Pattern Recognition 122 (2022) 108291

Contents lists available at ScienceDirect

Pattern Recognition

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

AE-Net: Fine-grained sketch-based image retrieval via attention-enhanced network

Yangdong Chen[a,a,1], Zhaolong Zhang[a,a,1], Yanfei Wang[a,a,1], Yuejie Zhang[a,*], Rui Feng[a], Tao Zhang[b,*], Weiguo Fan[c]

[a] School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
[b] School of Information Management and Engineering, Shanghai Key Laboratory of Financial Information Technology, Shanghai University of Finance and Economics, Shanghai, China
[c] Department of Business Analytics, Tippie College of Business, University of Iowa, USA

ARTICLE INFO

Article history:
Received 10 June 2020
Revised 21 August 2021
Accepted 30 August 2021
Available online 1 September 2021

Keywords:
Fine-grained sketch-based image retrieval (FG-SBIR)
Residual channel attention
Local self-spatial attention
Contrastive learning
Spatial sequence transformer

ABSTRACT

In this paper, we investigate the task of Fine-grained Sketch-based Image Retrieval (FG-SBIR), which uses hand-drawn sketches as input queries to retrieve the relevant images at the fine-grained instance level. The sketches and images come from different modalities, thus the similarity computation needs to consider both fine-grained and cross-modal characteristics. Existing solutions only focus on fine-grained details or spatial contexts, while ignoring the channel context and spatial sequence information. To mitigate such challenging problems, we propose a novel deep FG-SBIR model, which aims at inferring attention maps along channel dimension and spatial dimension, improving modules of channel attention and spatial attention, and exploring Transformer to enhance the model's ability for constructing and understanding spatial sequence information. We focus not only on the correlation information between two modalities of sketch and image, but also on the discrimination information inside the single modality. Mutual Loss is especially proposed to enhance the traditional triplet loss, and promote the internal discrimination ability of the model on a single modality. Extensive experiments show that our AE-Net obtains promising results on *Sketchy*, which is the largest public dataset available for FG-SBIR at present.

1. Introduction

With the popularity of mobile devices, it is an easy and efficient way for people to draw a sketch as the query modality. Sketching is an innate ability of people to express what they want. This kind of special modality has attracted wide research interests [1–3]. For image retrieval, using sketches to express visual cues and query intentions becomes more convenient than just using textual queries. Thus Sketch-based Image Retrieval (SBIR) has emerged deep potential in the practical application of computer vision, which can make image users use hand-drawn sketches to retrieve relevant natural color images in a large-scale image database.

Generally, existing approaches for SBIR mainly focus on category-level retrieval between sketches and images, which usually ignore intra-category variations. This level of retrieval is called Coarse-grained SBIR (CG-SBIR/Category-level SBIR), i.e., given a query sketch, a simple image recognition model is just utilized to retrieve the images with the same class labels [4]. However, such methods can not fully leverage the power of visual cues involved in sketches, and then ignore the fine-grained visual property of intra-category variation. Thus it is significant to retrieve the relevant images with the same characteristics at the fine-grained level such as pose, viewpoint, texture, and shape. Fine-grained SBIR (FG-SBIR) is emerging to solve such tough problems. FG-SBIR pays more attention to full visual details that are sketched and conveyed from free-hand sketches and natural color images. It deeply explores the similarities between sketches and images in a specific category at the fine-grained instance level. FG-SBIR is more likely to support practical business applications by providing such a specific interaction pattern that is more expressive than the generic way of browsing textual information.

During recent years, great progress has been made in the area of FG-SBIR. However, FG-SBIR still suffers from several challenging problems that need more attempts to get optimal solutions. First of all, there naturally exists a semantic gap between sketches and images, because they come from two different modalities. Compared to images, sketches mainly describe the shape or con-

* Corresponding author.
E-mail addresses: 18210240044@fudan.edu.cn (Z. Zhang), yjzhang@fudan.edu.cn (Y. Zhang), taozhang@mail.shufe.edu.cn (T. Zhang).
[1] Equal contribution.

https://doi.org/10.1016/j.patcog.2021.108291
0031-3203/

tour information of objects but miss the color and texture information. Secondly, sketches are more abstract than images. Hand-drawn sketches are usually drawn by non-experts, mainly based on their mental recall of the reference images displayed before drawing. Therefore, it is difficult for FG-SBIR models to deal with the global shape and local misalignment spatial information of objects simultaneously. In addition to the above challenges shared with CG-SBIR, FG-SBIR has the biggest challenge, how to establish the correspondence relations between sketches and images at the fine-grained level. When facing a query sketch, there are lots of candidate images in different categories. Moreover, there are many visually similar candidate images in the same category, and the right and wrong retrieval results may differ slightly in some local parts.

Current FG-SBIR models mainly focus on how to narrow the high-level semantic gap between sketches and images. Yu et al. [5] adopted a framework with three branches of deep Convolutional Neural Networks (CNNs). These branches shared the weights, and each of them was associated with its corresponding domain. To narrow the gap between such two modalities, each image was converted to an edge map first, and then the similarity was calculated directly between the sketch and the edge map. This approach not only increases the burden of data preprocessing, but also loses some important information and brings some new noises. By introducing spatial attention and HOLEF Loss, Song et al. [6] improved [5] through solving the local spatial misalignment by spatial attention. However, they only concerned the misalignment in high-level semantic space, which led to the loss of fine-grained details and could not be recovered at high level. Besides, [5] and [6] only carried out the task on small datasets that contained several classes. Hence, such models can not be well promoted in the real scenarios, because the categories of query sketches are abundant and each category contains a large number of images.

Based on the above observations, we propose a novel FG-SBIR model with Attention-enhanced Network (AE-Net). It pays more attention to the fine-grained details of sketches and images by using different attention mechanisms, that is, channel attention, self-attention, and spatial sequence attention. We introduce channel attention into our model and redesign it as **Residual Channel Attention** to acquire category discrimination. It aims to enhance the robustness and feature expression ability of the model through residual connection with parameters. We also explore another kind of spatial attention, i.e., self-attention, which can establish long region dependence among the pixels. However, limited by this kind of attention itself, it does not scale very well to a wider range of attention. Because the computational complexity for computing the weight map is $O(n^2)$, the calculation process is complex and difficult to obtain the optimal value. Moreover, when the gradient backpropagation is carried out, a large number of computational resources need to be consumed. Thus we propose **Local Self-attention** to reduce both the computational complexity and resource consumption, improve the calculation speed, and focus on more area. Since convolution extracts features by mixing cross-channel and spatial information, channel attention and spatial attention are utilized to emphasize the meaningful features along two main dimensions of the channel and spatial axis, and then can learn *"what"* and *"where"* on two dimensions respectively. Meanwhile, by observing the way that people draw sketches and look at images, e.g., from top to down and from left to right, we consider establishing the connection between these two manners. We introduce **Spatial Sequence Transformer** that uses Transformer to build up the relation between the spatial sequences of sketches and images and increase the model's ability to understand spatial sequence information. Through these attention mechanisms, we aim to narrow the huge misalignment between sketch modality and image modality, and improve the fine-grained feature representation power. Besides, existing models usually only focus on the correlation between two modalities, but ignore the distance relation in a single modality. To mitigate such a problem, we particularly propose Mutual Loss to overcome this deficiency.

Our contributions can be summarized as:

1. A novel FG-SBIR model with Attention-enhanced Network (AE-Net) is established, which pays more attention to the fine-grained details of the sketches and images.
2. We introduce three modules, i.e., the Residual Channel Attention module, Local Self-attention mechanism, and Spatial Sequence Transformer to mine the fine-grained details of the sketches and images in all dimensions.
3. Mutual Loss is proposed to improve the traditional Triplet Loss and restrain the distance relations among the sketches/images in a single modality.
4. Our model can achieve the promising performance on *Sketchy*, the largest public dataset available for FG-SBIR at present.

The rest of the paper is organized as follows. Section 2 briefly reviews some related works. In Section 3, we introduce the details of our new Attention-enhanced Fine-grained Sketch-based Image Retrieval framework. Section 4 gives our experimental results and analyses on the algorithm evaluation. Finally, we give the conclusions in Section 5.

2. Related work

2.1. Fine-grained SBIR (FG-SBIR)

The sketch is a kind of vivid data formation, which can be easily used to express ideas [7]. Therefore, free-hand sketch understanding tasks have been studied for several years in the field of computer vision [1]. Among these tasks, sketch-based image retrieval is a hot topic that attracts many research interests. It was proposed in the 1990s [8], and many researchers focused on Coarse-grained Sketch-based Image Retrieval (CG-SBIR) at that time. CG-SBIR is also called as Category-level SBIR, which focuses on category-level sketch-to-image retrieval. In that period, researchers usually used well-designed descriptors to extract the features of sketches and images [4]. Beyond these CG-SBIR researches, Li et al. [9] firstly used a Deformable Part-based Model (DPM) to solve the Fine-grained Sketch-based Image Retrieval (FG-SBIR) problem. Different from CG-SBIR, FG-SBIR focuses on instance-level sketch-to-image retrieval that needs to distinguish the slight differences among the images of the same category [10]. However, the representation ability of these hand-crafted descriptors is limited, which makes it difficult to achieve good performance on FG-SBIR.

Recently, some models are proposed to solve FG-SBIR by using deep learning [5,6,11]. Yu et al. [5] built up a dataset that contained sketch-photo pairs with detailed triplet annotations. They proposed a triplet network based on *Sketch-a-Net* [2] and used the edge maps of the images as a middle modality to narrow the differences between sketches and images. Song et al. [12] used a multi-task learning method to train their triplet network. Besides a triplet ranking loss, they also introduced an attribute prediction loss to make their network predict semantic attributes like whether a shoe is high-heeled. However, these previous FG-SBIR methods focus on coarse matching through deep cross-domain representation learning, and pay no attention to fine-grained details and spatial contexts [6]. Therefore, by introducing a spatial attention which focused on fine-grained details, Song et al. [6] attempted to address the above problems. They also used a Higher-order Learnable Energy Function (HOLEF) to improve the performance of their triplet network. Sangkloy et al. [3] also investigated the problem of FG-SBIR and proposed a triplet network with triplet loss and classification loss. They proposed Sketchy Database with 125 categories, trying to solve the problem over a wide range of categories but

