applied sciences

MDPI

Article

Revisiting Low-Resolution Images Retrieval with Attention Mechanism and Contrastive Learning

Thanh-Vu Dang †, Gwang-Hyun Yu † and Jin-Young Kim *

Department of ICT Convergence System Engineering, Chonnam National University, Gwangju 61186, Korea; 197796@jnu.ac.kr (T.-V.D.); 188799@jun.ac.kr (G.-H.Y.)

* Correspondence: beyondi@jnu.ac.kr

† These authors contributed equally to this work.

Abstract: Recent empirical works reveal that visual representation learned by deep neural networks can be successfully used as descriptors for image retrieval. A common technique is to leverage pre-trained models to learn visual descriptors by ranking losses and fine-tuning with labeled data. However, retrieval systems' performance significantly decreases when querying images of lower resolution than the training images. This study considered a contrastive learning framework fine-tuned on features extracted from a pre-trained neural network encoder equipped with an attention mechanism to address the image retrieval task for low-resolution image retrieval. Our method is simple yet effective since the contrastive learning framework drives similar samples close to each other in feature space by manipulating variants of their augmentations. To benchmark the proposed framework, we conducted quantitative and qualitative analyses of CARS196 (mAP = 0.8804), CUB200-2011 (mAP = 0.9379), and Stanford Online Products datasets (mAP = 0.9141) and analyzed their performances.

Keywords: image retrieval; representation learning; contrastive learning; self-supervised learning

Citation: Dang, T.-V.; Yu, G.-H.; Kim, J.-Y. Revisiting Low-Resolution Images Retrieval with Attention Mechanism and Contrastive Learning. *Appl. Sci.* **2021**, *11*, 6783. https://doi.org/10.3390/app11156783

Academic Editor: Joonki Paik

Received: 7 June 2021
Accepted: 20 July 2021
Published: 23 July 2021

1. Introduction

Even though high-quality images are popular, we get often degraded or low-resolution images due to careless photo taking of loose focusing or picture taking from afar. Thus image retrieving from low-quality images is requested in real applications because retrieval results for a degraded image are usually poor. In this paper, we propose a proper deep-learning-based framework that can straightly deal with low-resolution images. This study implemented three main modules for solving category image retrieval on low-resolution samples. First, an attention-based encoder network was employed to extract meaningful visual representations of images. Second, we manipulated a contrastive learning framework to obtain embeddings that are used for information retrieval. The purpose of contrastive learning is to find consistent representations of different resolution views augmented from the same source. Third, a model was trained end-to-end, including its encoder network and projection head with multiple loss functions, consisting of contrastive loss for maximizing agreement of different resolution versions of an identical image, cross-entropy loss for classification, and triplet loss for maximizing distance of negative pairs (different category) and minimizing the distance of positive pairs (same category). This section presents a brief review of previous studies about image retrieval, attention mechanism, and contrastive learning. To show the importance of this study, we examined some failure cases when using low-resolution images as queries for image retrieval, as shown in Figure 1. These examples highlight that low-resolution images are inferior for feature matching despite using a powerful pre-trained model.

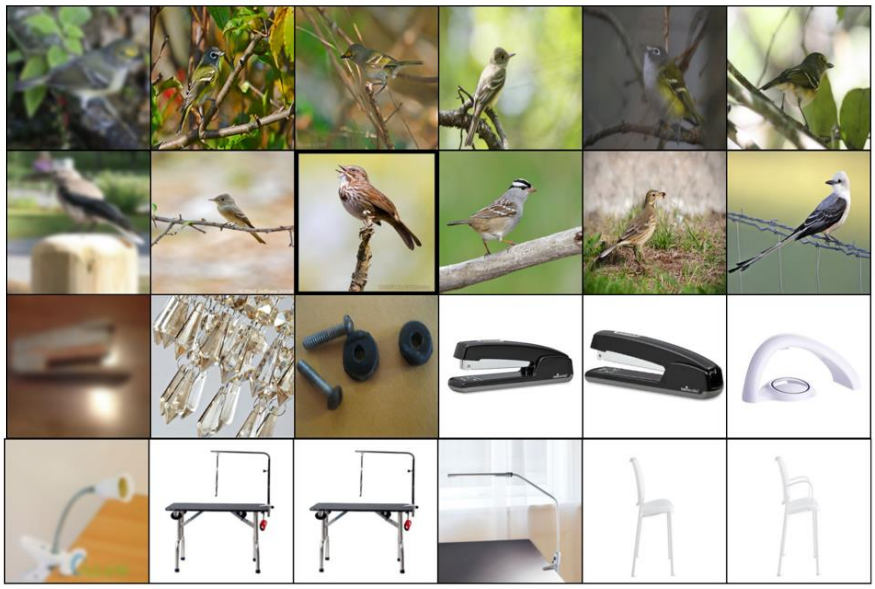

Figure 1. Failure cases when querying a retrieval system using low-resolution images. The base model is EfficientNet-b7 with features extracted from the last convolutional layer. The first image of each round is the query, and the next five images are the top 5 retrieval results, with labels of some images different from the label of the query image. Images in the first two rows belong to the CUB200-2011 dataset, while those of the last two rows belong to the Stanford Online Product dataset. First row: 1st image is query image from class "mourning warbler ", 2nd image from class "magnolia warbler", 3rd image from class "yellow-throated vireo", 4th image from class "white-crowned sparrow", 5th image from class "blue-headed vireo", 6th image from class "yellow-throated vireo". Second row: 1st image is query image from class "Clark nutcracker", 2nd image from class "Western Kingbird", 3rd image from class "song sparrow", 4th image from class "American pipits", 5th image from class "blue-headed vireo", 6th image from class "scissor-tailed flycatcher". Third row: 1st image is query image from class "stapler ", 2nd image from class "lamp", 3rd image from class "cabinet", 4th and 5th images from class "stapler", 6th image from class "kettle". Fourth row: 1st image is query image from class "lamp ", 2nd and 3rd image from class "cabinet", 4th image from class "lamp", 5th and 6th images from class "chair".

Image Retrieval: Image descriptors based on deep convolutional neural networks (CNNs) have been used as primary descriptors in various computer vision tasks, such as classification, semantic segmentation, and especially on image search [1,2]. Studies on image retrieval have progressively developed various methods to compress spatial feature maps into a vector-form descriptor. There are growing appeals for constituting image descriptors. Outputs from the final fully connected layers [1,3], the most activated convolutions [2], and generalized pooling of convolutions [4] have been utilized as image descriptors. Each descriptor has different functions, such as concentrating on informative regions or large receptive field regions. Consequently, modern methods have come up with ensemble techniques to boost the desired systems' performance. Early fusion blends descriptors across layers and trains an integrated model based on an end-to-end approach [3,5–7], while late fusion is a method in which individual models and features from multiple learners are entangled to form a compact global descriptor [8]. Recently, one of the major topics that show the growth in interest from numerous researchers in this field is attention mechanism.

Attention Mechanism: One of the current trends in designing neural network architecture is the attention mechanism [9–11]. The attention model integrates the concept of relevance by focusing only on the relevant aspects of a given input, which is useful for achieving a compelling performance of the task. These systems only focus on a relevant part of input useful for getting the required knowledge for working on a task and ignoring

irrelevant details. Approaches tackling the image retrieval problem include using attention structures introduced in [12–15].

