VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY

MAJOR: COMPUTER SCIENCE

COURSE: CS336.O11.KHTN - MULTIMEDIA INFORMATION RETRIEVAL

Sketch-based Image Retrieval System

Students:
Cuong Le Chi – 21520012
Anh Nguyen Tuan – 21520142
Bach Luong Toan – 21521845
Tri Phan Truong – 21520117
Thinh Nguyen Truong – 21520110

Supervisor:
PhD. Thanh Ngo Duc

Contents

List of Figures

1. Introduction

Sketch-based Image Retrieval (SBIR) is a method used to find images in databases that match a sketch provided by the user. This approach is particularly useful when textual descriptions of the desired image are difficult to formulate, making traditional keyword-based search methods inefficient or ineffective. SBIR systems allow users to quickly and intuitively search for images by simply drawing a rough sketch of what they are looking for. Nowadays, SBIR is gaining increased attention due to its ability to provide intuitive, efficient, and innovative solutions for image retrieval. This growth is fueled by technological advancements and the expanding role of visual communication in the digital landscape.

Input

- Query sketch: The user provides a sketch as input, which is typically a simplified representation of the desired image, focusing on shapes, outlines, and key features.
- Images collection.

Output

- Images from collection that relevant to query.

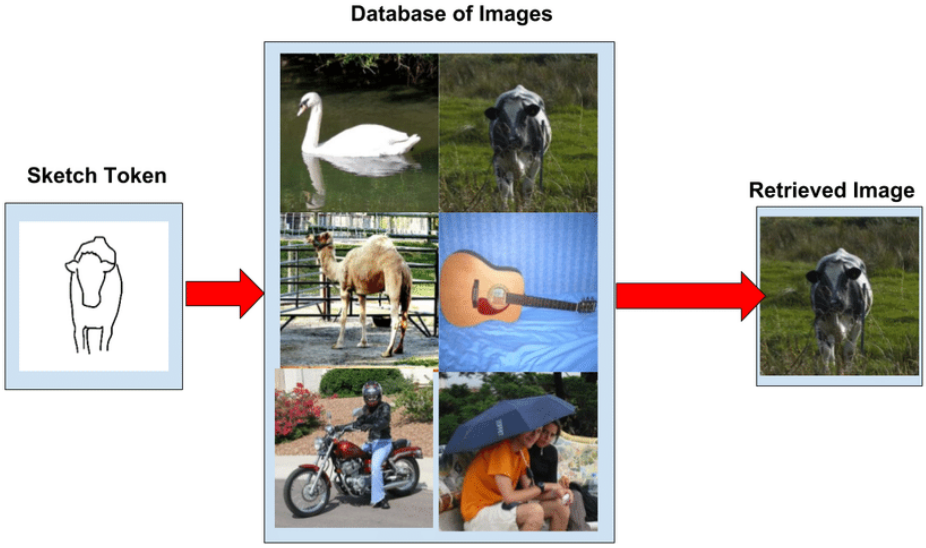

Figure 1: Illustration of Sketch-based Image Retrieval.

Zero-Shot Sketch-based Image Retrieval

Unlike traditional sketch-based image retrieval systems, which require examples of sketches corresponding to the images in the database during the training phase, Zero-Shot Sketch-based Image Retrieval (ZS-SBIR) systems are designed to work in scenarios where the sketch query may represent categories not seen during training. In other words, Zero-Shot Sketch-based Image Retrieval is a combination of Zero-Shot Learning and Sketch-based Image Retrieval.

