Revisiting Relevance Feedback for CLIP-based Interactive Image Retrieval

Ryoya Nara[1], Yu-Chieh Lin[2], Yuji Nozawa[2], Youyang Ng[2], Goh Itoh[2], Osamu Torii[2], and Yusuke Matsui[1]

[1] The University of Tokyo
[2] Kioxia Corporation

Abstract. Many image retrieval studies use metric learning to train an image encoder. However, metric learning cannot handle differences in users' preferences, and requires data to train an image encoder. To overcome these limitations, we revisit relevance feedback, a classic technique for interactive retrieval systems, and propose an interactive CLIP-based image retrieval system with relevance feedback. Our retrieval system first executes the retrieval, collects each user's unique preferences through binary feedback, and returns images the user prefers. Even when users have various preferences, our retrieval system learns each user's preference through the feedback and adapts to the preference. Moreover, our retrieval system leverages CLIP's zero-shot transferability and achieves high accuracy without training. We empirically show that our retrieval system competes well with state-of-the-art metric learning in category-based image retrieval, despite not training image encoders specifically for each dataset. Furthermore, we set up two additional experimental settings where users have various preferences: one-label-based image retrieval and conditioned image retrieval. In both cases, our retrieval system effectively adapts to each user's preferences, resulting in improved accuracy compared to image retrieval without feedback. Overall, our work highlights the potential benefits of integrating CLIP with classic relevance feedback techniques to enhance image retrieval.

Keywords: Interactive image retrieval · Relevance feedback · CLIP

1 Introduction

With the rapid growth of available images in recent years, image retrieval systems have become ubiquitous, particularly in e-commerce and internet search applications. Image retrieval systems receive a query image from a user, retrieve similar images, and return them to the user.

Many content-based image retrieval studies utilize metric learning [12,22,34]. However, in many cases, metric learning is insufficient because users may have various preferences for the returned images. For example, one may search for dog images by querying an image of a dog running in a park, and another may search for park images with the same query image. Simply mapping images into image features might not accommodate those varying preferences. Furthermore, metric learning requires abundant annotated data to train an image encoder.

To overcome these limitations of metric learning-based image retrieval methods, we revisit the classic relevance feedback technique by incorporating the Contrastive Language-Image Pre-training (CLIP) model [25]. We propose an image retrieval system that encodes images with a CLIP image encoder and employs relevance feedback to gather information about each user's preferences. The overall retrieval process is as follows:

1. Our retrieval system receives a query image from a user, retrieves images from a database, and returns them to the user.
2. The user reviews the returned images and provides binary feedback on each image, indicating whether the user prefers it.
3. The system learns the user's preference and returns images the user prefers.

While our retrieval system does require users to provide feedback on the returned samples, it is through the feedback that our system learns each user's unique preference, even when they vary greatly. Whenever a user provides a query image, our system learns the user's preference in real-time and returns relevant images that align with the preference. Moreover, our retrieval system does not necessitate training. By leveraging CLIP's zero-shot transferability and incorporating the user's feedback, we can achieve high retrieval accuracy. Furthermore, because our system does not rely on injecting preferences through text, it can handle diverse and complex queries while avoiding the limitations of multimodal retrieval methods.

We empirically show that our retrieval system achieves high accuracy in various settings. We evaluate our retrieval system by automatically generating users' feedback from dataset labels to simulate users' feedback. In category-based image retrieval settings, our retrieval system achieves competitive accuracy with state-of-the-art metric learning methods. Additionally, we set up two experimental settings where users have various preferences: one-label-based image retrieval and conditioned image retrieval. In these two experimental settings, our retrieval system achieves higher accuracy than image retrieval without relevance feedback.

In this work, we aim to cast a spotlight on relevance feedback again. Despite the significant advancements in deep learning techniques for image retrieval, relevance feedback has received less attention in recent years. However, our findings demonstrate that incorporating relevance feedback into modern deep learning-based image retrieval systems, such as those based on CLIP, can significantly improve their accuracy.

To sum up, our contributions are as follows:

- We propose CLIP-based interactive image retrieval with relevance feedback.
- We propose an evaluation method of image retrieval systems with relevance feedback to simulate users' feedback.
- With a realistic feedback size, our system achieves competitive results with supervised metric learning methods in category-based image retrieval, despite not training image encoders specifically for each dataset.

- We set up two experimental settings where users have various preferences. In both settings, our retrieval system improves accuracy from image retrieval without user feedback.
- With a realistic feedback size, our retrieval system achieves competitive results with state-of-the-art multimodal retrieval in conditioned image retrieval settings, despite not exploiting textual information.

2 Related Work

2.1 Metric Learning

Metric learning [7,9,12,15,35] is a major approach to training an image encoder for image retrieval systems. Metric learning trains an image encoder to map an image into an image feature so that semantically similar images are close together and dissimilar images are apart. Metric learning are utilized in image retrieval [15, 22], personal re-identification [6, 41], face recognition [20, 29, 37], landmark retrieval [39], and few-shot learning [30,31,38].

2.2 Interactive Image Retrieval

Interactive image retrieval systems enable users to inject their preferences into the retrieval system and obtain samples they prefer. Before the advent of deep learning, relevance feedback [11,26,40] had been a popular technique for learning each user's preference. Recently, multimodal retrieval [2, 28] has become mainstream of interactive image retrieval.

Relevance feedback [1, 13, 26, 40] is a technique for interactive retrieval systems. Each user provides feedback for the returned samples, and the retrieval system receives the user's feedback and executes retrieval again to meet their preferences better. In addition to one-time binary feedback, Ahmed [1] used scalers for multi-level feedback, and Wu *et al.* [40] studied retrieval systems where users provide feedback multiple times. Our work revisits relevance feedback. We focus on cases where users provide binary feedback just once.

Multimodal retrieval [2, 4, 28, 32, 33] enables users to inject their preferences through text. Multimodal retrieval addresses tasks such as composed image retrieval [4, 28] and conditioned image retrieval [2, 33]. For example, along with a query image of a long red dress, a user injects a text query of "I want something short and yellow." In response, the image retrieval system returns an image of a short yellow dress.

2.3 Contrastive Language-Image Pre-training (CLIP)

Contrastive language-image pre-training (CLIP) [25] is a pre-trained vision-language model trained on large-scale image-text pairs. CLIP consists of an image encoder that embeds images into a feature space and a text encoder that embeds strings into the same feature space. CLIP models achieve impressive results on various downstream tasks [3, 8, 21, 24, 27], and many studies leverage CLIP as powerful feature extractors.

3 Approach

First, we explain the overall retrieval pipeline in Sec. 3.1. Next, we describe how we develop a retrieval algorithm that adapts to each user's preference in Sec. 3.2. Finally, we explain how to accurately assess whether our retrieval system aligns with user preferences in Sec. 3.3.

3.1 Retrieval Pipeline

We execute retrieval twice for each query image $q \in \mathcal{I}$. Here, $\mathcal{I}$ is a space representing all possible images. We consider two databases: $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{I}$. First, our retrieval system executes retrieval for database images $\mathcal{X}_1$. Next, it updates the retrieval algorithm and executes retrieval for database images $\mathcal{X}_2$. In actual application, the database images in both retrievals are the same: $\mathcal{X}_1 = \mathcal{X}_2$, i.e., we have just one single database $\mathcal{X}$ and execute retrieval for $\mathcal{X}$ twice. We introduce the two databases here for a fair comparison, as we will explain in Sec. 3.3.