This study employed a simple contrastive learning framework [16] for image retrieval but investigated an effective architecture for the feature extractor. As a result, we found that the Visual Transformer [11] model is especially effective for visual representation learning. Inspired by Transformer architecture [10] from the natural language processing domain, ViT is proposed as a promising model that naturally integrates a self-attention mechanism to solve computer vision problems. Self-attention is introduced to visual tasks to interpret the correlation among pixels where a high attention score between two visual patches indicates their strong relation and vice versa. Although ViT is not the first method to implement self-attention for the visual task [17–19], it is remarkable thanks to its overwhelming results and efficiency in hardware accelerators, in addition to its simplified implementation. When pre-trained using large-scale datasets and transferred to multiple recognition benchmarks, ViT outperforms state-of-the-art convolution-based neural networks [11,20,21]. However, Transformers lack some inductive biases compared to CNNs, such as translation equivariance, and thus training on sufficient amounts of data is recommended. Otherwise, the self-attention mechanism inherits receptive field properties from CNNs and considers wide regions even from low layers.

Contrastive learning: Learning visual representation is mostly a label-driven task, where learnable feature extractors are trained to optimize objective functions that involve the label of samples, such as categories and pairs of negative and positive samples. The success of such tasks requires large amounts of labeled data [22–24], which is not always available and is often very expensive to acquire. However, unsupervised visual representation learning remains an unexploited area in computer vision research. Recently, a considerable research effort has been put into methods to enhance vision systems without providing a large amount of full supervision. In particular, this effort is characterized by advances in self-supervised learning with a contrastive loss function [25–27]. Self-supervised learning frameworks formulate pretext learning tasks that leverage unlabeled data to learn high-level semantic visual representations useful for the downstream task of interest. For example, pretext tasks such as predicting orientation of rotated image [28], filling in a missing patch [29], or jigsaw re-ordering [30] are beneficial for downstream tasks such as recognition and semantic segmentation because high-level concepts of objects (e.g., shape and texture) are encoded when solving the pretext task. Precisely, self-supervised representation learning's underlying concept is maximization of the mutual information between different views of the data [31–33].

The main theme of contrastive learning is an instance discrimination task. An image and its augmentation are taken to be in the same class (positives), and all other images are considered to be of different classes (negatives). Noticeably, contrastive loss objective function and aggressive data augmentation are the other two key factors influencing the success of self-supervised representation learning [16,32]. In addition, Ref. [34] showed that contrastive loss encourages consistent representation of augmented view and matches prior distribution. There are two main practical advantages of contrastive learning. First, the agreement is estimated between only the learned representations of various views, which lie on a lower dimensional space than the original one. Second, various views can be chosen to capture different aspects and modalities of the data with plenty of modeling flexibility [35,36]. These properties can be especially beneficial for feature matching that helps to retrieve information in retrieval tasks.

Contribution: Our research aimed to find a solution for the challenging problem of category image retrieval tasks on low-resolution images. The problem can be hypothesized into a general question about learning effective visual representations in an embedding space where similar images with different resolutions are kept close to each other and dissimilar ones are placed far away from each other. In this paper, we present a framework that consists of contrastive learning trained over the Visual Transformer encoder (ViT). The benefits of using contrastive learning are expected to maximize positive pairs' agreement,

which are samples from the same class or augmenting resolution samples from the same source, via contrastive loss. However, it is usually a huge challenge when training a contrastive framework from scratch since it requires large-batch training for a long period [16]. Therefore, a possible solution to the problem at hand is proposed in this paper. We used a powerful pre-trained encoder to extract visual representations and fine-tune contrastive learning to learn embeddings for feature matching. Our contribution can be summarized as follows:

1. We adapted the Visual Transformer to the image retrieval task when the embedded vectors were calculated using attention weights. The main advantage of this method is that the attention mechanism of the ViT model helps to focus more on an object of interest when comparing two images.
2. We addressed the problem of retrieval with degraded samples such as low-resolution. We proposed using a contrastive learning framework to learn an embedded space where the same samples are close together with respect to Euclidean distance.
3. We conducted extensive experiments on CARS196, Stanford Online Products, and CUB200-2011 datasets under various circumstances. Both quantitative and qualitative results show that the proposed framework is efficient.

2. Materials and Methods

This study introduces an effective yet simple framework for image retrieval. The feature descriptor is extracted from a backbone network and goes further through a projection module to become embedded vectors for retrieval. We study the behavior of the output representation space when training with a contrastive loss, in particular, how augmenting impacts the space properties and the performance of image retrieval on low-resolution inputs. Our framework is illustrated in Figure 2. This section demonstrates our framework with three modules. A feature extractor module extracts a visual representation of a given image, while a projection head trained in a contrastive approach helps to map visual representation to an embedding space so that the similarity of samples can be calculated. Finally, we introduce an auxiliary module with classification loss and triplet loss, which significantly enhances the category retrieval's performance.

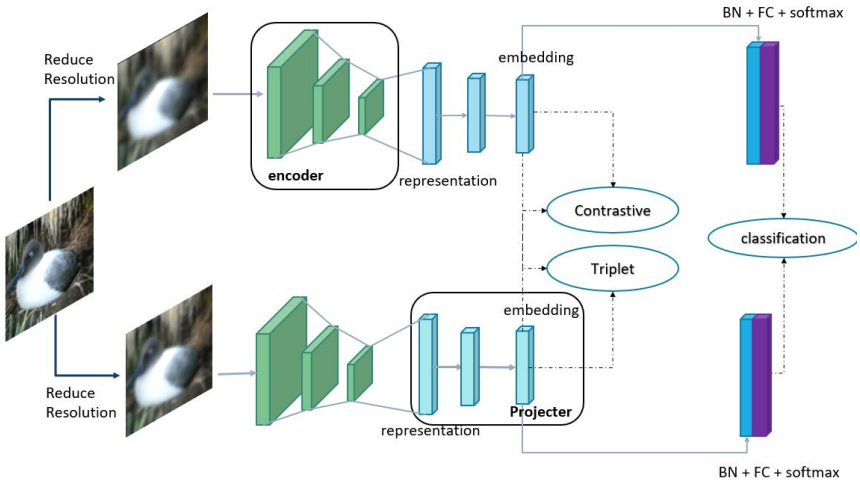

Figure 2. The overall architecture of our proposed framework. The framework is described using five major computing steps. (1) An input image is augmented into two views with different resolutions for contrastive learning. (2) Two augmented images go through a shared encoder module to extract representation vectors. (3) A projection head is added on top of the encoder model to perform nonlinear mapping into an embedding space so that (4) contrastive and triplet loss can be established to maximize the similarity of positive pairs and dissimilarity of negative pairs. (5) We add a linear layer with softmax activation followed by batch normalization to calculate all samples' class probability from two branches.

2.1. Feature Extractor

We manipulate the Visual Transformer [11] model to extract a visual representation of given images. The main component of the ViT model is the self-attention encoder module, which implements Transformer architecture [10] in the most standard way. According to the original version of ViT, our feature extractor involves three main steps.

Patch embedding: We split an image into a sequence of patches and map each patch to a D dimensions embedding space. Precisely, we put an image $x \in \mathbb{R}^{C \times W \times H}$ through D 2d convolutions with the kernel size of $P \times P$ and stride of P, resulting in a feature map with the size $D \times N \times N$, then flattening the feature map into a sequence of N^2 latent vectors with a constant size of D. In the above configuration, $N^2 = H \times W/P^2$ is the number of embedded patches, where $(H,\ W)$ represents the resolution of the original image, and $(P,\ P)$ is the resolution of an image patch. Apart from the embedded patches, the ViT model adds an extra learnable class embedding for classification tasks, and the results obtained from using this class token are referred to as "ViT-class" in our study. To maintain the position of the patches after flattening, we follow the standard way by adding a learnable 1D positional embedding into each patch, and this positional embedding does not share weights across patches.

