Enhanced Image Retrieval Using Multiscale Deep Feature Fusion in Supervised Hashing

Amina Belalia , Kamel Belloulata * , Adil Redaoui

Posted Date: 19 November 2024

doi: 10.20944/preprints202411.1422.v1

Keywords: content-based image retrieval; Hashing code; deep learning; multiscale feature extract; deep supervised hashing

Article

Enhanced Image Retrieval Using Multiscale Deep Feature Fusion in Supervised Hashing

Amina Belalia [1], Kamel Belloulata [2,*] and Adil Redaoui [2]

[1] School of Computer Sciences, Sidi bel Abbes 22000, Algeria; a.belalia@esi-sba.dz
[2] RCAM Laboratory, Telecommunications Department, Sidi Bel Abbes University, Sidi bel Abbes 22000, Algeria; kamel.belloulata@univ-sba.dz
* Correspondence: kamelbelloulata99@gmail.com ; Tel.: +213-7-7371-5910

Abstract: In recent years, deep network-based hashing has emerged as a prominent technique, especially within image retrieval by generating compact and efficient binary representations. However, many existing methods tend to solely focus on extracting semantic information from the final layer, neglecting valuable structural details that encode crucial semantic information. As structural information plays a pivotal role in capturing spatial relationships within images, we propose the enhanced image retrieval using Multiscale Deep Feature Fusion in Supervised Hashing (MDFF-SH), a novel approach that leverages multiscale feature fusion for supervised hashing. The balance between structural information and image retrieval accuracy is pivotal in image hashing and retrieval. Striking this balance ensures both precise retrieval outcomes and meaningful depiction of image structure. Our method leverages multiscale features from multiple convolutional layers, synthesizing them to create robust representations conducive to efficient image retrieval. By combining features from multiple convolutional layers, MDFF-SH captures both local structural information and global semantic context, leading to more robust and accurate image representations. Our model significantly improves retrieval accuracy, achieving higher Mean Average Precision (MAP) than current leading methods on benchmark datasets such as CIFAR-10, NUS-WIDE and MS-COCO with observed gains of 9.5%, 5% and 11.5%, respectively. This study highlights the effectiveness of multiscale feature fusion for high-precision image retrieval.

Keywords: content-based image retrieval; Hashing code; deep learning; multiscale feature extract; deep supervised hashing

1. Introduction

The surge in high-dimensional multimedia data, driven by advancements in computer networks and social media platforms, underscores the need for efficient storage and retrieval solutions [1–3]. Approximate Nearest Neighbor (ANN) search [4] has emerged as a pivotal area of study in computer vision and information retrieval, a technique essential for reducing storage requirements and improving search efficiency in high-dimensional spaces. Hashing [5] has garnered considerable attention as a potent strategy within ANN search, which transforms high-dimensional data into compact binary codes while preserving spatial relationships between data points. Deep hashing techniques [6–9] have been devised to concurrently learn visual features and binary hash codes, enriching encoded information with semantic context. Deep hashing approaches have emerged as powerful tools for simultaneous feature learning and binary code generation. Unlike conventional hashing methods that rely on independently trained hash functions and quantization algorithms, deep hashing techniques adopt an end-to-end framework to extract semantic representations and construct binary hash codes. These end-to-end frameworks enable the cohesive construction of semantic representations and binary hash codes, surpassing traditional hashing techniques that rely on separate hash functions and quantization steps. While recent advancements in deep hashing have shown promise in information retrieval [10–14], there remains a need for further improvements, particularly in retaining local structural details within images. Most deep hashing approaches emphasize high-level features

 doi:10.20944/preprints202411.1422.v1

from fully connected (FC) layers [15–17], leading to a dearth of local feature information due to the global nature of these representations. Integrating multi-level features that capture both local and global details has shown promise for enhancing retrieval accuracy [18–20]. Despite attempts to fuse multi-feature representations, existing methods often fall short of achieving true end-to-end compatibility between feature representation and binary hash coding.

To address these limitations and to effectively harness the complementary nature of deep multi-scale features, this paper introduces Multiscale Deep Feature Fusion for High-Precision Image Retrieval through Supervised Hashing (MDFF-SH). Leveraging ResNet50 [21], convolutional multi-scale features are aggregated from images of varying sizes and fused within corresponding convolutional layers to yield robust representations. Inspired by the feature pyramid network [22], this fusion process incorporates top-down pathways and lateral connections, enabling exploration of both top-layer semantic and bottom-layer spatial features (Fig.1). Moreover, the MDFF-SH adapts hashing results based on different scale features, enhancing retrieval recall without sacrificing precision. In summary, the key contributions of this paper are as follows:

1. **Dual-Scale Approach:** We propose a dual-scale approach that considers both feature and image sizes to preserve semantic and spatial details. Moreover that compensates for the loss of high-level features and ensures the generated hash codes are more discriminative and informative.
2. **Multi-Scale Feature Fusion:** MDFF-SH learns hash codes across multiple feature scales and fuses them to generate final binary codes, enhancing retrieval performance.
3. **End-to-End Learning:** Our MDFF-SH model integrates joint optimization for feature representation and binary code learning within a unified deep framework.
4. **Superior Performance:** Extensive experiments on three well-known datasets demonstrate that MDFF-SH surpasses state-of-the-art approaches in retrieval performance.

2. Related works

Hashing techniques have gained significant popularity in image retrieval due to their minimal storage requirements and fast processing capabilities [23], [24]. The primary purpose of hashing is to map high-dimensional data into low-dimensional hash codes, ensuring that similar data points have minimal Hamming distances while dissimilar points have maximized distances.

Hashing methods are categorized into supervised [25], [26] and unsupervised [27–31] approaches, based on the use of labeled data. Unsupervised hashing methods [32–35] focus on learning hash functions using unlabeled training samples to transform input images into binary codes. Locality-Sensitive Hashing (LSH) [36] is among the most well-known unsupervised methods, with other significant approaches like Spectral Hashing (SH) [33] and Iterative Quantization (ITQ) [34] being prominent in the field.

In contrast, supervised hashing techniques leverage labeled data to learn hash functions, often yielding higher accuracy compared to unsupervised methods. Supervised Hashing with Kernels (KSH) [37] is notable for employing kernel methods to create nonlinear hash functions. Minimal Loss Hashing (MLH) [38] uses structured SVMs to define an objective for learning hash functions. Supervised Discrete Hashing (SDH) [28] refines the objective function to produce high-quality hash codes without relaxation.