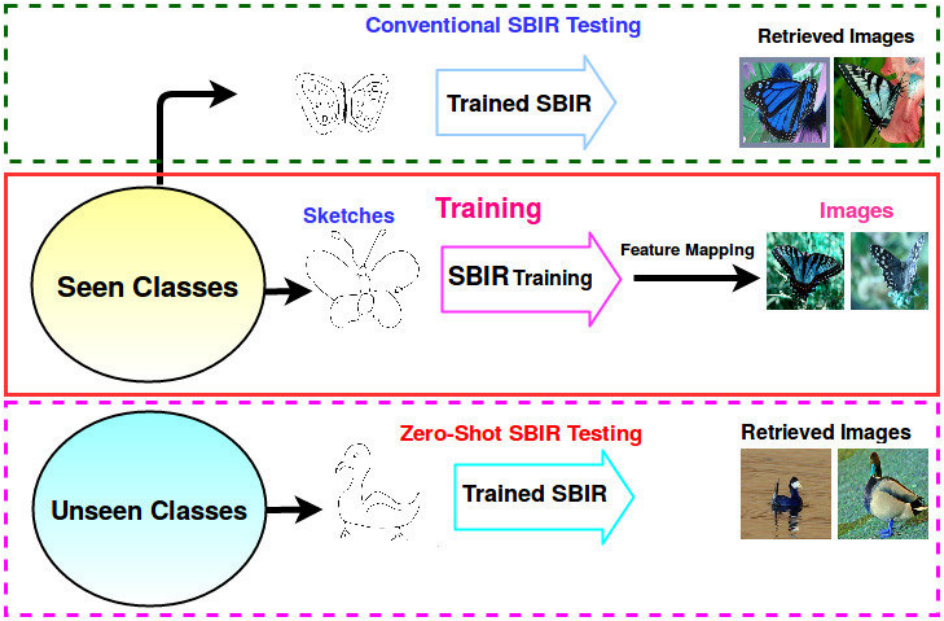

Figure 2: Illustration of Zero-Shot Sketch-based Image Retrieval.

2. Retrieval Pipeline

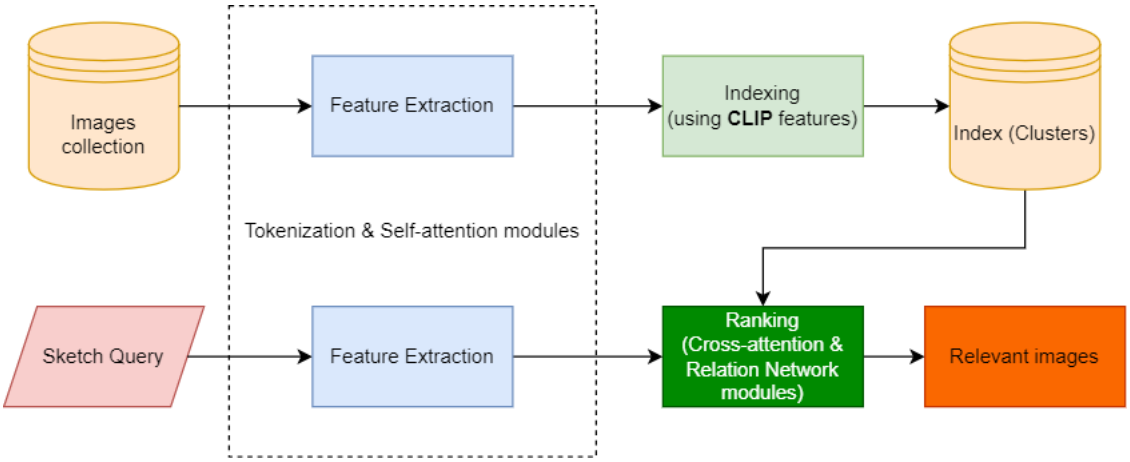

Figure 3: The pipeline of our retrieval system.

We use the pipeline and the workflow of the system as shown in Figure 3. Some information about the pipeline is as follows:

- Sketch Query: The user submits a sketch.
- Images Collection: A set of images from which the system will retrieve.
- Feature Extraction: Each image in the collection goes through a feature extraction module, where important information and features of the image are summarized.
- Indexing: After feature extraction, the system creates an index for the images, based on features encoded by CLIP.
- Index (Clusters): Classifying images into clusters to optimize the retrieval process.
- Ranking: The features of the sketch query and the image indices are compared and ranked.
- Relevant Images: The system returns the highest-ranked results, which are considered the most relevant images to the sketch query provided by the user.

