A Trademark Retrieval Method Based on Self-Supervised Learning

Kailang Hu
Zhengzhou University
No. 100 Science Avenue, Zhengzhou
Henan 450001, China
hukailang@gs.zzu.edu.cn

Yixiao Lu
Yes Energy LLC
1877 Broadway #606, Boulder
CO 80302, United States
yixiao.lu@yesenergy.com

Huibing Li
Zhengzhou University
No. 100 Science Avenue, Zhengzhou
Henan 450001, China
bin3012924536@gs.zzu.edu.cn

Xuan Song
Zhengzhou University
No. 100 Science Avenue, Zhengzhou
Henan 450001, China
songxuan@zzu.edu.cn

Abstract

Trademark retrieval is a frequently used task in intellectual property protection. Utilizing efficient trademark retrieval methods can improve retrieval efficiency, reduce manual review costs, and effectively prevent trademark infringement. While the diversity and complexity of trademark, as well as the scarcity of labeled data, challenge existing retrieval methods, we propose a revised trademark retrieval system based on self-supervised learning.Our method revises MoCoV2 self-supervised learning framework by introducing a hard sample selection strategy to enhance the model's performance and its capacity of feature representation. We also integrate attention mechanisms and multi-stage feature fusion to improve the model's ability to capture significant visual elements in trademark and multi-scale features. We conducted evaluation and comparison experiments on the METU dataset. The experimental results indicate that our method achieves better performance on the metrics of NAR (Normalized Average Rank) and MAP@ 100 (Mean Average Precision at 100) compared to the state-of-the-art method, proving the effectiveness of the proposed trademark retrieval method.

Keywords: ***Trademark Retrieval, Self-supervised Learning, Feature fusion, Intellectual property protection***

1. Introduction

In contemporary economic and business activities, protecting trademarks from similar confusion infringement has become increasingly important. When trademark holders want to register new trademarks and protect their brand interests, applicants need to submit the trademark to the patent office for registration review. The patent office will examine whether the new trademark has textual and graphical similarities or semantic confusion with registered trademarks in the database.

According to the 2023 report data from the World Intellectual Property Organization (WIPO)[1], a total of 15.5 million trademark applications were submitted globally in 2022, with an annual increase. To address this massive volume of trademark applications and registrations, the human-involved application reviewing process is not efficient enough. Therefore, a Large-Scale Trademark Retrieval (LSTR) system for similar trademark searches can benefit both reviewers and applicants. For reviewers, it can quickly eliminate similar or confusing trademarks, reducing workload and increasing review speed. For applicants, it reminds potential duplicate trademarks before submission, lowering the cost of potential disqualified applications and overwhelming revisions.

The development of LSTR systems can be summarized into three stages. (1) **Manual category labeling stage.** Early LSTR systems were based on category label patterns, namely the Vienna Classification System[2], where researchers decomposed trademarks into several categories and subgroups based on their graphic elements, and category labels were assigned to each graphic element, enabling retrieval through searching for relevant label codes. (2) **CBIR stage based on hand-crafted features.** It measures similarity based on images' content rather than metadata that manual labeling uses, which is prior to manual category labeling methods as a more efficient way. (3) **CBIR**

[1] Source: https://www.wipo.int/edocs/pubdocs/en/wipo-pub-941-2023-en-world-intellectual-property-indicators-2023.pdf

[2] Source: https://www.wipo.int/classifications/vienna/en/

stage based on deep learning features. With deep learning features, it have been proven to have advantages in accuracy and extracting high-level semantic information [36] than hand-crafted features, thus being widely used in recent work.

CBIR methods based on deep features mainly rely on supervised metric learning patterns, which requires large-scale datasets with positive and negative sample annotations to support supervised training. However, there is currently a lack of large-scale labeled datasets in the field of trademarks. This has led to the phenomenon that most work in trademark retrieval relies on pre-trained models from general domains, where potential to improve retrieval performance is limited.

Recent studies [46, 7, 16] indicate that self-supervised learning methods can extract effective invariant features from large-scale unlabeled data, facilitating the transfer of existing general-domain models to downstream tasks. Based on this understanding, it is feasible to use existing large amounts of unlabeled trademark data for self-supervised training instead of supervised training, which makes it possible to train components with trainable parameters for enhancing the representation capability of trademark retrieval models.

In this paper, we propose a trademark retrieval method based on self-supervised contrastive learning. This method is an improvement on the MoCoV2 contrastive learning framework, introducing a hard negative sample selection strategy to accelerate convergence, balance the number of positive and negative samples, and prevent the model from converging to trivial solution. In addition, to enhance model's representation capability, We also introduce a lightweight attention mechanism and multi-scale feature fusion method in the framework encoder. To prove that the proposed method is valid, we conduct comparative experiments on the METU trademark dataset.

The remaining part of this paper is orgnized as follows. Section 2 discusses the related literature in trademark retrival area and self-supervised learning algorithms that we mainly use. In Section 3, we propose a A large-scale trademark retrieval framework. In Section 4, we measure the proposed method in two datasets and show the results of evaluation metrics. Finally, we conclude our work in Section 5.

2. Related Work

In this section, we review the related literatures on existing trademark retrieval methods and self-supervised learning methods, where we propose some revisions to improve the models' performance.

2.1. Trademark retrieval methods

Early LSTR systems based on Manual category labeling relied on predefined graphic element code systems. In this pattern, domain experts were responsible for identifying distinctive graphic elements in trademark images and mapping them to code labels in the predefined classification standard, achieving retrieval results by searching for images corresponding to the code labels. However, this trademark retrieval pattern still relied on professionals for manual extraction, and the increasing complexity of trademark designs adds difficulty for the classification coding system to describe elements such as color, texture features, and artistic styles. Therefore, subsequent LSTR systems gradually replaced the classification coding system with content-based image retrieval (CBIR) to achieve more efficient and adaptive retrieval solutions.

CBIR methods achieves retrieval based on the content of the image itself rather than external information. Its concept is to process the image through algorithms to obtain distinctive descriptors, measure the differences between descriptors to determine the similarity between images, and then achieve retrieval based on similarity ranking. CBIR methods based on manually designed features mainly rely on global features formed by low-level image features such as texture, color, and shape, as well as local descriptors formed by geometric edge key points as image descriptors. Examples include global features composed of circularity, aspect ratio[12], gradient histogram[11], color edge gradient histogram[31], and complex designed descriptors such as shape context descriptor[34], SIFT[28], Fourier descriptor[19], and local self-similarity factor[6]. With the rise of deep learning, these manually designed feature extraction methods have gradually been replaced by deep learning methods, as the latter show greater advantages in terms of accuracy and high-level semantic information [36].

Most trademark retrieval work in CBIR based on deep learning features uses pre-trained models. For example, Cemal Aker et al. [1] attempted to use output features from the fully connected layer of pre-trained models for trademark retrieval, demonstrating significantly better retrieval performance compared to hand-crafted features. Subsequent works [26, 40] proved the effectiveness of using mainstream backbone networks, i.e., VGG [37] and ResNet [17], combined with deep learning feature aggregation patterns. Tursun et al. [40] employed various widely-used aggregation methods for deep features, such as SPoC[2], CRoW[23], and R-MAC[38], and attempted to use attention mechanisms to suppress the focus on text elements within trademarks to achieve better retrieval results. Lan et al. [26] tried using local binary patterns to aggregate feature maps from intermediate convolutional layers of pre-trained networks, further improving retrieval performance, but at the cost of introducing excessively high aggregation costs, lack-

ing scalability.