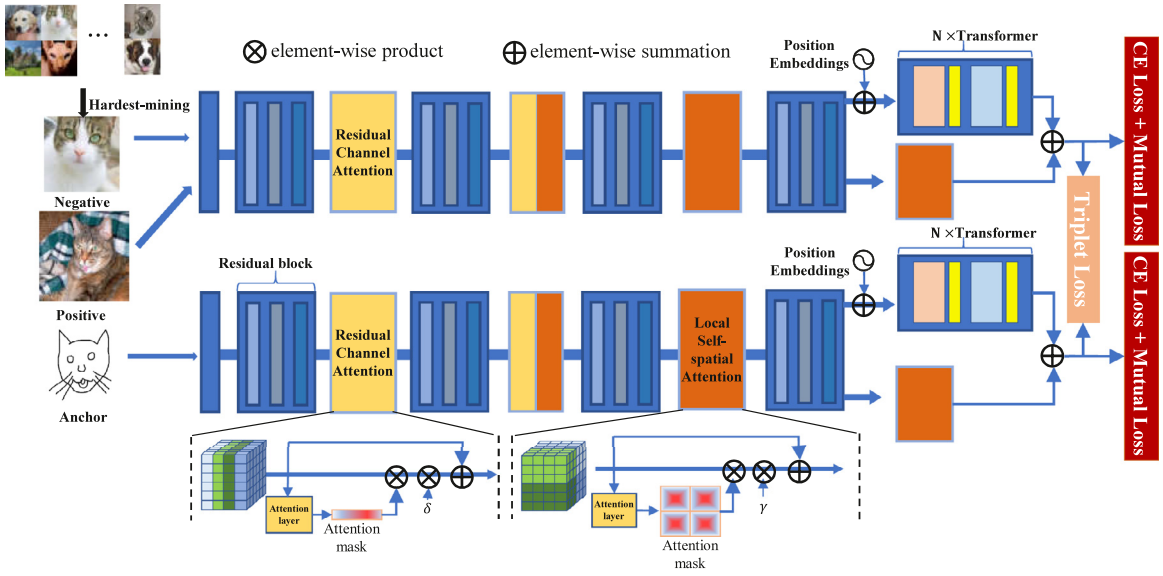

Fig. 1. The architecture of the proposed Attention-enhanced FG-SBIR model. It takes the sketches and images as input and extracts their features with our proposed attention mechanisms, **Residual Channel Attention** and **Local self-attention**, and the **Spatial Sequence Transformer**. The whole model is optimized with Triplet Loss, Cross-entropy Loss (CELoss), and the proposed **Mutual Loss**.

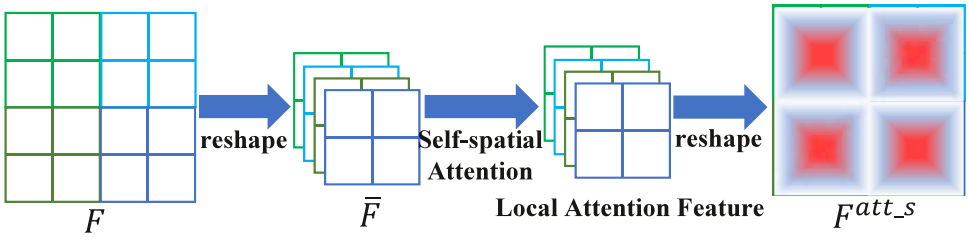

Fig. 2. An illustration for Local Self-attention.

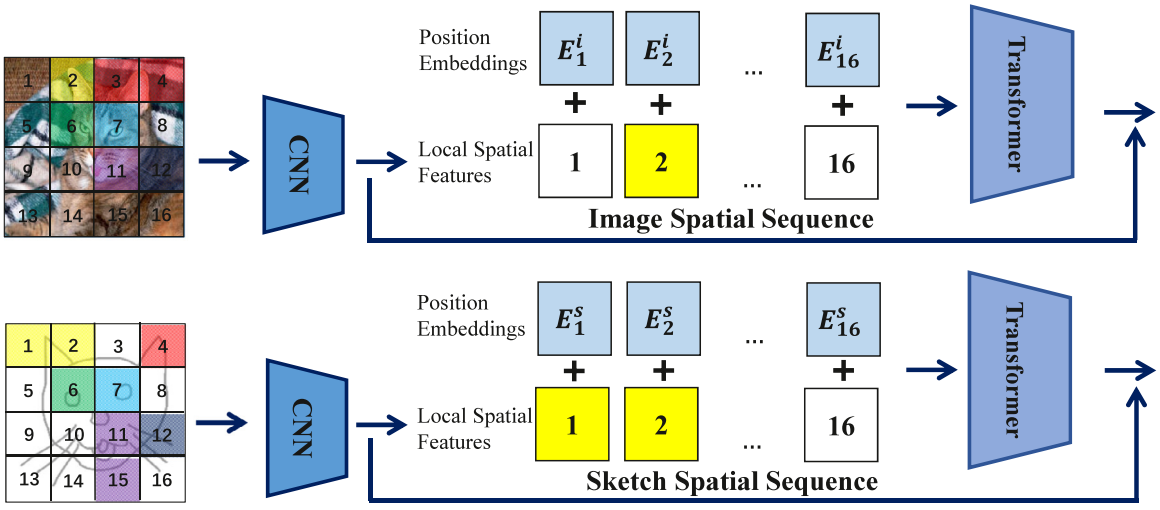

Fig. 3. An illustration of Spatial Sequence Transformer. The image (sketch) spatial sequence is composed by the image (sketch) spatial features and position embeddings.

not limited to a specific category. Lin et al. [11] further analyzed the limitation of previous works. They argued that a complex pre-training process was often needed, and using the edge map as an intermediate representation was difficult in practice for the models that were usually sensitive to the quality of the edge maps. They also explored the impact of several loss functions on the FG-SBIR task, such as Spherical Loss, Central Loss, Classification Loss, and Softmax Loss. Besides using information from sketches and images, Wang et al. [13] further explored the use of information from text descriptions. They proposed a Deep Cascaded Cross-modal Ranking Model (DCCRM) that used all the beneficial multimodal information, including sketches, images, and text descriptions. Bhunia et al. [14] proposed a semi-supervised framework, which explored to utilize unlabelled photos to mitigate the limited upper bound of sketch data.

Recently, zero-shot learning has attracted increasing attention for a number of computer vision tasks, such as SBIR [15,16]. To adapt to the zero-shot learning settings, Zero-shot SBIR (ZS-SBIR)

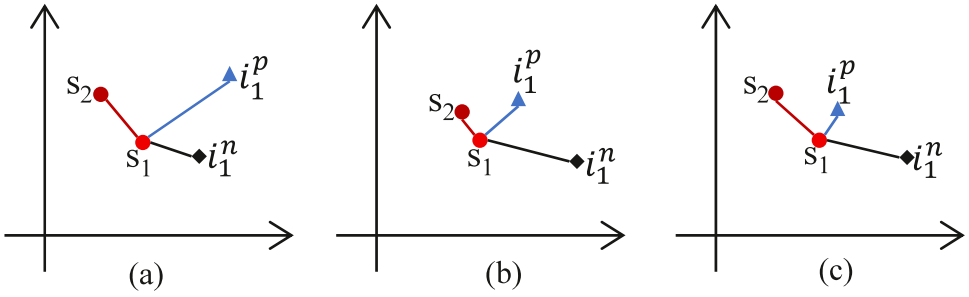

Fig. 4. An illustration of **Mutual Loss**. s_1 and s_2 represent two sketches, while i_1^p and i_1^n denote the positive image and the negative image of s_1. Different distance situations are shown in (a) distance relations without training; (b) distance relations after being trained with $L_{triplet}$ and L_{cat}; and (c) distance relations after being trained with all these three losses.

works often employ generative models[17] and mine semantic information [15,16] to cover the categories that do not appear in the training set.

2.2. Attention mechanism

In the human visual system, people do not attempt to process the entire scene at once. Instead, to capture the visual structure better, they use a series of partial glimpses to focus on the salient parts selectively. Thus attention is a very important and promising feature. The attention mechanism was first proposed in the field of neural machine translation [18]. Due to its good performance in many neural language processing tasks, it is widely used in many other fields. Especially in the field of computer vision, many attention processing attempts have been made to improve the performance of CNNs in the specific visual tasks, such as image/video captioning [19–21], image segmentation [22,23], fine-grained image classification [24,25], and especially sketch-based image retrieval [6,26].

Spatial-wise attention is one of the most widely used attention types. This attention is modeled as spatial probabilities that re-weight the feature maps [27]. By using the attention mechanism, Xu et al. [19] taught their image captioning model to focus on different parts of an image when generating a sentence. Similarly, Yao et al. [20] used an attention model to let their video captioning model see the most relevant frame when generating different words. In the fine-grained image classification task, Peng et al. [24] proposed Object-Part Attention Model (OPAM), which contained object-level attention and part-level attention model. The object-level attention localized objects of images for learning object features, while the part-level attention model selected the discriminative parts for learning the subtle and local features.

Beyond spatial-wise attention, channel-wise attention is also an attention type that is widely used. Each CNN filter performs as a pattern detector, and each channel of a feature map corresponds to the filter. Therefore, channel-wise attention can be viewed as selecting the important semantic attributes [27]. Hu et al. [28] proposed Squeeze-and-Excitation network (SENet) to model the relation between the channels. Woo et al. [29] further combined the spatial- and channel-wise attention together in their Convolutional Block Attention Module (CBAM). In the field of image captioning, Chen et al. [27] proposed Spatial- and Channel-wise Attention-based Convolutional Neural Network (SCA-CNN), which used spatial and channel attention to select the important regions and semantic attributes.

Self-attention is another well-known attention mechanism. Since it has been proposed in Transformer [30], self-attention brings great progress to natural language processing. It is also widely used in the field of computer vision [22,25,31–33]. For image recognition, self-attention is used to enhance the feature extraction ability of the neural networks. Zhou et al. [25] developed two-dimensional relative self-attention, replaced the convolutions with self-attention, and got competitive results. However, they found the result was better when combining the self-attention and convolution operator. Zhao et al. [31] considered two forms of self-attention, i.e., pairwise self-attention and patchwise self-attention. They found that their attention networks outperformed the convolutional counterparts and might have more generalization ability. For image generation, Zhang et al. [32] introduced self-attention to establish the relation between two different pixels. This enabled their model to generate an image by using long range interactions among the pixels. For scene segmentation, Fu et al. [22] used self-attention in both spatial and channel dimensions. In spatial dimensions, the attention model learned interdependencies of the pixels that belonged to the same class. In channel dimensions, the attention model modeled the relations between different channels. Parmar et al. [34] restricted the self-attention mechanism to attend to local neighbourhoods. Through local self-attention, the computation could be accelerated in parallel, which meant a larger scale of images could be handled.

Also, some attempts are trying to apply attention mechanism in the SBIR field. Song et al. [6] introduced a spatial attention which enabled their model to become sensitive to the fine-grained visual details. However, they experimented on a small dataset. The real-life scenarios are very complex, which include different kinds of query sketches and large-scale candidate images, and require the model to pay attention to the types and fine-grained differences of retrieval targets. Thus it is hard to directly generalize such attention models to real-life scenarios. What's more, although Song et al. [6] used a skip connection for attention, they did not specifically study the setting of attention weight. For the attention mask, they used the 1×1 convolution to generate the mask for the feature map but not the self-attention mechanism. Lei et al. [26] introduced a co-attention model to capture the common features between sketches and images. The co-attention model first used a channel-wise attention to select discriminative features of each input and learn attention masks, and then generated the final co-mask by element-wise multiplying the above masks.

