. 1 shows architecture of my model. And following is detailed introduction.

A. Text Feature Extraction

To begin with, I use the encoding metric of CLIP to extract text feature, which is used to input into generator.

B. Generator

Due to the complexity of the network in SalGan's paper and the limited computing resources, I have appropriately simplified the network and constructed it as follows:

(1) A **Conv-VGG** layer, which helps the model extract and represent useful features from the raw image data [5];
(2) A **Max Pooling** layer, which reduces the size of the feature map by selecting the maximum value within the local receptive field [6];
(3) A Conv-VGG layer again, but reduce the size of this layer appropriately;
(4) An **Upsampling** layer, which increases the spatial resolution of the input feature map [7];
(5) A **Conv-Scratch** layer, which is similar to Conv-VGG layer.
(6) A **text features conv** layer and **fusion features conv** layer, which are used to extract information of text and combination of image and text, respectively;
(7) A **Sigmoid** output layer, which converts the fused features into the final output image.

The generator structure makes full use of the advantages of deep learning in image and text feature extraction, and generates images associated with the input text description through effective feature fusion.

C. Discriminator

Similar to generator, I simplify the design in SalGan paper, and my architecture is as followed:

(1) A **Conv-VGG** layer;
(2) A **Max Pooling** layer;
(3) A **Fully Connecter** layer, which is responsible for the integration and convergence of features and also plays a key role in the decision-making process of the network [8];

The discriminator serves as a critical component that challenges the generator to produce increasingly realistic data. Through a **continuous adversarial process**, it guides the generator towards better performance and ensures the generation of high-quality, realistic synthetic data.

This paper was produced by Qi Siyuan, 521030910012, Shanghai Jiao Tong University, majoring in CS(IEEE)
Finished in December, 2023.

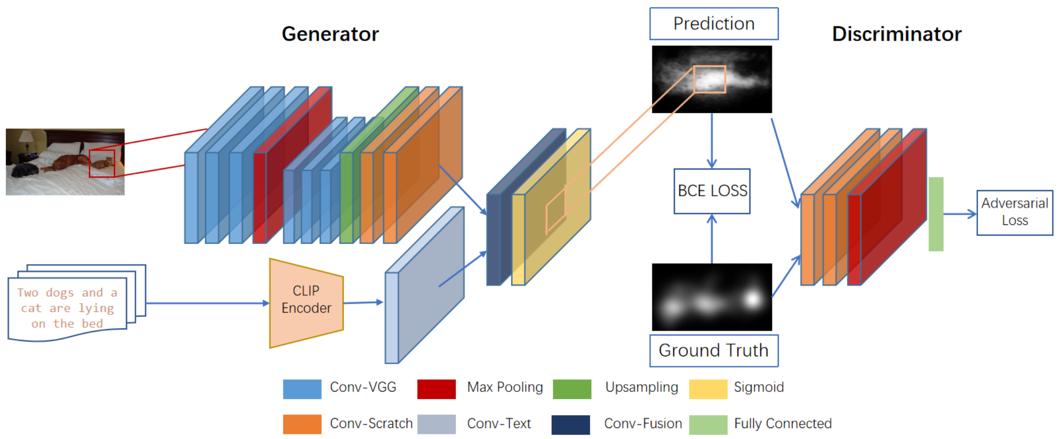

Fig. 1: Architecture of CLIP-SalGan

D. Loss Function

For my model, I mainly use the following two loss function:

(1) **Content Loss**: Due to the involvement of multiple pixels in the process of generating saliency maps, it is more appropriate to treat each prediction value as independent of the others. Therefore, we can apply an element-wise sigmoid to each output of the final layer. This allows us to interpret pixel-wise predictions as probabilities of independent binary random variables. In this case, an appropriate loss function is **binary cross-entropy (BCE)** [9], which is the average of individual binary cross-entropy values across all pixels:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{j=1}^{N}(S_j \log(\hat{S}_j) + (1 - S_j)\log(1 - \hat{S}_j)) \tag{1}$$