Encoder: Encoder is a computational block consisting of a multi-head attention module [10] and an MLP with two consecutive linear layers. Input and output of encoder module are both embedded vectors of batches. LayerNorm is applied before feeding embedded vectors into the attention module and MLP. The multi-head attention module expands the model's ability to jointly focus on different positions, thus providing different representation subspaces of pair $(key\ \boldsymbol{K},\ query\ \boldsymbol{Q}, value\ \boldsymbol{V})$ from different attention heads:

$$multihead = Concat(head_1,\ \ldots, head_h), \tag{1}$$

where each head is a context vector from scale dot-product attention.

$$head_i = softmax\left(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^T}{\sqrt{d}}\right)\boldsymbol{V}_i, \tag{2}$$

$\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent a query, key, and value, respectively, calculated inside the transformer architecture that encodes information from the image's patches with the self-attention mechanism to mutually attend to each other. d is the dimensions of patch embeddings; Equation (2) employs d to scale the attention scores. The encoder module is illustrated in Figure 3.

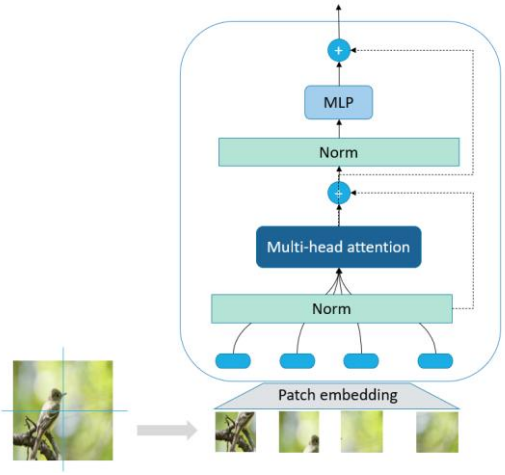

Figure 3. Transformer encoder: the module is combined with three key processes: normalization, multi-head attention, and non-linear mapping (MLP).

Visual descriptor: We develop a novel visual descriptor for the corresponding image based on the attention mechanism. Our technique naturally arises from attention maps of the ViT model. We generate visual descriptors by combining the final embeddings of informative regions selected by ranking attention weights.

Given an image $\boldsymbol{x} \in \mathbb{R}^{C\times W\times H}$, let $\boldsymbol{z}^{(i)} = \left[z_1^{(i)}, z_2^{(i)}, \ldots, z_{N^2}^{(i)}\right]$ be the list of embedded patches resulting from the i^{th} encoder. Obviously, the first list of embedded patches $\boldsymbol{z}^{(0)}$ is spawned from the summation of linear projections and positional embedding as described above, $\boldsymbol{z}_0 = [embed(x_1) + position\_embed(x_1), \ldots, embed(x_{N^2}) + position\_embed(x_{N^2})]$. The ith embedding layer is the output from the ith encoder module, while input is the $(i-1)$th embedding layer $\boldsymbol{z}^i = encoder(\boldsymbol{z^{i-1}})$. Let a_p be a joint self-attention weight of a local patch x_p, $p = 1, \ldots, N^2$, then our visual descriptor for an image $\boldsymbol{x}$ is the ranking weighted sum of its patch embeddings,

$$\boldsymbol{z}(K) = \sum_i^K a_i \boldsymbol{z^i}, \tag{3}$$

where K is the desired rank, $\boldsymbol{a} = \sigma([a_1, \ldots, a_{N^2}])$ is a permutation of the list of joint attention weights, and $\boldsymbol{z} = \sigma([z_1, \ldots, z_{N^2}])$ is the corresponding list of embedded patches' output from the final encoder and sorted with the same order. When σ is the arrangement from greatest to least, our visual descriptor is the weighted sum of the most K attentive regions.

According to ViT architecture, each encoder block has its own self-attention maps from the multi-head attention. The self-attention mechanism allows ViT to interpret information across the entire image, even in the early layers. In the early layers, some heads consistently focus on small areas, while others attend to most parts of the image, indicating that the ability to unite information globally is already in use inside the early layers. Meanwhile, the attention regions from all heads tend to be wider when going through higher layers, showing that the model aims to capture global information at higher layers. This ability is analogous to the receptive field concept in CNNs, which is the strength of convolutional layers capable of integrating both local and global information.

Instead of considering the correlation among patches of an image, we investigate parts of the image that should be attended to and extract their corresponding embedded patches. As highlighted in the equation, the embedded patches are output from the final encoder block, but the attention weights are jointly measured across multi-head attention modules. Let $\boldsymbol{A}^{(i)} \in [0,\ 1]^{h\times(N^2\times N^2)}$ be the attention map calculated inside the multi-head attention module of the ith encoder, where h is the number of attention heads, and N^2 is the number of patches as earlier mentioned. Then, $\boldsymbol{A}^{(i)}$ is a collection of attention maps, which are symmetric matrices of size $N^2 \times N^2$ whose coefficients estimate the degree of attention between two patches. To derive the joint attention map, we first average all attention maps for different heads to obtain an attention map responsible for a layer. To account for residual connections, we add an identity matrix to the attention map and re-normalize the weights,

$$\boldsymbol{Ana}^{(i)} = normalize\left(\frac{1}{h}\sum_j^h A_j^{(i)} + I_{N^2\times N^2}\right), \tag{4}$$

where $\boldsymbol{Ana}^{(i)} \in [0,\ 1]^{N^2\times N^2}$ is the normalized average attention map from the ith layer, $I_{N^2\times N^2}$ is an identity matrix of size $N^2 \times N^2$, and $normalize(\boldsymbol{X}) = \boldsymbol{X}/\sum_{i,j} X_{ij}$. Finally, the joint attention map is obtained by multiplying the attention maps across all layers.

$$\boldsymbol{Ajoint} = \prod_i^L \boldsymbol{Ana}^{(i)}, \tag{5}$$

where $\boldsymbol{Ajoint} \in [0,\ 1]^{N^2\times N^2}$ and L is the number of encoder blocks. To generate the visual descriptors, we require only the attention weights assigned to image patches; these

attention weights attend to themselves and can be attained by extracting diagonal of the joint attention matrix $\boldsymbol{a} = [a_1, \ldots, a_{N^2 \times N^2}] = diagonal(\boldsymbol{Ajoint})$. The attention weights derived here are exactly the weights involved in Equation (3). The aforementioned method is referred to as attention rollout and was introduced in [11]. Figure 4 illustrates our approach to extract visual representations from the ViT model.

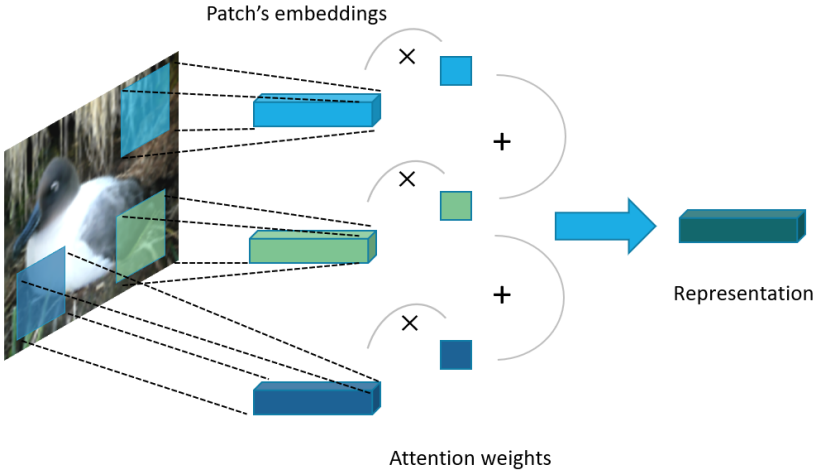

Figure 4. Extracting attention representation from Visual Transformer model: The output of the final encoder module of the ViT model includes $N \times N$ patch's embeddings and $N^2 \times N^2$ attention weights. The attention representation of an image is calculated by summation of the patches' embeddings weighted by the attention scores.