In advance, we prepare database features. We use visual encoder $\phi : \mathcal{I} \to \mathbb{R}^D$ and get database features in the first retrieval $\mathcal{V}_1 \coloneqq \{\phi(x) \mid x \in \mathcal{X}_1\} \subset \mathbb{R}^D$, and database features in the second retrieval $\mathcal{V}_2 \coloneqq \{\phi(x) \mid x \in \mathcal{X}_2\} \subset \mathbb{R}^D$.

When a user provides a query image q, our retrieval system encodes q into D-dimensional feature as $\mathbf{u} \coloneqq \phi(q) \in \mathbb{R}^D$. Our retrieval system executes retrieval for $\mathcal{V}_1$ to obtain similar features to $\mathbf{u}$. Let us denote the top-M similar samples to $\mathbf{u}$ from $\mathcal{V}_1$ as $\mathcal{W}_1 \subseteq \mathcal{V}_1$. We write this K-NN (Nearest Neighbor) retrieval operation as a function form in Eq. (1):

$$\mathcal{W}_1 = \psi(\mathbf{u}, M, \mathcal{V}_1) \coloneqq M\text{-}\operatorname*{argmax}_{\mathbf{w} \in \mathcal{V}_1} \frac{\mathbf{u}^\top \mathbf{w}}{\|\mathbf{u}\|_2 \|\mathbf{w}\|_2}. \tag{1}$$

Note that $|\mathcal{W}_1| = M$. Our retrieval system returns $\mathcal{W}_1$ to the user.

Next, the user provides binary feedback to each sample in $\mathcal{W}_1$. We formulate the feedback as Eq. (2):

$$\mathcal{F} = \{(\mathbf{w}, b) \mid \mathbf{w} \in \mathcal{W}_1\}. \tag{2}$$

Here, $b \in \{0, 1\}$ is the user's feedback. If the user prefers a returned sample $\mathbf{w}$, they provide $b = 1$ to $\mathbf{w}$. Otherwise, they provide $b = 0$. We can regard $\mathcal{F}$ as a labeled dataset where each sample $\mathbf{w}$ has a binary label b.

With $\mathcal{F}$, we update the retrieval algorithm ψ to $\tilde{\psi}$. Next, our retrieval system executes retrieval for $\mathcal{V}_2$ and retrieves top-K sample $\mathcal{W}_2 \subset \mathcal{V}_2$:

$$\mathcal{W}_2 = \tilde{\psi}(\mathbf{u}, K, \mathcal{V}_2). \tag{3}$$

$\mathcal{W}_2$ represents the final returned samples. We aim to update the retrieval algorithm so that each returned sample in $\mathcal{W}_2$ is preferable for the user. Moreover, we aim to choose the visual encoder ϕ to achieve high retrieval performance in various user preferences and datasets.

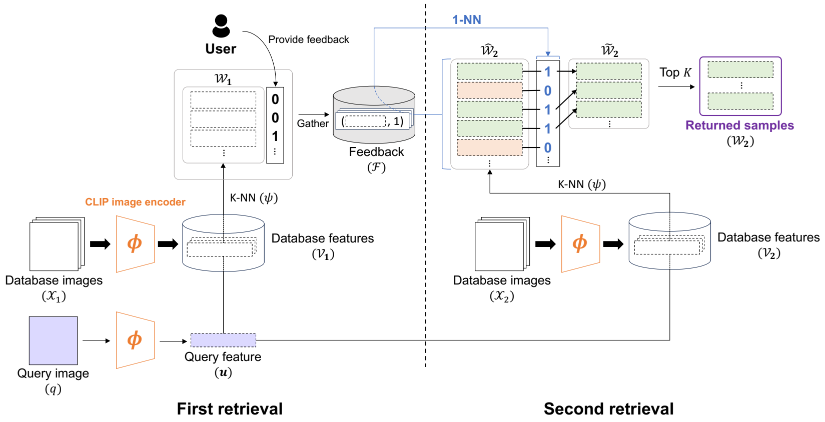


Fig. 1: Overall of our proposed method. The updated retrieval $\tilde{\psi}$ is a retrieval algorithm that retrieves and returns $\mathcal{W}_2$ from $\mathcal{V}_2$.

3.2 Proposed Method

Fig. 1 illustrates the overview of our proposed method. We propose a CLIP-based retrieval system with relevance feedback. When updating the retrieval algorithm, our retrieval system predicts whether each sample $\mathbf{w} \in \mathcal{W}_2$ is preferable for the user according to the information in $\mathcal{F}$.

We use CLIP visual encoder as our retrieval system's encoder ϕ. We aim to update the retrieval algorithm for various datasets and users' preferences, so we must extract appropriate information from images with an image encoder. Therefore, we utilize off-the-shelf CLIP because CLIP achieves surprising performance in many image recognition tasks without additional training.

After the first retrieval with the feedback $\mathcal{F}$, our retrieval system prepares a binary classifier $f : \mathbb{R}^D \to \{0, 1\}$ to predict the user's preference. There could be several options for f, but in our case, f is a simple 1-NN classifier over $\mathcal{F}$:

$$f(\mathbf{a}) = b_* \quad \text{where} \quad (\mathbf{y}_*, b_*) = \psi(\mathbf{a}, 1, \mathcal{F}), \tag{4}$$

Here, we use a slight abuse notation for ψ. This form of f enables us to update the retrieval algorithm online without requiring any additional training time.

In the second retrieval phase, our retrieval system utilizes f to execute the updated retrieval $\tilde{\psi}$ described in Eq. (3). First, our retrieval system executes K-NN retrieval as follows:

$$\hat{\mathcal{W}}_2 = \psi(\mathbf{u}, \hat{K}, \mathcal{V}_2). \tag{5}$$

Here, $\hat{K}$ is a sufficiently large value. Our retrieval system then refines $\hat{\mathcal{W}}_2$ by asking f to obtain the returned samples' candidates $\tilde{\mathcal{W}}_2$ as Eq. (6):

$$\tilde{\mathcal{W}}_2 = \{\mathbf{w}_2 \in \hat{\mathcal{W}}_2 \mid f(\mathbf{w}_2) = 1\}. \tag{6}$$

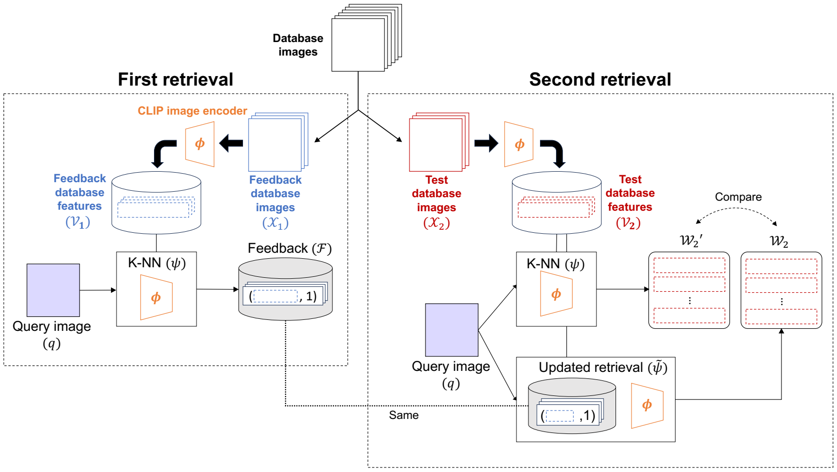

Database images
First retrieval
Second retrieval
CLIP image encoder
ϕ
Feedback database features $(\mathcal{V}_1)$
Feedback database images $(\mathcal{X}_1)$
Test database images $(\mathcal{X}_2)$
Test database features $(\mathcal{V}_2)$
Compare
$\mathcal{W}_2{}'$
$\mathcal{W}_2$
K-NN (ψ)
Feedback $(\mathcal{F})$
(, 1)
Query image (q)
Updated retrieval $(\tilde{\psi})$
(,1)
Same

Fig. 2: Evaluation of retrieval algorithm with relevance feedback. We omit the detail of the second retrieval process described in Fig. 1.