Recently, some studies have used supervised learning to fine-tune general domain models to improve performance. Perez et al. [30] attempted to fine-tune pre-trained models for task transfer by optimizing classification loss functions. For this purpose, they constructed a large visual trademark database and a large semantic trademark database. By training two VGG branch networks to learn visual and semantic features separately and then fusing them, they achieved better results than single-branch networks. Lan et al. [25] also built a trademark dataset containing 647 categories and used the triplet loss function [10] from metric learning for fine-tuning. However, these studies not only have not open-sourced their training sets, but also essentially treat the trademark retrieval task as a closed-set classification problem with many samples and few categories, rather than an open-set retrieval problem with few samples and many categories (few-shot problem). Therefore, it is difficult for subsequent work to extend based on these studies.

With the development of self-supervised learning techniques, recent research has begun to explore using self-supervised learning frameworks that do not require manually labeled data to fine-tune pre-trained models, in order to fully utilize effective information in existing unlabeled datasets. As an important method of self-supervised learning, contrastive learning has demonstrated superior performance in multiple domains by learning to distinguish between similar and dissimilar sample pairs to obtain effective feature representations. Cao et al. [4] first introduced the self-supervised learning instance discrimination framework to trademark retrieval, constructing positive and negative sample pairs to fine-tune models using existing large-scale unlabeled data, learning discriminative feature representations.

2.2. Self-supervised learning methods

Self-supervised learning, as a method between unsupervised learning and supervised learning [32], is applicable to scenarios with unstructured and unlabeled data. Unlike unsupervised learning, which extracts useful information from the inherent structure of data, self-supervised learning attempts to generate pseudo-labels or supervisory signals from data through pretext tasks, guiding the model to learn essential feature representations of the data. These feature representations often capture key attributes of the data and remain relatively stable under various transformations or perturbations. Self-supervised learning is typically categorized into three modes based on the source of supervisory signals: context-based, contrastive learning-based, and generative model-based[15]. Since contrastive learning is conceptually similar to metric learning methods commonly used in image retrieval, this work primarily references contrastive-based self-supervised learning methods.

As the dominant paradigm in self-supervised learning, contrastive learning has achieved extensive development in the field of computer vision in recent years. Early self-supervised methods based on contrastive learning were inspired by instance discrimination tasks. Z Wu et al. [46] treated each image instance as a category for unlabeled classification task data, constructed a feature pool called Memory Bank to store encoded features of all instance images, obtained different views through data augmentation to construct positive sample pairs, while features of randomly selected images from other categories were used to form negative sample pairs, thus creating contrastive samples. They trained the model to extract image features and classification capabilities by maximizing the similarity between positive samples while minimizing the similarity with negative samples through the contrastive learning framework's loss function NCELoss. Subsequently, He Kaiming et al. [16] proposed the MoCo architecture for self-supervised learning using a contrastive momentum encoder, referencing the ideas from this work. The MoCo architecture replaced the Memory Bank structure with a dictionary queue structure, viewing contrastive learning as a dictionary query task. It utilized a momentum encoder to dynamically update the dictionary during training, making more effective use of a large number of negative samples and improving training efficiency. Chen et al. [7] later proposed a new architecture called SimCLR. Unlike MoCo's approach of updating the dictionary with a momentum encoder, this architecture adopted an end-to-end training method, directly using data from the training mini-batch as negative sample candidates, eliminating additional data structures during training. Additionally, SimCLR improved data augmentation strategies and introduced a projection head structure to enhance feature separability, significantly improving the model's classification performance. This structure was widely adopted in subsequent works, becoming one of the foundations of self-supervised architectures. In their subsequent SimCLRV2 work, they also revealed the effectiveness of combining self-supervised learning with a small amount of labeled data in semi-supervised learning.

In addition to the aforementioned works that directly utilize positive and negative samples for contrastive learning, there are also some efforts that attempt to use other contrastive information to achieve optimization purposes. For example, the SwAV proposed by Caron et al. [5] introduces clustering priors into self-supervised learning, using cluster centers as objects for comparison, leveraging the contrastive information formed between different clusters. The BYOL proposed by Grill et al. [14] and the SimSiam proposed by Chen et al. [9] are based on the idea of self-distillation, focusing on the self-contrastive information formed by the consistency between different views or augmented samples of the same sample. The Barlow Twins proposed by Zbon-

tar et al. [48] and its improved work VICReg [3] are based on the idea of feature decorrelation, achieving implicit contrast by maximizing the independence between different dimensions.

3. Methodology

In this chapter, we will provide a detailed introduction to our proposed large-scale trademark retrieval method based on self-supervised learning.

Our method uses self-supervised learning to learn essential feature representations from a large-scale unlabeled trademark dataset to obtain a trained encoder for similarity retrieval of trademark images. We construct positive and negative sample pairs from the dataset through a self-supervised learning framework, learning their contrastive information to train the feature representation capability of the encoder network. This method is based on revised MoCoV2[8] self-supervised learning framework for several reasons. (1) The contrastive learning paradigm is more applicable for the objective of distinguish between similar and dissimilar samples [46] and avoids learning overly simple or meaningless feature representations. (2) MoCoV2 introduces a dynamic dictionary queue mechanism that can store a large number of negative samples, allowing training with smaller batch sizes, which is extremely important when hardware capacity is limited. (3) The dictionary queue is conducive to applying hard negative sample mining methods, leaving room for further improving the efficiency of feature learning and the model's discriminative ability.

As for business values, the proposed method can also accelerate examination speed, accuracy, and consistency, helping rights holders better protect their intellectual property. It fully utilizes existing unlabeled trademark data, reducing time consumption while improving retrieval accuracy, significantly enhancing examination efficiency. Therefore, it can accelerate examination speed, accuracy, and consistency, helping rights holders better protect their intellectual property.

The self-supervised learning framework structure, training process, and evaluation process used in this paper's method will firstly be introduce in Section 3.1. Subsequently, we will focus on three key improvements: Section 3.2 discusses data augmentation strategies, Section 3.3 explores the design of loss functions and regularization terms, and Section 3.4 introduces methods for hard negative samples selection. Finally, in Section 3.5, we will elaborate on the design details of the encoder network used in the self-supervised framework.

3.1. Overview of Self-supervised Learning Architecture

The proposed method in this paper uses MoCoV2 as the self-supervised learning framework. As shown in Figure 1, MoCoV2 treats the contrastive learning problem as a dictionary query problem. The core idea is to use the input image as a query and compare it with keys composed of other samples to learn good feature representations.

Framework Components. The framework primarily consists of components such as an encoder, momentum encoder, queue dictionary, hard negative sample selection method, projection head, and contrastive loss function. Among these, the encoder f_q (also known as the query encoder) and the momentum encoder f_k (also known as the key encoder) share the same network architecture, both capable of mapping input trademark images into low-dimensional feature representations. However, these two encoders differ significantly in their parameter update strategies: the query encoder uses standard backpropagation, directly updating parameters based on the gradient of the contrastive loss. In contrast, the key encoder adopts a momentum update mechanism, with its parameter updates as Formula 1,

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{1}$$

where θ_k and θ_q represent the parameters of the momentum encoder and encoder respectively, $m \in [0, 1)$ is the momentum coefficient, which determines the rate at which the momentum encoder parameters follow the encoder updates. This momentum encoder with slowly updating parameters, combined with a continuously updated dictionary queue, ensures the maintenance of a stable and consistent negative sample pool. This allows the encoder to continuously extract contrastive information from the most recently generated negative samples, thereby continuously optimizing its feature representation capability.