2.2. Contrastive Learning Framework

We train projection heads on top of the ViT model in a contrastive way to maximize the agreement of positive pairs in an embedding space. The contrastive learning framework can be decomposed into four modules: data augmentation, backbone network, projection head, and contrastive objective function.

Data augmentation: This study concentrates on low-resolution image retrieval, and thus the augmentation method we applied here only varies the input's resolution. A stochastic data augmentation module transforms the input data randomly, resulting in two correlated views of the same sample but with different resolutions. In this work, we apply random cropping followed by a low-pass filter. The low-pass filter used here is the Gaussian blur with a fixed size of the kernel and random kernel standard deviation.

Base encoder: The input to the base encoder network is augmented data, and output is a representation vector. We mainly use ViT as our base encoder model; however, we also experiment with other encoder networks, such as BiT [20] and EfficientNet [21]. In the case of BiT and EfficientNet encoders, we extract the final convolutional layer before fully connected layers and then apply adaptive average pooling to obtain a 1-D representation vector. This study uses $\boldsymbol{z} = Encoder(\boldsymbol{x})$ as a notation for the representation vector. As a result, the dimension of the representation vector varies based on encoder architectures. The base encoder is visualized as two "Encoder" blocks as in Figure 2.

Projection head: The projection head consists of a non-linear mapping that maps representation vectors into an embedded space where the similarity between samples can be measured. As suggested in [16], we use an MLP with one hidden layer to formulate the projection head. This study uses $\boldsymbol{e} = Projector(\boldsymbol{z})$ as a notation for the embedded vector. The projection head is depicted in Figure 2 as "Projector" blocks with a hidden layer. The size of the hidden layer is the same as the number of dimensions of embedded vectors.

Contrastive objective function: A contrastive function is defined so that minimizing it results in maximizing the agreement between positive pairs; in other words, it is meant to pull similar samples closer to each other and push dissimilar samples in the opposite direction. We measure the similarity within a minibatch of N random samples. Following

the setting of the data augmentation module, two different resolution versions of the original input are generated inside a minibatch, resulting in $2N$ data points. Within a multiview minibatch, let $i \in I \equiv \{1, 2, \ldots, 2N\}$ be the index of an arbitrary augmented sample, and let $j(i) \in I$ be the index of the other augmented sample obtained from the same source sample. In self-supervised contrastive learning [16], the loss function for a positive pair of examples (i, j) is defined as

$$L^{self} = \sum_{i \in I} L_i^{self} = -\sum_{i \in I} \log \frac{\exp\left(sim\left(e_i, e_{j(i)}\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim(e_i, e_k)/\tau)}, \tag{6}$$

where e_i is embedding obtained from the projection head, and τ is the temperature scaling factor. This loss is infoNCE loss that maximizes a lower bound on mutual information of two observations.

Regarding the presence of labels, supervised contrastive losses [37] can be used and also be generalized to an arbitrary number of positive pairs. The loss function takes the following form:

$$L^{sup} = \sum_{i \in I} L_i^{sup} = \sum_{i \in I} -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(sim\left(e_i, e_p\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim(e_i, e_k)/\tau)}, \tag{7}$$

where $P(i) \equiv \left\{p \in A(i) : y_p = y_i\right\}$ is the set of indices for all positives in the multiview batch. In addition to the augmented version of the anchor, supervised contrastive loss considers the same label samples within the minibatch as a positive pair.

2.3. Auxiliary Module: Classification Loss and Triplet Loss

Contrastive loss maximizes agreement between augmented versions derived from the same source without implicitly sampling negative pairs [16]. To achieve a better performance, many negative pairs are sampled to ensure the convergence of the contrastive objective function. For example, [16] used a batch size of 8196. In that case, 16,382 negative samples per positive pair were given from both augmentation views, and the same conditions were applied in [37] with a batch size of 6144. Using a large batch size is a computational burden and hard to train with regular optimizations [38,39]. In this study, instead of using a large batch size, we leverage the samples' label to implicitly generate pairs of negative and positive samples. We also found that training with implicit labels or supervised training is standard practice to learn embedded vectors for category image retrieval.

As proposed in previous literature on image retrieval, softmax cross-entropy loss and ranking loss, such as triplet loss, are used for end-to-end training of a CNN backbone [6] or to fine-tune model triplet loss based on a classifier trained with cross-entropy loss [5,8]. In this study, we train the model with auxiliary classification loss to maximize inter-class distance and utilize triplet loss to rank the embeddings of inter-class pairs over intra-class pairs. We add label smoothing [40] and temperature scaling [41] in the auxiliary cross-entropy loss function to prevent overconfidence and to learn better embedding.

$$L^{CRE} = -\frac{1}{2N} \sum_{i=1}^{2N} log \frac{\exp\left(\frac{c_{iy(i)}}{\tau}\right)}{\sum_{j}^{M} \exp\left(\frac{c_{ij}}{\tau}\right)}, \tag{8}$$

where $2N$ is the batch size, including augmented samples, M is the number of classes, and τ is the temperature scaling factor. $c_i = W^T z_i + b$ is logits of sample ith obtained by adding a trainable liner layer over the embedded vector e_i.

Additionally, we add triplet loss to the objective function. Minimizing the triplet loss in the embedding space results in instances with the same label, and its augmentations should be closer together to form well-separated clusters. The version of triplet loss used

in this study is online triplet mining using the hard-batch strategy [42]. For each sample a in the batch, we can select the hardest positive and the hardest negative samples within the batch when forming the triplets to be used for computing the loss.

$$L^{triplet} = \sum_{i=1}^{P} \sum_{a=1}^{K} \left[m + \max_{\mathrm{p}=1\ldots\mathrm{K}} D\left(e_{ia}, e_{ip}\right) - \min_{j=1..P, n=1..K, j\neq i} D\left(e_{ia}, e_{jn}\right) \right], \tag{9}$$

where m is margin and P and K are the number of classes and the number of samples in these classes, calculated within a minibatch. The final loss function for end-to-end training of our framework is the weighted summation of a contrastive loss and the two auxiliary losses.

$$L = \alpha L^{self} + (1-\alpha) L^{sup} + \beta \times L^{CRE} + \gamma L^{triplet}, \tag{10}$$

where $\alpha, \beta, \gamma \in \{0, 1\}$ are hyperparameters to control the presence of each loss.

3. Results

We evaluated our proposed framework on category image retrieval tasks with low-resolution queries. This section gives a brief overview of the datasets that were used in this experiment, and then we give implementation detail such as model configurations and training settings; finally, we present quantitative results of the ranking recall and show visualization results.