Finally, our retrieval system picks up the top K elements.

$$\mathcal{W}_2 = \tilde{\mathcal{W}}_2[1:K]. \tag{7}$$

We put Eqs. (5) to (7) all together to implement $\tilde{\psi}$ in Eq. (3).

3.3 Evaluation

Fig. 2 illustrates the overall evaluation framework. As we explained, in the practical application of relevance feedback, the database features in the first and second retrieval are the same: $\mathcal{X}_1 = \mathcal{X}_2$ and $\mathcal{V}_1 = \mathcal{V}_2$. However, we could not correctly compare the performance of ψ and $\tilde{\psi}$ in this way, because our retrieval system knows the ground-truth label of whether the user prefers each of the top-M returned samples in the database of the first retrieval.

To evaluate the update of retrieval algorithm performance correctly, we revisit the classic evaluation method of relevance feedback, called the test and control method [5, 13]. First, we split all the database images into two subsets: feedback database images and test database images. In the first retrieval, we use the feedback database images to obtain samples used for relevance feedback. In the second retrieval, we use the test database images to evaluate retrieval accuracy. That is, the feedback database images are $\mathcal{X}_1$, and the test database images are $\mathcal{X}_2$. Here, $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$.

After splitting, we construct $\mathcal{V}_1$ and $\mathcal{V}_2$ as described in Sec. 3.1. To compare the retrieval performance of ψ and $\tilde{\psi}$, we execute retrieval for $\mathcal{V}_2$ with ψ.

$$\mathcal{W}_2' = \psi(\mathbf{u}, K, \mathcal{V}_2). \tag{8}$$

$\mathcal{W}_2'$ means samples returned from $\mathcal{V}_2$ with the simple K-NN. Finally, we compare $\mathcal{W}_2$ and $\mathcal{W}_2'$ and measure whether the returned samples are refined.

When we execute the splitting, we set the size of the feedback and test database to the same. Also, we adopt a stratified sampling approach, where the ratios of the labels in each subset are equal. We split the dataset in this way because the test and control method is unbiased only if the two database datasets have equivalent numbers and distributions of samples, as Hull [13] says.

4 Experiment

Although our retrieval system requires users' feedback, we automatically generate pseudo-feedback for evaluation purposes. We use labeled datasets in each task to automatically generate a user's feedback for each returned sample. In this work, we assume that a user could provide binary feedback correctly. That is, we generate positive feedback for positive samples and negative feedback for negative ones.

In our experiment, we consider evaluating accuracy of our retrieval algorithm with relevance feedback. First, we consider category-based image retrieval (Sec. 4.1), where a user prefers samples with the same label as the query image. Next, we consider two retrieval settings which considers diverse user preferences: one-label-based image retrieval (Sec. 4.2) and conditioned image retrieval (Sec. 4.3). We devise the two retrieval tasks from the same motivation as the GeneCIS benchmark [32], where users have various intentions.

Throughout all our experiments, we split the evaluation dataset into three groups: query images, feedback database images $\mathcal{X}_1$, and test database images $\mathcal{X}_2$. In all experimental settings, the ratio of these subset sizes is $1:2:2$. We conduct retrieval tasks on ten different splittings and calculate the average and standard deviation of Recall@K. We execute all experiments in a single Tesla V100 GPU. Images are resized to 256×256 and then cropped to 224×224 at the center to input them into the model. We set $\hat{K} = |\mathcal{V}_2|$. Furthermore, if we could not obtain any returned samples' candidates $\tilde{\mathcal{W}}$ for one query, we view this trial as a failure and calculate Recall@K as 0.

4.1 Category-based Image Retrieval

Category-based image retrieval is the most common retrieval task. We calculate Recall@K based on whether each returned sample's label is identical to the query's. After the first retrieval, we generate binary feedback for each returned sample based on whether its label is the same as the query's. We use two datasets: CUB-200-2011 [36] and Cars-196 [17]. Each sample in the two datasets has a single label.

We compare Recall@K of our retrieval system with metric-learning-based image retrieval. We choose triplet loss [35] as simple metric learning and HIST [18] as state-of-the-art metric learning. As a metric-learning-specific procedure, we follow the common practice of metric learning [23] and split the whole labels into two groups in advance: the training labels and the evaluation labels. Image encoders for metric learning methods are trained with images of the training labels.

Table 1: Recall@K in category-based image retrieval. We use ResNet-50 as an encoder architecture. We set M to 50 here. Note that Recall@K of HIST [18] differs from that reported in the original paper because the experimental condition differs.

Dataset	Method	Feedback	Training	$K = 1$	$K = 2$	$K = 4$	$K = 8$
CUB-200-2011	CLIP			46.2±1.1	59.9±0.8	73.3±1.1	84.4±0.9
	Triplet		✓	61.7±0.6	73.0±0.3	82.7±0.4	90.1±0.3
	HIST		✓	**67.9**±0.9	78.4±0.8	86.4±0.8	91.9±0.4
	Ours	✓		64.2±1.1	**78.6**±1.1	**88.5**±0.7	**93.8**±0.4
Cars-196	CLIP			69.5±1.0	80.8±1.0	89.1±0.8	94.6±0.6
	Triplet		✓	83.4±0.8	90.0±0.9	94.2±1.0	96.7±1.1
	HIST		✓	85.2±0.9	91.0±0.6	94.6±0.4	97.0±0.2
	Ours	✓		**86.4**±0.6	**94.7**±0.5	**98.0**±0.5	**99.0**±0.3

That is all for the metric-learning-specific procedure, and we handle only the remaining images (with the evaluation labels) as an evaluation dataset below. We execute the dataset splitting and obtain query images, feedback database images $\mathcal{X}_1$, and test database images $\mathcal{X}_2$. We use the same database images and query images to evaluate each retrieval system. When calculating Recall@K of metric-learning-based image retrieval, we encode test database images $\mathcal{X}_2$ with the trained encoder and construct test database features. Next, we encode each query image q into a D-dimensional feature with the same encoder, execute K-NN retrieval, and calculate Recall@K.