(2) **Adversarial Loss**: When updating the parameters of the generation function, the stability and convergence speed of the adversarial training can be improved by using the loss function of the **combination** of the discriminator's error and cross-entropy relative to Ground Truth, so the final loss function in the adversarial training can be expressed as followed:

$$\mathcal{L} = \alpha\mathcal{L}_{BCE} + L(D(I, \hat{S}), 1) \tag{2}$$

Then loss for discriminator is given as followed:

$$\mathcal{L}_D = L(D(I, S), 1) + L(D(I, \hat{S}), 0) \tag{3}$$

E. Hyper-Parameters Setting

Since the training of SalGan model takes so much time, I reviewed the literature to set the model's hyperparameters sensibly, and then skipped the fine-tuning process. Table. I gives the hyper-parameters I use.

TABLE I: Hyper-Parameters Setting

Epoches	**Batch Size**	**Learning Rate**	**Adam Beta**
50	16/32	0.002	(0.5, 0.999)

IV. Training Process

A. First Attempt

Firstly I trained the SalGan model on the total dataset, and divide the data into training, validation, and test sets in the ratio of **70%, 15%, and 15%** respectively.

It's worth noting that **different textual descriptions of the same image should be ensured to be assigned to the same set** within the training, validation, and test sets. Doing that can ensure that the model effectively learns useful information, prevents overfitting, and avoids having images in the test set that are already present in the training set.

After completing the training, I separately tested the model's saliency performance on the test set for salient text (Type 2), non-salient text (Type 3), general description (Type 4 & Type 1), pure images, and the entire dataset, obtaining the Table II:

Score Type	AUC	sAUC	CC	NSS
Type 2	0.7631	0.6317	0.5485	0.4188
Type 3	0.6956	0.5978	0.3128	0.4232
Type 4 & Type 1	0.7486	0.6245	0.5671	0.3408
Pure	0.7829	0.6416	0.6632	0.2497
ALL	0.7521	0.6262	0.5489	0.3389

TABLE II: Result for First Model

The results are disappointing; it seems that the model performs better on inputs without text, with only the NSS score showing improvement when text is included.

Furthermore, the most prominent demonstration of text-guided effects is observed when inputting salient and non-salient textual descriptions for the same image and **assessing whether there are noticeable differences** in the detected saliency maps. Fig. 2 shows one example in test set using my model.

Althouth my model doesn't predict the non-salient object "white lamps", at least different text descriptions lead to different outputs, meaning text does play a role in **guiding**.

The poor performance of my model may be because text features are too weak compared to image features. Therefore,

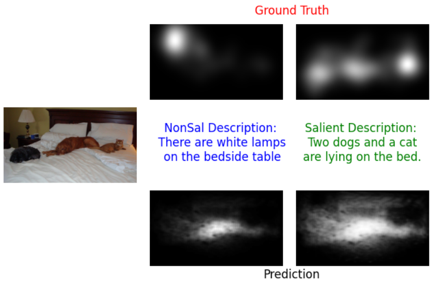

Fig. 2: Performance on Test Set

the guidance function is also weak because image features dominate.

Considering that, I retrain the model on different data segmentations for three groups:

(1) **Salient Group.** In this case I use Type2 data along with pure pictures($300 + 300 = 600$ in total);
(2) **Non-salient Group.** In this case I use Type3 data along with pure pictures($300 + 300 = 600$ in total);
(3) **General Group.** In this case I use Type4 & Type1 data along with pure pictures($(300+300)*2 = 1200$ in total);

Next, I will display performance on each group.

B. Salient Group VS Non-Salient Group

From a comparative perspective, in this scenario, we focus on the performance of the two models on **salient text and non-salient text**, which can be seen in Table. III in the appendix, where in each grid the left is score for Salient group while the right is for Non-salinet group.

And Fig. 3 shows performance of the two group models. More examples can be seen in Fig. 6 in the appendix.