Different from existing FG-SBIR approaches, our model not only concerns the high-level semantic space, but also mines important features from low level to high level along two main dimensions of channel and space. To construct a fine-grained sketch-image correlation, we focus on key features and suppress unnecessary ones through attention mechanisms. Our work improves channel attention and spatial attention mechanisms for robust attention module to overcome the shortcomings of their original attention patterns.

2.3. Deep embedding learning

Sketch-based Image Retrieval (SBIR) can be regarded as an extension of image retrieval, and they both aim at answering such a question: How similar are the given queries and the images?

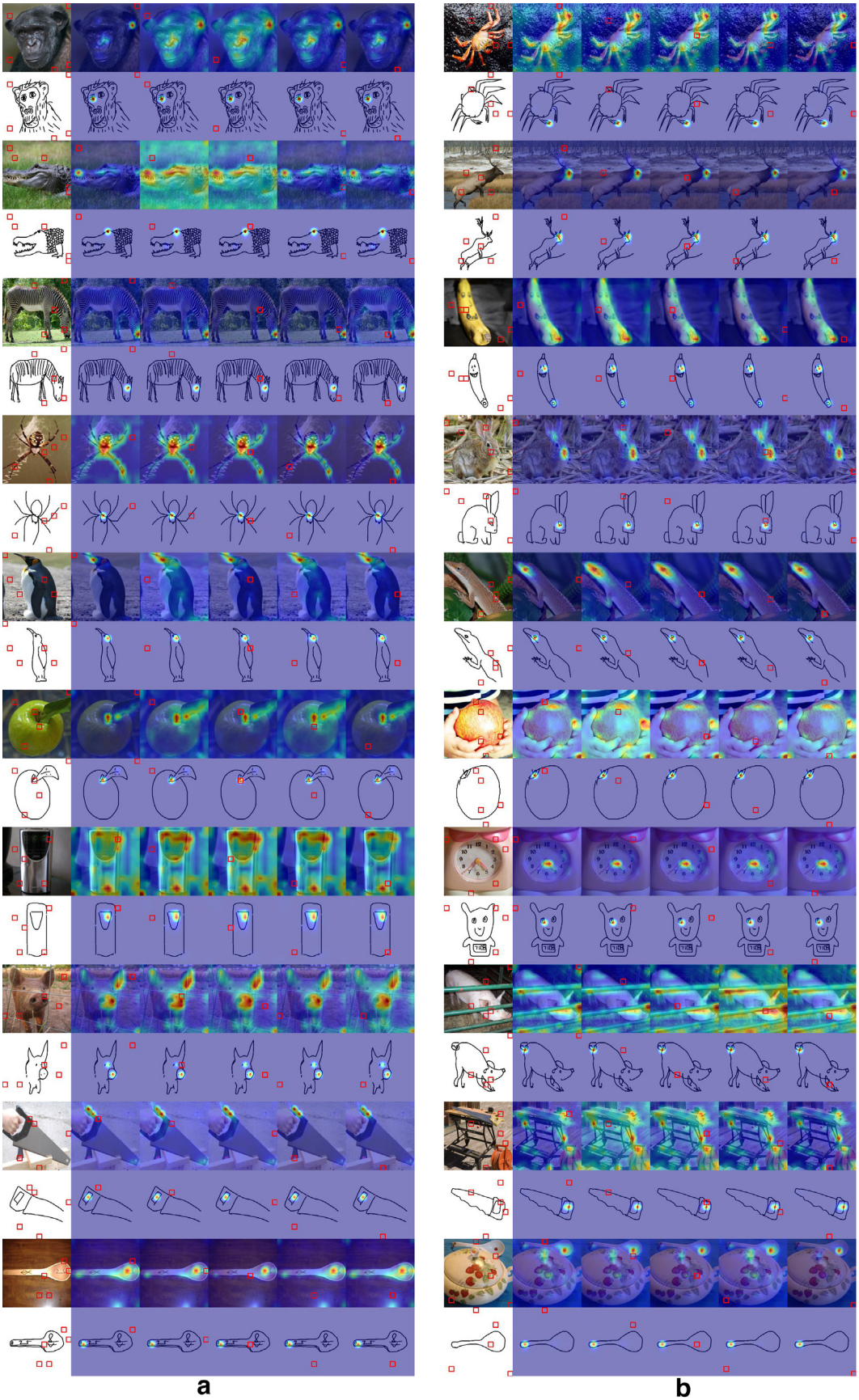

Fig. 5. The visualization of some self-attention maps for given regions (each region represents the receptive field pertaining to an output neuron in the feature map). For each sketch-image pair, 5 regions are randomly selected and the attention maps corresponding to them are shown in the back. The selected regions are marked in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

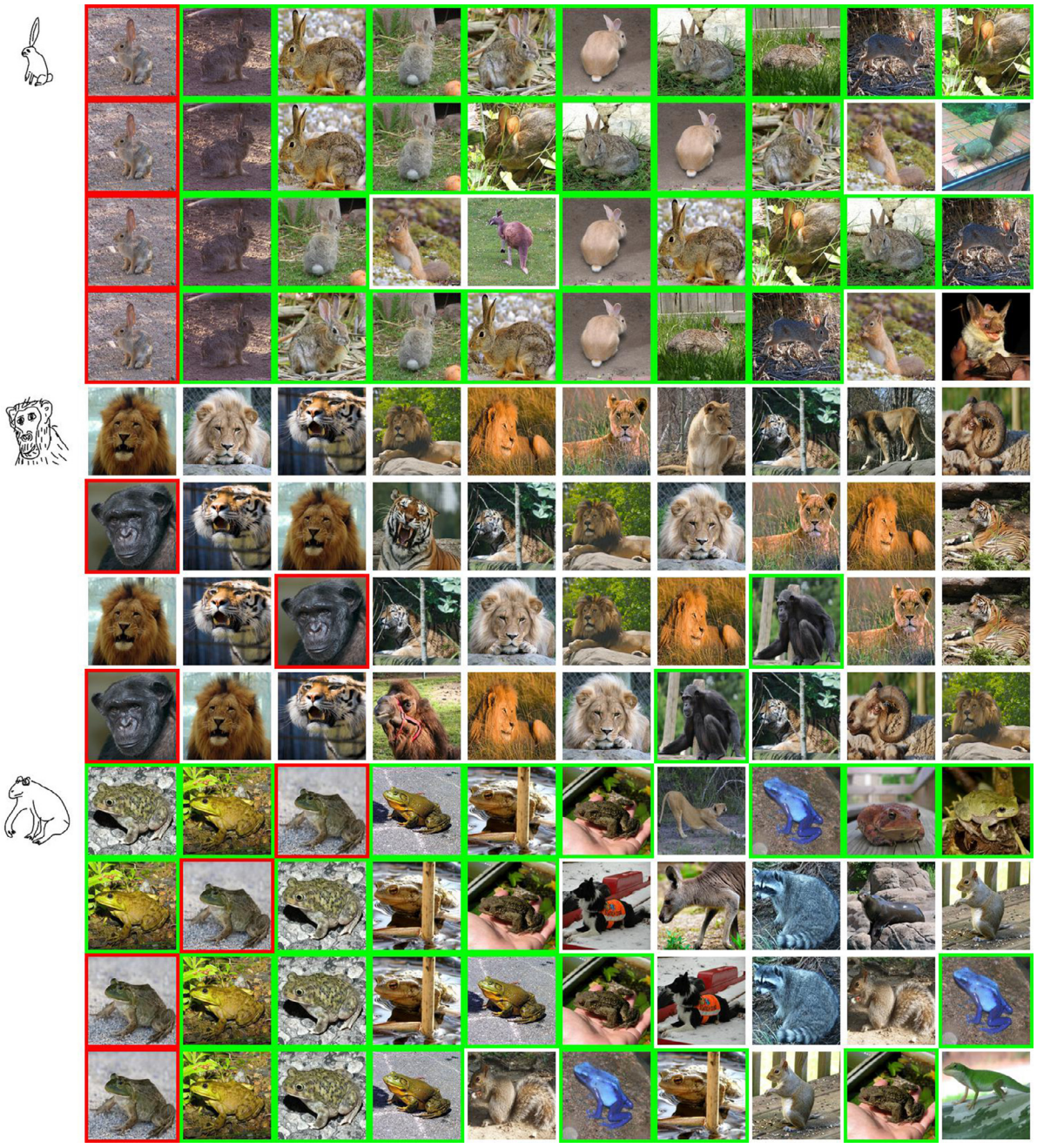

Fig. 6. Some retrieval results obtained by our **Baseline** model, **Local Self-attention** model, **Residual Channel Attention** model, and **Spatial Sequence Transformer** model.

To solve this problem, Siamese Network and Triplet Network have been proposed successively. The Siamese Network with the contrastive loss forces all positive pairs to be close, while the negatives should be separated by a given margin. However, using the fixed margin will limit the embedding space to a very small range. This motivates the Triplet Network with triplet loss, which only requires negative images to be farther than any positive images. Recently, some deep learning models [3,5,6] with triplet networks were proposed to solve the FG-SBIR problem and achieved outstanding performance. These models indicated that three-branch CNNs with triplet ranking loss could perform better than the two-branch CNNs with pairwise contrastive loss through experimental evaluation. The difference between the two models in [3,5] is whether the networks share weights, that is, the network is Siamese or heterogeneous. The network of [5] is Siamese, because it serves as a unified feature extractor for sketches and edge maps. However, the network of [3] is heterogeneous, because it does not do image edge extraction preprocessing and the branches of the network are trained directly on two modalities. Inspired by the thoughts of [3], our model is heterogeneous, which does not need to carry out the preprocessing operations such as edge graph extraction and then avoids the disadvantages such as losing information and introducing noises.

Wu et al. [35] showed that sample selection in embedding learning played an equal or a more important role than the loss. As we apply the Triplet Network in our model, we focus on the triplet sampling methods only. Usually, the sampling method can be divided into off-line sampling and on-line sampling. Off-line sampling is adopted in many previous SBIR works. For example, Yu et al. [5] generated the triplet list using a well-designed strat-

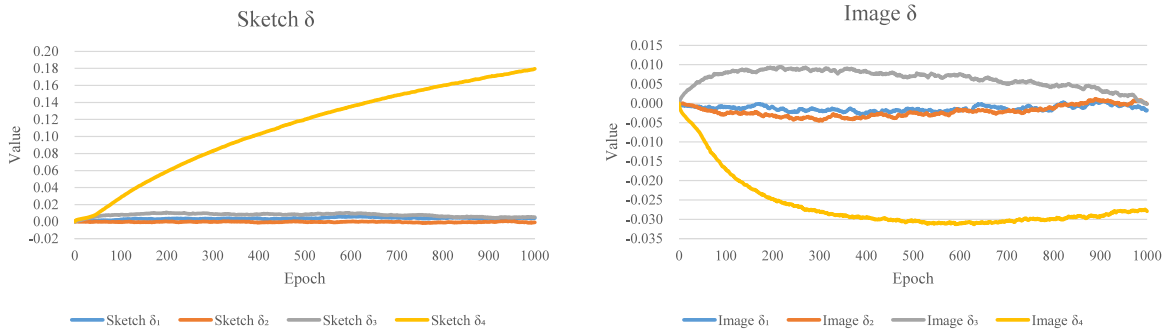

Fig. 7. The changing curves of δ with epochs.