We use ResNet-50 as a backbone of all image encoders to compare Recall@K equally. To obtain a visual encoder trained with triplet loss [35], we execute training ourselves following the implementation of prior work [15]. For HIST [18], the authors publicize their pre-trained models to each dataset, so we use them in our experiment. We fix the feedback size M to 50.

Result Tab. 1 shows the results. Our retrieval system utilizes relevance feedback and achieves competitive results with state-of-the-art metric learning methods, despite not training image encoders specifically for each dataset. Recall@1 of HIST is higher than that of our retrieval system in both datasets, but our retrieval system surpasses HIST in Recall@$2, 4, 8$ in both datasets.

These results mean that simple user feedback for each query is enough to enable the CLIP image encoder to achieve competitive retrieval accuracy with metric learning methods. Although the feedback size $M = 50$ is large for practical applications, this experiment reveals that we can build accurate retrieval systems by combining CLIP and realistic feedback size.

4.2 One-label-based Image Retrieval

One-label-based image retrieval is an experimental setting simulating a user focusing on one of the query image's characteristics. Each user expects that the returned samples have the same characteristics on which the user focuses. For example, consider a user providing a query image of a boy with a hat standing before a door. The user hopes to obtain a hat image, not paying attention to other characteristics such as a door and a boy. In this case, the user provides positive feedback to images of a hat, and our retrieval aims to adapt to the user's preference and return images of a hat. Metric learning cannot handle this task because it does not take each user's preference as input.

To simulate this setting, we use three datasets in which each image has multiple labels: MIT-States [14], Fashion200k [10], and COCO 2017 Panoptic Segmentation [16, 19]. For each query image, we focus on one of the query image's labels. We view the label as the characteristic each user focuses on, and regard samples with the focused label as positive when generating user feedback and calculating Recall@K. For example, when a query image has labels of "boy," "hat," and "door," and we focus on the label "hat," we simulate a user who provides the query image aiming to obtain images of a hat. In this case, we generate positive feedback to samples with the label "hat" and regard such samples as positive when calculating Recall@K.

For each query image q that has L labels $\{l_i\}_{i=1}^{L}$, we execute trials L times. In i^{th} retrieval, we focus on the label l_i, generate user feedback, and calculate Recall@K as explained above.

We use test images in MIT-States [14] and Fashion200k [10] and evaluation data of COCO 2017 Panoptic Segmentation [16, 19]. We view each dataset as an evaluation dataset and execute the dataset splitting. We regard attributes and nouns for each image of MIT-States and Fashion200k as labels, and annotated objects of each COCO 2017 Panoptic Segmentation image as labels. We fix the feedback size M to 50. As backbones of CLIP image encoders, we choose ViT-B/32 and ResNet-50.

Result Tab. 2 shows the results. Our retrieval system successfully improves Recall@K with relevance feedback in all datasets and encoder architectures by a significant margin. With the relevance feedback, Recall@1 improves by up to 9.5%. These results mean that our retrieval system successfully adapts to each user's preference in our simulation experiment. Fig. 3 illustrates one example of our retrieval system's trial with CLIP ViT-B/32 in COCO dataset. In this case, we focus on a label "banana," simulating a user who provides the query image in Fig. 3 searching for banana images. With relevance feedback, our retrieval system can return images with the label "banana" more accurately.

Regarding the encoder architecture, ViT-B/32 surpasses ResNet-50 in Recall@K. In terms of Recall@1, ViT-B/32 ourperforms ResNet-50 by 1.8% in Fashion200k, and 2.1% in MIT-States. These results suggest that CLIP image encoders with large architectures work better in one-label-based image retrieval settings.

4.3 Conditioned Image Retrieval

Conditioned image retrieval is an experimental setting simulating a user who searches for similar images to a query image but differs from it in some aspects.

Table 2: Recall@K of one-label-based image retrieval with CLIP visual encoder.

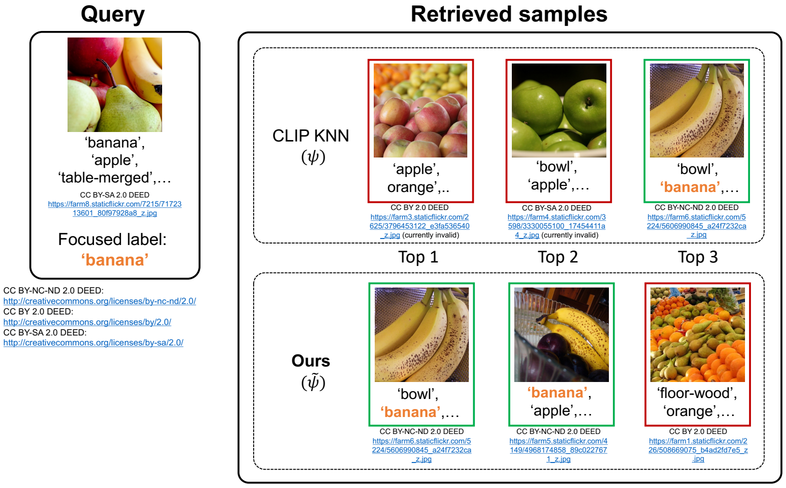
Dataset	Arch	Feedback	$K = 1$	$K = 2$	$K = 4$	$K = 8$
Fashion200k	ViT-B/32		67.9±0.8	75.8±0.8	82.4±0.8	87.5±0.8
		✓	**76.1**±0.6	**83.5**±0.5	**87.8**±0.4	**90.1**±0.5
	R50		66.1±0.6	74.5±0.8	81.2±0.6	86.6±0.6
		✓	**74.3**±0.9	**82.3**±0.7	**87.2**±0.5	**89.6**±0.4
MIT States	ViT-B/32		40.4±0.4	51.8±0.2	62.7±0.2	73.1±0.2
		✓	**49.9**±0.6	**63.5**±0.7	**74.3**±0.5	**81.6**±0.4
	R50		37.9±0.2	49.0±0.2	60.2±0.3	70.7±0.3
		✓	**47.1**±0.4	**60.4**±0.4	**71.7**±0.3	**79.4**±0.3
COCO	ViT-B/32		49.3±0.8	63.1±0.7	75.6±0.6	85.0±0.4
		✓	**58.3**±0.8	**73.7**±0.9	**85.6**±0.6	**92.7**±0.4
	R50		49.7±0.8	63.3±0.5	75.4±0.6	84.8±0.4
		✓	**58.4**±1.0	**73.8**±0.7	**85.5**±0.6	**92.5**±0.5

For example, a user provides a ripe apple image, but they want unripe apple images. In this case, the user provides positive feedback to images of an unripe apple, and our retrieval system aims to adapt to the user's preference and return unripe apple images. Metric learning cannot handle this task because it does not take such users' preferences as input.

To simulate this setting, we follow previous conditioned image retrieval studies [2,32,33] and use MIT-States [14]. Each image in MIT-States has one adjective label and one noun label. We use the same experimental settings as previous conditioned image retrieval studies. For each query image q with the adjective label l_{adj} and noun label l_{noun}, we choose one adjective label l'_{adj} that is different from l_{adj}. We view images that have both l'_{adj} and l_{noun} as those the user prefers, and regard samples with l'_{adj} and l_{noun} as positive when generating and calculating Recall@K. For example, when a query image has $l_{\text{adj}} =$ "ripe" and $l_{\text{noun}} =$ "apple," and we choose $l'_{\text{adj}} =$ "unripe," we simulate a user who provides a ripe apple image aiming to obtain images of an unripe apple. In this case, we automatically generate positive feedback to samples with labels "unripe" and "apple," and regard such samples as positive when calculating Recall@K.