The emergence of deep neural networks has propelled the development of deep hashing algorithms [12,16,39–44], which outperform traditional methods by using rich feature representations. Pairwise and triplet-based similarity preservation are common strategies in these methods to utilize label information. CNN-based Hashing (CNNH) [16] extracts features using CNNs but separates feature learning from hash function training, limiting feedback integration. Deep Pairwise-supervised Hashing (DPSH) [12] uses a Bayesian approach to model relationships between pairwise labels and hash codes, optimizing this relationship for better learning outcomes. HashGAN [39] employs a Wasserstein GAN to generate hash codes while utilizing pairwise similarities within a Bayesian

 doi:10.20944/preprints202411.1422.v1

framework. Zhuang et al. [40] developed a binary CNN classifier leveraging triplet loss to maintain semantic relationships. Deep Triplet Quantization (DTQ) [41] incorporates triplet-based quantization in a supervised learning framework for joint optimization of quantization and feature learning. In Supervised Semantics-Preserving Hashing (SSDH) [42], hash functions are embedded as a fully connected layer, with training focused on minimizing classification error. Wang et al. [43] offered a comprehensive framework for distance-preserving linear hashing extended to deep learning, where the fully connected layer's features support hashing. Shen et al.'s Similarity-Adaptive Deep Hashing (SADH) [44] uses outputs from fully connected layers to refine a similarity graph matrix for enhanced hash code learning.

Traditional approaches often rely on high-level features, typically from the final fully connected layers. However, capturing diverse features for a more comprehensive representation is crucial. Multi-level image retrieval methods address this need. Lin et al. introduced Discriminative Deep Hashing (DDH) [14], integrating end-to-end learning and multi-scale feature extraction from convolution-pooling layers. Yang et al. [45] developed Feature Pyramid Hashing (FPH), a dual-pyramid framework for learning detailed and semantic features for fine-grained retrieval. Redaoui et al. proposed Deep Feature Pyramid Hashing (DFPH) [46] for leveraging multi-level visual and semantic data, and Deep Supervised Hashing with Multiscale Feature Fusion (DSHFMDF) [47], which extracts and combines multiscale features from various convolutional layers for robust image retrieval. Ng et al. [48] introduced Multi-Level Supervised Hashing (MLSH), which separately trains tables at different feature levels to enhance both structural and semantic representation.

2.1. Problem Definition

Let $X = \{x_i\}_{i=1}^{N} \in \mathbb{R}^{d \times N}$ represent a training dataset with N images, where $Y = \{y_i\}_{i=1}^{N} \in \mathbb{R}^{K \times N}$ are the associated ground truth labels for the x_i samples, and K denotes the number of classes. To express the semantic similarities between images, we use a pairwise label matrix $S = \{s_{ij}\}$, where $s_{ij} \in \{0, 1\}$ indicates whether images x_i and x_j are semantically related $s_{ij} = 1$ or not $s_{ij} = 0$. The objective of deep hashing is to learn a function $f : x \mapsto B \in \{-1, 1\}^L$, which maps each input image x_i to a binary code $b_i \in \{-1, 1\}^L$, where L represents the length of the binary code.

2.2. Model Architecture

The architecture of the proposed MDFF-SH model, depicted in Figure 1, is structured to achieve high-efficiency and high-precision image retrieval through five main components: (1) feature extraction, (2) feature reduction, (3) feature fusion, (4) hash coding, and (5) classification. This modular approach ensures a cohesive understanding of each component and how they contribute to the model's overall functionality.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 November 2024 doi:10.20944/preprints202411.1422.v1

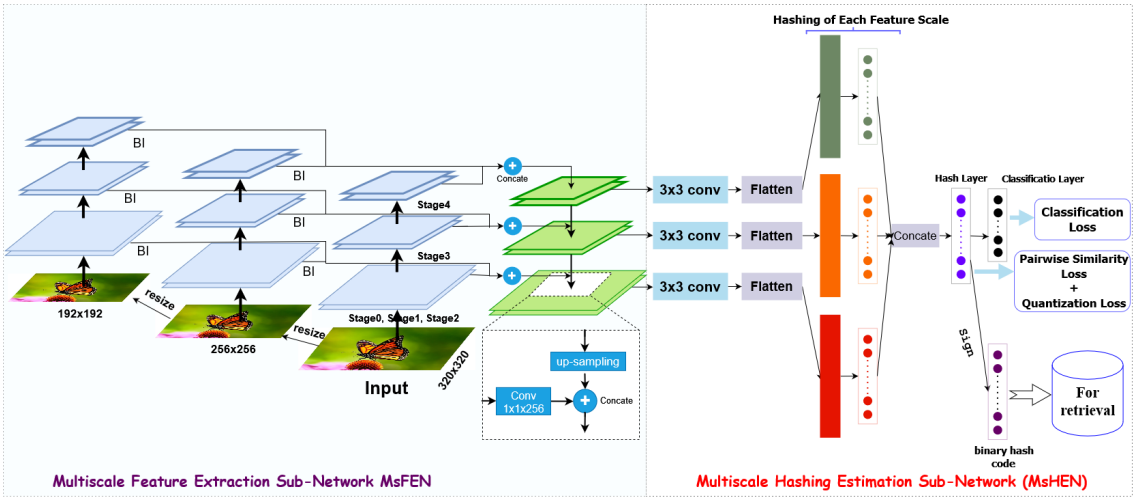


Figure 1. Enhanced Image Retrieval Using Multiscale Deep Feature Fusion in Supervised Hashing (MDFF-SH).

1. **Feature Extraction:** The initial feature extraction stage is crucial for gathering informative details from the input image. In MDFF-SH, the ResNet50 network serves as the backbone due to its capability to capture complex and distinguishing image features. Each layer in ResNet50 is designed to capture image details at increasing levels of abstraction, making it an ideal foundation for extracting both structural and semantic features.