3.1. Datasets

We report performance on three popular datasets widely used for category-level image retrieval. CUB200-2011 [43] dataset contains 11,788 images representing 200 bird classes. CARS196 [44] dataset consists of 16,185 images corresponding to 196 classes. Stanford Online Products [45] contains 120 k online product images of 22,634 categories. Such data are prone to fine-grain tasks, especially CARS196 and CUB200-2011, which contain many images that share the same properties of an object across categories such as shape and color. In the contrastive learning framework, we apply an identical augmentation for all three datasets. First, we randomly crop part of the input image with the ratio ranges from 0.5 to 1, then apply Gaussian kernel to generate two blurred images originated from the same source; the kernel size is set to 23, but the variance is randomized from 1 to 5 to generate multiresolution samples. At the end of the preprocessing step, we resize the augmented images to a fixed size of 224×224. The training and testing set for each dataset are separated as per the default settings provided in the datasets package. We further split 20% from the training set to form a validation set. The sampling strategy is similar for all the three datasets. In the retrieval phase, the samples from the test set without blurry augmentation are used to form a gallery and samples from the evaluation set with the above augmentations are used as queries.

3.2. Implementations

All experiments are implemented using Pytorch on a Titan V GPU with 12 GB memory. We use Python as the programing language for all experiments, including structuring model architecture, loading dataset, and evaluations. The deep learning framework used in this study is Pytorch. The source code is available at https://github.com/Ka0Ri/Contrastive-learning-for-image-retrieval (accessed on 20 May 2021).

We use BiT [20] and EfficientNet [21] as the backbone network for comparison with the ViT [11] model. All the models are fine-tuned based on pre-trained weights on ImageNet. The comparison of the size of models is given in Table 1. For more details, we use ViT-B-16, the base version with 12 encoder layers, 12 self-attention heads per layer, and patchify an image by 16×16. We use BiT-M-R152 $\times$ 4 architecture, a varied version of ResNet version 2 with 152 layers and a width factor of 4. BiT models maintain almost the same architecture as the original ResNet version 2, except for replacing batch normalization with group normalization and weight standardization. The BiT comes up with three versions, S,

M, and L, where the pre-trained dataset scales up from S to L. The scale of BiT depends on which ResNet model is used; in our study, two factors are considered: depth factor, which reflects how deep the model is going to be (50, 101, 152 layers), and width factor, which defines how many channels are used in a residual model ($\times 1$, $\times 2$, $\times 4$). EfficientNets are a family of models resulting from extensive search by neural architect search to balance performance and computational resources usage. EfficientNet's scaling factors include depth, width, and resolution of a pre-trained dataset of use to maximize the model accuracy for any given resource constraints. EfficientNet is scaled up from MobileNet (B0) into seven versions from B1 to B7 where the number of parameters gradually increased from 5.3 M to 66 M. We experiment with EfficienNet and BiT along with ViT architecture to clarify the effectiveness of ViT's attention-based mechanism over residual-based and depth-wise convolution-based mechanisms. In this study, we use Efficient-B7 for a fair comparison with other types of architectures. ViT, BiT, and EfficientNets are models that challenge the recently developed state of the art in regards to the large-scale ImageNet dataset.

Table 1. Parameters count, FLOPS, and ImageNet Top-1 accuracy for ViT, BiT, and EfficientNet. The parameters and FLOPS are reported by "thop" library given an image of size $224 \times 224 \times 3$.

Model Name	Params	FLOPS	Top-1 (%)
ViT-B-16	86 M	16.85 G	84.15
BiT-M-R152x4	68 M	68.59 M	85.39
Efficient-B7	66 M	87.07 M	84.3

We consider the setup generic for training hyperparameters in all experiments unless otherwise stated in the separated experiments. In the training phase, we use the AdamW optimizer [46] with cosine-annealing learning-rate schedule during the training process, the weight decay is set to 10^{-6}, and the other parameters of the optimizer are set to default. The learning is also warmed up during the first ten epochs from 0 to 0.001. The model is trained for 100 epochs with a batch size of 64.

3.3. Quantitative Results

We save the model's weights that attain the lowest loss value or the highest accuracy for the validation set in the case of using the auxiliary classification loss. The model's weights are loaded into the corresponding model to extract embedded vectors for performing image search in the retrieval phase. The search strategy used in our study is simply an exhaustive search using L_2 similarity. Note that the embedded vectors are normalized in a unit sphere so that the similarity can be calculated simply by the dot product. For quantitative evaluation, the embedded vectors from the test set are used to build a gallery and embedded vectors extracted from the validation set act as queries. It is rational to apply a more effective searching strategy rather than an exhaustive comparison using L2 for image retrieval; however, we focus on improving the searching space and consider effective search methods as future works. The matching score for a query is measured by ranking recall as follows.

$$R@k(x) = \begin{cases} 1 & |\{z \in R_k | y(z) = y(x)\}| > 0 \\ 0 & |\{z \in R_k | y(z) = y(x)\}| = 0 \end{cases}, \tag{11}$$

where x is a query image and R_k is a set of top k retrieval results. The main focus of this experiment is to calculate the average recall value over queries from the validation set, and such a higher recall value shows better performance. In addition, we also calculate the mean average precision (mAP) for comparison purposes.

Most experiments were carried out with a basic setting as follows: the ViT-B-16 model is fine-tuned and used to perform image retrieval on the CUB200-2011 dataset with supervised contrastive loss function ($\alpha = 0$), auxiliary classification loss ($\beta = 1$), and triplet loss function ($\gamma = 1$); the embedding dimension is set to 128, and the representation vector is

extracted from the class token. First, we show the performance of the proposed framework using different datasets. Then we make an ablation study about the effectiveness of the backbone networks, the number of embedding dimensions, and loss components in the subsequent sections.

3.3.1. Experimental Results of Different Datasets

This section sets a benchmark for our framework on CUB200-2011, CARS196, and SOP datasets. Recall results are reported for the first five ranks on CUB200-2011 and CARS196 datasets, and they are reported in recall of rank 1, 10, 100, 500, and 1000 on the SOP dataset with the instance-level label. The experimental results show that our approach achieves a recall of 0.9414, 0.8541, and 0.9806 with the first ranking and mAP of 0.9379, 0.8804, and 0.9141 on the CUB200-2011, CARS196, and SOP datasets, respectively, details are given in Table 2. SOP dataset gives two types of labels: class labels with 22,634 categories and super-class labels with 12 categories. The super-class labels indicate the type of product, while the class label varies according to each product, that is, different views of the same product. Our result also shows a high recall value with instance image retrieval, and we obtain recall at the first rank of 0.947 on the SOP dataset.

Table 2. Recall@k and mAP results from CUB200-2011, CARS196, and SOP datasets.

Dataset	Recall@k					mAP
	1	2	3	4	5	
CAR196	0.8541	0.9218	0.9661	0.9739	0.9817	0.8804
CUB200-2011	0.9414	0.9687	0.9765	0.9765	0.9765	0.9379
SOP super-class	0.9806	0.9891	0.9932	0.9952	0.9966	0.9141
	1	10	100	500	1000	
SOP class	0.9470	0.9867	0.9962	0.9986	0.9986	

3.3.2. The Effectiveness of Backbone Network

Table 3 shows image retrieval performances as a result of different backbone encoder networks. The highest mAP value is obtained from ViT architecture (0.9379) which proves that representations encoded by ViT perform better than representations extracted from other state-of-the-art architectures, such as BiT (mAP = 0.904) and EfficientNets (mAP = 0.7906). Our results demonstrate that attention-based architecture performs better than ResNet-based architectures in regards to studies about image retrieval [1,3,8]. It is well understood that a big model commonly results in better performance. However, these results are not biased since we selected those architectures with the approximate number of parameters.

Table 3. Recall@k and mAP comparison between different encoding models.