For each q with l_{adj} and l_{noun}, we try all possible adjective labels $\mathcal{L}_{\text{adj}} \backslash \{l_{\text{adj}}\}$. Here, $\mathcal{L}_{\text{adj}}$ is a set of all adjective labels. When we define $\{l^i_{\text{adj}}\}_{i=1}^{N} \coloneqq \mathcal{L}_{\text{adj}} \backslash \{l_{\text{adj}}\}$ ($N = |\mathcal{L}_{\text{adj}} \backslash \{l_{\text{adj}}\}|$), we execute trials N times for q. In i^{th} retrieval, we choose l^i_{adj}, generate binary feedback, and calculate Recall@K as explained above.

Additionally, we compare our retrieval system with multimodal retrieval. We choose GeneCIS [32] as the state-of-the-art multimodal retrieval addressing conditioned image retrieval on MIT-States. GeneCIS consists of visual encoder $\phi'_v : \mathcal{I} \to \mathbb{R}^D$ and multimodal encoder $\phi'_{\text{mm}} : \mathcal{I} \times \mathcal{T} \to \mathbb{R}^D$. Here, $\mathcal{T}$ is a set of all text. We reimplement GeneCIS and calculate Recall@K in the same query images and test database images $\mathcal{X}_2$ as those used to evaluate our retrieval


Query
'banana',
'apple',
'table-merged',…
CC BY-SA 2.0 DEED
https://farm8.staticflickr.com/7215/7172313601_80f97928a8_z.jpg
Focused label:
'banana'
Retrieved samples
CLIP KNN
(ψ)
'apple',
orange',..
CC BY 2.0 DEED
https://farm3.staticflickr.com/2625/3796453122_e3fa536540_z.jpg (currently invalid)
'bowl',
'apple',…
CC BY-SA 2.0 DEED
https://farm4.staticflickr.com/3598/3330055100_17454411a4_z.jpg (currently invalid)
'bowl',
'banana',…
CC BY-NC-ND 2.0 DEED
https://farm6.staticflickr.com/5224/5606990845_a24f7232ca_z.jpg
Top 1
Top 2
Top 3
Ours
($\tilde{\psi}$)
'bowl',
'banana',…
CC BY-NC-ND 2.0 DEED
https://farm6.staticflickr.com/5224/5606990845_a24f7232ca_z.jpg
'banana',
'apple',…
CC BY-NC-ND 2.0 DEED
https://farm5.staticflickr.com/4149/4968174858_89c0227671_z.jpg
'floor-wood',
'orange',…
CC BY 2.0 DEED
https://farm1.staticflickr.com/226/508669075_b4ad2fd7e5_z.jpg

CC BY-NC-ND 2.0 DEED:
http://creativecommons.org/licenses/by-nc-nd/2.0/
CC BY 2.0 DEED:
http://creativecommons.org/licenses/by/2.0/
CC BY-SA 2.0 DEED:
http://creativecommons.org/licenses/by-sa/2.0/

Fig. 3: An example of one-label-based image retrieval in COCO.

Table 3: Recall@K of conditioned image retrieval for MIT States. Note that Recall@K of GeneCIS [32] differs from that reported in the original paper because the experimental condition differs.

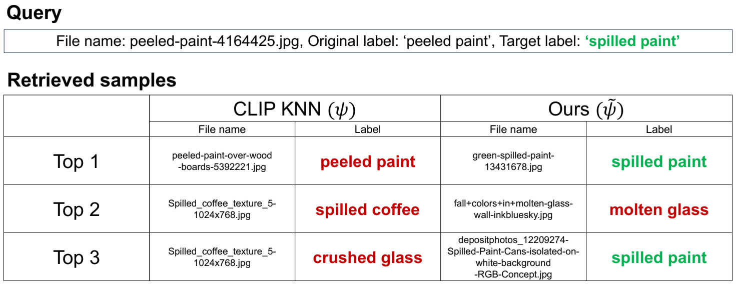
Arch	Method	Feedback	$K=1$	$K=2$	$K=4$	$K=8$
ViT-B/16	CLIP	-	4.28±0.14	7.99±0.23	14.1±0.3	23.4±0.3
	GeneCIS	Text	**17.3**±0.6	**26.7**±0.7	**38.0**±0.7	**50.7**±0.7
	Ours	Binary	**17.3**±0.9	**26.7**±1.1	36.1±1.0	42.7±0.9
R50x4	CLIP	-	4.16±0.08	7.72±1.2	13.7±0.2	22.9±0.2
	GeneCIS	Text	15.7±1.0	24.2±1.9	34.8±1.0	**47.2**±0.8
	Ours	Binary	**16.4**±0.8	**25.8**±1.0	**35.0**±0.8	41.6±0.8

system. We obtain test database features as $\mathcal{V}'_2 \coloneqq \{\phi'_v(x) \mid x \in \mathcal{X}_2\}$. Consider the query image q has l_{adj} and l_{noun}, and we choose $l'_{\mathrm{adj}} (\neq l_{\mathrm{adj}})$. We generate the query feature $\mathbf{u}_{\mathrm{mm}} \coloneqq \phi'_{\mathrm{mm}}(q, l'_{\mathrm{adj}})$. Note that each adjective label is represented as text: $l'_{\mathrm{adj}} \in \mathcal{T}$. We execute retrieval and return samples $\mathcal{W}_{\mathrm{mm}}$ as follows:

$$\mathcal{W}_{\mathrm{mm}} = \psi(\mathbf{u}_{\mathrm{mm}}, K, \mathcal{V}'_2). \tag{9}$$

We view test images of MIT-States as an evaluation dataset and execute the dataset splitting. We fix M to 50. We choose ResNet-50x4 and ViT-B/16 as CLIP visual encoder backbones. The authors publicize the pre-trained models of GeneCIS [32], and we use them for comparison.

Result Tab. 3 shows the results. We successfully improve Recall@K with relevance feedback from simple CLIP K-NN retrieval without relevance feedback



Query

File name: peeled-paint-4164425.jpg, Original label: ‘peeled paint’, Target label: **‘spilled paint’**

Retrieved samples

	CLIP KNN (ψ)		Ours ($\tilde{\psi}$)	
	File name	Label	File name	Label
Top 1	peeled-paint-over-wood -boards-5392221.jpg	**peeled paint**	green-spilled-paint-13431678.jpg	**spilled paint**
Top 2	Spilled_coffee_texture_5-1024x768.jpg	**spilled coffee**	fall+colors+in+molten-glass-wall-inkbluesky.jpg	**molten glass**
Top 3	Spilled_coffee_texture_5-1024x768.jpg	**crushed glass**	depositphotos_12209274-Spilled-Paint-Cans-isolated-on-white-background -RGB-Concept.jpg	**spilled paint**

Fig. 4: An example of conditioned image retrieval.

by a large margin. Furthermore, when we provide a realistic amount of relevant feedback ($M = 50$), our retrieval system achieves competitive accuracy with state-of-the-art multimodal retrieval. Recall@1 of ours and GeneCIS are almost equal in both encoder architectures. These results mean that our retrieval system adapts to each user's preference without textual information in this setting. We demonstrate an example trial of our retrieval system (CLIP ViT-B/16) in Fig. 4. In the first case, we choose an adjective label "spilled," simulating a user who provides the peeled paint image searching for spilled paint images. With relevance feedback, our retrieval system can return images with the labels "spilled" and "paint" more accurately.