   MDFF-SH systematically collects features from distinct levels of ResNet50. This includes low-level features, capturing fine details such as edges and textures, and high-level features, encapsulating semantic attributes. This multi-level approach ensures that the image representation integrates both granular details and overall semantic meaning.
2. **Multiscale Feature Focus:** The model's multiscale feature extraction focuses on layers from several convolutional blocks—specifically, the final layers of 'conv3', 'conv4', and 'conv5' blocks, along with the fully connected layer *fc1*. Lower-level layers like 'conv1' and 'conv2' are excluded to optimize memory usage, as their semantic contribution is limited. The selected layers effectively capture a balanced mix of structural and semantic information, providing a comprehensive representation of the image that includes both low- and high-level characteristics.
3. **Feature Reduction:** After extraction, the dimensionality of the multiscale features is reduced to retain discriminative power without excessive computational overhead. Using a 1x1 convolutional kernel, the model combines features across levels in a linear manner, creating a streamlined yet rich representation. This step enhances the depth and robustness of the features while minimizing redundancy.
4. Feature Fusion: In the fusion stage, the reduced features from different levels are combined to produce a unified representation. By merging both low- and high-level information, the fusion layer enables the model to construct an image representation that captures local structures alongside global context. This fusion provides a robust basis for generating binary codes that reflect a detailed and semantically rich image profile.
5. **Hash Coding:** To generate the final hash codes, the fused feature representation undergoes nonlinear transformations through hash layers, each of which outputs binary codes of the desired length L. This transformation ensures that the binary codes retain the core characteristics of the images in a compact and retrieval-optimized format.
6. **Classification:** The classification layer, which corresponds to the number of classes in the dataset, assigns the generated hash codes to specific image categories. This final component allows MDFF-SH to distinguish among classes based on learned binary representations, reinforcing the network's retrieval effectiveness.

   Through this structured architecture presented in Table 1, MDFF-SH captures both local and

 doi:10.20944/preprints202411.1422.v1

global image information, resulting in a powerful and compact feature representation that is tailored to high-precision image retrieval.

Table 1. Summary of the feature extraction network. Layers marked with '#' are used for feature extraction. ReLU and Batch Normalization layers are omitted for simplicity.

Conv Block	Layers	Kernel Sizes	Feature Dimensions
1	Conv2D, Conv2D#, MaxPooling	$64 \times 3 \times 3$, $64 \times 3 \times 3$	224×224
2	Conv2D, Conv2D#, MaxPooling	$128 \times 3 \times 3$, $128 \times 3 \times 3$	112×112
3	Conv2D, Conv2D, Conv2D, Conv2D#, MaxPooling	$256 \times 3 \times 3$, $256 \times 3 \times 3$, $256 \times 3 \times 3$, $256 \times 3 \times 3$	56×56
4	Conv2D, Conv2D, Conv2D, Conv2D#, MaxPooling	$512 \times 3 \times 3$, $512 \times 3 \times 3$, $512 \times 3 \times 3$, $512 \times 3 \times 3$	28×28
5	Conv2D, Conv2D, Conv2D, Conv2D#, MaxPooling	$512 \times 3 \times 3$, $512 \times 3 \times 3$, $512 \times 3 \times 3$, $512 \times 3 \times 3$	14×14

After extracting features from multiple scales, we employ a 1x1 convolutional kernel to reduce dimensionality while preserving discriminative information. This process enhances feature depth and robustness and eliminates redundancy.

Subsequently, a fusion layer composed of 1024 nodes integrates these multi-scale features, combining low-level structural details with high-level semantic information. This fusion step creates a comprehensive image representation that balances fine-grained local structures with broader contextual understanding.

To generate compact binary hash codes, we apply a nonlinear mapping through multiple hash layers, each with L nodes. This nonlinear transformation effectively encapsulates key image characteristics into binary codes. The concatenated hash code representation is further refined in the final hashing layer to ensure semantic integrity and discriminative power.

Finally, a classification layer with neurons corresponding to the number of classes is employed to categorize images based on their learned representations. The discriminative nature of the hash codes enables accurate image classification.

By integrating multi-scale features and a well-structured architecture, our model generates diverse and informative hash codes. These hash codes effectively capture both local details and global context, leading to improved retrieval performance and accurate image classification.

2.3. Loss Functions and Learning Rule

To ensure that the generated hash codes effectively preserve semantic similarity, our MDFF-SH method combines three distinct loss functions: pairwise similarity loss, quantization loss, and classification loss. These losses are harmonized to support efficient and effective training.

2.3.1. Pairwise Similarity Loss

The MDFF-SH method is designed to maintain similarity between pairs of input samples within Hamming space. Pairwise similarity is evaluated by calculating the inner product between hash codes b_i and b_j , defined as dist $dist_H\left(b_i, b_j\right) = \frac{1}{2} b_i^T b_j$. Given a set of binary codes $B = \{b_i\}_{i=1}^N$ and pairwise labels $S = \{s_{ij}\}$, the probability of the pairwise labels is represented as:

$$p(s_{ij}|B) = \begin{cases} \sigma(w_{ij}) & \text{if } s_{ij} = 1 \\ 1 - \sigma(w_{ij}) & \text{if } s_{ij} = 0 \end{cases} \tag{1}$$

where $\sigma(w_{ij}) = \frac{1}{1+e^{-w_{ij}}}$ and $w_{ij} = \frac{1}{2} b_i^T b_j$

This formulation implies that a larger inner product $\langle b_i, b_j \rangle$ corresponds to a smaller $dist_H\left(b_i, b_j\right)$ and a higher value of $p\left(1|b_i, b_j\right)$. When $s_{ij} = 1$, the binary codes b_i and b_j are considered similar.

 doi:10.20944/preprints202411.1422.v1

The optimization problem then becomes minimizing the negative log-likelihood over labels in S, resulting in

$$J_1 = -\log p\left(S|B\right) = -\sum_{s_{ij}\in S}\left(s_{ij}w_{ij} - \log(1+e^{w_{ij}})\right) \tag{2}$$

This objective aims to minimize the Hamming distance between similar samples while maximizing the distance between dissimilar samples, aligning with the principles of similarity-based hashing.

2.3.2. Quantization Loss

In practical applications, binary hash codes are commonly used for measuring similarity. However, optimizing discrete hash codes directly within a neural network can be challenging. To address this, we employ a continuous approximation for hash coding. Let u_i denote the output of the hash layer, with b_i defined as $b_i = \text{sgn}\left(u_i\right)$. To minimize the gap between continuous and discrete representations, we introduce quantization loss as a secondary objective:

$$J_2 = \sum_{i=1}^{Q}|b_i - u_i|_2^2 \tag{3}$$

where Q represents the mini-batch size.