Model Name	Recall@k					mAP
	1	2	3	4	5	
ViT-B-16	0.9414	0.9687	0.9765	0.9765	0.9765	0.9379
BiT-M-R152x4	0.9296	0.9648	0.9726	0.9765	0.9804	0.9040
Efficient-B7	0.7226	0.8242	0.8828	0.9062	0.9257	0.7906

3.3.3. The Effectiveness of the Number of Embedding Dimensions

A typical experiment in previous studies about information retrieval aimed to analyze the impact of the number of dimensions of embeddings. After a series of experiments, it is found that the dimensions of 256 produce the best performance with the mAP of 0.955 and the first rank recall of 0.9453, as shown in Table 4. These findings are consistent with previous research on contrastive learning [16,26,37], which shows that the embeddings' dimensions should be either 128 or 256. It is also consistent with studies about image

retrieval using the embeddings search strategy [3,6,8]. We also note that the size of a hidden layer in the projection head is equal to the dimensions of the embedded vector.

Table 4. Recall@k and mAP comparison using different numbers of embedding dimensions.

Dimensions	Recall@k					mAP
	1	2	3	4	5	
64	0.9296	0.9843	0.9843	0.9843	0.9843	0.9507
128	0.9414	0.9648	0.9765	0.9765	0.9804	0.9438
256	0.9453	0.9726	0.9843	0.9843	0.9882	0.9555
512	0.9101	0.9687	0.9882	0.9882	0.9960	0.9339
1024	0.9257	0.9570	0.9726	0.9726	0.9765	0.9336

3.3.4. The Effectiveness of Loss Components

We study the effectiveness of loss components that impact the performance of image retrieval. We add auxiliary loss functions such as triplet loss and classification loss into the contrastive loss to train the model end-to-end. In particular, we experiment with a tuple of three parameters $(\alpha, \beta, \gamma) \in \{0, 1\}^3$ that characterize the presence of self-contrastive loss ($\alpha = 1$), otherwise supervised contrastive loss ($\alpha = 0$), the presence of classification loss ($\beta = 1$), and triplet loss ($\gamma = 1$). As mentioned in the previous section, the classification loss used in this study is the cross-entropy loss with label smoothing ($p = 0.1$) and temperature scaling ($\tau = 0.5$). Otherwise, the temperature scaling factor in contrastive loss is set to 0.5. In addition, the margin parameter in triplet loss is set to 1. Table 5 demonstrates that the case of using supervised contrastive loss in combination with classification and triplet loss achieves the best performances ($mAp = 0.9379$). The experiment also shows that stand-alone contrastive loss is not enough to accomplish the category image retrieval task, as shown in the cases of self-supervised contrastive loss ($mAP = 0.6026$) and supervised contrastive loss ($mAP = 0.6214$). In addition, the effectiveness of classification loss is higher than that of triplet loss, as shown in the case of (self-)contrastive loss combined with classification loss ($mAP = 0.9120$, $mAP = 0.8929$), compared to the case of (self-)contrastive loss combined with triplet loss ($mAP = 0.8445$, $mAP = 0.8910$). The results confirm that classification loss is a good option to supply category information for image retrieval tasks.

Table 5. Recall@k and mAP comparison between using different loss function settings of ViT model.

Loss Components	Recall@k					mAP
	1	2	3	4	5	
$\alpha = 1, \beta = 0, \gamma = 0$	0.6210	0.7070	0.7382	0.7890	0.8164	0.6026
$\alpha = 0, \beta = 0, \gamma = 0$	0.6484	0.7265	0.7773	0.7890	0.8046	0.6214
$\alpha = 1, \beta = 0, \gamma = 1$	0.8437	0.8984	0.9375	0.9492	0.9570	0.8445
$\alpha = 0, \beta = 1, \gamma = 0$	0.8789	0.9531	0.9726	0.9726	0.9726	0.8929
$\alpha = 1, \beta = 1, \gamma = 0$	0.9023	0.9531	0.9726	0.9765	0.9765	0.9120
$\alpha = 0, \beta = 1, \gamma = 1$	0.9414	0.9687	0.9765	0.9765	0.9765	0.9379
$\alpha = 1, \beta = 1, \gamma = 1$	0.9296	0.9804	0.9804	0.9843	0.9882	0.9268
$\alpha = 0, \beta = 0, \gamma = 1$	0.8867	0.9296	0.9531	0.9531	0.9531	0.8910

3.3.5. The Effectiveness of Attention Mechanism

Finally, we analyze the effectiveness of attention embeddings, which is the key proposal in our study. ViT architecture is built upon the attention mechanism, where each image patch has its own attention weight to determine which region should be focused. In this study, we also investigate the effectiveness of the attention mechanism by analyzing the number of "decisive" patches corresponding with the highest attention weights. Table 6 illustrates that our simple method achieves a better result using 25 decisive patches (recall@1 = 0.9492, mAP = 0.9475). The case of only one decisive patch means that only

the most attentive region is extracted, while the case of 128 decisive patches reflects that the average attention of all patch's embeddings is extracted. We compare our findings with a previous study [6] to address our results, as shown in the CGD row. The study from [6] is a typical study about image retrieval where a ResNet-based model is fine-tuned using combined global descriptors, and this study achieved significant performance on a broad range of datasets. We verify that our proposed framework outperforms CGD in both terms of ranking recall and mAP. Together, the present findings confirm that using attention patch's embedding is slightly more robust than class's embedding, as suggested in [12]. To conclude this section, we show that fine-tuning the ViT model with contrastive learning provides substantially better results than the direct use of features extracted from the pre-trained model, that is, an mAP of 0.54 without fine-tuning compared to an mAP of 0.6214 when fine-tuned with supervised contrastive learning. This phenomenon can be extended to other models as well.

Table 6. Recall@k and mAP comparison of different representation output via ViT-based encoder.

Model Name	Recall@k					mAP
	1	2	3	4	5	
ViT-class	0.94140625	0.96875	0.976563	0.976563	0.976563	0.937995
ViT-1	0.8984375	0.949219	0.960938	0.972656	0.976563	0.897331
ViT-128	0.92578125	0.96875	0.980469	0.992188	0.992188	0.918958
ViT-25	0.94921875	0.988281	0.992188	0.992188	0.996094	0.947578
CGD	0.875	0.923611	0.947917	0.958333	0.96875	0.857604
No fine-tuning	0.55078125	0.648438	0.6875	0.726563	0.757813	0.540013

3.4. Qualitative Analysis

From the quantitative results presented in the previous section, we verify that our best model for a retrieval system is the model with 25 decisive patch embeddings trained with supervised contrastive loss, classification loss, and triplet loss. This section illustrates some qualitative analyses obtained from our method. First of all, we describe that our framework works well across fine-grain datasets, typically CUB200-2011, CARS196, and SOP. Figure 5 shows the retrieval results when the queries are images with mild resolution reduction. It is important to highlight the fact that a successful retrieval system should show that retrieval results match the query in different views, such as varied lighting conditions or points of view. This property partially manifests in our results, for example, a green car in a different viewpoint or the same type of car but in two different colors, red and blue, as illustrated in Figure 5.

To clarify the purpose of our study, we consider the case where strong resolution reduction is used, as depicted in Figure 6. The result from Figure 6 is concrete proof that our framework can deal with low-resolution images. We speculate that the result might be due to contrastive learning to fine-tune the model with both multiresolution and fine resolution samples. The study may raise concerns about superior performance, which can be addressed by using super-resolution preprocessing. However, training additional models for super-resolution is impractical because of the demand for high computational resources. In addition, in our findings, we believe that the effectiveness of descriptors for feature matching is a result of careful fine-tuning of the model with proper augmentation.