Training Process. We use unlabeled trademark images from the METU dataset(see Section 4.1) for training. Each batch of images undergoes two different data augmentations and is input into the encoder and momentum encoder respectively. The anchor sample (also called Query) output by the encoder and the positive sample (also called Positive Key) output by the momentum encoder form a positive sample pair, representing different views of the same trademark. Meanwhile, we select negative samples (also called Negative Keys) from the dictionary queue through a hard negative selection strategy, forming negative sample pairs with the anchor samples, representing differences between different trademarks. These positive and negative sample pairs are input into the contrastive loss function to calculate the loss value, guiding the model to learn to distinguish between similar and dissimilar trademarks. After each training step, the positive samples of the current batch are stored in the dictionary queue as candidates for negative samples in subsequent batches, while updating the dictionary content to ensure the timeliness of negative sample features. Through this process, the model gradually learns to extract essential features of trademarks and accurately distinguish

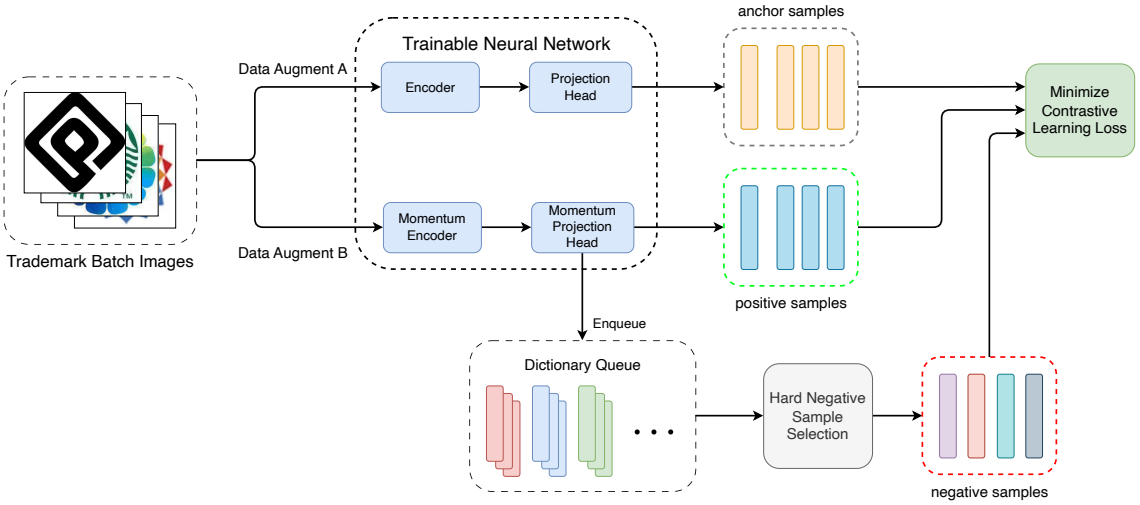


Figure 1: This work is based on an overview of the self-supervised learning framework. The framework includes an encoder, a momentum encoder, a projection head, a dictionary queue, a hard sample selection method, and the InfoNCE loss function. Training data is augmented to obtain different views of the same instance, which are input into the encoder and momentum encoder to obtain anchor samples and positive samples, forming positive sample pairs. Anchor samples and other batch samples from the dictionary queue selected through hard negative sample selection form negative sample pairs. Feature representation is learned through the InfoNCE loss function, and samples from each training batch are stored in the dictionary queue for negative sample selection in subsequent batches.

similarities, thereby improving the accuracy and efficiency of trademark retrieval.

Evaluation Process. After training is completed, we use the encoder in the framework to map trademarks into fixed-length feature vectors. These feature vectors reflect the key elements in the trademark images, and the degree of difference between these feature vectors can reflect the similarity between the original trademark images. We encode all trademarks into feature vectors and store them in a feature pool. When retrieving trademarks, we measure the difference between the feature vector of the query trademark and the features in the pool according to the specified metric method, and sort them in ascending order of difference. We then select the top N as the most similar images to achieve the purpose of trademark retrieval.

3.2. Data Augmentation Strategy

We adopted the data augmentation strategy from the MoCoV2 work and made targeted adjustments based on the characteristics of trademark confusion cases. Given that a significant proportion of trademark similarities stem from rotational changes in graphics, we introduced a new augmentation method on top of the original strategy: a random rotation of -90 to 90 degrees with a probability of 0.15. This improvement aims to enhance the model's robustness to rotational changes, thereby better identifying such similar cases. Additionally, since the images in the METU dataset have been pre-processed and cropped, and the information within the retained area is crucial for determining trademark similarity, we adjusted the random cropping strategy. We increased the minimum scale of the random cropping range relative to the original image from 0.2 to 0.5 to ensure more key information is retained. Through these targeted modifications, our data augmentation strategy not only retains the advantages of MoCoV2 but also better aligns with the specific requirements of the trademark recognition task.

3.3. Loss Function and Regularization Term

MoCoV2 uses InfoNCE loss as the contrastive loss function, which is based on Noise Contrastive Estimation and the principle of mutual information maximization. Its purpose is to maximize the similarity between positive samples while minimizing the similarity between negative samples, thereby training the encoder's feature extraction capability. The formula is as Equation 2,

$$L_{\text{NCE}} = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)} \tag{2}$$

where q represents the query feature, k_+ represents the positive sample feature, k_i represents the negative sample feature, τ is the temperature hyperparameter, and K is the number of negative samples.

To further enhance the training effect, we introduced the CO2 regularization term[45] and the KoLeo regularization term[35].

CO2 Regularization Term. The CO2 regularization term is a consistency loss designed for contrastive learning, aiming to address the "class collision" problem—where

traditional methods tend to treat all negative samples as equally dissimilar, while in reality, some of these samples may have similar semantic content. The core idea of CO2 is to encourage consistent similarity distributions between query samples and positive samples with respect to negative samples.

Specifically, CO2 introduces a bidirectional consistency loss to constrain the following two aspects: the similarity distribution between anchor samples and negative samples, and the similarity distribution between positive samples and negative samples. This bidirectional constraint ensures that the model can more comprehensively understand the relationships between samples during the learning process, thereby improving the quality and discriminative ability of feature representations. The mathematical expression of the CO2 regularization term is as Equation 3.

$$L_{con} = \frac{1}{2}(\mathrm{KL}(P||Q) + \mathrm{KL}(Q||P)) \tag{3}$$

In this expression, $\mathrm{KL}(\cdot)$ is the symmetric Kullback-Leibler (KL) divergence, P and Q represent the similarity distributions of positive samples and query samples with negative samples, respectively, defined as Equation 4.

$$\begin{aligned} P(i) &= \frac{\exp(p \cdot n_i/\tau_{con})}{\sum_k \exp(p \cdot n_k/\tau_{con})} \\ Q(i) &= \frac{\exp(q \cdot n_i/\tau_{con})}{\sum_k \exp(q \cdot n_k/\tau_{con})} \end{aligned} \tag{4}$$

Here, p is the positive sample, q is the query sample, n_i is the i-th negative sample, and τ_{con} is the temperature parameter. By introducing this regularization term, we expect the model to learn more robust and generalized visual representations.