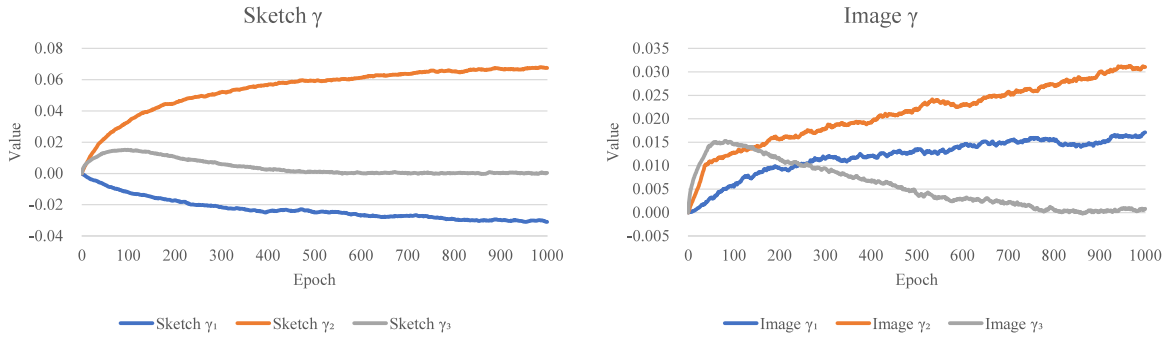

Fig. 8. The changing curves of γ with epochs.

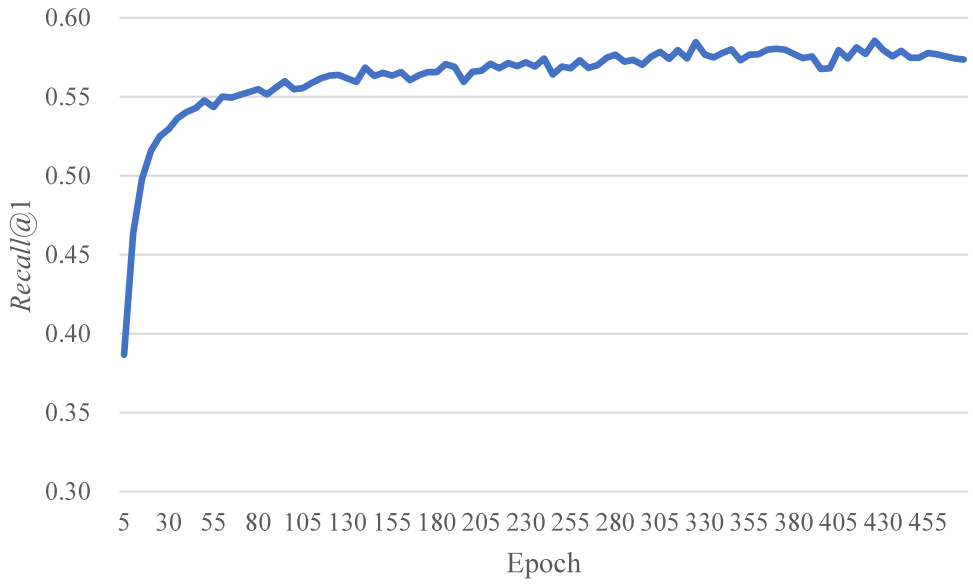

Fig. 9. The changing *Recall*@1 curve with epochs for CG-SBIR.

egy and [3] also sampled a training list before the training phase. This kind of method generates the triplets at the very beginning and will not change them during the whole training stage, which is not flexible enough and can hardly handle all the possible triplet cases. On-line sampling, which can alleviate the above problems, is rarely used in the SBIR task, but such a method shows a unique performance advantage in tasks like face recognition and Person ReID. FaceNet [36] first used the semi-hard negative mining, and afterward, such strategy becomes widely adopted.

3. AE-Net: attention-enhanced FG-SBIR framework

3.1. Overview of our framework

Our Attention-enhanced Network (AE-Net) for FG-SBIR takes a triplet network as the basic framework. It contains different CNN branches for the input of sketches and images respectively. AE-Net can be formulated as AENET$(s, i; \theta_s, \theta_i)$, which takes sketches s and images i as input and outputs their feature vectors. Figure 1 shows

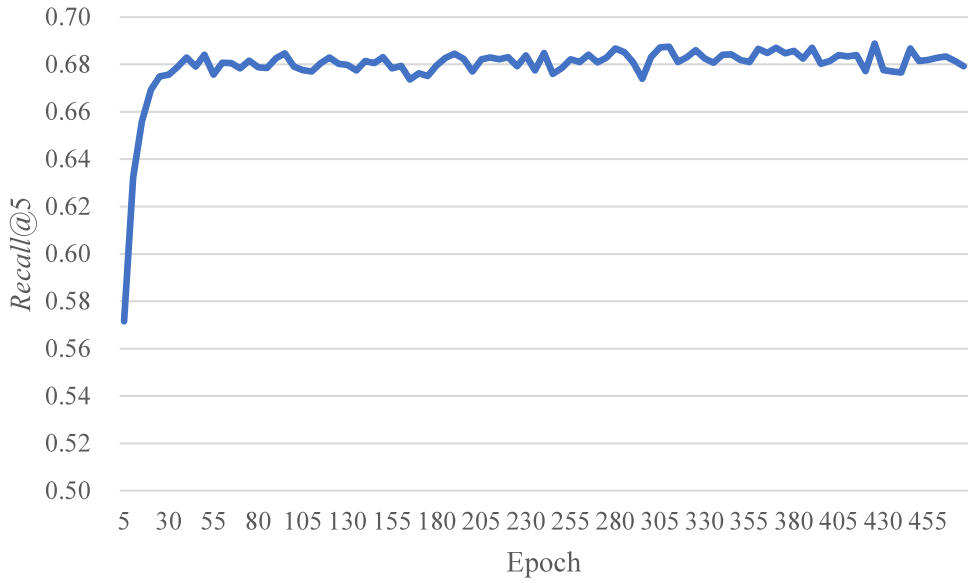

Fig. 10. The changing *Recall*@5 curve with epochs for CG-SBIR.

the architecture of the proposed model. Firstly, the branch networks extract deep features from input. Residual Channel Attention, Local Self-spatial Attention, and Spatial Sequence Transformer are then proposed to enhance our model's fine-grained feature representation ability. Finally, the features are fed into Triplet Loss, Cross-entropy Loss, and our proposed Mutual Loss to acquire the ranking order.

3.2. On-line sampling

We use on-line sampling when generating the triplets for training. The core idea is to generate the triplets in a mini-batch. Suppose we have a training mini-batch $\mathcal{B} = \{(s_i, i_i^+)\}_{i=1}^m$, containing m pairs of sketch s_i and its matching image i_i^+. For each sketch-image pair (s_i, i_i^+), we select K images as the negative images $i_1^-, \ldots, i_K^-$ from other sketch-image pairs in the mini-batch $\mathcal{B}$. These negative images and the target sketch-image pairs constitute the triplets. The maximum value of K is $m-1$, which means we take every possible triplet into account. We adopt the basic sampling method, **hard-mining**. However, sampling every possible triplet is not necessary or effective, so we also apply the **hardest-mining** method [36] besides of hard-mining. Our hardest sample mining can be regarded as selecting the most confusing sample in a mini-batch. We first compute the Euclidean distance between the features of the given sketch and all the candidate images $(i_1^-, \ldots, i_{m-1}^-)$, and pick up the image with the shortest distance. Thus, we will finally get m triplets in a mini-batch. This makes sense as if we make the hardest case meet the conditions, the others will meet naturally.

3.3. Channel attention and residual channel attention

Given an intermediate feature map $F \in \mathbb{R}^{C\times H\times W}$, each channel of F is considered as a feature detector [29]. Channel attention mainly focuses on "*what*", which is the specific semantic of input image. The destination of channel attention is to produce 1-D channel scores of features. Inspired by the model in [29], we use the same way to compute the channel attention map. Both average-pooled and max-pooled features are used to produce a channel attention map. Commonly, average-pooling is used to aggregate the spatial information, and max-pooling gathers another important clue about distinctive object features to infer better channel-wise attention [29]. With the purpose of finding fine-grained feature differences among images in FG-SBIR, both average-pooled and max-pooled features are used.

The first step of channel attention is to aggregate spatial information of a feature map. Through the average-pooling and max-pooling operations, we can obtain two different spatial context descriptors, F_{avg}^c and $F_{\max}^c$. Both of them are then fed into a shared network which is composed of Multi-layer Perceptron (MLP) with one hidden layer. After acquiring the mapped features of two descriptors through the shared network, we use the element-wise summation to merge the output feature vectors, as shown in Eq. (1).

$$\begin{aligned} M_c(F) &= \sigma\left(MLP(AvgPool(F)) + MLP(MaxPool(F))\right) \\ &= \sigma\left(W_1(W_0(F_{avg}^c)) + W_1(W_0(F_{max}^c))\right) \end{aligned} \tag{1}$$

where σ represents the sigmoid function; W_0 and W_1 denotes the weights of MLP in different layers; and $M_c(*)$ is the mapping function learned by the attention module.

The resulting attention mask M_c is a probability map obtained by normalizing the score vector using the sigmoid function. The attention feature map F^{att_c} is the element-wise product computed by the attention mask and input feature map, which is defined as:

$$F^{att\_c} = M_c \otimes F \tag{2}$$

However, due to the networks being trained in two modalities respectively, the attention mask may be a little imprecise. Compared to the original feature map F, the attention feature map F^{att_c} may introduce noises and lose some useful information. This kind of attention uses average-pooling and max-pooling to compress input features, which is fatal for fine-grained feature extraction because of losing fine-grained information. The information can not be effectively passed to the next layer, and thus the network loses the expressive ability to extract fine-grained features. For the original channel attention in practice, we find that the result is unstable and worse than the baseline. To overcome this problem, a residual connection with a learnable parameter is introduced to connect the input of the attention network directly to the attention feature map. We call this attention pattern as residual channel attention. The final attention feature map with a residual connection can be computed as:

$$F^{att\_c} = F + \delta(M_c \otimes F) \tag{3}$$

where δ is a learnable parameter and initialized as 0 at the beginning of training. Different from [6], we set the weight of attention as a learnable parameter. Thus, the original feature map and the imprecise attention feature map are combined automatically by self-learning. The networks can first learn on the raw features, and then gradually learn to increase the weight of attention features. The process of training our AE-Net with Residual Channel Attention is summarized in Algorithm 1.