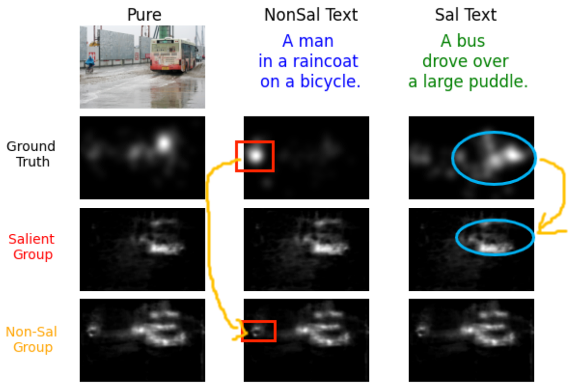

Fig. 3: Performance for Two Group Models

Obviously we can see that the model trained on NonSalient text can capture the text information effectively. As shown in the figure, "**A man in a raincoat**" is successfully predicted.

C. General Group

The model in this group can be widely applied in everyday life, given an overall description of a picture, it can effectively predict its salient parts. As a comparison, we can contrast the performance of this model with input that has a general text description and input without a text description.

Similarly, I put the score result in the appendix(Table. IV), and Fig. 4 shows performance of the model.

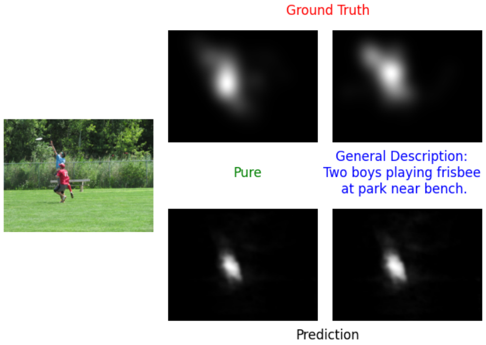

Fig. 4: Performance for General Model

It can be observed that there is little difference in the model's output when inputting pure images as opposed to images with a general scene description. However, in practice, the overall description does not significantly alter the focus of our attention, leading to similar saliency maps.

V. Result Analysis

A. Loss during Training

For models trained on different dataset, I recorded the corresponding loss and ploted the loss curves in Fig. 5.

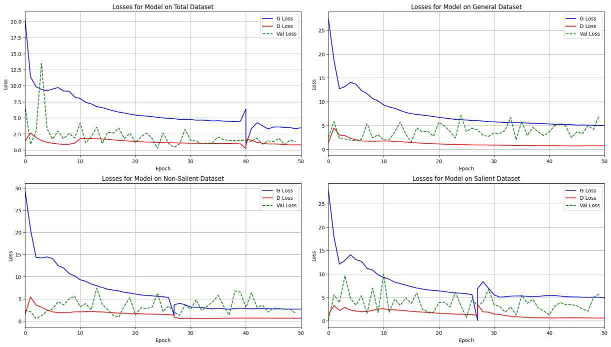

Fig. 5: Loss of Different Training Process

We can see from the curves that the generator's loss **rapidly decreases** in the early stage, with the descent rate **gradually slowing down**, and it essentially **converges** around 50 epochs. That is because in the initial stages, the generator might learn some simple features or patterns, leading to a rapid loss reduction. But over time, the generator may encounter more challenging aspects of the data distribution, causing the loss decrease rate to slow down.

About the discriminator, its loss **remains relatively low** and, overall, it is in a **continuous decreasing trend**. That indicates that the discriminator can relatively easily distinguish between real and generated samples. And continuously decreasing discriminator loss suggests that the generator is consistently improving the quality of generated samples, although there is still room for improvement.

The loss on the validation set, on the other hand, **fluctuates up and down**, which could be due to a smaller number of samples in the validation set or differences in the data distribution between the training and validation sets. It could also result from training randomness, such as random batch selection or data augmentation strategies.

B. Scores on Test Set

About the model trained on total dataset, AUC, sAUC, and CC metrics show acceptable performance, but the NSS score is too low. This indicates that the model I trained cannot accurately mimic human gaze patterns. To improve this score, it may be necessary to adjust hyperparameters and optimize the design of the loss function, such as combining BCE loss and MSE loss, in order to enhance performance in the NSS metric.