KoLeo regularization term. The purpose of the KoLeo regularization term is to promote uniform distribution of features in a batch within the spherical space, which is considered beneficial for subsequent quantization steps and improves the performance of similarity search in high-dimensional data. It is based on the Kozachenko-Leonenko differential entropy estimator, with Equation 5,

$$L_{\text{KoLeo}} = -\frac{1}{n}\sum_{i=1}^{n}\log(\rho_{n,i}) \tag{5}$$

where $\rho_{n,i}$ is the distance from the i-th sample to its nearest neighbor within a batch.

To balance the impact of the loss function and the regularization term on the learning objective, we set weight parameters λ_a and λ_b for these two regularization terms, as shown in Equation 6,

$$L = L_{\text{NCE}} + \lambda_a \times L_{con} + \lambda_b \times L_{\text{KoLeo}} \tag{6}$$

where the selected value of λ_a is 0.02, and the value of λ_b is 0.05.

3.4. Hard Negative Sample Selection

According to Robinson et al. [33], using too many easy negative samples with low similarity to the query sample in the contrastive learning process can easily lead to degenerate solutions, resulting in model performance degradation. On the other hand, using hard samples with high similarity to the query sample for contrastive learning can not only reduce the possibility of degenerate solutions but also accelerate training and make the model converge more easily. Therefore, we introduced a hard negative sample selection method to the dictionary queue to filter out hard negative samples from the negative sample pool and exclude the influence of easy negative samples.

Specifically, inspired by the method proposed by Zhu et al. [51], we use an online negative sample selection method to evaluate the similarity between query samples and negative samples in the dictionary queue during training. Unlike the L2 distance used in their work, we believe that using cosine similarity to measure the similarity between samples better aligns with the sensitivity to vector direction in CBIR, and offers better computational efficiency when dealing with high-dimensional feature vectors, making it suitable for large-scale trademark datasets. When selecting negative samples, we calculate the cosine similarity between the anchor sample and negative samples, excluding negative samples with similarity below a specified threshold and retaining only the remaining negative samples for subsequent optimization. In our experiments, we set the threshold to 0.4. As shown in the ablation experiment results on the bottom side of Figure 6, this simple online hard negative sample selection method significantly improved the model's performance.

3.5. Encoders

The encoder is primarily consist of the feature extraction network (extractor) and the feature fusion network, as shown in Figure 2.

Feature Extraction Network

The feature extraction network is used to extract key features from trademark images, typically built on pre-trained off-the-shelf networks. To facilitate fair comparison with previous work, we made improvements on the ResNet50 backbone to fully leverage the effects of self-supervised learning. Starting from trademark similarity confusion cases, we believe that the basis for most similarity judgments lies in the fine-grained feature elements in images, such as subtle differences in graphic shape details and texture colors. This means that the feature vectors extracted by the encoder need to reflect these fine-grained feature elements. For the feature extractor, it should be able to focus on the most distinctive features while suppressing the impact of noise. Although traditional convolutional neural net-

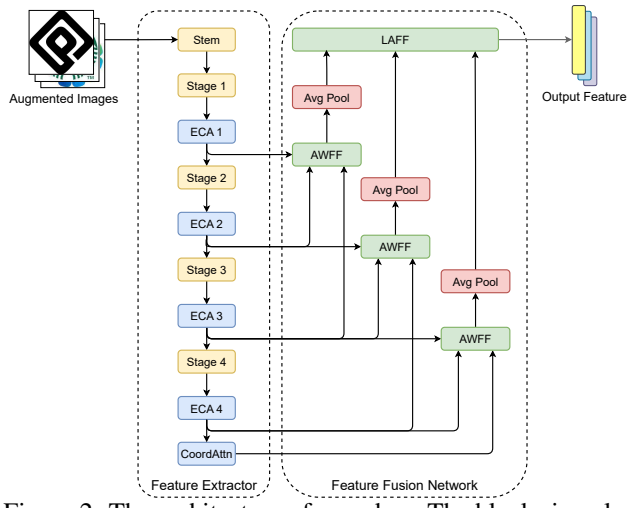


Figure 2: The architecture of encoders. The blocks in color of blue are attention-based, ahd those in color of green are for feature fusion.

works perform well in image classification tasks, they may have limitations in handling such fine-grained visual differences. To address this, we introduce two attention modules into the original network architecture: the ECA (Efficient Channel Attention) Module and the CA (Coordinate Attention) Module.

ECA Module. In trademark retrieval tasks, the specific color schemes and texture patterns contained in images are key elements for determining similarity. We believe that this crucial information is distributed across different channels of the feature map. To effectively capture and utilize this information, we introduce a channel attention mechanism to integrate cross-channel information and adjust the importance of each channel, thereby highlighting key feature elements in trademark images.

Among the various channel attention methods, we chose the ECA (Efficient Channel Attention) module proposed by Wang et al. [44] to improve our backbone network. ECA is a lightweight channel attention mechanism that, compared to the classic channel attention network SENet, avoids dimensionality reduction, significantly reduces computational complexity while maintaining the same performance, and can adaptively adjust the kernel size of the field of view. Its structure is shown in Figure 3. After the feature map input, it first undergoes Global Average Pooling (GAP) to capture global information of the entire feature map. Then, the resulting 1×1 scale feature is input into a 1D convolutional layer to model the interdependencies between channels. The output of the convolution is then input into a Sigmoid activation function to normalize the output between 0 and 1, thus obtaining channel weights. The channel weights are multiplied channel-wise with the original input feature map, and finally, the enhanced feature map is output.

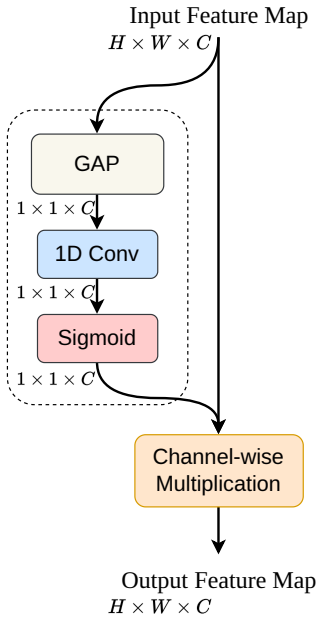


Figure 3: ECA module.

The cross-channel information interaction range of ECA is constrained by the convolution kernel size in 1D convolution layer. According to the findings of related work on grouped convolution [21, 47, 49], under the premise of a fixed number of groups, high-dimensional channels tend to benefit from larger convolution receptive fields, while low-dimensional channels are more suitable for smaller receptive fields. This observation inspired ECA to construct a function mapping from the number of channel features to the convolution kernel size, with the specific formula being Equation 7,

$$k = \psi(C) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{\text{odd}} \tag{7}$$

where k is the size of the convolution kernel, C is the number of feature channels, γ and b are parameters set to 2 and 1 respectively, and $|t|_{odd}$ represents the nearest odd number to t. In our experiments, we added the ECA module after each Stage of the ResNet50 backbone, and calculated the size of the 1D convolution kernel according to the output channel number of each Stage to enhance the network's perception of color and texture. We believe that in the lower Stages, the ECA module assists in enhancing basic texture and edge information of the image; while in the higher Stages, the ECA module focuses on capturing complex shape and semantic information.