Algorithm 1 Training AE-Net with Residual Channel Attention.

Input: Training dataset: $D = \{s, i\}_{i=1}^{n}$, Number of epochs: E
Output: AE-Net with Residual Channel Attention: $\text{AENET}_{\text{rca}}(s, i; \theta_s, \theta_i)$

for $t = 1 : E$ **do**
  Sample a mini-batch of m sketch-image pairs $\{(s, i)\}_{i=1}^{m}$ from the training dataset D.
  Calculate the feature maps F_0^s and F_0^i using the first convolutional layer of $\text{AENET}_{\text{rca}}$.
  for $l = 1 : 4$ **do**
    Calculate the feature maps F_l^s and F_l^i using the lth layer of $\text{AENET}_{\text{rca}}$ according to the previous feature maps F_{l-1}^s and F_{l-1}^i.
    Calculate the attention feature maps F_l^s and F_l^i according to Eqs. (1) and (3).
  end for
  Calculate Triplet Loss and Classification Loss and do back-propagation.
  Update the parameters θ_s and θ_i according to the gradient.
end for
Output AE-Net with Residual Channel Attention $\text{AENET}_{\text{rca}}(s, i; \theta_s, \theta_i)$.

3.4. Self-attention and local self-attention

Different from channel attention, self-attention mainly focuses on the relations between the spatial regions. The purpose of self-attention is to produce a 2-D similarity matrix $M_s \in \mathbb{R}^{N\times N}$ by using the inter-spatial relations of features. Inspired by the model in [29], we use the same way to compute the spatial attention map. By introducing self-attention into our framework, the networks can efficiently model the relations among widely separated spatial regions.

To calculate the attention map, the image feature F is first resized to $x \in \mathbb{R}^{C\times N}, N = H \times W$, and then x is mapped into different feature space $q, k, q(x) = W_q x, k(x) = W_k x (W_q, W_k \in \mathbb{R}^{\bar{C}\times C}, \bar{C} = \frac{C}{8})$. The attention map can be calculated as:

$$M_s^{j,i} = \frac{exp(s_{i,j})}{\sum_{i=1}^{N} exp(s_{i,j})}, s_{i,j} = q(x_i)^T k(x_j) \tag{4}$$

where $M_s^{i,j}$ indicates the similarity between the features of the ith region and jth region. The attention feature map F^{att_c} is the multiplication of the attention mask and the mapped input feature map. Here, we define the result of the attention layer as $R = (R_1, R_2, \ldots, R_j, \ldots, R_N) \in \mathbb{R}^{C\times N}$, and each element R_j can be calculated as:

$$R_j = \sum_{i=1}^{N} M_s^{j,i} v(x_i), v(x_i) = W_v x_i \tag{5}$$

where $W_v \in \mathbb{R}^{C\times C}$ is the weight.

In addition, the output of the attention layer is multiplied by a learnable parameter and added to the input feature map. Therefore, the final result can be computed as:

$$F^{att\_s} = F + \gamma(R) \tag{6}$$

where γ is initialized as 0 and plays the same role as δ in Eq. (3).

Spatial attention is proposed to solve the problem of long-range dependence on images. Because of the inherent calculation way of self-spatial attention, it does not scale very well to the wider range of attention. When we want to use it on larger space, i.e., the lower-level features, we need to focus on more regions. The calculation process becomes complex and is difficult to optimize, as the computational complexity for the weight map is $O(hw)$, where h, w denotes the height and width of the feature map. Moreover, when the gradient backpropagation is carried out, it will consume a large number of computing resources. Thus we propose Local Self-attention to overcome this shortcoming.

When we achieve the attention calculation in a large space, we pay more attention to the relations between the current region and its neighboring regions, and its relations with distant regions can be made up in higher levels of attention. Based on this observation, we reduce the computational complexity by dividing an entire large space into small areas for attention processing. It is worth noting that by dividing the space, we can acquire attention in different areas at the same time, in which parallel acceleration can be achieved through dimensional transformation and shared attention weights. More specifically, the feature map is first transformed to $\overline{F} \in \mathbb{R}^{\overline{N}\times\overline{H}\times\overline{W}}$, where $\overline{H} = \frac{H}{l}, \overline{W} = \frac{W}{l}, \overline{N} = l^2$, and l is the length of divided spatial sides. Secondly, self-attention is applied to the feature map $\overline{F}$. Finally, the result is recovered to the original shape of the input feature, as shown in Fig. 2. The process of training our AE-Net with Local Self-attention is summarized in Algorithm 2.

Algorithm 2 Training AE-Net with Local Self-attention.

Input: Training dataset: $D = \{s, i\}_{i=1}^{n}$, Number of epochs: E.
Output: AE-Net with Local Self-attention: $\text{AENET}_{\text{lsa}}(s, i; \theta_s, \theta_i)$

for $t = 1 : E$ **do**
  Sample a mini-batch of m sketch-image pairs $\{(s, i)\}_{i=1}^{m}$ from the training dataset D.
  Calculate the feature maps F_0^s and F_0^i using the first convolutional layer of $\text{AENET}_{\text{lsa}}$.
  for $l = 1 : 4$ **do**
    Calculate the feature maps F_l^s and F_l^i using the lth layer of $\text{AENET}_{\text{lsa}}$ according to the previous feature maps F_{l-1}^s and F_{l-1}^i.
    if $l = 2$ **then**
      Do transfer on the feature maps F_{l-1}^s and F_{l-1}^i: $\mathbb{R}^{H\times W} \mapsto \mathbb{R}^{N\times\overline{H}\times\overline{W}}, \overline{H} = H/2, \overline{W} = W/2$.
    end if
    if $l > 1$ **then**
      Calculate the attention feature maps F_l^s and F_l^i according to Eqs. (4)–(6).
    end if
  end for
  Calculate Triplet Loss and Classification Loss and do back-propagation.
  Update the parameters θ_s and θ_i according to the gradient.
end for
Output AE-Net with Local Self-attention $\text{AENET}_{\text{lsa}}(s, i; \theta_s, \theta_i)$.

3.5. Spatial sequence transformer

Inspired by the manner of drawing sketches, we consider constructing the sketch-image correspondence relation in the spatial sequence. Since the use of Transformer in BERT has become ubiquitous in Natural Language Processing recently, Transformer has shown its simple idea and extraordinary effect [37]. Here, we nov-

elly combine it with CNN for modeling the relation between sketch spatial sequence and image spatial sequence.

It is a very novel way to make sketches align with images on the spatial sequence. We will take one specific sketch-image pair to illustrate the spatial sequences and how we align these sequences. As shown in Fig. 3, there is an image of a cat and its corresponding sketch. They are divided into several patches (16 in Fig. 3), and these patches compose the spatial sequences. Each patch has its semantic information, and the patches that represent the same semantic attribute are colored the same. For example, in the image, the 2nd patch represents the left ear of the cat, while in the sketch the 1st and 2nd patches both represent the left ear. Therefore, there would be a misalignment between the 1st patch of the image and the sketch. We introduce Transformer to alleviate this misalignment by suppressing the unnecessary patches like the above 1st patch of the image.

We first describe the structure of the Transformer encoder part, and later introduce how we adopt it into our model. The transformer encoder mainly consists of Multi-Head Attention and Position-wise Feed-Forward Network [30]. Given an input X, the model first projects it into three latent spaces and outputs three feature matrices, namely, Q, K, V. This can be formulated as:

$$Q = XW^Q, K = XW^K, V = XW^V \tag{7}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d_{\text{model}} \times h \cdot d_v}$. Then these matrices are evenly split into h segments along the feature dimension (the 2nd dimension). Finally the multi-head attention is computed as follows.

$$\begin{aligned} \text{MultiHead(X)} = \text{MultiHead(Q, K, V)} &= [\text{head}_1, \ldots, \text{head}_\text{h}]W^Q \\ \text{head}_\text{i} &= \text{Attention}(Q_i, K_i, V_\text{i}) \\ \text{Attention}(Q_i, K_i, V_i) &= \text{softmax}(\tfrac{Q_i K_i^T}{\sqrt{d_k}})V_i \end{aligned} \tag{8}$$

where $Q = [Q_1, \ldots, Q_h]$, $K = [K_1, \ldots, K_h]$, $V = [V_1, \ldots, V_h]$. Then the output of the multi-head attention layer is added with the original input and fed into the feed-forward network. The feed-forward network is implemented by stacking two fully connection layers and a ReLU activation layer between them.

$$\text{FFN}(X') = \text{ReLU}(X'W_1 + b_1)W_2 + b_2 \tag{9}$$

where X' denotes the output of the previous layer; and W_1, W_2, b_1, b_2 denotes the parameters of the fully connected layer.

Besides, Layer Normalization (LN) is applied after these blocks. The transformer encoder contains several layers of the above blocks. Thus, the process of the lth block X_l can be formulated as follows.

$$X'_l = \text{MultiHead}(\text{LN}(X_{l-1})) + X_{l-1} \tag{10}$$

$$X_l = \text{FFN}(\text{LN}(X'_l)) + X'_l \tag{11}$$

where X_l is the output the lth block.

The input of Transformer is usually a sequence of word embedding when it is used to model natural language. Here, we apply it over the feature map extracted by the last layer of the CNN. To make the feature map fit the form of the Transformer input, we regard the original feature map as a spatial sequence feature map and flatten it to a 1-D sequence, i.e., $\mathbb{R}^{C \times H \times W} \mapsto \mathbb{R}^{N \times C}$, $N = H \times W$, where N is the length of the spatial sequence, and C is the dimension of the feature vector of each patch. In addition to the feature, position embedding is also used. For each local patch, its position embedding has the same shape as it. Specifically, the position embedding of the ith local patch can be calculated as:

$$PE(i, 2j) = \sin(pos/10000^{2i/d_{model}}) \tag{12}$$

$$PE(i, 2j+1) = \cos(pos/10000^{2i/d_{model}}) \tag{13}$$

where $2j$ and $2j+1$ are the dimension of the patch.

In our model, we stack two layers of Transformer encoder after the last layer of the CNN, and add a residual connection between the input and output feature of the transformer. These hyper-parameters are set as $d_{\text{model}} = C$, $h = 8$, and $d_k = d_v = 64$. The process of training our AE-Net with Spatial Sequence Transformer is summarized in Algorithm 3.

Algorithm 3 Training AE-Net with Spatial Sequence Transformer.