Regarding the other three models trained on specific subsets, they did not perform well on the specific dataset as I had hoped. Anyway, different models can produce varying saliency maps for the same image and different text inputs, which to some extent serves the purpose of text-guided attention. Similar to the previous assignment, the poor performance of the small model is attributed to the insufficient amount of data, preventing the model from adequately learning meaningful information, resulting in high variability and consequently, subpar performance.

VI. Limitations

Regarding the model I trained, there are several limitations:

(1) The Gan model struggles to effectively incorporate textual features. Originally, SalGan model used only image features for training. Howerer I make use of CLIP to extract text features and integrate both types of features into the generator. There might be better fusion methods, such as using inner product operations, and so on.

(2) Hyper-parameters setting. Due to limited computational resources and the difficulty of training GAN models, I did not fine-tune the hyperparameters, and there may be better combinations of hyperparameters.

(3) Not take advantage of fixation maps. The dataset includes fixation maps, which represent human gaze attention, but it is challenging to incorporate them into training of Gan model effectively, resulting that the model can't be able to utilize this information.

VII. Conclusion

In this document, I propose CLIP-SalGan, a powerful text-guided saliency model. Given a image with its different text description, the model can predict the corresponding saliency map effectively. I fully leveraged the advantages of the Gan model in image-to-image tasks, and achieved good results in all four evaluation metrics. I hope that my model can be of assistance to fellow computer vision researchers, and I also aspire to make significant advancements in saliency prediction.

VIII. Appendix

Score Type	AUC	sAUC	CC	NSS
Type 2	0.6968/0.6969	0.5984/0.5982	0.3948/0.4104	0.2618/0.2798
Type 3	0.6393/0.6379	0.5695/0.5688	0.2165/0.2354	0.2504/0.2763
Pure	0.7103/0.7179	0.6050/0.6088	0.4432/0.4861	0.1582/0.1779

TABLE III: Result for Sal & Non-Sal Model

Score Type	AUC	sAUC	CC	NSS
Pure	0.7380	0.6189	0.5327	0.2005
Type1 & Type4	0.7179	0.6090	0.4759	0.2574

TABLE IV: Result for General Model

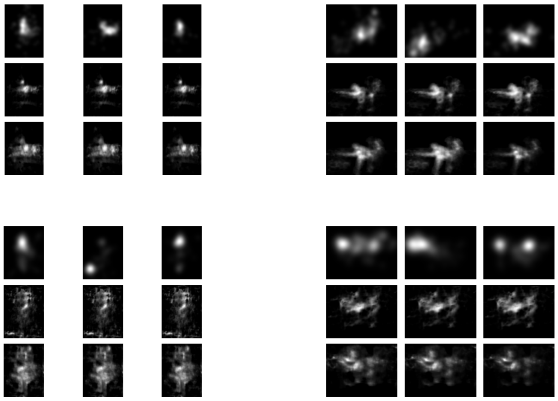

Fig. 6: Performance of CLIP-SalGan on Some Other Images

References

[1] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[2] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PMLR, 2021: 8748-8763.

[3] Pan J, Ferrer C C, McGuinness K, et al. Salgan: Visual saliency prediction with generative adversarial networks[J]. arXiv preprint arXiv:1701.01081, 2017.

[4] Li G, Yu Y. Deep contrast learning for salient object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 478-487.

[5] Tammina S. Transfer learning using vgg-16 with deep convolutional neural network for classifying images[J]. International Journal of Scientific and Research Publications (IJSRP), 2019, 9(10): 143-150.

[6] Graham B. Fractional max-pooling[J]. arXiv preprint arXiv:1412.6071, 2014.

[7] Kundu S, Mostafa H, Sridhar S N, et al. Attention-based image upsampling[J]. arXiv preprint arXiv:2012.09904, 2020.

[8] Basha S H S, Dubey S R, Pulabaigari V, et al. Impact of fully connected layers on performance of convolutional neural networks for image classification[J]. Neurocomputing, 2020, 378: 112-119.

[9] Jadon S. A survey of loss functions for semantic segmentation[C]//2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB). IEEE, 2020: 1-7.