CA Module. We also considered the degree of information expression in different spatial positions of trademark features and introduced Coordinate Attention (CA) from Hou et al. [18] at the end of the backbone network. The CA module focuses on capturing long-range spatial depen-

dencies and position information in feature maps, which effectively complements the ECA module that mainly focuses on inter-channel relationships. In the trademark retrieval task, this complementarity manifests as: the ECA module enhances the model's perception of specific visual elements in trademarks (such as colors, textures), while the CA module helps the model understand the spatial arrangement and global structure of these elements within the entire trademark. The CA module performs spatial feature modeling along the horizontal and vertical directions of the image to generate position-sensitive attention maps. Specifically, given an input feature map $X \in \mathbb{R}^{C\times H\times W}$, the CA module first generates feature descriptors z_h and z_w for the two directions through pooling operations along the horizontal and vertical directions as Equation 8.

$$\begin{aligned} z_h &= \text{Pool}_h(X) \in \mathbb{R}^{C\times H\times 1} \\ z_w &= \text{Pool}_w(X) \in \mathbb{R}^{C\times 1\times W} \end{aligned} \tag{8}$$

these descriptors representing the feature distribution in horizontal and vertical directions are processed through a series of convolutions and activation operations to obtain attention weights a_h and a_w for horizontal and vertical directions as Equation 9.

$$\begin{aligned} a_h &= \sigma(f([z_h; z_w])) \in \mathbb{R}^{C\times H\times 1} \\ a_w &= \sigma(g([z_h; z_w])) \in \mathbb{R}^{C\times 1\times W} \end{aligned} \tag{9}$$

Here, f and g are 1D convolution mapping functions, σ represents the Sigmoid activation function, and $[\cdot;\cdot]$ denotes feature concatenation. Finally, the output Y of the CA module is obtained by multiplying the original feature map with the generated attention weights as Equation 10.

$$Y = X \cdot a_h \cdot a_w \tag{10}$$

As shown in the ablation experiment table on the right side of Figure 4.4, by incorporating these two attention mechanisms, the model's retrieval evaluation metric performance has significantly improved, while the overall computational overhead has not increased substantially. This demonstrates the effectiveness of improving the model through lightweight attention mechanisms.

Feature Fusion Network

Considering that trademark design typically includes multi-scale and multi-level visual elements, ranging from simple local shape textures to advanced global semantic information, we believe that a single-level feature representation is insufficient to comprehensively capture the key elements of trademarks. Therefore, we constructed a feature fusion network that receives output features from ECA at various stages and the CA module at the end of the backbone network to integrate information from different scales and abstraction levels, forming an effective feature vector. Our feature fusion network is consist of AWFF and LAFF.

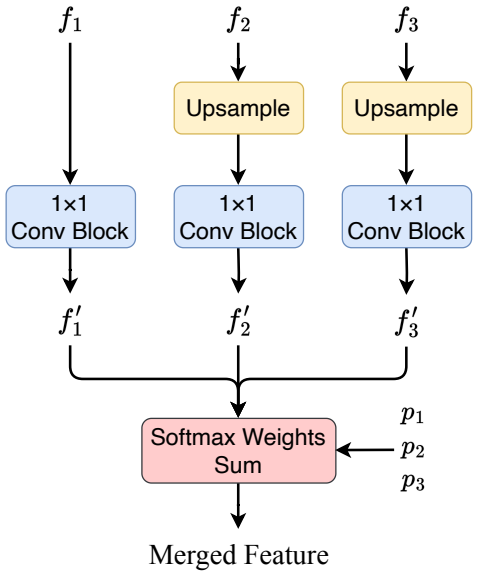


Figure 4: AWFF module structure.

Adaptive Weighted Feature Fusion (AWFF) Block. We introduce a module called AWFF for preliminary fusion of features from different levels. The core idea is to adaptively fuse multi-scale features through learnable weight parameters. As shown in Figure 4, given features f_1, f_2, f_3 from stages, we use the large-scale feature map f_1 from the lowest layer as a reference and upsample the other two feature maps to align their scales. Then, we input these three feature maps into a 1D convolutional layer to adjust their channel numbers, followed by batch normalization and ReLU activation functions, obtaining f_1', f_2', f_3' respectively. For feature fusion, considering computational complexity, we do not use self-attention or cross-attention mechanisms with numerous parameters. Instead, we use three learnable parameters normalized by Softmax as feature attention weights for weighted fusion of the processed features, as shown in Equation 11,

$$O = h(p_1) * f_1' + h(p_2) * f_2' + h(p_3) * f_3' \tag{11}$$

where p_i is a learnable parameter, $h(\cdot)$ is the Softmax function, and f_1', f_2', f_3' are the features of the three inputs after upsampling, 1D convolution, batch normalization, and activation function processing. In our experiments, we input the output features of the four stages of the feature extraction network into this module in groups of three adjacent features in sequence, setting the output feature channel numbers to 256, 512, and 1024. AWFF adaptively adjusts the importance of each feature map, and the ablation experiment results in the table show that AWFF significantly

improves the performance of various metrics with only a small cost.

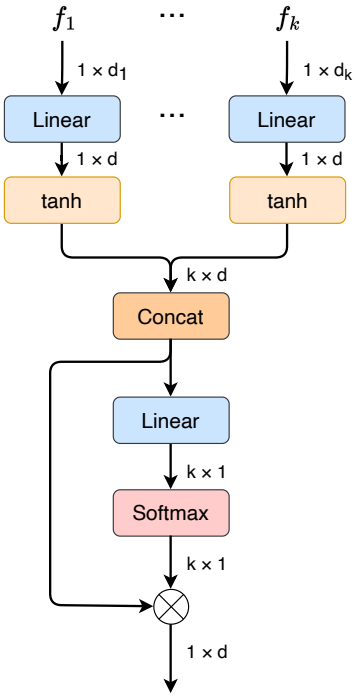


Figure 5: LAFF module structure.

Lightweight Attention Feature Fusion (LAFF) Block. We use the LAFF module proposed by Hu et al. [20] to fuse the feature branches preliminarily fused by AWFF into the final feature vector. Compared to the commonly used multi-head self-attention (MHSA), LAFF is not only simpler and has fewer parameters, but also demonstrates better fusion effects when facing features with high heterogeneity. Its structure is shown in Figure 5. Given the input 1D features $\{f_1, f_2, ..., f_k\}$, where $f_i \in \mathbb{R}^{1\times d_i}$, they are adjusted through a linear layer to dimensions of $1 \times d$, where d is the specified number of hidden layer neurons. Then, the tanh activation function is used to map the feature range to [-1, 1] and add nonlinear expression. These features are concatenated to form a $k \times d$ dimensional feature input to a $d \times 1$ linear layer, and a $k \times 1$ weight vector is generated through Softmax. Finally, this weight vector is element-wise multiplied with the concatenated features to obtain the final fused feature. In our experiments, we obtain one-dimensional features by applying average pooling with a target scale of 1 to the output features of AWFF, and input them into the LAFF module with $d = 512$ hidden layer neurons to fuse and obtain the final vector. Ablation experiments show that the LAFF module significantly improves evaluation metrics, proving its effectiveness.