Finally, we once again address the effectiveness of the attention mechanism, as demonstrated in Figure 7. The attention mechanism grants the ability to focus on the region of interest, that is, the region that contains objects, as displayed in Figure 6. Attention is particularly important when investigating visual objects in a distracting background. The experimental results herein aim to verify that with the help of attention, the retrieval results can be more accurate in fine-grain category-image retrieval and even in sophisticated cases where semantic vision may not be distinguished.

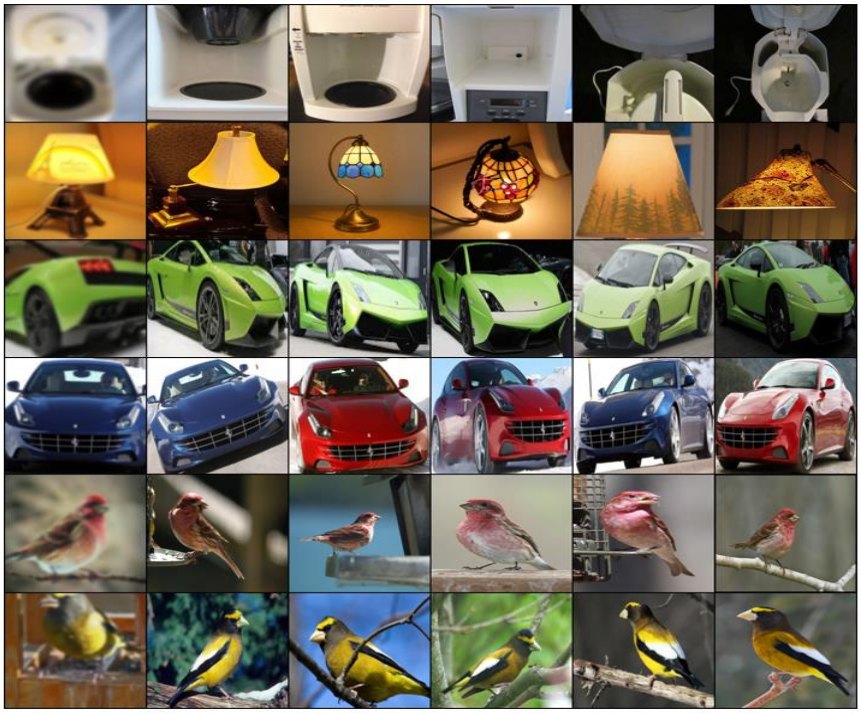

Figure 5. Top 5 retrieval results on mild resolution reduction samples. The first image in each row is the query image. The first two rows show samples from the SOP dataset, the next two rows show samples from the CARS196 dataset, and the last two rows show samples from the CUB200-2011 dataset.

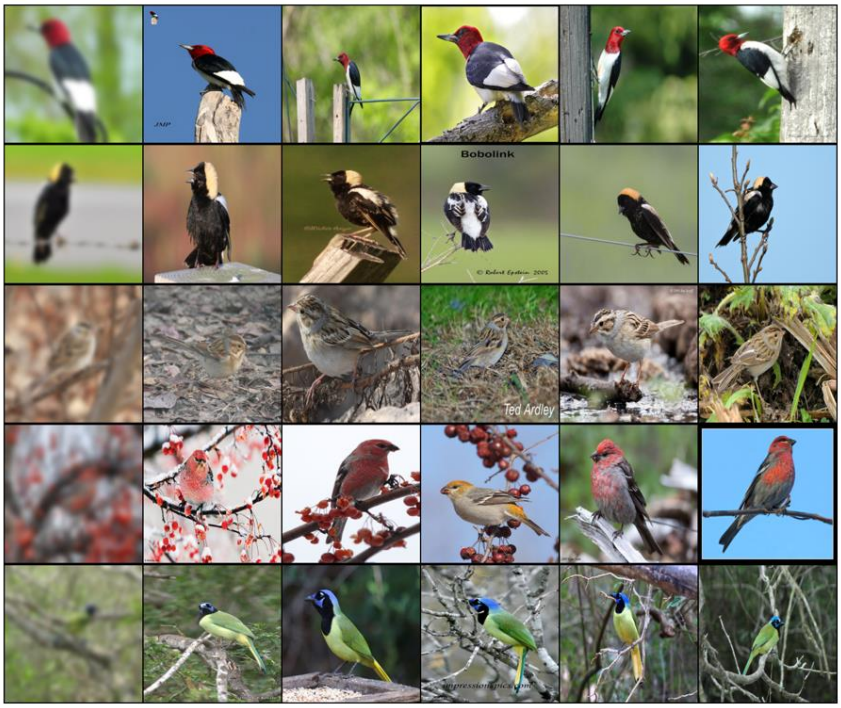

Figure 6. Top 5 retrieval results on aggressive resolution reduction samples from the CUB200-2011 dataset. The first image in each row is the query image.

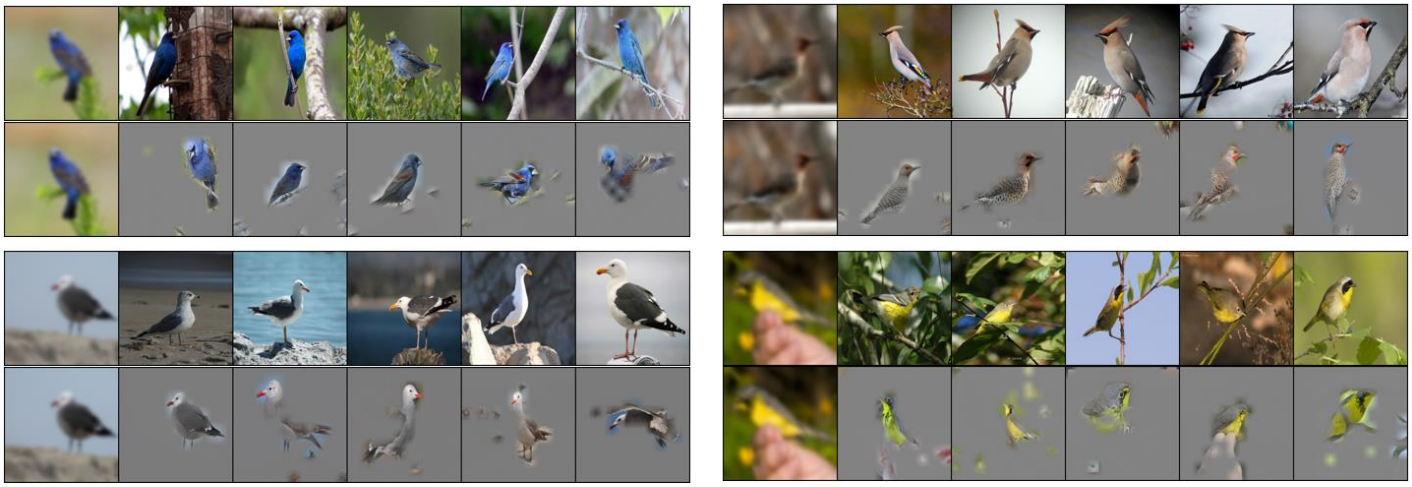

Figure 7. Top 5 retrieval results of samples from the CUB200-2011 dataset. The first image on each row is the query image. The figure shows comparisons between using attention embeddings (the second row) and without attention (the first row).