5 Additional Analysis

5.1 Architectures of CLIP image encoders and the feedback size

In category-based image retrieval settings, we change the architecture of CLIP image encoder and evaluate how our retrieval system improves the retrieval accuracy. We choose ResNet-50, ViT-B/32, and ViT-L/14 as backbone architectures and calculate Recall@K in each setting. Also, we vary the feedback size (M) and assess the relationship between the feedback size and the retrieval accuracy. We vary M among 10, 25, and 50.

Fig. 5 shows the results of comparison among various kinds of CLIP and M. Larger M leads to better Recall@K. We can infer that our retrieval system obtains more information from the larger feedback. Moreover, the larger architecture of CLIP image encoder achieves higher accuracy, implying that the larger CLIP image encoder extracts more appropriate information from images.

In particular, CLIP ViT-L/14 performs exceptionally well for $M = 10$, which is truly a realistic and user-friendly feedback size. Combined with the results described in Sec. 4.1, it is shown that we can achieve a practical interactive image retrieval system with CLIP ViT-L/14 and a smaller feedback size.

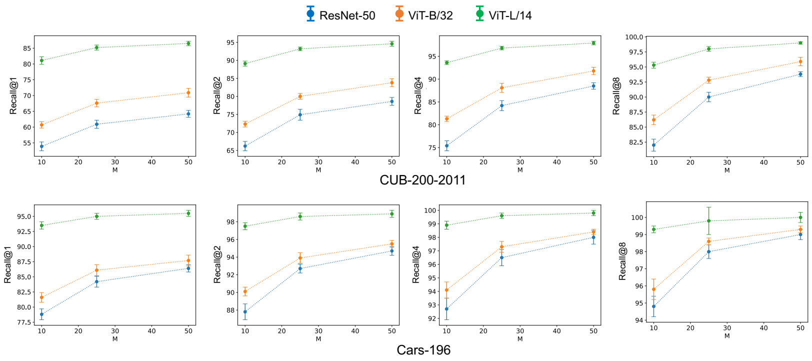


Fig. 5: Comparison among various kinds of CLIP and M.

5.2 Retrieval accuracy and the number of positive feedback

We examine the influence of positive samples in the first retrieval on the retrieval performance in the second retrieval. We choose category-based image retrieval settings, where the architecture of CLIP visual encoder is ViT-B/32 and $M = 50$. We count the positive feedback in the first retrieval, which is represented as $|\{(\mathbf{w}, b) \in \mathcal{F} \mid b = 1\}|$. At the same time, we calculate MAP@R of the returned samples in the second retrieval. MAP@R is calculated as follows [23]:

$$\text{MAP@R} = \frac{1}{R} \sum_{i=1}^{R} P(i). \tag{10}$$

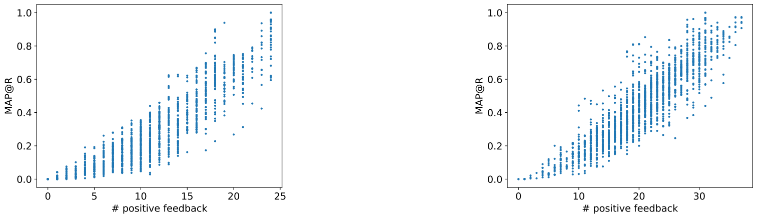
$$P(i) = \begin{cases} \text{precision at } i \text{ (if the } i^{\text{th}} \text{ returned sample is positive)}, \\ 0 \text{ (otherwise)}. \end{cases} \tag{11}$$

Here, R is the total number of positive samples in $\mathcal{X}_2$. We execute the retrieval for all query images and collect pairs of the number of positive feedback in the first retrieval and MAP@R of the second retrieval.

Fig. 6 shows the results for each dataset. As we can see, a positive correlation exists between the number of positive feedback and MAP@R. These results suggest that more positive feedback leads to higher retrieval performance. When we have positive samples in the user feedback, we can accurately predict whether each sample in the second retrieval is preferable.

5.3 Retrieval runtime

We examine the retrieval runtime of our retrieval system. We choose category-based image retrieval settings. We choose ViT-B/32 as the backbone architecture of CLIP image encoder. We measure the runtime in each process: encoding, CLIP K-NN, and the updated retrieval. We can formulate each process as $\phi(q)$,



(a) CUB-200-2011 **(b)** Cars-196

Fig. 6: Relashinship between the number of positive feedback and MAP@R. Each data point represents one query.

Table 4: Average runtime of each process per query (ms).

Dataset	Encoding ($\phi(q)$)	K-NN retrieval ($\psi(\mathbf{u}, K, \mathcal{V}_2)$)	Updated retrieval ($\tilde{\psi}(\mathbf{u}, K, \mathcal{V}_2)$)
CUB200	10.2±0.6	0.0874±0.0006	0.271±0.003
Cars-196	10.3±0.2	0.0875±0.0019	0.272±0.006

$\psi(\mathbf{u}, K, \mathcal{V}_2)$, and $\tilde{\psi}(\mathbf{u}, K, \mathcal{V}_2)$ respectively. We set $\hat{K} = |\mathcal{V}_2|$, the most time consuming settings. We use PyTorch libraries and execute all calculations in GPU memory. We use the same computational resources as Sec. 4.

Tab. 4 shows the results. Our updated retrieval $\tilde{\psi}$ takes only 3 times overhead compared to the simple K-NN retrieval ψ. The encoding takes much longer than the retrieval, so the difference in the retrieval runtime can be ignored in our retrieval system. We discuss the retrieval runtime theoretically in the supplementary material.

5.4 Limitation

One limitation of our work is that we utilize pre-existing datasets to create simplified experimental scenarios that mimic user feedback. However, real-world human users may possess a wider range of subjective preferences that cannot be fully replicated by current methods. To assess interactive image retrieval systems with greater accuracy, a novel dataset that captures the nuances of human users' preferences is imperative.

6 Conclusion

Our study proposed a CLIP-based interactive image retrieval system to overcome the shortcomings of metric learning. Our retrieval system receives binary feedback from each user, updates the retrieval algorithm, and returns images the user prefers. Our retrieval system adapts to any user preference and works well without training an image encoder. This paper revisited relevance feedback and integrated it with CLIP, suggesting a powerful baseline for interactive image retrieval. To the best of our knowledge, we are the first to explore the possibility of retrieval with modern vision-language models and relevance feedback. Our paper will throw a spotlight on relevance feedback again.