The specific implementation of the pipeline and the techniques applied will be discussed in the **Methodology** section.

Our source code is available at https://github.com/LTBach/ZSE-SBIR_CLIP.git.

3. Methodology

3.1. Feature Extraction

We using **Tokenization** and **Self-Attention** modules of ZSE-SBIR [1] in order to extract the important features of each images in the collection.

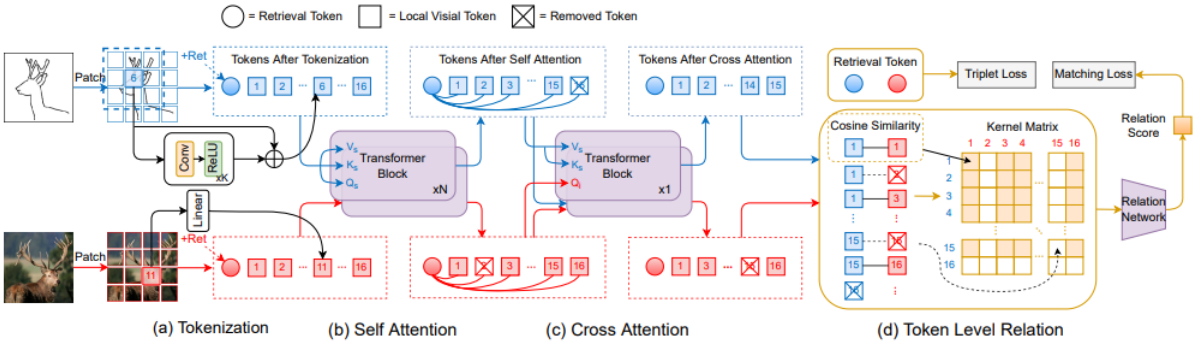

Figure 4: Network overview.

3.1.1. Tokenization

Given a query sketch $S \in \mathbb{R}^{h\times w\times c}$ and a gallery image $I \in \mathbb{R}^{h\times w\times c}$ to be matched, tokenizing them into a sequence of visual tokens by using the same approach proposed in ViT [4] where images are evenly partitioned into non-overlapping patches, followed by a projection head mapping them into $S' \in \mathbb{R}^{n\times d}$ and $I' \in \mathbb{R}^{n\times d}$. However, this tokenization is not friendly to sketches, which are typically composed of sparse strokes. To address this issue, the ZSE-SBIR's authors proposed a learnable tokenizer, which transforms a given sketch $S \in \mathbb{R}^{h\times w\times c}$ into a sequence of visual embeddings $X \in \mathbb{R}^{n\times d}$. Specifically, the tokenizer is made up of a stack (K = 4) of convolution layers (Conv) with various kernel size, each followed by a non-linear activation (σ=ReLU): $X = [\sigma(\texttt{Conv}(S))]_{\times 4}$. Essentially, it can enlarge the receptive field when constructing visual tokens through hierarchical convolution, thereby better preserving structural cues from nearby regions. Additionally, a residual connection is introduced to rectify the vanilla tokens, then the final token embedding is: $X = X + S'$.

3.1.2. Self-Attention

Different from the vanilla ViT [4], the ZSE-SBIR's authors replace the vanilla class token with a retrieval token [`Ret`] to facilitate the retrieval task by capturing a global representation of an image. Specifically, the retrieval token is initialized as a trainable d-dimensional token embedding [`Ret`] $\in \mathbb{R}^d$. During the model inference, all visual tokens including the retrieval

token [Ret] interact with each other through the multi-head self attention (MSA) modules, followed by MLP blocks. Thus, we get self-attention features.

3.2. Indexing

CLIP (Contrastive Language-Image Pretraining) is one of the state-of-the-art methods in the Text-Image Retrieval task. However, we do not use it directly for retrieval but will employ CLIP embeddings to aid the Indexing process. We perform the clustering step to be able to use them. This means we use CLIP features to cluster images in database.