Input: Training dataset: $D = \{s, i\}_{i=1}^n$, Number of epochs: E
Output: AE-Net with Spatial Sequence Transformer: $\text{AENET}_{\text{sst}}(s, i; \theta_s, \theta_i)$

for $t = 1 : E$ **do**
  Sample a mini-batch of m sketch-image pairs $\{(s, i)\}_{i=1}^m$ from the training dataset D.
  Calculate the feature maps F_0^s and F_0^i using the convolutional layers of $\text{AENET}_{\text{sst}}$.
  Flatten the feature maps F^s and F^i, i.e., $\mathbb{R}^{C \times H \times W} \mapsto \mathbb{R}^{N \times C}$, $N = H \times W$.
  Feed the feature maps into the transformer layers and get the output feature maps F^s and F^i.
  Reshape the feature maps F^s and F^i to the original shape, i.e., $\mathbb{R}^{N \times C} \mapsto \mathbb{R}^{C \times H \times W}$.
  Calculate the final feature maps F^s and F^i: $F^s = F^s + F_0^s$, $F^i = F^i + F_0^i$.
  Calculate Triplet Loss and Classification Loss and do back-propagation.
  Update the parameters θ_s and θ_i according to the gradient.
end for
Output AE-Net with Spatial Sequence Transformer $\text{AENET}_{\text{sst}}(s, i; \theta_s, \theta_i)$.

3.6. Mutual loss

Given an input triplet (s_i^a, i_i^p, i_i^n) corresponding to a query sketch s_i^a, a positive image i_i^p, and a negative image i_i^n, a triplet loss can be defined as:

$$L_{triplet}(s^a, i^p, i^n) = max(0, m + D(F(s^a), F(i^p)) - D(F(s^a), F(i^n))) \tag{14}$$

where $F(\cdot)$ presents the output of the corresponding network branch; m is a margin to control the anchor-positive and anchor-negative distance; and $D(\cdot, \cdot)$ denotes the distance evaluation function, typically Euclidean distance calculation function. A successful SBIR system needs to consider both the semantics of query sketches (object categories) and fine-grained details (location, shape, perspective, and other attributes). Thus the Cross-Entropy Loss (CELoss) is also introduced to improve the performance when training the model, which is formulated as follows.

$$L_{cat} = -\sum_{i=1}^{n} y_i \log(\hat{y}_t) \tag{15}$$

Although good performance can be achieved by using the hardest sample mining, the above loss does not consider the relations in a single modality. Generally, the distance between two images is not limited. During training, especially for semantically similar images, the distance between two images is reduced, which results in a decrease of retrieval accuracy. Therefore, we propose Mutual Loss to reduce the error rate and ensure the retrieval accuracy by restricting the distance between two images, which is defined as:

$$L_{mutual}(x_i, x_j) = max(0, m - D(x_i, x_j)) \tag{16}$$

where m is a margin to control the distance between two images in the single modality; and $D(\cdot,\cdot)$ denotes the Euclidean distance function. x_i and x_j enumerates the features of the instances in the same modality, i.e., computing the mutual loss in the sketch and image modality, respectively.

An illustration of Mutual Loss is shown in Fig. 4, where s_1 and s_2 represents two semantically similar sketches, i_1^p and i_1^n denotes positive and negative images of s_1. Figure 4(a) shows the distance relation without training. Figure 4(b) shows the training result with triplet loss and classification loss, i.e., without mutual loss. After training, s_1 is close to i_1^p and s_2, which results in the short distance between s_2 and i_1^p and leads to an incorrect retrieval result. Figure 4(c) shows the training result with triplet Loss, classification loss, and mutual loss. s_2 is far from s_1, which prevents the situation in Fig. 4(b). Similar to sketch modality, mutual loss is also applied to image modality.

Since our model contains two network branches that are trained on their corresponding modalities, the total loss function can be defined as:

$$L = \alpha L_{triplet} + \lambda(L_{cat}^s + L_{cat}^i) + \beta(L_{mutual}^s + L_{mutual}^i) \tag{17}$$

where L_*^s and L_*^i denote the sketch and image modality respectively; and α, λ, β represent the weights of losses.

4. Experiments

We evaluate our AE-Net against several state-of-the-art FG-SBIR models on the largest fine-grained dataset *Sketchy* [3]. Next, we will describe the experimental configuration and component analysis of our AE-Net in detail.

4.1. Datasets

In our experiments, we use the largest dataset *Sketchy* [3] for FG-SBIR. It contains 12,500 unique images of objects and 75,471 human sketches drawn according to the images. These images are evenly separated in 125 categories, and each of them corresponds to at least 5 sketches. All the images and sketches are resized to the same size as 256×256. Our model is evaluated on the testing set that contains 6312 query sketches and 1250 images distributed in 125 categories. The rest of the data is used as the training set.

4.2. Evaluation metric

We use *Recall*@K as the performance evaluation metric in our experiments. Given a query sketch, *Recall*@K is equal to 1 if its relevant image is included in the top-*K* retrieved images, or equal to 0 if there is no relevant image in the top-*K* retrieved images. This metric can measure the ranking position of the target image in top-*K* results. Because there is only one correct image corresponding to the query sketch in the dataset, we pay more attention to the indicator of *Recall*@1, which shows the frequency of top-1 correct retrieval result in the testing set.

4.3. Experimental settings

Our model is implemented on *Pytorch*. We use *ResNet* [38] as the basic model for the three branches. Each branch is pre-trained on *ImageNet* [39] before being fine-tuned on *Sketchy*. Here, in our triplet network model, two image branches still share their weights. Thus our model has a set of weights for sketch modality and image modality respectively. We use Adam algorithm to optimize our model. We totally train it for 1000 epochs and select the best model by testing it every 5 epochs according to the *Recall*@1. The initial learning rate is 1×10^{-5}, and the mini-batch size is 64. At the training stage, there are no additional data augmentation operations other than resizing sketches and images. The number of layers for Transformer is set to 2. The margin m in triplet loss is set as 0.3. The weights of losses are set as $\alpha = 50$, $\lambda = 1$. We have conducted experiments with mini-batch to verify its feasibility. However, as shown in Table 1, on *ResNet*50 baseline, in the case of fixed seeds we get the best *Recall*@1 46.45% with batch size 16, compared to 48.92% with batch size 32 and 49.78% with batch size 64. This proves that the mini-batch size can significantly affect the performance of the model. We also try to use amp to save the memory. However, the best *Recall*@1 drops dramatically to 36.51%. This proves that the structure of Transformer is not suitable for amp and the distributed data parallel mechanism in *PyTorch*. It is very difficult to train Transformer with amp.

Table 1
The *Recall*@1 results of *ResNet*50 baseline with different batch size.

Batch Size	16	32	64
***Recall*@1**	46.45%	48.92%	49.78%

4.4. Ablation study

In our AE-Net, two new attention mechanisms are introduced simultaneously, i.e., residual channel attention and local self-spatial attention. Besides, Transformer is introduced to construct the relations between spatial sequences of sketches and images. A Mutual loss is also used to solve the inherent flaws of the original triplet loss and classification loss. To evaluate the contribution of each component, we make several comparisons.

4.4.1. Comparison on different attention models

We make several comparisons among different models with different attention models. Specifically, they are listed as follows.

- **Baseline**: The baseline models only adopt the basic *ResNet*18/ *ResNet*50/ *ResNet*101 as backbone for each branch.
- **Channel Attention**: We add the original channel attention model after each block of the basic network.
- **Residual Channel Attention**: We replace the original channel attention model with our proposed residual channel attention.
- **Self-attention**: We find that this sample self-attention model costs huge computing resources, which is far beyond our capacity. Therefore, we only use it after the last two blocks of the basic network.
- **Local Self-attention**: We use local self-attention after the second block of the basic network, and self-attention after the last two blocks.
- **CS Attention**: This attention model represents the combination of the above residual channel attention and local self-attention. It is worth noting that residual channel attention is added after the first two blocks, while local self-attention is added after the last two blocks. By doing this, the model can first learn to select the important semantic attributes and then learn to pay attention to the key regions.
- **Transformer**: We add Transformer after the last block of the basic network. We first encode the feature maps, which are output from the last block of the network, into path sequences, and then use Transformer to match the spatial sequence information for relieving spatial misalignment.
- **Attention and Transformer**: We add all the improved attention models and Transformer into our model. Specifically, we use residual channel attention after the first two blocks, local self-attention after the last two blocks, and Transformer after the last self-attention model.

Table 2
The experimental results of different components.

Model	Component	*Recall*@1	*Recall*@5
Quadruplet Network [40]	Quadruplet_MT_V2	42.16%	-
*ResNet*18	Baseline	43.92%	79.56%
	CS Attention	45.07%	79.71%
	Transformer	44.64%	79.94%
	Attention and Transformer	**45.95%**	**80.86%**
*ResNet*50	Baseline	51.23%	84.70%
	Channel Attention	50.74%	84.55%
	Residual Channel Attention	51.69%	84.87%
	Self-attention	51.32%	84.57%
	Local Self-attention	51.47%	85.33%
	CS Attention	51.72%	86.14%
	Transformer	52.09%	86.32%
	Attention and Transformer	**52.19%**	**86.29%**
*ResNet*101	Baseline	52.33%	85.83%
	CS Attention	54.06%	86.59%
	Transformer	52.47%	85.47%
	Attention and Transformer	**54.59%**	**86.37%**

Table 3
The experimental results on Mutual Loss.

Model	Loss	*Recall*@1	*Recall*@5
Quadruplet Network	Quadruplet_MT_V2	42.16%	-
*ResNet*18	Baseline	43.92%	79.56%
	Hardest-mining+ Mutual Loss	**44.84%**	**79.34%**
*ResNet*50	Hard-mining/$\alpha = 1$	48.55%	84.30%
	Hardest-mining/$\alpha = 1$	49.37%	84.00%
	Hard-mining/$\alpha = 50$	51.23%	84.70%
	Hardest-mining/$\alpha = 50$	51.45%	84.69%
	Hardest-mining+ Mutual Loss	**52.01%**	**85.50%**
*ResNet*101	Baseline	52.33%	85.83%
	Hardest-mining+ Mutual Loss	**52.82%**	**85.39%**

All these models are trained with triplet loss, classification loss, and hard-mining. The related experimental results are shown in Table 2.

As far as we know, the current best *Recall*@1 is obtained by Quadruplet Network [40] that is based on *ResNet*18, and reaches 42.16%. Although the *DCCRM*(S+I+D) model in [13] achieved better results, they used more modality information, the text descriptions, than other methods. Therefore, we do not consider it in this section. To be fair, we also use *ResNet*18 as our basic network branch. The *Recall*@1 of our baseline model with *ResNet*18 is 43.92%, which makes around 2% improvement compared to the Quadruplet Network. Table 2 shows that all of our baseline models are far superior to most of the available state-of-the-art models. In the following, we will mainly discuss the results achieved by the models based on *ResNet*50.