2.3.3. Classification Loss

To support robust learning of multiscale features across the network, we employ cross-entropy loss for classification, which helps the model correctly categorize input samples. The classification loss is given by:

$$J_3 = -\sum_{i=1}^{Q}\sum_{k=1}^{K} y_{i,k}\log(p_{i,k}) \tag{4}$$

where $y_{i,k}$ denotes the true label and $p_{i,k}$ represents the softmax output of the i-th training sample for the k-th class.

In conclusion, the total loss function combines the pairwise similarity, quantization, and classification losses, as follows:

$$J = J_1 + \beta J_2 + \gamma J_3 \tag{5}$$

where β and γ are balancing parameters that control the contributions of quantization and classification losses, respectively.

3. Experiments

This section evaluates the performance of MDFF-SH and its variations on three extensive public datasets: CIFAR-10, NUS-WIDE, and MS-COCO. Our objective is to demonstrate the effectiveness of the proposed method compared to several leading hashing approaches. We begin with an overview of these datasets and follow with the experimental setup. Section 3.3 details the evaluation metrics and baseline methods. Finally, we present the results, including a comparative analysis with state-of-the-art hashing techniques.

3.1. Datasets

CIFAR-10,krizhevsky2009learning: This dataset consists of 60,000 color images across ten object classes, with each class containing 6,000 images sized 32×32 pixels. Following the protocol in [50], we randomly select 1,000 images (100 per class) as the query set, with the remaining images serving as the database. From the database, we sample 5,000 images (500 per class) as the training set.

NUS-WIDE,chua2009nus: Comprising 123,287 color images (40,504 validation images and 82,783 training images), this dataset includes images labeled with one or more of 80 categories. For our

 doi:10.20944/preprints202411.1422.v1

experiments, we randomly select 5,000 images as the query set, 10,000 as the training set, and use the remaining images as the database.

MS-COCO,lin2014microsoft: is a dataset consists 123287 colour images (40504 validation and 82783 training). Each image is labelled by one or more of 80 categories. We randomly selected 5,000 images as queries points and 10000 images as the training datasets. While the rest of the images as the database.

3.2. Experimental Settings

The MDFF-SH method is implemented using the PyTorch deep learning framework, and we initialize the network parameters with the ResNet50 convolutional model pretrained on the ImageNet dataset [53]. All experiments are conducted using the RMSProp optimizer [54] with a learning rate of 1×10^{-5} and a batch size of 32. Hyperparameters are set as follows: $\alpha = 0.01$ and $\beta = 0.1$.

3.3. Evaluation Metrics and Baselines

To assess the performance of our image retrieval method and facilitate comparisons with alternative approaches, we use the following metrics:

1. Mean Average Precision (MAP) results,
2. Precision-Recall (PR) curves,
3. Precision at top retrieval levels (P@N), and
4. Precision within a Hamming radius of 2 (P@H $\leq$ 2).

The MDFF-SH method is compared with a selection of traditional and state-of-the-art methods, including five unsupervised methods: LSH [4], SH [33], SGH [55], ITQ [34], PCAH [56], as well as two supervised hashing methods: SDH [28] and KSH [37]. Additionally, we include nine deep supervised hashing methods: CNNH [16], DNNH [8], DCH [57], DHN [7], HashNet [58], DHDW [59], DPH [60], LRH [50], and MFLH [61]. For multi-label datasets such as MS-COCO and NUS-WIDE, two samples are considered similar if they share one or more semantic labels.

3.4. Results

Table 2 presents a comparison of MAP results for our method and competing hashing methods on CIFAR-10 and NUS-WIDE with hash code lengths of 12, 24, 32, and 48 bits. Our MDFF-SH method consistently outperforms all other methods. Specifically, compared to the best traditional hashing method, SDH [28], MDFF-SH achieves an average MAP improvement of 52.7% and 23.55% on CIFAR-10 and NUS-WIDE, respectively. Deep hashing methods generally perform better than classical methods due to their ability to generate more robust feature representations. For CIFAR-10 and NUS-WIDE, MDFF-SH delivers an average MAP increase of 9.58% and 2.95%, respectively, over the second-best method, MFLH [61], across all hash code lengths. For example, the MAP values of MDFF-SH at different lengths are enhanced by 52.6%, 52.5%, 53.3%, and 52.4%, respectively, compared to the SDH method. The MAP of the MDFF-SH method is also significantly improved compared to the deep hash method. These results indicate the capability of MDFF-SH to produce high-quality hash codes for efficient image retrieval.

Table 2 shows the performance of MDFF-SH on the MS-COCO dataset. MDFF-SH achieves superior retrieval performance at all code lengths compared to all baseline methods. As a multi-label dataset, MS-COCO presents a more complex semantic structure than single-label datasets, which poses a greater challenge for maintaining semantic integrity in hash codes. For example, the MDFF-SH method improves the MAP values over different lengths of hash codes by 7.4%, 11.7%, 14.1%, and 15.2%, respectively, compared with the DCH method. Nonetheless, MDFF-SH achieves the best results, underscoring the effectiveness and robustness of the proposed approach for high-precision image retrieval in complex datasets.

 doi:10.20944/preprints202411.1422.v1

Table 2. Mean Average Precision (MAP) of Hamming ranking for different number of bits on CIFAR-10 and NUS-WIDE. The MAP values are calculated on the top 5,000 retrieval images for the NUS-WIDE dataset.