4. Conclusions

This study introduced a simple yet effective framework for category image retrieval. We exploited the Visual Transformer architecture to take advantage of the self-attention mechanism to enhance the robustness of representation vectors. Additionally, we solved the low-resolution image retrieval problem by using contrastive learning with a proper augmentation strategy. The solution proposed here addresses only the case of low-resolution samples; however, the same framework can be applied to other degraded image retrieval systems if a suitable augmentation method is defined. We guarantee the effectiveness of our approach through extensive experiments, both quantitative and qualitative, on several public datasets. However, the limitation of this study is that the retrieval system was only analyzed at the category level. The lack of evaluations at the instance level makes the model a bit inferior to the general-purpose retrieval system, and we wish to tackle this challenge in future studies.

Author Contributions: Conceptualization, J.-Y.K. and G.-H.Y.; methodology, T.-V.D.; software, G.-H.Y.; validation, J.-Y.K. and G.-H.Y.; formal analysis, G.-H.Y.; investigation, T.-V.D.; writing—original draft preparation, T.-V.D.; writing—review and editing, J.-Y.K.; visualization, T.-V.D.; supervision, J.-Y.K.; project administration, J.-Y.K.; funding acquisition, G.-H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the BK21 FOUR Program (Fostering Outstanding Universities for Research, 5,199,991,714,138) funded by the Ministry of Education (MOE, Korea) and the National Research Foundation of Korea (NRF).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets involved in this study are public data and available online. The CUB200-2011 can be found at http://www.vision.caltech.edu/visipedia/CUB-200.html (accessed on 20 May 2021). The CARS196 dataset can be found at https://ai.stanford.edu/~jkrause/cars/car_dataset.html (accessed on 20 May 2021). The Stanford Online Products dataset can be found at https://cvgl.stanford.edu/projects/lifted_struct/ (accessed on 20 May 2021).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish.

References

1. Babenko, A.; Slesarev, A.; Chigorin, A.; Lempitsky, V. Neural Codes for Image Retrieval. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 584–599.
2. Tolias, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2015**, arXiv:1511.05879.
3. Albert, G.; Almazán, J.; Revaud, J.; Larlus, D. Deep Image Retrieval: Learning Global Representations for Image Search. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 241–257.
4. Artem, B.; Lempitsky, V. Aggregating deep convolutional features for image retrieval. *arXiv* **2015**, arXiv:1510.07493.
5. Filip, R.; Tolias, G.; Chum, O. Fine-tuning CNN Image Retrieval with no Human Annotation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway, NJ, USA, 2019; Volume 41, pp. 1655–1668.
6. Jun, H.; Ko, B.; Kim, Y.; Kim, I.; Kim, J. Combination of multiple global descriptors for image retrieval. *arXiv* **2019**, arXiv:1903.10663.
7. Revaud, J.; Almazán, J.; Rezende, R.; Souza, C.R. Learning with average precision: Training image retrieval with a listwise loss. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.
8. Albert, G.; Almazan, J.; Revaud, J.; Larlus, D. End-to-end learning of deep visual representations for image retrieval. *Int. J. Comput. Vis.* **2017**, *124*, 237–254.
9. Dzmitry, B.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
10. Ashish, V.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
11. Alexey, D.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M. An image is worth 16x16 words: Transformers for image recognition at scale. In Proceedings of the International Conference on Learning Representations, Lisbon, Portuagal, 7–8 October 2021.
12. El-Nouby, A.; Neverova, N.; Laptev, I.; Jégou, H. Training Vision Transformers for Image Retrieval. *arXiv* **2021**, arXiv:2102.05644.
13. Kim, W.; Goyal, B.; Chawla, K.; Lee, J.; Kwon, K. Attention-based ensemble for deep metric learning. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 736–751.
14. Chen, B.; Deng, W. Hybrid-attention based decoupled metric learning for zero-shot image retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
15. Socratis, G.; Boutalis, Y.; Chatzichristofis, S. Investigating the Vision Transformer Model for Image Retrieval Tasks. *arXiv* **2021**, arXiv:2101.03771.
16. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 13–18 July 2020.
17. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
18. Nicolas, C.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
19. Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; Shlens, J. Stand-Alone Self-Attention in Vision Models. *arXiv* **2019**, arXiv:2010.11929.
20. Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; Houlsby, N. Big Transfer (BiT): General Visual Representation Learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
21. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
22. Alexander, K.; Zhai, X.; Beyer, L. Revisiting self-supervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
23. Pham, H.; Dai, Z.; Xie, Q.; Luong, M.-T.; Le, Q. Meta pseudo labels. *arXiv* **2020**, arXiv:2003.10580.
24. Zoph, B.; Ghiasi, G.; Lin, T.-Y.; Cui, Y.; Liu, H.; Cubuk, E.; Le, Q. Rethinking Pre-training and Self-training. *arXiv* **2020**, arXiv:2006.06882.
25. Tian, Y.; Krishnan, D.; Isola, P. Contrastive multiview coding. *arXiv* **2019**, arXiv:1906.05849.
26. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv* **2018**, arXiv:1807.03748.
27. Phuc, L.-K.; Healy, G.; Smeaton, A. Contrastive representation learning: A framework and review. *IEEE Access* **2020**, *8*, 193907–193934.
28. Spyros, G.; Singh, P.; Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv* **2018**, arXiv:1803.07728.
29. Phillip, I.; Zoran, D.; Krishnan, D.; Adelson, E. Learning visual groups from co-occurrences in space and time. *arXiv* **2015**, arXiv:1511.06811.
30. Mehdi, N.; Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 69–84.
31. Hjelm, D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv* **2018**, arXiv:1808.06670.

32. Michael, T.; Djolonga, J.; Rubenstein, P.; Gelly, S.; Lucic, M. On mutual information maximization for representation learning. *arXiv* **2019**, arXiv:1907.13625.
33. Wang, T.; Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 13–18 July 2020.
34. Chen, T.; Li, L. Intriguing Properties of Contrastive Losses. *arXiv* **2020**, arXiv:2011.02803.
35. Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; Isola, P. What makes for good views for contrastive learning. *arXiv* **2020**, arXiv:2005.10243.
36. Senthil, P.; Gupta, A. Demystifying contrastive self-supervised learning: Invariances. *arXiv* **2020**, arXiv:2007.13916.
37. Prannay, K.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishna, D. Supervised contrastive learning. *arXiv* **2020**, arXiv:2004.11362.
38. Yang, Y.; Gitman, I.; Ginsburg, B. Large batch training of convolutional networks. *arXiv* **2017**, arXiv:1708.03888.
39. Shirish, K.N.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; Tang, P.T.P. On large-batch training for deep learning: Generalization gap and sharp minima. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
40. Christian, S.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
41. Chuan, G.; Pleiss, G.; Sun, Y.; Weinberger, K. On Calibration of Modern Neural Networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
42. Hermans, A.; Leibe, B.; Beyer, L. In Defense of the Triplet Loss for Person Re-Identification. *arXiv* **2017**, arXiv:1703.07737.
43. Peter, W.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. *Caltech-UCSD Birds 200*; California Institute of Technology: Pasadena, CA, USA, 2010.
44. Krause, J.; Stark, M.; Deng, J.; Li, F.-F. 3D Object Representations for Fine-Grained Categorization. In Proceedings of the 4th IEEE Workshop on 3D Representation and Recognition, Sydney, NSW, Australia, 2–8 December 2013.
45. Song, H.O.; Xiang, Y.; Jegelka, S.; Savarese, S. Deep Metric Learning via Lifted Structured Feature Embedding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
46. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6 May 2019.