It can be observed that almost all the attention models adopted in our model can bring an improvement on *Recall*@1 compared with the baseline model. The performance of **Channel Attention** (50.74%) is lower than that of **Baseline** (51.23%), especially the value of *Recall*@1. This proves our claim that using average-pooling and max-pooling may lose much fine-grained information. In contrast, our proposed **Residual Channel Attention** achieves 51.69% making around 1% improvement on *Recall*@1 compared to **Baseline** (51.23%) and **Channel Attention** (50.74%). The residual channel attention uses a residual connection with a learnable parameter between the original feature map and the attention feature map. With the help of this connection, our model can mitigate the loss of fine-grained information caused by the average-pooling and max-pooling operations. In practice, we also adopt sample self-attention after different layers of *ResNet*50 to improve its final performance. However, it is found that the computing resources required for this sample self-attention is so huge (>48 GB GDDR, 4 GPUs, each 12GB GDDR) that exceeds the maximum we can bear. We can only use it in a smaller spatial area, i.e., after the last two layers (about 17GB GDDR). To exploit this kind of attention in a wider spatial area, **Local Self-attention** is proposed, a generalization of **Self-attention**, which obtains 51.47% *Recall*@1 that can be viewed as a small improvement compared to **Self-attention** (51.32%). Meanwhile, there is only a small increase in resource consumption (about 17.5GB GDDR). Self-attention can capture long range interaction among the pixels [33]. Our Local self-attention can also achieve that goal by first building up the relations locally in the previous blocks and then capturing long range relations from the later blocks. This not only alleviates the shortage of computing resources but also achieves a better result on *Recall*@1. When we combine Residual Channel Attention with Local Self-attention, which obtains **CS Attention**, the *Recall*@1 can increase to 51.72%. Such a result is better than all the other models that use one attention mechanism alone. These attention models learn different beneficial patterns and work well with each other. This shows that our model can first select useful semantic attributes, and then focus on key regions. The *Recall*@1 of **Transformer** reaches 52.09%, outperforming all those above models adopting the attention mechanism. This shows the validity of our model. By aligning the spatial sequences of sketches and images, our model can further mine the fine-grained details. When we integrate all the components into our model and obtain **Attention and Transformer**, the *Recall*@1 can increase to 52.19%.

Due to limited resources, we only evaluate the vital components based on *ResNet*16 and *ResNet*101, i.e., CS Attention, Transformer, and Attention and Transformer. **CS Attention** (45.07% for *ResNet*18 based and 54.06% for *ResNet*101 based) achieves around 1% improvement compared with **Baseline** (43.92% for *ResNet*18 based and 52.33% for *ResNet*101 based). **Transformer** (44.64% for *ResNet*18 based and 52.47% for *ResNet*101 based) brings slight increment against **Baseline**, while the final model, **Attention and Transformer** (45.95% for *ResNet*18 based and 54.59% for *ResNet*101 based), can bring around 2% improvement. These results again indicate that our proposed attention models can learn useful attributes and work well together. From the results for *Recall*@5, we can draw the same conclusion. All of them verify the effectiveness of the proposed model.

4.4.2. Comparison on mutual loss and sampling method

In this section, we deliver comparison experiments on different aspects: (1) Hard-mining and Hardest-mining; (2) Different weights of Triplet Loss; and (3) Training with additional Mutual Loss. If not specifically noted, the models are all trained with Triplet Loss and Classification Loss. The related experimental results are shown in Table 3.

We mainly report the results obtained by the *ResNet*50 based model. The basic model trained by the **hard-mining** sampling method with the hyper-parameter α set as 1 achieves 48.55% *Recall*@1. When we use the **hardest-mining** sampling method instead of the hard-mining sampling method, the model gets a better result, 49.37% *Recall*@1. This may be due to the overfitting problem of hard-mining sampling, for it uses every possible triplet in a mini-batch during training. However, hard-mining sampling is more stable and converges slightly faster in practice. Different loss weights have a great influence on the retrieval result. By increas-

Table 4
The comparison results with existing approaches.

Category	Model	*Recall*@1
Siamese Network [3]	GN Siamese	27.36%
	AN Siamese	21.36%
Triplet Network [3]	GN Triplet	37.10%
	GN Triplet w/o Cat	22.78%
TC-Net [11]	TC-Net	40.02%
Quadruplet Network [40]	Quadruplet_MT	38.21%
	Quadruplet_MT_V2	42.16%
DCCRM [13]	DCCRM(S+I)	40.16%
	DCCRM(S+I+D)	46.20%
Human [3]	-	54.27%
Our Model	**Attention and Transformer (*ResNet*18)**	**45.95%**
	Attention and Transformer (*ResNet*50)	**52.19%**
	Attention and Transformer (*ResNet*101)	**54.59%**

ing the weight value of triplet loss properly, the power of our model can be effectively improved. Compared with the baseline model, our model with the hyper-parameter $\alpha = 50$ can achieve better results whether it is trained with hard-mining or hardest-mining. This shows that Triplet Loss plays an important role to help the model figure out the fine-grained differences among similar sketches and images. The best result is obtained by training with additional **Mutual Loss** and hardest-mining. It increases *Recall*@1 to 52.01% and *Recall*@5 to 85.50%, which respectively brings 0.56% and 0.81% promotion against the best model with hardest-mining only. This reflects the effectiveness of our proposed Mutual Loss.

4.5. Comparisons with existing approaches

To further demonstrate the effectiveness of our AE-Net, we make comparisons with several typical approaches. The related experimental results are shown in Table 4.

- **Siamese Network** [3]: This network consists of two asymmetric branches corresponding to sketch and image. The branch of *GN Siamese* is based on *GoogLeNet*, while the branch of *AN Siamese* is based on *AlexNet*. Both of them are trained with Siamese and classification loss.
- **Triplet Network** [3]: This network consists of three branches corresponding to sketch, positive image, and negative image, which contains two models of *GN Triplet* and *GN Triplet w/o Cat*. Different from *GN Triplet* that is trained with triplet and classification loss, *GN Triplet w/o Cat* is trained only with triplet loss.
- **TC-Net** [11]: This network adopts the same triplet network framework as [5], sharing the weights of the networks in sketch and image branches. However, it is trained by various loss functions that are Spherical Loss, Central Loss, Classification Loss, and Softmax Loss.
- **Quadruplet Network** [40]: This network is similar to Triplet Network. However, the branches share weights for both sketch and image, and are based on the *ResNet*18 architecture. This model is trained by two stages, firstly with classification loss on *Sketchy* and then with triplet loss on the same dataset while mining different types of triplets.
- **DCCRM** [13]: This model uses the same triplet network as [3], where the networks in sketch branch and image branch are independent. Additionally, the model explores to use all the beneficial information, that is sketches, images, and text descriptions.

For a given query sketch, there are usually a few visually similar images. Higher accuracy of ranking, especially the accuracy of top-1 retrieval, can provide a better indication of the degree to which the model distinguishes the fine-grained subtle differences among candidate images. It can be observed from Table 4 that our model achieves the best performance, and significantly outperforms all the other existing models on *Sketchy*. Our model with *ResNet18* as basic backbone can achieve 45.95% *Recall*@1, which delivers 18.59%, 8.85%, 5.93%, and 3.79% absolute increases against *GN Siamese* [3], *GN Triplet* [3], *TC-Net* [11], and *Quadruplet_MT_V2* [40]. It only performs worse than *DCCRM* [13], which uses more modality information than ours. However, the text descriptions are not always available and hard to obtain, which makes it difficult for this model to extend to other datasets or real-life scenarios. Most of the above methods are trained with complex pre-training strategies. *Siamese Network* [3] and *Triplet Network* [3] are first trained to classify the sketches and images, and then fine-tuned on the *Sketchy* database [3]. *DCCRM* [13] are trained on sketch-like images, before being fine-tuned on real sketches. Compared with them, our model only uses the pre-trained weights on *ImageNet* [39], which shows the superiority and convenience of our model. With the layers of our basic backbone going deeper, from *ResNet18* to *ResNet101*, the performance of the model is getting better. Our model based on *ResNet*50 and *ResNet*101 all surpass the previous methods. The best *Recall*@1 can reach 54.59%, which is obviously superior to all the other existing methods. Except for the extraction ability of the basic model becoming stronger, our proposed attention mechanisms also play important roles. They can capture more useful fine-grained details based on the more exact features.

4.6. Visualization and qualitative analysis

In this section, we give some visualization examples of the attention maps and retrieval results obtained by our model. In Fig. 5, we visualize the attention map generated from the penultimate self-attention layer of **Local Self-attention** in Table 2. There are 15 examples of good retrieval results (the corresponding images are ranked at the first in the retrieval results) and 5 examples of poor retrieval results (the corresponding images are not ranked at the first). Those examples of poor results are shown in the last 5 rows of Fig. 5(a). For each image-sketch pair, we randomly choose 5 regions and show the attention maps corresponding to them. The spatial attention maps show the relevance of corresponding area regions in the image for the given patch. These regions are marked in red. The attention maps show that self-attention can build up the long range dependence among the regions. It can enhance the feature of a given region with the important regions that contribute a lot to retrieval. We also find self-attention in the image network tends to model the relations among a wide range of regions, such as the whole head of the *crocodile* and the whole body and feet of the *spider*, while in the sketch network it mainly focuses on several key regions. This is mainly due to the difference between these two modalities. The images are rich in color and texture, but the sketches only have strokes. Despite this, our self-attention model can still align the same important regions of these two kinds of inputs.

From the good retrieval results, we can observe that our self-attention models can make the sketch network and image network focus on the same vital parts, like the face of the *ape*, the nose of the *pig*, the body of the *banana*, etc. This can significantly narrow the domain gap between the sketches and images. From the bad retrieval results (the last 5 rows of Fig. 5(b)), we can see that the self-attention focuses on the wrong regions. However, that is reasonable because the noises from the environment occupy most of the space and the right item only occupies a tiny space. We also give some comparisons between the good and bad results belonging to the same category, which are shown in the last 5 rows of

Fig. 5(a) and (b). These results are sound evidence of the above conclusions.

Additionally, we also offer some qualitative results of our AE-Net in Fig. 6. For each sketch, we visualize the top-10 retrieval results obtained by our different models. They are **Baseline** model, **Local Self-attention** model, **Residual Channel Attention** model, and **Spatial Sequence Transformer** model in each row from top to down. The true match images are marked by red and the images belong to the same category as the given sketch are marked in green, while the others are marked by white. The retrieval results of the *rabbit* sketch show that our model can not only do well for fine-grained sketch-based retrieval but also for coarse-grained. Those results almost all belong to the *rabbit* category. By comparing the retrieval results of *frog* obtained by different models, we can see that our attention models can improve the retrieval results. There are some bad results, but they all have the same posture as the *frog* sketch. It can be observed that our model with Attention and Transformer does a better job of eliminating the ambiguity of subtle visual details. However, there are still some failure cases, like *ape* in Fig. 6. Although our attention models can rank the ground-truth image first on the list, the other images are almost all from different categories, not even the same category as the given sketch. This may due to the quality of the sketches. The given *ape* sketch contains too much hair, which may mislead the model into giving the wrong results.