	CIFAR-10 (MAP)				NUS-WIDE (MAP)			
Method	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
SH [33]	0.127	0.128	0.126	0.129	0.454	0.406	0.405	0.400
ITQ [34]	0.162	0.169	0.172	0.175	0.452	0.468	0.472	0.477
KSH [37]	0.303	0.337	0.346	0.356	0.556	0.572	0.581	0.588
SDH [28]	0.285	0.329	0.341	0.356	0.568	0.600	0.608	0.637
CNNH [16]	0.439	0.511	0.509	0.522	0.611	0.618	0.625	0.608
DNNH [8]	0.552	0.566	0.558	0.581	0.674	0.697	0.713	0.715
DHN [7]	0.555	0.594	0.603	0.621	0.708	0.735	0.748	0.758
HashNet [58]	0.609	0.644	0.632	0.646	0.643	0.694	0.737	0.750
DPH [60]	0.698	0.729	0.749	0.755	0.770	0.784	0.790	0.786
LRH [50]	0.684	0.700	0.727	0.730	0.726	0.775	0.774	0.780
MFLH [61]	0.726	0.758	0.771	0.781	0.782	0.814	0.817	0.824
MDFF-SH	**0.811**	**0.854**	**0.874**	**0.880**	**0.828**	**0.854**	**0.866**	**0.887**

Table 3. Mean Average Precision (MAP) of Hamming ranking for different number of bits on MS-COCO. The MAP values are calculated on the top 5,000 retrieval images.

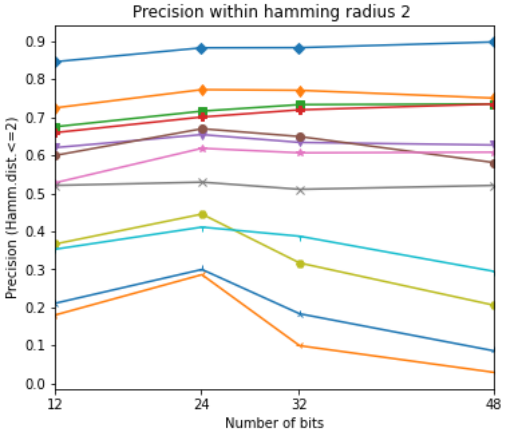
	MS-COCO (MAP)			
Method	16 bits	32 bits	48 bits	64 bits
SGH [55]	0.362	0.368	0.375	0.384
SH [33]	0.494	0.525	0.539	0.547
PCAH [56]	0.559	0.573	0.582	0.588
LSH [4]	0.406	0.440	0.486	0.517
ITQ [34]	0.613	0.649	0.671	0.680
DHN [7]	0.608	0.640	0.661	0.678
HashNet [58]	0.642	0.671	0.683	0.689
DCH [57]	0.652	0.680	0.689	0.690
DHDW [59]	0.655	0.681	0.695	0.702
MDFF-SH	**0.726**	**0.797**	**0.830**	**0.842**

Figures 2(**a**) and 3(**a**) present the precision curves for P@H=2, demonstrating that our MDFF-SH method consistently outperforms other techniques by achieving the highest precision within this Hamming radius. Although a slight decline in P@H=2 performance is observed as the code length increases, MDFF-SH maintains strong retrieval accuracy, indicating its ability to focus on relevant points within a Hamming radius of 2, even with longer hash codes.

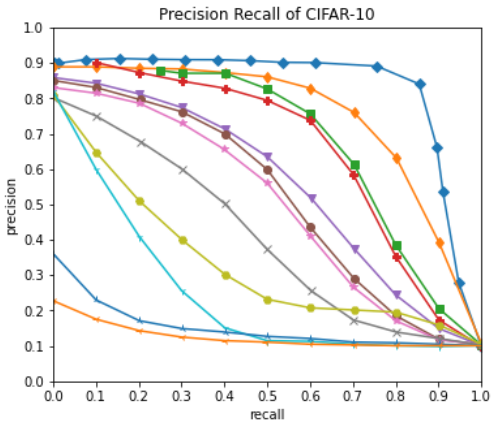
Additionally, Figures 2(**b**), 3(**b**), 2(**c**), and 3(**c**) compare the Precision-Recall and precision-at-top-results performance of MDFF-SH with other methods. In particular, Figures 2(**c**) and 3(**c**) show that MDFF-SH achieves the highest precision with 48-bit codes across varying numbers of returned samples, especially in the range of 100 to 1,000. Furthermore, Figures 2(**b**) and 3(**b**) illustrate that MDFF-SH achieves notably high precision at low recall levels—a crucial feature for precision-first retrieval systems widely used in practical applications. Overall, these results underscore the superior performance of MDFF-SH compared to other methods evaluated.

Figures 2(**a**) and 3(**a**) display the precision curves for P@H=2, clearly showing that our MDFF-SH method outperforms other approaches by achieving the highest precision within this Hamming radius. While a slight decrease in P@H=2 performance occurs as code length increases, this result highlights MDFF-SH's effectiveness in concentrating on relevant points within a Hamming radius of 2, even with longer hash codes.

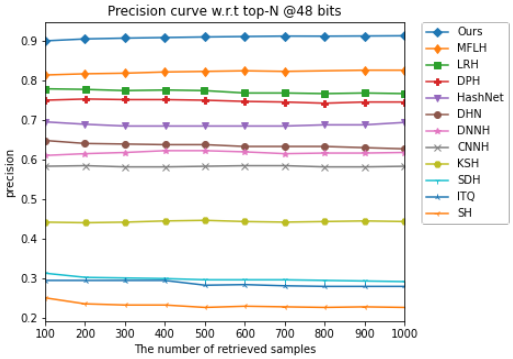
 doi:10.20944/preprints202411.1422.v1



(**a**) Precision within hamming radius 2



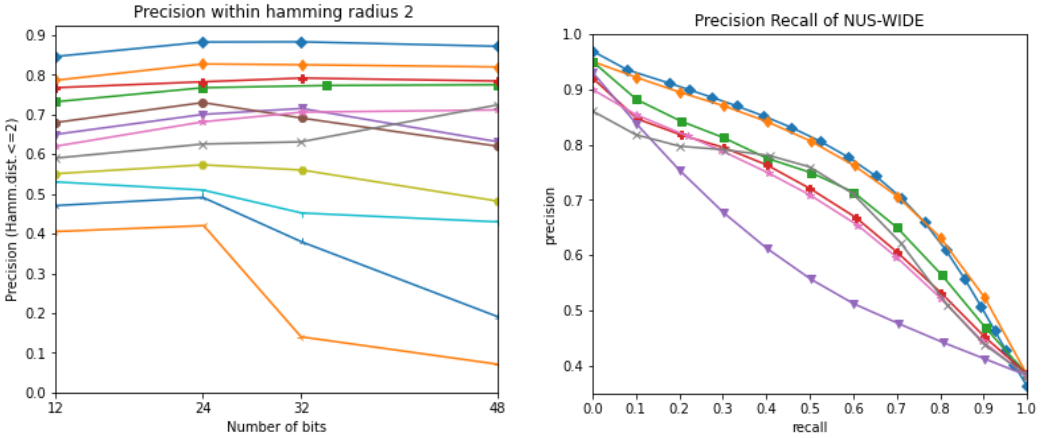
(**b**) Precision Recall Curve on 48 bits