4. Experimental Results

This chapter provides a detailed description of the experimental evaluation process for our proposed method. We designed a series of comprehensive experiments to validate the effectiveness and superiority of our approach. In Section 4.1, we introduce the datasets and evaluation metrics used, providing a foundation for interpreting subsequent experimental results. Section 4.2 elaborates on the network parameter settings and training conditions, ensuring the reproducibility of the experiments. In Section 4.3, we present the experimental results of our proposed method on evaluation metrics and conduct a comparative analysis with existing state-of-the-art methods to highlight the advantages of our approach. To gain an in-depth understanding of the contributions of each component in our proposed method, Section 4.4 presents detailed ablation experiments. Section 4.5 visually demonstrates the practical performance of our method through qualitative analysis.

4.1. Evaluation on Datasets

Dataset. Our work utilized two datasets, namely CNT (China Trademark) and METUv2.

CNT (China Trademark) is a trademark dataset we constructed that includes similarity information annotations. It is used to implement pre-training of the encoder to stabilize the subsequent self-supervised training process. We use web crawlers to collect 80,000 trademark review documents with a total of 252,000 images from the China Trademark website. We first filtered out trademark image pairs cited in documents judged as similar from these review documents, then removed invalid trademarks that were damaged or lacked similar trademark citations. Next, we eliminated duplicate trademarks by calculating file hash values and merged their similar trademark citation images. Finally, we manually filtered out trademarks containing only text elements from these trademark images and cropped the remaining trademarks to remove excess white edges and highlight the main content. Ultimately, we obtained 14,715 valid trademark data, which were categorized into 3,734 similarity sets, each represented by a numerical code. The correspondence between images and sets is recorded in a table. It is worth noting that pre-training on CNT has minimal impact on the final evaluation metrics.

We use METUv2 dataset for subsequent self-supervised training and retrieval effect evaluation. METUv2 dataset was constructed by Tursun et al. [42], which has 923,343 trademarks. The dataset is consist of 922,926 unlabeled images and 417 images with category labels as the query set. The unlabeled dataset is used for self-supervised training, while the labeled query set is categorized into 35 groups based on similarity. In each group of the query set, there are 10 to 14 images, which are similar to each other based on domain experts' opinions. The query set is used to evaluate

the retrieval model's performance.

Evaluation Metrics. We adopted the same evaluation metrics as in Tursun's work [42], namely mAP@k (mean Average Precision at k) and NAR (Normalized Average Rank). mAP effectively combines the precision and recall of the retrieval system, and its formula is shown as Equation 12,

$$\text{mAP@}k = \frac{1}{|Q|} \sum_{q \in Q} \text{AP@}k(q) \tag{12}$$

where Q is the size of the query set, $AP@k$ is the Average Precision@k for each query q.

$AP@k$ is shown as Equation 13,

$$\text{AP@}k(q) = \frac{\sum_{i=1}^{k} P(i) \times rel(i)}{\min(m, k)} \tag{13}$$

wherein $P(i)$ represents the precision at the i-th position (Precision at i), equal to the number of relevant images in the top i results divided by i. In this formula, rel(i) is a binary function that equals 1 if the item at rank i is relevant, and 0 otherwise. This helps to only count the precision for relevant items. The parameter k represents the number of top results considered in the evaluation.

NAR represents the average ranking of the retrieved image among all images, used to evaluate the overall ranking quality of the model, shown as Equation 14,

$$\text{NAR} = \frac{1}{N \cdot N_{rel}} \left(\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel}+1)}{2} \right) \tag{14}$$

where N represents the size of the entire dataset, N_{rel} is the number of relevant images with the same label as the image to be retrieved, and R_i is the similarity ranking of these relevant images. Regarding the significance of evaluation metrics, we generally consider that the higher the value of mAP@k, the better the retrieval performance of the model's top k results, and the smaller the value of NAR, the better the model's overall data retrieval performance.

4.2. Training-Evaluation Protocol

The training process of the proposed method is consist of two phases in chronological order: the encoder pre-training phase and the self-supervised training phase.

Pre-training phase. In this phase, we fine-tune the model using the CNT dataset and employ CosFaceLoss[43] for training. For parameters, we scale the image size to 160×160, set the learning rate to 4e-4, total learning epochs to 40, and batch size to 64.

Self-supervised Training Phase. In this phase, we use the METUv2 dataset for self-supervised learning on the MoCoV2 framework, employing the InfoNCE loss function with regularization term for training. For parameters, we similarly scale the images to 160×160, set the learning rate to 2.5e-4, and configure a total of 30 learning epochs with a batch size of 256. The MoCoV2 dictionary queue length is set to 65536, and the output feature dimension of the Projection Head is set to 128.

Throughout the training process, we employ the low-cost CAME[29] as the optimizer. The learning rate scheduling strategy combines warm-up and cosine annealing as proposed by Loshchilov et al. [27] Specifically, during the initial 5 epochs of training, the learning rate gradually increases to the set initial value, a process known as warm-up. In the subsequent training phase, the learning rate gradually decreases following the shape of a cosine function, a process known cosine annealing. This strategy allows the model to maintain stability in the early stages of training and gradually fine-tune in the later stages to achieve optimal results. Notably, during the self-supervised training phase, we enable the restart strategy: every 10 epochs constitute one cosine cycle, and at the beginning of each new cycle, the learning rate is reset to 0.9 times that of the previous cycle. Due to hardware limitations, we adopt BF16 mixed-precision training, which allows us to maximize the number of images processed per batch while ensuring training quality.

Evaluation Phase. To apply the trained models on new samples, following the proposal in [7], we set the batch size to 1 and remove the Projection Head, which is unnecessary for the evaluation phase, using only the output of the last layer of the encoder as the trademark feature vector. We enable PCA whitening as a post-processing step for evaluation data to reduce redundancy between channels, using 20,000 randomly sampled images from METUv2 to learn PCA, selecting 256 principal components to retain. Faiss was used to calculate and sort the distances of similar feature vectors based on the FlatIP index. Details of our evaluation results are presented in Table 1 and Figure 6.

4.3. Comparative Analysis

We compare our proposed method based on self-supervised learning and hard sample selection with recent state-of-the-art trademark retrieval methods on the METU trademark dataset, as shown in Table 1. Inspired by the work of [41], we categorize these methods into four types: handcrafted features, fine-tuning pre-trained deep features, pre-training single-pass, and self-supervised learning. Our method belongs to the last category and achieves state-of-the-art results in NAR and MAP@100. Additionally, our method maintains a relatively low feature dimension of 256, which is consistent with previous SOTA methods.