4.7. Parameter analysis

We analyze the changing curves of the two parameters δ and γ with epochs as shown in Figs. 7 and 8. The identification number of the curve represents which block the parameter controls. In general, the absolute values of δ and γ increase with epochs, especially the weight of the specific block (i.e., the second curve for γ and the forth for δ), which means the attention mechanism becomes more and more important as the training proceeds. Besides, the absolute values are not very big, which proves the importance of the residual connection.

4.8. Extension experiment

We extend our method to Coarse-grained SBIR(CG-SBIR) on the Quickdraw[2] dataset. The changing curves of *Recall*@1 and *Recall*@5 with epochs are shown as below. It can be seen that the best *Recall*@1 reaches around 0.58 and the best *Recall*@5 reaches around 0.68, which indicate the significant effect of our model in CG-SBIR, though it is not specially designed for CG-SBIR (Fig. 9 and 10).

5. Conclusion and future work

We propose a novel deep model with channel attention, spatial attention, and Transformer for FG-SBIR. By introducing the residual channel attention module, our model can focus on the subtle differences between sketches and images, and calculate deep features including fine-grained and high-level semantics. By introducing the self-attention module, our model can build up the long region dependence among the pixels. By introducing Spatial Sequence Transformer, the misalignment of the sketch sequences and image sequences are alleviated. Besides, the proposed Mutual Loss further enhances the robustness of our model. Our work has positive implications for tasks like cross-modal image retrieval and contrastive learning. Our future work will focus on network lightweight and miniaturization while ensuring the effectiveness of retrieval.

[2] https://quickdraw.withgoogle.com/data

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.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61976057, No. 62172101), the Science and Technology Commission of Shanghai Municipality (No.21511101000, No. 20511101203, No. 20511102702, No. 20511101403, No.19DZ2205700), the Science and Technology Major Project of Commission of Science and Technology of Shanghai (No.2021SHZDZX0103), Shanghai Natural Science Foundation (No. 19ZR1417200), and Humanities and Social Sciences Planning Fund of Ministry of Education of China (No. 19YJA630116). Yangdong Chen, Zhaolong Zhang and Yanfei Wang are co-first authors. Yuejie Zhang and Tao Zhang are corresponding authors.

References

[1] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects? ACM Trans. Graph. (TOG) 31 (4) (2012) 1–10.
[2] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, T. Hospedales, Sketch-a-net that beats humans, arXiv preprint arXiv:1501.07873(2015).
[3] P. Sangkloy, N. Burnell, C. Ham, J. Hays, The sketchy database: learning to retrieve badly drawn bunnies, ACM Trans. Graph. (TOG) 35 (4) (2016) 1–12.
[4] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, Sketch-based image retrieval: benchmark and bag-of-features descriptors, IEEE Trans. Vis. Comput. Graph. 17 (11) (2010) 1624–1636.
[5] Q. Yu, F. Liu, Y.-Z. SonG, T. Xiang, T. Hospedales, C.C. Loy, Sketch me that shoe, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 799–807.
[6] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, T.M. Hospedales, Deep spatial-semantic attention for fine-grained sketch-based image retrieval, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5551–5560.
[7] P. Xu, Deep learning for free-hand sketch: a survey, arXiv preprint arXiv:2001.02600(2020).
[8] T. Kato, T. Kurita, N. Otsu, K. Hirata, A sketch retrieval method for full color image database-query by visual example, in: Proceedings. 11th IAPR International Conference on Pattern Recognition, IEEE, 1992, pp. 530–533.
[9] Y. Li, T.M. Hospedales, Y.-Z. Song, S. Gong, Fine-grained sketch-based image retrieval by matching deformable part models, in: British Machine Vision Conference (BMVC), 2014.
[10] S. Wang, J. Zhang, T.X. Han, Z. Miao, Sketch-based image retrieval through hypothesis-driven object boundary selection with HLR descriptor, IEEE Trans. Multimedia 17 (7) (2015) 1045–1057.
[11] H. Lin, Y. Fu, P. Lu, S. Gong, X. Xue, Y.-G. Jiang, TC-Net for iSBIR: triplet classification network for instance-level sketch based image retrieval, in: Proceedings of the 27th ACM International Conference on Multimedia (ACM MM), 2019, pp. 1676–1684.
[12] J. Song, Y.-Z. Song, T. Xiang, T. Hospedales, X. Ruan, Deep multi-task attribute–driven ranking for fine-grained sketch-based image retrieval, in: Proceedings of the 27th British Machine Vision Conference (BMVC), 2016, p. 132.1.
[13] Y. Wang, F. Huang, Y. Zhang, R. Feng, T. Zhang, W. Fan, Deep cascaded cross–modal correlation learning for fine-grained sketch-based image retrieval, Pattern Recognit. 100 (2020) 107148.
[14] A.K. Bhunia, P.N. Chowdhury, A. Sain, Y. Yang, T. Xiang, Y.-Z. Song, More photos are all you need: semi-supervised learning for fine-grained sketch based image retrieval, arXiv preprint arXiv:2103.13990(2021).
[15] Y. Shen, L. Liu, F. Shen, L. Shao, Zero-shot sketch-image hashing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3598–3607.
[16] Q. Liu, L. Xie, H. Wang, A.L. Yuille, Semantic-aware knowledge preservation for zero-shot sketch-based image retrieval, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3662–3671.
[17] V.K. Verma, A. Mishra, A. Mishra, P. Rai, Generative model for zero-shot sketch-based image retrieval, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2019, pp. 704–713.
[18] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473(2014).
[19] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International Conference on Machine Learning (ICML), 2015, pp. 2048–2057.
[20] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, A. Courville, Describing videos by exploiting temporal structure, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4507–4515.

[21] J. Wang, W. Wang, L. Wang, Z. Wang, D.D. Feng, T. Tan, Learning visual relationship and context-aware attention for image captioning, Pattern Recognit. 98 (2020) 107075.
[22] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3146–3154.
[23] P. Zhang, W. Liu, H. Wang, Y. Lei, H. Lu, Deep gated attention networks for large-scale street-level scene segmentation, Pattern Recognit. 88 (2019) 702–714.
[24] Y. Peng, X. He, J. Zhao, Object-part attention model for fine-grained image classification, IEEE Trans. Image Process. 27 (3) (2017) 1487–1500.
[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2921–2929.
[26] J. Lei, Y. Song, B. Peng, Z. Ma, L. Shao, Y.-Z. Song, Semi-heterogeneous three-way joint embedding network for sketch-based image retrieval, IEEE Trans. Circuits Syst. Video Technol. (2019).
[27] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T.-S. Chua, SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5659–5667.
[28] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141.
[29] S. Woo, J. Park, J.-Y. Lee, I. So Kweon, CBAM: convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[31] H. Zhao, J. Jia, V. Koltun, Exploring self-attention for image recognition, arXiv preprint arXiv:2004.13621(2020).
[32] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-attention generative adversarial networks, arXiv preprint arXiv:1805.08318(2018).
[33] I. Bello, B. Zoph, A. Vaswani, J. Shlens, Q.V. Le, Attention augmented convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 3286–3295.
[34] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, D. Tran, Image transformer, in: International Conference on Machine Learning, PMLR, 2018, pp. 4055–4064.
[35] C.-Y. Wu, R. Manmatha, A.J. Smola, P. Krahenbuhl, Sampling matters in deep embedding learning, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2840–2848.
[36] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815–823.
[37] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805(2018).
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.
[40] O. Seddati, S. Dupont, S. Mahmoudi, Quadruplet networks for sketch-based image retrieval, in: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (ICMR), ACM, 2017, pp. 184–191.

Yangdong Chen received the B.S. degree in Computer Science from Fudan University, Shanghai, China, in 2019. He is currently a Ph.D. student in School of Computer Science, Fudan University, Shanghai, China. He is a member of Institution of Media Computing in School of Computer Science. His research interest is cross-modal and cross-domain information analysis, processing, and generation.

Zhaolong Zhang received the B.S. degree in Software Engineering from Tongji University, Shanghai, China, in 2018. He is currently a master student in School of Computer Science, Fudan University, Shanghai, China. He is a member of Institution of Media Computing in School of Computer Science. His research interest is multimedia information analysis, processing, and modeling.

Yanfei Wang received the B.S. degree in Computer Science from Sun Yat-sen University, Guangzhou, China, in 2017. He is currently a master student in School of Computer Science, Fudan University, Shanghai, China. He is a member of Institution of Media Computing in School of Computer Science. His research interest is cross-media retrieval and image synthesis/translation, including sketch-based image retrieval, multi-view/multimodal correlation learning, and sketch synthesis.

Yuejie Zhang received the B.S. degree in Computer Software, the M.S. degree in Computer Application, and the Ph.D. degree in Computer Software and Theory from Northeastern University, Shenyang, China, in 1994, 1997 and 1999, respectively. She was a Postdoctoral Researcher at Fudan University, Shanghai, China, from 1999 to 2001. In 2001, she joined Department of Computer Science and Engineering (now School of Computer Science), Fudan University as an Assistant Professor, and then become Associate Professor and Full Professor. Her research interests include multimedia/cross-media information analysis, processing, and retrieval, and machine learning.

Rui Feng received the B.S. degree in Industrial Automatic from Harbin Engineering University, Haerbin, China, in 1994, the M.S. degree in Industrial Automatic from Northeastern University, Shenyang, China, in 1997, and the Ph.D. degree in Control Theory and Engineering from Shanghai Jiaotong University, Shanghai, China, in 2003. In 2003, He joined Department of Computer Science and Engineering (now School of Computer Science), Fudan University as an Assistant Professor, and then become Associate Professor and Full Professor. His research interests include multimedia information analysis and processing, and machine learning.

Tao Zhang received the B.S. and M.S. degree in Automation Control, and the Ph.D. degree in System Engineering from Northeastern University, Shenyang, China, in 1992, 1997 and 2000, respectively. He was a Postdoctoral Researcher at Fudan University, Shanghai, China, from 2001 to 2003. In 2003, he joined School of Information Management and Engineering, Shanghai University of Finance and Economics as an Associate Professor and then become Full Professor. His research interests include big data analysis and mining, system modeling and optimization.

Weiguo Fan received the B.S. degree in information and control engineering from the Xi'an Jiaotong University, Xian, China, in 1995, the M.S. degree in computer science from the National University of Singapore in 1997, and the Ph.D. degree in AI and Information Systems from the University of Michigan, Ann Arbor, in 2002. He is currently Henry Tippie Chaired professor of business analytics at the University of Iowa. He has published more than 200 refereed articles in many premier IT/IS journals and conferences such as TKDE, PR, TOIT, WWW, SIGIR, CIKM, AAAI, and KDD. His research interests include information retrieval, data mining, text mining, Web mining, and pattern recognition.