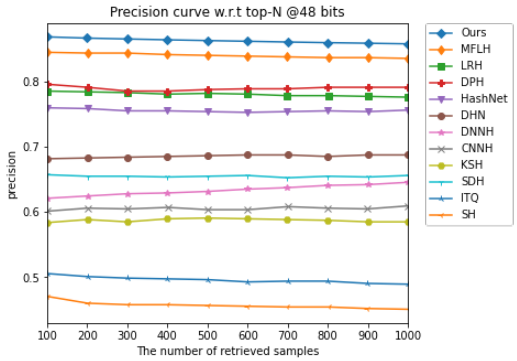
(**c**) Precision curve w.r.t top-N @48 bits

Figure 2. The comparison results on the CIFAR-10 dataset under three evaluation metrics.

 doi:10.20944/preprints202411.1422.v1



(**a**) Precision within hamming radius 2 (**b**) Precision Recall Curve on 48 bits



(**c**) Precision curve w.r.t top-N @48 bits

Figure 3. The comparison results on the NUS-WIDE dataset under three evaluation metrics.

In summary, our MDFF-SH method consistently outperforms the compared methods across various evaluation metrics, underscoring its superiority in image retrieval tasks. To visually illustrate its effectiveness in eliminating irrelevant images, we present Figure 4, showcasing the retrieval accuracy of different image categories in the CIFAR-10 dataset using MDFF-SH with 48-bit binary codes. The figure features query images in the first column, while the subsequent columns display images retrieved using MDFF-SH. This example reinforces our approach's capability to precisely retrieve pertinent images, further substantiating its practical utility.

 doi:10.20944/preprints202411.1422.v1

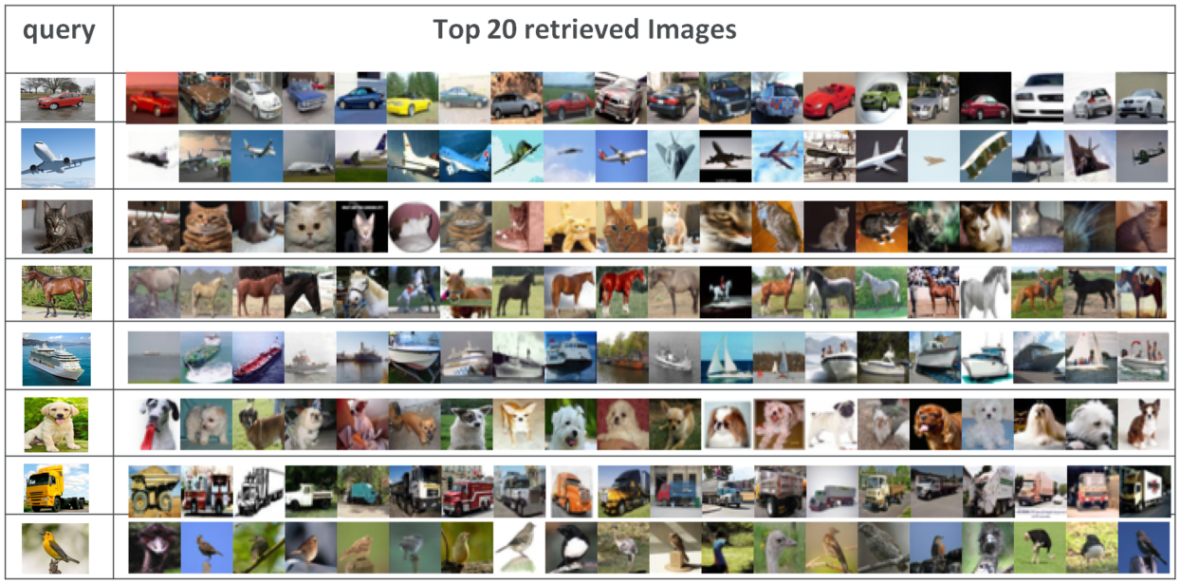


Figure 4. Presented are the top 20 retrieved results from the CIFAR-10 dataset, utilizing MDFF-SH with 48-bit hash codes. The first column showcases the query images, while the subsequent columns display the retrieval results generated by MDFF-SH.

In summary, our MDFF-SH method consistently surpasses the compared techniques across multiple evaluation metrics, affirming its superior performance in image retrieval tasks. To visually demonstrate its effectiveness in filtering out irrelevant images, Figure 4 presents the retrieval accuracy for various image categories within the CIFAR-10 dataset using MDFF-SH with 48-bit binary codes. In this figure, query images are shown in the first column, followed by images retrieved through MDFF-SH in the subsequent columns. This example highlights our method's ability to accurately retrieve relevant images, underscoring its practical value in real-world applications.

3.5. Ablation Studies

(1) Ablation Studies on Multi-Level Image Representations for Enhanced Hash Learning: To investigate the impact of multi-level image representations on hash learning, we conducted ablation studies. Unlike many existing methods that primarily focus on semantic information extracted from the final fully connected layers, we explored the contribution of structural information from various network layers.
Table 4 presents the retrieval performance on the CIFAR-10 dataset using different feature maps. We observed that features from the fc1 layer yielded the highest mAP of 75.8%, emphasizing the importance of high-level semantic information. However, using features from Conv 3-5 resulted in an average mAP of 62.5%, highlighting the significance of low-level structural details. Our proposed MDFF-SH approach outperformed all other configurations, achieving an average mAP of 85.5%, demonstrating the effectiveness of combining multi-scale features for enhanced retrieval performance.

Table 4. Mean Average Precision (mAP) for different feature scales with various bit lengths on CIFAR-10.