References

1. Ahmed, A.: Implementing relevance feedback for content-based medical image retrieval. IEEE Access (2020) 3
2. Anwaar, M.U., Labintcev, E., Kleinsteuber, M.: Compositional learning of image-text query for image retrieval. In: WACV (2021) 3, 10
3. Baldrati, A., Bertini, M., Uricchio, T., Del Bimbo, A.: Conditioned image retrieval for fashion using contrastive learning and CLIP-based features. In: ACM Multimedia Asia (2021) 3
4. Baldrati, A., Bertini, M., Uricchio, T., Del Bimbo, A.: Effective conditioned and composed image retrieval combining clip-based features. In: CVPR (2022) 3
5. Chang, Y., Cirillo, C., Razon, J.: Evaluation of feedback retrieval using modified freezing, residual collection and test and control groups. The SMART retrieval system-experiments in automatic document processing (1971) 6
6. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: A deep quadruplet network for person re-identification. In: CVPR (2017) 3
7. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR (2005) 3
8. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary detection via vision and language knowledge distillation. In: ICLR (2022) 3
9. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR (2006) 3
10. Han, X., Wu, Z., Huang, P.X., Zhang, X., Zhu, M., Li, Y., Zhao, Y., Davis, L.S.: Automatic spatially-aware fashion concept discovery. In: ICCV (2017) 9
11. Harman, D.: Relevance feedback revisited. In: ACM SIGIR (1992) 3
12. Hoffer, E., Ailon, N.: Deep metric learning using triplet network (2014) 1, 3
13. Hull, D.: Using statistical testing in the evaluation of retrieval experiments. In: ACM SIGIR (1993) 3, 6, 7
14. Isola, P., Lim, J.J., Adelson, E.H.: Discovering states and transformations in image collections. In: CVPR (2015) 9, 10
15. Kim, S., Kim, D., Cho, M., Kwak, S.: Proxy anchor loss for deep metric learning. In: CVPR (2020) 3, 8
16. Kirillov, A., Lin, T.Y., Caesar, H., Girshick, R., Dollá'r, P.: Microsoft coco: Panoptic segmentation challenge (2017) 9
17. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: ICCVW (2013) 7
18. Lim, J., Yun, S., Park, S., Choi, J.Y.: Hypergraph-induced semantic tuplet loss for deep metric learning. In: CVPR (2022) 7, 8
19. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 9
20. Meng, Q., Zhao, S., Huang, Z., Zhou, F.: Magface: A universal representation for face recognition and quality assessment. In: CVPR (2021) 3
21. Mokady, R., Hertz, A., Bermano, A.H.: Clipcap: Clip prefix for image captioning. arXiv preprint arXiv:2111.09734 (2021) 3

22. Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S.: No fuss distance metric learning using proxies. In: ICCV (2017) 1, 3
23. Musgrave, K., Belongie, S.J., Lim, S.N.: A metric learning reality check. In: ECCV (2020) 7, 13
24. Nakata, K., Ng, Y., Miyashita, D., Maki, A., Lin, Y.C., Deguchi, J.: Revisiting a knn-based image classification system with high-capacity storage. In: ECCV (2022) 3
25. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021) 2, 3
26. Rui, Y., Huang, T., Ortega, M., Mehrotra, S.: Relevance feedback: a power tool for interactive content-based image retrieval. IEEE TCSVT (1998) 3
27. Sain, A., Bhunia, A.K., Chowdhury, P.N., Koley, S., Xiang, T., Song, Y.Z.: CLIP for All Things Zero-Shot Sketch-Based Image Retrieval, Fine-Grained or Not. In: CVPR (2023) 3
28. Saito, K., Sohn, K., Zhang, X., Li, C.L., Lee, C.Y., Saenko, K., Pfister, T.: Pic2word: Mapping pictures to words for zero-shot composed image retrieval. CVPR (2023) 3
29. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR (2015) 3
30. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR (2018) 3
31. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: A good embedding is all you need? In: ECCV (2020) 3
32. Vaze, S., Carion, N., Misra, I.: Genecis: A benchmark for general conditional image similarity. In: CVPR (2023) 3, 7, 10, 11
33. Vo, N., Jiang, L., Sun, C., Murphy, K., Li, L.J., Fei-Fei, L., Hays, J.: Composing text and image for image retrieval - an empirical odyssey. In: CVPR (2019) 3, 10
34. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: CVPR (2019) 1
35. Weinberger, K.Q., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: NIPS (2005) 3, 7, 8
36. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200 (2010) 7
37. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV (2016) 3
38. Wertheimer, D., Tang, L., Hariharan, B.: Few-shot classification with feature map reconstruction networks. In: CVPR (2021) 3
39. Weyand, T., Araujo, A., Cao, B., Sim, J.: Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In: CVPR (2020) 3
40. Wu, H., Lu, H., Ma, S.: Willhunter: interactive image retrieval with multilevel relevance. In: ICPR (2004) 3
41. Xiao, T., Li, S., Wang, B., Lin, L., Wang, X.: Joint detection and identification feature learning for person search. In: CVPR (2017) 3

Appendix

A Implementation Details

To increase the reproducibility of our experiments and make them more understandable, we provide additional information and explanations about our experimental conditions that we omit due to space limitations.

A.1 Category-based image retrieval

When training an encoder with metric learning, we follow the common practice of metric learning and treat the former half of the labels as the training labels and the latter half as the evaluation labels. Fig. A illustrates the dataset usage in our category-based image retrieval experiment in the case of CUB-200-2011. CUB-200-2011 has 200 labels in total. Therefore, we use the first 100 labels for training and the second 100 labels for evaluation. We treat images of the second 100 labels as an evaluation dataset, and execute the dataset splitting to obtain three subsets: query images, feedback database images, and test database images.

Since feedback database images are reserved for applying relevance feedback as in our proposed approach, when evaluating metric-learning-based image retrieval approaches, only query images and test database images have been used.

A.2 One-label-based image retrieval

For MIT-States and Fashion200k datasets, some images have been excluded in our evaluation because their captions, i.e., the set of labels, are not qualified. The details are as below.

First, we exclude rare captions that are not shared by enough images. Specifically, if a caption occurs less than 5 times in the evaluation dataset, where 5 is the minimal number of samples for doing a 1:2:2 splitting introduced in Sec. 4, the few images with that caption are not included in our experiments. For example, there are only two images are with the "molten orange" caption in XXX dataset, so the two images are excluded in our experiments.

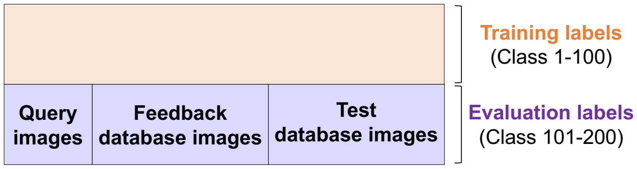


Fig. A: Dataset usage in case of CUB-200-2011.

Table A: Dataset details of category-based image retrieval

Dataset	# labels	# images
CUB-200-2011	100	5924
Cars-196	98	8131

Table B: Dataset details of one-label-based image retrieval

Dataset	# labels	# images	types of labels	# queries
Fashion200k	1258	10609	adjective, noun	9722±5
MIT-States	146	10443	adjective, noun	4178±0
COCO	133	5000	noun	6967±196

Second, when using MIT-States, we exclude samples with the adjective label "adj," which has no meaning as an adjective. This is the same implementation as the prior conditioned image retrieval studies.