Method	DIM ↓	NAR ↓	MAP@100 ↑
hand-crafted features			
Feng *et al.* [13]	6,224	0.083	-
Tursun *et al.* [39]	10,000	0.062	-
fine-tuned off-the-shelf deep features			
Perez *et al.* (vis) [30]	4,096	0.066	-
Perez *et al.* (con) [30]	4,096	0.063	-
Perez *et al.* (vis, con) [30]	4,096	0.047	-
Yavuz *et al.* (Smooth-AP Loss, S.Color)	512	0.040	-
pre-trained single pass [50, 24]			
SPoC [2, 40]	256	0.120	18.7
CRoW [23, 40]	256	0.140	19.8
R-MAC [38]	256	0.072	24.8
MAC [38, 40]	512	0.120	21.5
Jimenez [22, 40]	256	0.093	21.0
CAM MAC [40]	256	0.064	22.3
ATR MAC [40]	512	0.056	24.9
ATR R-MAC [40]	256	0.063	25.7
ATR CAM MAC [40]	512	0.040	25.1
MR-R-SMAC w/UAR [41]	256	0.028	31.0
self-supervised learning			
Cao *et al.* [4]	128	0.051	-
Ours	256	**0.025**	**37.4**

Table 1: Our method compared to related work.

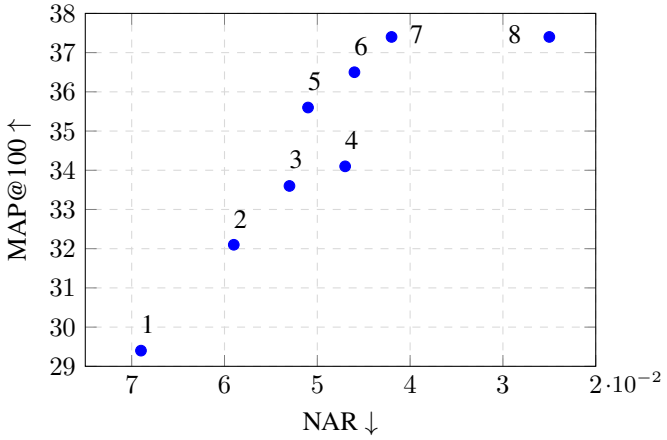
4.4. Ablation Study

To gain a deeper understanding of the importance and contribution of each component in our proposed method, we conducted a series of ablation experiments. These experiments aimed to decompose the components used in the method, set up different component combination cases, and sequentially evaluate the impact of each component on overall performance to validate our design choices. We performed all ablation experiments on METUv2, using the same training and testing protocols as in the main experiments to maintain consistency. Evaluation metrics included Normalized Average Rank (NAR) and Mean Average Precision at 100 (MAP@100).

We divide the proposed method into the following components: Self-Supervised Learning (SSL), Hard Negative Sample Selection (HNSS), Regularization Terms (Reg), Efficient Channel Attention (ECA), Coordinate Attention (CA), Adaptive Weighted Feature Fusion (AWFF), LightWeight Atentional Feature Fusion (LAFF), and PCA Whitening (pcaW). We observe performance changes by progressively adding these components. Figure 6 shows the detailed results of the ablation experiments.

From the experimental results, we can clearly see the contribution of each component to performance. From the experimental results, we can clearly see the contribution of each component to performance.

To begin with, using only self-supervised learning (SSL) as the baseline model, we achieved a MAP@100 of 29.4 and a NAR of 0.069. After introducing hard negative sample selection (HNSS) method, the performance significantly improved, which indicates that emphasizing challenging samples can bring substantial improvements to the effectiveness of self-supervised learning. With the addition of regularization (Reg), the model's generalization ability was enhanced, reflecting the important role of feature consistency and distribution uniformity in contrastive learning tasks.



Case	Components								NAR ↓	MAP@100 ↑
	SSL	HNSS	Reg	ECA	CA	AWFF	LAFF	pcaW		
1	✓								0.069	29.4
2	✓	✓							0.059	32.1
3	✓	✓	✓						0.053	33.6
4	✓	✓	✓	✓					0.047	34.1
5	✓	✓	✓	✓	✓				0.051	35.6
6	✓	✓	✓	✓	✓	✓			0.046	36.5
7	✓	✓	✓	✓	✓	✓	✓		0.042	37.4
8	✓	✓	✓	✓	✓	✓	✓	✓	**0.025**	**37.4**

Figure 6: Ablation experiments for various components of the method proposed in this paper. The case ID is correspondent between the scatter plot and the table. In the plot, the combination of lower NAR (right) and higher MAP@100 (upper) means better performance.

Furthermore, the improvement by introducing efficient channel attention (ECA) mechanism proves the significance of channel attention on enhancing trademark image feature extraction. The coordinate attention (CA) caused a slight increase in NAR but brought a significant improvement to MAP@100, indicating the important impact of spatial attention on trademark retrieval effectiveness. AWFF and LAFF further improved performance, highlighting their advantages in feature fusion.

Finally, using PCA whitening (pcaW) significantly improved NAR performance without reducing MAP@100 performance, which means the critical role of PCA whitening in removing feature redundancy to improve dimensional effectiveness.

4.5. Qualitative Results

To intuitively demonstrate the effectiveness of our proposed method, we selected three typical query samples and

Table 2: Retrieval effect diagram of sample query images from the METU dataset. The leftmost column shows the sample query images, and the right side displays the rank and similarity of images of the same category as the query retrieved by the method in this paper.

Query	Same Category Samples									
										
Rank	1	2	3	5	8	11	18	34	89	201
Sim	0.997	0.864	0.777	0.755	0.741	0.719	0.716	0.695	0.666	0.631
										
Rank	1	2	3	4	6	9	13	44	82	184
Sim	0.972	0.883	0.844	0.772	0.682	0.678	0.675	0.660	0.636	0.620
										
Rank	1	2	4	7	42	57	231	655	1102	1586
Sim	0.928	0.770	0.747	0.717	0.658	0.642	0.534	0.505	0.489	0.453

Table 3: Query sample compare with category-inconsistent images

Query	Category-inconsistent Images		
			
Rank	3	34	40

presented the retrieval results obtained using our method, as shown in Table 2. In the table, the leftmost column shows three trademark query samples from the METU query set, while the right side displays 10 related samples of the same category retrieved from the dataset, along with their similarity rankings and normalized cosine similarities relative to the query samples.

From the results, we can observe several important informations: These related samples of the same category as the query samples are ranked high and are relatively concentrated within the top 100 rankings, which proves the broad applicability of our method in trademark retrieval for different elements. Secondly, the retrieval results in the second row indicate that our method is insensitive to changes in text content. This means that our method can selectively identify and focus on key element areas while suppressing text content with minor contributions [40]. In addition, as shown in the Table 3, even when the retrieval results in the third row are relatively less satisfactory, the category-inconsistent images ranked relatively high still exhibit significant visual similarity to the query samples. This phenomenon is particularly evident in the retrieval results of the third row, fully proving that our model can effectively capture the core visual features of trademarks.

These qualitative results strongly demonstrate the effectiveness and practicality of the method proposed in this paper. Our method not only accurately identifies similar trademarks but also exhibits the ability to handle complex visual features, providing a powerful solution for trademark retrieval and similarity analysis tasks.

5. Conclusion

This paper proposes a large-scale trademark retrieval method using self-supervised learning to address the lack of annotated training data. By leveraging self-supervised contrastive learning on unannotated trademark data, our approach enables the encoder to learn essential feature representations and improve retrieval performance.

Our key contribution is an improved MoCoV2 framework with a hard negative sample selection strategy, which boosts model representability and mitigates degenerate solutions. We further optimize the encoder by integrating lightweight attention mechanisms (ECA, CA) and multi-scale feature fusion techniques (AWFF, LAFF), enabling better focus on cross-channel and spatial information. Experimental results on the METUv2 dataset demonstrate state-of-the-art performance, with NAR of 0.025 and mAP@100 of 37.4. Ablation studies validate the effectiveness of each component.