Method	CIFAR-10 (MAP)			
	12 Bits	24 Bits	32 Bits	48 Bits
$fc1$	0.710	0.761	0.775	0.788
$conv3$–5	0.580	0.595	0.639	0.688
MDFF-SH	0.811	0.854	0.874	0.880

 doi:10.20944/preprints202411.1422.v1

(2) Ablation Studies on the Objective Function: To assess the impact of different loss components in our objective function, we conducted ablation studies on the CIFAR-10 dataset using the MDFF-SH model. We evaluated the performance of the model when either the Pairwise Quantization Loss ($\beta = 0$, MDFF-SH-J3) or the Classification Loss ($\gamma = 0$, MDFF-SH-J2) was excluded. As shown in Table 5, the inclusion of both J2 and J3 resulted in an 8.55% performance improvement. This finding highlights the importance of both quantization loss, which minimizes quantization error, and classification loss, which preserves semantic information, for generating high-quality hash codes.

Table 5. mAP results for different variants of the objective function on CIFAR-10.

Method	CIFAR-10 (MAP)			
	12 Bits	24 Bits	32 Bits	48 Bits
MDFF-SH-J2	0.667	0.812	0.830	0.852
MDFF-SH-J3	0.656	0.742	0.785	0.796
MDFF-SH	0.811	0.854	0.874	0.880

4. Conclusions and Future Work

This paper introduces a novel end-to-end framework, Multiscale Deep Feature Fusion for High-Precision Image Retrieval through Supervised Hashing (MDFF-SH), designed to generate robust binary codes. Our approach optimizes three key components: similarity loss, quantization loss, and semantic loss, to effectively integrate structural information into hash representations. By leveraging multiscale features, MDFF-SH achieves a balance between structural detail and retrieval accuracy, leading to improved recall and precision.

Extensive experiments on standard image retrieval datasets demonstrate the superior performance of MDFF-SH compared to state-of-the-art methods. In future work, we aim to extend this approach to medical imaging, where the presence of multi-scale objects could benefit significantly from our method's ability to capture both fine-grained and coarse-grained details.

The scalability of our model makes it adaptable to various computer vision tasks, providing robust feature representations that have the potential to advance a wide range of applications.

Author Contributions: Conceptualization, A.R. and K.B.; methodology, A.B., A.R. and K.B.; software, A.R.; validation, A.B. and K.B.; formal analysis, A.B., K.B., and R.A.; investigation, A.B. and K.B.; writing—original draft preparation, A.B. and K.B.; writing—review and editing, K.B. and A.R.; visualization, A.R. and A.B.; supervision, K.B. and A.B.; project administration, K.B.; funding acquisition, K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: http://www.cs.toronto.edu/~kriz/cifar.html, https://paperswithcode.com/datasets (all accessed on 31 July 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MDFF-SH	Multiscale Deep Feature Fusion for High-Precision Image Retrieval through Supervised Hashing
FPN	Feature Pyramid Network
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network

 doi:10.20944/preprints202411.1422.v1

References

1. Yan, C.; Shao, B.; Zhao, H.; Ning, R.; Zhang, Y.; Xu, F. 3D room layout estimation from a single RGB image. *IEEE Transactions on Multimedia* **2020**, *22*, 3014–3024.
2. Yan, C.; Li, Z.; Zhang, Y.; Liu, Y.; Ji, X.; Zhang, Y. Depth image denoising using nuclear norm and learning graph model. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **2020**, *16*, 1–17.
3. Li, S.; Chen, Z.; Li, X.; Lu, J.; Zhou, J. Unsupervised variational video hashing with 1d-cnn-lstm networks. *IEEE Transactions on Multimedia* **2019**, *22*, 1542–1554.
4. Gionis, A.; Indyk, P.; Motwani, R.; others. Similarity search in high dimensions via hashing. Vldb, 1999, Vol. 99, pp. 518–529.
5. Wang, J.; Zhang, T.; Sebe, N.; Shen, H.T.; others. A survey on learning to hash. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *40*, 769–790.
6. Erin Liong, V.; Lu, J.; Wang, G.; Moulin, P.; Zhou, J. Deep hashing for compact binary codes learning. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2475–2483.
7. Zhu, H.; Long, M.; Wang, J.; Cao, Y. Deep hashing network for efficient similarity retrieval. Proceedings of the AAAI conference on Artificial Intelligence, 2016, Vol. 30.
8. Lai, H.; Pan, Y.; Liu, Y.; Yan, S. Simultaneous feature learning and hash coding with deep neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3270–3278.
9. Cakir, F.; He, K.; Bargal, S.A.; Sclaroff, S. Hashing with mutual information. *IEEE transactions on pattern analysis and machine intelligence* **2019**, *41*, 2424–2437.
10. Li, Q.; Sun, Z.; He, R.; Tan, T. Deep supervised discrete hashing. *Advances in neural information processing systems* **2017**, *30*.
11. Yue, C.; Long, M.; Wang, J.; Han, Z.; Wen, Q. Deep quantization network for efficient image retrieval. Proc. 13th AAAI Conf. Artif. Intell., 2016, pp. 3457–3463.
12. Li, W.J.; Wang, S.; Kang, W.C. Feature learning based deep supervised hashing with pairwise labels. *arXiv preprint arXiv:1511.03855* **2015**.
13. Lu, J.; Liong, V.E.; Zhou, J. Deep hashing for scalable image search. *IEEE transactions on image processing* **2017**, *26*, 2352–2367.
14. Lin, J.; Li, Z.; Tang, J. Discriminative Deep Hashing for Scalable Face Image Retrieval. IJCAI, 2017, pp. 2266–2272.
15. Jiang, Q.Y.; Li, W.J. Asymmetric deep supervised hashing. Proceedings of the AAAI conference on artificial intelligence, 2018, Vol. 32.
16. Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised hashing for image retrieval via image representation learning. Twenty-eighth AAAI conference on artificial intelligence, 2014.
17. Shen, F.; Gao, X.; Liu, L.; Yang, Y.; Shen, H.T. Deep asymmetric pairwise hashing. Proceedings of the 25th ACM international conference on Multimedia, 2017, pp. 1522–1530.
18. Li, Y.; Xu, Y.; Wang, J.; Miao, Z.; Zhang, Y. Ms-rmac: Multiscale regional maximum activation of convolutions for image retrieval. *IEEE Signal Processing Letters* **2017**, *24*, 609–613.
19. Tolias, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879* **2015**.
20. Seddati, O.; Dupont, S.; Mahmoudi, S.; Parian, M. Towards good practices for image retrieval based on CNN features. Proceedings of the IEEE international conference on computer vision workshops, 2017, pp. 1246–1255.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.
22. Zhao, Y.; Han, R.; Rao, Y. A new feature pyramid network for object detection. 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS). IEEE, 2019, pp. 428–431.
23. Jin, Z.; Li, C.; Lin, Y.; Cai, D. Density sensitive hashing. *IEEE transactions on cybernetics* **2013**, *44*, 1362–1371.
24. Andoni, A.; Indyk, P. Near-optimal hashing algorithms for near neighbor problem in high dimension. *Communications of the ACM* **2008**, *51*, 117–122.
25. Kulis, B.; Darrell, T. Learning to hash with binary reconstructive embeddings. *Advances in neural information processing systems* **2009**, *22*.