3.2.1. CLIP

The main reason we use CLIP is that it achieves SOTA performance. Additionally, we need to encode images into a vector space where images with similar content will have closely situated vectors. Another reason is that the features from our dataset are too large, exceeding the memory capacity required for traditional clustering: our features have a size of (196, 768), whereas CLIP's are only (512,).

3.2.2. Clustering method

We chose K-means for clustering after extracting CLIP embeddings. During experimentation, we selected the number of centroids to be 125. The representation vector of each cluster is created by calculating the mean of self-attention features of all images in that cluster.

3.3. Ranking

Utilizing **Cross-Attention** and **Relation Network** modules of ZSE-SBIR to rank and find relevant images. First, using Cross-Attention and Relation Network modules to ranking clusters' representation vectors to get top relevant cluster(s). Then, using these 2 modules one more time to ranking images in the top relevant cluster(s) and return retrieval results to the users.

3.3.1. Cross-Attention

The self-attention module learns an informative tokenbased representation of each image. To estimate local visual correspondences between sketch and photo tokens, we resort to cross attention. The idea is to find pair-wise connections between visual tokens from different modalities, i.e., sketch and photo. This can be achieved by swapping the sketch query and image query, resulting in the new Query, Key and Value tuples. In this way, sketch token embeddings are updated by the information from photo tokens. Photo token embeddings can be obtained in the same way.

3.3.2. Relation Network

Inspired by Relation Network proposed in [2], the ZSE-SBIR's authors incorporate a relation network in their framework to estimate the matching score of a particular sketch-photo pair (S, I), based on their associated local correspondence kernel matrix $M^{S,I}$. Specifically, our relation network $R_\psi(.)$ is a stack of two FC-ReLU-Dropout layers that can produce a relation score in the range of $(0, 1)$:

$$r(S, I) = \texttt{sigmoid}(R_\psi(M^{S,I}))$$

Unlike concatenating global image features in [2], their relation network conducts reasoning on local token similarities, thereby has the opportunity to learn which (set of) token correspondences (embedded in $M^{S,I}$) to prioritize during matching. In the end, retrieval can be performed by ranking gallery images according to their relation scores. Thus the result of Relation Network is score(s) that show how much the image(s) relevance to the query sketch.

3.4. Losses

Triple loss: Given a triplet $<S_i, I_i^+, I_i^->$ where S_i is an anchor sketch, I_i^+ is a photo with the same label to S_i while I_i^- from a different class.
Target: minimized positive pair $<S_i, I_i^+>$ and push the anchor S_i away from the negative instance I_i^-.
Using triple loss on retrieval token `Ret` that is used as the global feature of sketches and photos, thus the triplet loss is defined as:

$$L_{tri} = \frac{1}{T}\sum_{i=1}^{T} \max(||\texttt{Ret}(S_i) - \texttt{Ret}(I_i^+)|| - ||\texttt{Ret}(S_i) - \texttt{Ret}(I_i^-)|| + m, 0)$$

T is the total number of triplets, and m is the margin.
Mean square error: In addition, to measure a sketch-photo pair belongs same class or not through kernel based relation network, we use the matching loss like MSE on total numbers of query sketches and candidate photos.

$$L_{re} = \sum_{i=1}^{N}\sum_{j=1}^{H}(r_{i,j} - 1(y_i == y_j)^2)$$

N: number of query sketches, m candidate photos, $r = 1$ when matched and $r = 0$ otherwise.
Overall loss is summed as:

$$L = L_{tri} + L_{re}$$

4. Evaluation

4.1. Dataset

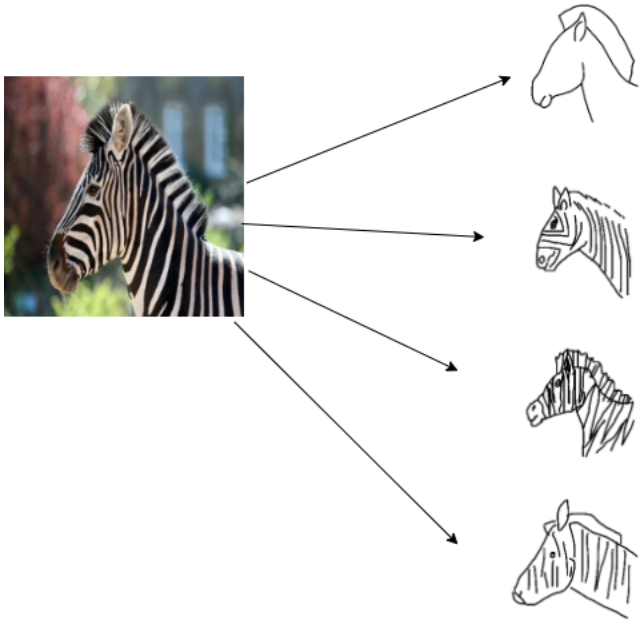

Figure 5: Dataset illustration.

We using Sketchy Ext that is an extended version of Sketchy dataset [3]. Each image is resized at 256x256. Sketchy dataset includes 125 categories (100 photos per class and 5-8 corresponding sketches per photo). We expanded by adding 60,502 photos, resulting in an enlarged photo gallery with 70,002 images, average: 560 images per class. About sketch query there is totally 75,481 image outline correspondingly 12500 photos above, average: 6 sketches per photo.

4.2. Category-level ZSE-SBIR comparison results

Table 1 [1] compare ZSE-SBIR method with several baselines, including ZSIH [5], CC-DG [6], DOODLE [7], SEM-PCYC [8], SAKE [9], SketchGCN [10], StyleGuide [11], PDFD [12], DSN [13], BDA-SketRet [14], SBTKNet [15], Sketch3T [16], TVT [17] and ViT-Ret/ViTVis [4] adapted by the authors. ViT-Ret means replacing the class token in ViT with a retrieval token used for matching; while ViT-Vis uses the visual tokens for matching. It should be noted that all the baselines, except CC-DG [5], StyleGuide [11] and ViT variants [4], employ external semantic information, whereas ZSE-SBIR only relies on the learned visual correspondences between sketch-photo pairs.

Evaluation metrics: Mean average precision (mAP), precision on top 100 (Prec@100) mAP is a standard metric for evaluating the performance of ranking models in information

Table 1: Category-level ZS-SBIR comparison results. "ESI": External Semantic Information. "-": not reported.

Method	ESI	$\mathbb{R}^D$	Sketchy Ext	
			mAP	Prec@100
ZSIH	✓	64	0.254	0.340
CC-DG	×	256	0.311	0.468
DOODLE	✓	256	0.359	-
SEM-PCYC	✓	64	0.349	0.463
SAKE	✓	512	0.547	0.692
SketchGCN	✓	300	0.382	0.538
StyleGuide	×	200	0.376	0.484
PDFD	✓	512	0.661	0.781
ViT-Vis	×	512	0.410	0.569
ViT-Ret	×	512	0.483	0.637
DSN	✓	512	0.583	0.704
BDA-SketRet	✓	128	0.437	0.514
SBTKNet	✓	512	0.553	0.698
Sketch3T	✓	512	0.575	-
TVT	✓	384	0.648	0.796
Our-RN (paper)	×	512	0.698	0.797
Our-Ret (paper)	×	512	0.736	0.808

retrieval and image retrieval tasks. It's a measures overall ranking performance and allows comparison between models. While mAP focuses on the entire list, Prec@100 is accuracy of the top 100 results providing additional insights into a model's ability to retrieve relevant items at the forefront.

Results:

From Table 1, we can see that ZSE-SBIR [1] achieving competitive or better results without extra semantic data (text or labels).

4.3. System evaluation

Implement: To evaluate the system, we employ 375 sketch queries derived from the sketch dataset Ext, ensuring 3 queries for each of the 125 classes. Each query is independently evaluated to measure the system's overall performance.

Evaluation metrics: We evaluate on three perspectives: mAP@100, Prec@100 (because dataset have exactly 100 relevant images, Prec@100 also equivalent to Recall) and average search time per query (s/query).