Our method benefits intellectual property offices by improving trademark application processing efficiency and ac-

curacy, while helping applicants better understand existing trademarks to reduce rejection risks. It also supports market regulation by identifying potential infringement.

Despite the significant progress, there are still potential for future improvement. Future work is expected to proceed in three aspects as follows.

(1) **Feature extraction network** is expected to have futher improvements, mainly by using more powerful backbone networks such as Vision Transformer and EfficientNetV2, and applying model compression and knowledge distillation techniques to reduce the number of model parameters and computational complexity. This will help further enhance the model's performance and efficiency, making it more suitable for large-scale trademark retrieval tasks. (2) **Feature fusion techniques** still has potential to be revised. Although the current AWFF and LAFF modules have achieved good results, there is still room for optimization. We will explore more advanced feature fusion methods, such as dynamic feature fusion or attention-guided fusion mechanisms, to better integrate multi-scale and multi-level feature information. (3) **Ensembling our method with other modalities**, such as text descriptions or trademark metadata, to achieve a more comprehensive and robust trademark retrieval method may be useful. Multimodal fusion will bring additional semantic information to trademark retrieval, helping address cases that are difficult to distinguish based solely on visual features.

References

[1] C. Aker, O. Tursun, and S. Kalkan. Analyzing deep features for trademark retrieval. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2017. 2

[2] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015. 2, 11

[3] A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 4

[4] J. Cao, Y. Huang, Q. Dai, and W.-K. Ling. Unsupervised trademark retrieval method based on attention mechanism. *Sensors*, 21(5):1894, 2021. 3, 11

[5] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 3

[6] K. Chatfield, J. Philbin, and A. Zisserman. Efficient retrieval of deformable shape classes using local self-similarities. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 264–271. IEEE, 2009. 2

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3, 10

[8] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 4

[9] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 3

[10] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 3

[11] G. Ciocca and R. Schettini. Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recognition*, 34(8):1639–1655, 2001. 2

[12] J. P. Eakins, J. D. Edwards, K. J. Riley, and P. L. Rosin. Comparison of the effectiveness of alternative feature sets in shape retrieval of multicomponent images. In *Storage and retrieval for media databases 2001*, volume 4315, pages 196–207. SPIE, 2001. 2

[13] Y. Feng, C. Shi, C. Qi, J. Xu, B. Xiao, and C. Wang. Aggregation of reversal invariant features from edge images for large-scale trademark retrieval. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, pages 384–388. IEEE, 2018. 11

[14] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 3

[15] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3

[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2, 3

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[18] Q. Hou, D. Zhou, and J. Feng. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13713–13722, 2021. 7

[19] S. Hsieh and K.-C. Fan. Multiple classifiers for color flag and trademark image retrieval. *IEEE Transactions on image processing*, 10(6):938–950, 2001. 2

[20] F. Hu, A. Chen, Z. Wang, F. Zhou, J. Dong, and X. Li. Lightweight attentional feature fusion: A new baseline for text-to-video retrieval. In *European conference on computer vision*, pages 444–461. Springer, 2022. 9

[21] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi. Deep roots: Improving cnn efficiency with hierarchical filter

groups. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1231–1240, 2017. 7

[22] A. Jimenez, J. M. Alvarez, and X. Giro-i Nieto. Class-weighted convolutional features for visual instance search. *arXiv preprint arXiv:1707.02581*, 2017. 11

[23] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I 14*, pages 685–701. Springer, 2016. 2, 11

[24] J. Kim and S.-E. Yoon. Regional attention based deep feature for image retrieval. In *BMVC*, page 209, 2018. 11

[25] T. Lan, X. Feng, L. Li, and Z. Xia. Similar trademark image retrieval based on convolutional neural network and constraint theory. In *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2018. 3

[26] T. Lan, X. Feng, Z. Xia, S. Pan, and J. Peng. Similar trademark image retrieval integrating lbp and convolutional neural network. In *Image and Graphics: 9th International Conference, ICIG 2017, Shanghai, China, September 13-15, 2017, Revised Selected Papers, Part III 9*, pages 231–242. Springer, 2017. 2

[27] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 10

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 2

[29] Y. Luo, X. Ren, Z. Zheng, Z. Jiang, X. Jiang, and Y. You. Came: Confidence-guided adaptive memory efficient optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4442–4453, 2023. 10

[30] C. A. Perez, P. A. Estévez, F. J. Galdames, D. A. Schulz, J. P. Perez, D. Bastías, and D. R. Vilar. Trademark image retrieval using a combination of deep convolutional neural networks. In *2018 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2018. 3, 11

[31] R. Phan and D. Androutsos. Content-based retrieval of logo and trademarks in unconstrained color image databases using color edge gradient co-occurrence histograms. *Computer Vision and Image Understanding*, 114(1):66–84, 2010. 2

[32] V. Rani, S. T. Nabi, M. Kumar, A. Mittal, and K. Kumar. Self-supervised learning: A succinct review. *Archives of Computational Methods in Engineering*, 30(4):2761–2775, 2023. 3

[33] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020. 6

[34] M. Rusinol, D. Aldavert, D. Karatzas, R. Toledo, and J. Lladós. Interactive trademark image retrieval by fusing semantic and visual content. In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*, pages 314–325. Springer, 2011. 2

[35] A. Sablayrolles, M. Douze, C. Schmid, and H. J'egou. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018. 5

[36] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1482–1491, 2017. 2

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[38] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015. 2, 11

[39] O. Tursun, C. Aker, and S. Kalkan. A large-scale dataset and benchmark for similar trademark retrieval. *arXiv preprint arXiv:1701.05766*, 2017. 11

[40] O. Tursun, S. Denman, S. Sivapalan, S. Sridharan, C. Fookes, and S. Mau. Component-based attention for large-scale trademark retrieval. *IEEE Transactions on Information Forensics and Security*, 17:2350–2363, 2019. 2, 11, 12

[41] O. Tursun, S. Denman, S. Sridharan, and C. Fookes. Learning regional attention over multi-resolution deep convolutional features for trademark retrieval. In *2021 IEEE international conference on image processing (ICIP)*, pages 2393–2397. IEEE, 2021. 10, 11

[42] O. Tursun and K. Sinan. A challenging big dataset for benchmarking trademark retrieval. In *IAPR Conference on Machine Vision and Applications*, page 28, 2015. 9, 10

[43] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018. 10

[44] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020. 7

[45] C. Wei, H. Wang, W. Shen, and A. Yuille. Co2: Consistent contrast for unsupervised visual representation learning. *arXiv preprint arXiv:2010.02217*, 2020. 5

[46] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 2, 3, 4

[47] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 7

[48] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021. 4

[49] T. Zhang, G.-J. Qi, B. Xiao, and J. Wang. Interleaved group convolutions. In *Proceedings of the IEEE international conference on computer vision*, pages 4373–4382, 2017. 7

[50] L. Zheng, Y. Yang, and Q. Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2017. 11

[51] W. Zhu, J. Liu, and Y. Huang. Hnssl: Hard negative-based self-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4778–4787, 2023. 6