A.3 Conditioned image retrieval

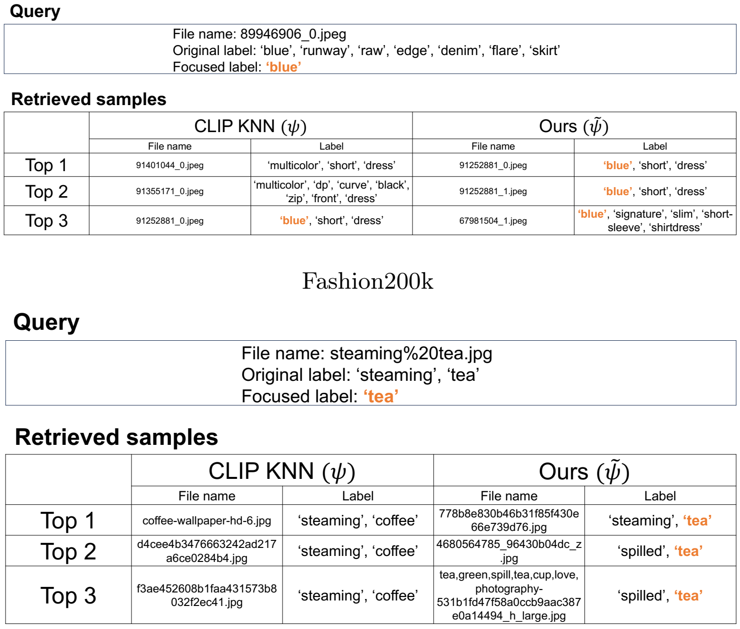
To have MIT-States dataset for conditioned image retrieval evaluation, we first remove the images with unqualified captions as described in Sec. A.2. Additionally, when a query image has a noun label l_{noun} and an adjective label l_{adj}, we choose a different adjective label l'_{adj} so that there is at least one sample that has l'_{adj} and l_{noun}.

B Dataset Details

We describe the details of the evaluation datasets that we use in our experiments in Tabs. A to C. We also attach information about the number of queries in Tabs. B and C. We change the splitting way ten times and calculate the average and standard deviation of the number of queries because the number of queries depends on query images' labels.

Table C: Dataset details of conditioned image retrieval

Dataset	# labels	# images	types of labels	# queries
MIT-States	146	10443	adjective, noun	15368±2



Query

File name: 89946906_0.jpeg
Original label: ‘blue’, ‘runway’, ‘raw’, ‘edge’, ‘denim’, ‘flare’, ‘skirt’
Focused label: **‘blue’**

Retrieved samples

	CLIP KNN (ψ)		Ours ($\tilde{\psi}$)	
	File name	Label	File name	Label
Top 1	91401044_0.jpeg	‘multicolor’, ‘short’, ‘dress’	91252881_0.jpeg	**‘blue’**, ‘short’, ‘dress’
Top 2	91355171_0.jpeg	‘multicolor’, ‘dp’, ‘curve’, ‘black’, ‘zip’, ‘front’, ‘dress’	91252881_1.jpeg	**‘blue’**, ‘short’, ‘dress’
Top 3	91252881_0.jpeg	**‘blue’**, ‘short’, ‘dress’	67981504_1.jpeg	**‘blue’**, ‘signature’, ‘slim’, ‘short-sleeve’, ‘shirtdress’

Fashion200k

Query

File name: steaming%20tea.jpg
Original label: ‘steaming’, ‘tea’
Focused label: **‘tea’**

Retrieved samples

	CLIP KNN (ψ)		Ours ($\tilde{\psi}$)	
	File name	Label	File name	Label
Top 1	coffee-wallpaper-hd-6.jpg	‘steaming’, ‘coffee’	778b8e830b46b31f85f430e66e739d76.jpg	‘steaming’, **‘tea’**
Top 2	d4cee4b3476663242ad217a6ce0284b4.jpg	‘steaming’, ‘coffee’	4680564785_96430b04dc_z.jpg	‘spilled’, **‘tea’**
Top 3	f3ae452608b1faa431573b8032f2ec41.jpg	‘steaming’, ‘coffee’	tea,green,spill,tea,cup,love,photography-531b1fd47f58a0ccb9aac387e0a14494_h_large.jpg	‘spilled’, **‘tea’**

MIT-States

Fig. B: Examples of one-label-based image retrieval.

C Examples of one-label-based image retrieval

We provide additional examples of one-label-based image retrieval in Fig. B. In the case of Fashion200k, our retrieval system successfully retrieves images with “blue” labels more accurately than image retrieval without relevance feedback.

D Theoretical and Additional Runtime Analysis

We provide a theoretical explanation of the retrieval runtime difference between K-NN (ψ) and the updated retrieval ($\tilde{\psi}$). As in Eq. (8), simple K-NN calculates cosine similarities $|\mathcal{V}_2|$ times. In contrast, the updated retrieval executes cosine similarities $\hat{K}M$ times in Eq. (6). Along with the operation of Eq. (5), the updated retrieval calculates cosine similarities $\hat{K}M + |\mathcal{V}_2|$ times. Therefore, the updated retrieval is $1 + \frac{\hat{K}M}{|\mathcal{V}_2|}$ times slower than the simple K-NN. In the experiment of Sec. 5.3, we set $M = 50$ and $\hat{K} = |\mathcal{V}_2|$, so the updated retrieval runtime could be 51 times longer than the simple K-NN retrieval runtime. In our experiment, we execute retrieval in GPU memory. GPUs are good at simple parallel computation, and since the number of data we are dealing with in this paper can be fitted into GPU memory, there is not much difference in the retrieval runtime.

Table D: Comparison of Recall@1 and retrieval runtime (ms) among various $\hat{K}$. $|\mathcal{V}_2|$ is around 2000 in CUB-200-2011, and around 3000 in Cars-196.

Dataset	Method	$\hat{K}$	Retrieval runtime (ms)	Recall@1
CUB-200-2011	ψ	-	0.0874±0.0006	51.9±1.0
	$\tilde{\psi}$	30	0.232±0.006	69.3±1.6
		100	0.253±0.005	70.8±1.4
		$\|\mathcal{V}_2\|$	0.272±0.006	70.9±1.4
Cars-196	ψ	-	0.0875±0.0019	72.9±0.9
	$\tilde{\psi}$	30	0.255±0.004	87.2±1.0
		100	0.266±0.005	87.7±0.9
		$\|\mathcal{V}_2\|$	0.272±0.006	87.7±0.9

To explore retrieval runtime further, we try smaller $\hat{K}$ and observe retrieval accuracy and speed. We set $\hat{K} = 30, 100, |\mathcal{V}_2|$ and compare retrieval runtime and Recall@1. Tab. D shows the results. Larger $\hat{K}$ leads to longer retrieval runtime, this can be expected from our theoretical explanation. Additionally, setting smaller $\hat{K}$ slightly lower Recall@1.

E Future Works

This paper only focuses on one-time binary relevance feedback. We will achieve more complex interactive image retrieval tasks by exploring multiple-level and multi-time relevance feedback and combining them with modern deep learning.

We also need to evaluate whether the returned samples match the user's preferences by having actual users provide feedback on the returned samples. In this study, we automatically generated feedback using labels in the dataset, assuming that users could correctly provide binary feedback. However, users may provide incorrect feedback with a certain probability. Experiments with users are necessary to evaluate whether our retrieval system can handle such cases.