Results: Clustering significantly influences retrieval time. We have 125 clusters, equivalent

Table 2: Compare search entire images and our approach using CLIP and Kmean.

	mAP@100	mPrec@100	average_seach_time
Search the entire database	0.746	0.649	71.160
Search through our method	0.744	0.634	1.214

to the number of classes. However, the number of images in each cluster can differ, leading to faster retrieval times for clusters with fewer images. We can adjust the cluster count to achieve more optimal times, but this will require a trade-off between retrieval speed and accuracy. During our experiments, we found that the accuracy post-clustering is sufficiently good and can be practically deployed (Table 2).

5. Conclusions

As can be seen, the method we have presented achieves stable search performance. The application of the strengths of CLIP (Contrastive Language-Image Pretraining) combined with clustering techniques also significantly improves the time and ability of retrieval. Furthermore, the method is very practical and natural, as the way humans store information in the brain is similar. In the future, we plan to expand our system to integrate additional tools for sound and text. We also have plans to extend into the real world, supporting fields like photography and surveillance systems.

References

[1] Fengyin Lin and Mingkang Li and Da Li and Timothy Hospedales and Yi-Zhe Song and Yonggang Qi.Zero-Shot Everything Sketch-Based Image Retrieval, and in Explainable Style.In CVPR, 2023.

[2] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In CVPR, 2018.

[3] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawnbunnies. ACM TOG, 2016.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In ICLR, 2021.

[5] Yuming Shen, Li Liu, Fumin Shen, and Ling Shao. Zero-shot sketch-image hashing. In CVPR, 2018.

[6] Kaiyue Pang, Ke Li, Yongxin Yang, Honggang Zhang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Generalising fine-grained sketch-based image retrieval. In CVPR, 2019.

[7] Sounak Dey, Pau Riba, Anjan Dutta, Josep Llados, and YiZhe Song. Doodle to search: Practical zero-shot sketchbased image retrieval. In CVPR, 2019.

[8] Anjan Dutta and Zeynep Akata. Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval. In CVPR, 2019.

[9] Qing Liu, Lingxi Xie, Huiyu Wang, and Alan L Yuille. Semantic-aware knowledge preservation for zeroshot sketch-based image retrieval. In ICCV, 2019.

[10] Zhaolong Zhang, Yuejie Zhang, Rui Feng, Tao Zhang, and Weiguo Fan. Zero-shot sketch-based image retrieval via graph convolution network. In AAAI, 2020.

[11] Titir Dutta, Anurag Singh, and Soma Biswas. Styleguide: zero-shot sketch-based image retrieval using style-guided image generation. IEEE TMM, 2020.

[12] Cheng Deng, Xinxun Xu, Hao Wang, Muli Yang, and Dacheng Tao. Progressive cross-modal semantic network for zero-shot sketch-based image retrieval. IEEE TIP, 2020.

[13] Zhipeng Wang, Hao Wang, Jiexi Yan, Aming Wu, and Cheng Deng. Domain-smoothing network for zero-shot sketchbased image retrieval. IJCAI, 2021.

[14] Ushasi Chaudhuri, Ruchika Chavan, Biplab Banerjee, Anjan Dutta, and Zeynep Akata. Bda-sketret: Bi-level domain adaptation for zero-shot sbir. arXiv preprint arXiv:2201.06570, 2022.

[15] Osman Tursun, Simon Denman, Sridha Sridharan, Ethan Goan, and Clinton Fookes. An efficient framework for zero-shot sketch-based image retrieval. Pattern Recognition, 2022.

[16] Aneeshan Sain, Ayan Kumar Bhunia, Vaishnav Potlapalli, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketch3t: Test-time training for zero-shot sbir. In CVPR, 2022.

[17] Jialin Tian, Xing Xu, Fumin Shen, Yang Yang, and Heng Tao Shen. Tvt: Three-way vision transformer through multimodal hypersphere learning for zero-shot sketch-based image retrieval. In AAAI, 2022.