26. Liu, H.; Ji, R.; Wu, Y.; Liu, W. Towards optimal binary code learning via ordinal embedding. Thirtieth AAAI conference on artificial intelligence, 2016.
27. Wang, J.; Wang, J.; Yu, N.; Li, S. Order preserving hashing for approximate nearest neighbor search. Proceedings of the 21st ACM international conference on Multimedia, 2013, pp. 133–142.
28. Shen, F.; Shen, C.; Liu, W.; Tao Shen, H. Supervised discrete hashing. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 37–45.
29. Salakhutdinov, R.; Hinton, G. Semantic hashing. *International Journal of Approximate Reasoning* **2009**, *50*, 969–978.
30. Zhang, S.; Li, J.; Jiang, M.; Yuan, P.; Zhang, B. Scalable discrete supervised multimedia hash learning with clustering. *IEEE Transactions on Circuits and Systems for Video Technology* **2017**, *28*, 2716–2729.
31. Lin, M.; Ji, R.; Liu, H.; Sun, X.; Wu, Y.; Wu, Y. Towards optimal discrete online hashing with balanced similarity. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, Vol. 33, pp. 8722–8729.
32. Jegou, H.; Douze, M.; Schmid, C. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence* **2010**, *33*, 117–128.
33. Weiss, Y.; Torralba, A.; Fergus, R. Spectral hashing. *Advances in neural information processing systems* **2008**, *21*.
34. Gong, Y.; Lazebnik, S.; Gordo, A.; Perronnin, F. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence* **2012**, *35*, 2916–2929.
35. Liu, W.; Wang, J.; Kumar, S.; Chang, S.F. Hashing with graphs. Icml, 2011.
36. Datar, M.; Immorlica, N.; Indyk, P.; Mirrokni, V.S. Locality-sensitive hashing scheme based on p-stable distributions. Proceedings of the twentieth annual symposium on Computational geometry, 2004, pp. 253–262.
37. Liu, W.; Wang, J.; Ji, R.; Jiang, Y.G.; Chang, S.F. Supervised hashing with kernels. 2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012, pp. 2074–2081.
38. Norouzi, M.; Fleet, D.J. Minimal loss hashing for compact binary codes. ICML, 2011.
39. Cao, Y.; Liu, B.; Long, M.; Wang, J. Hashgan: Deep learning to hash with pair conditional wasserstein gan. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1287–1296.
40. Zhuang, B.; Lin, G.; Shen, C.; Reid, I. Fast training of triplet-based deep binary embedding networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5955–5964.
41. Liu, B.; Cao, Y.; Long, M.; Wang, J.; Wang, J. Deep triplet quantization. Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 755–763.
42. Yang, H.F.; Lin, K.; Chen, C.S. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *40*, 437–451.
43. Wang, M.; Zhou, W.; Tian, Q.; Li, H. A general framework for linear distance preserving hashing. *IEEE Transactions on Image Processing* **2017**, *27*, 907–922.
44. Shen, F.; Xu, Y.; Liu, L.; Yang, Y.; Huang, Z.; Shen, H.T. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *IEEE transactions on pattern analysis and machine intelligence* **2018**, *40*, 3034–3044.
45. Yang, Y.; Geng, L.; Lai, H.; Pan, Y.; Yin, J. Feature pyramid hashing. Proceedings of the 2019 on International Conference on Multimedia Retrieval, 2019, pp. 114–122.
46. Redaoui, A.; Belloulata, K. Deep Feature Pyramid Hashing for Efficient Image Retrieval. *Information* **2023**, *14*. doi:10.3390/info14010006.
47. Redaoui, A.; Belalia, A.; Belloulata, K. Deep Supervised Hashing by Fusing Multiscale Deep Features for Image Retrieval. *Information* **2024**, *15*. doi:10.3390/info15030143.
48. Ng, W.W.; Li, J.; Tian, X.; Wang, H.; Kwong, S.; Wallace, J. Multi-level supervised hashing with deep features for efficient image retrieval. *Neurocomputing* **2020**, *399*, 171–182.
49. Krizhevsky, A.; Hinton, G.; others. Learning multiple layers of features from tiny images **2009**.
50. Bai, J.; Li, Z.; Ni, B.; Wang, M.; Yang, X.; Hu, C.; Gao, W. Loopy residual hashing: Filling the quantization gap for image retrieval. *IEEE Transactions on Multimedia* **2019**, *22*, 215–228.
51. Chua, T.S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; Zheng, Y. Nus-wide: a real-world web image database from national university of singapore. Proceedings of the ACM international conference on image and video retrieval, 2009, pp. 1–9.
52. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. European conference on computer vision. Springer, 2014, pp. 740–755.

53. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; others. Imagenet large scale visual recognition challenge. *International journal of computer vision* **2015**, *115*, 211–252.
54. G Hinton, N.S.; Swersky, K. Overview of Mini Batch Gradient Descent. Computer Science Department, University of Toronto, 2015.
55. Jiang, Q.Y.; Li, W.J. Scalable graph hashing with feature transformation. Twenty-fourth international joint conference on artificial intelligence, 2015.
56. Wang, J.; Kumar, S.; Chang, S.F. Semi-supervised hashing for large-scale search. *IEEE transactions on pattern analysis and machine intelligence* **2012**, *34*, 2393–2406.
57. Cao, Y.; Long, M.; Liu, B.; Wang, J. Deep cauchy hashing for hamming space retrieval. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1229–1237.
58. Cao, Z.; Long, M.; Wang, J.; Yu, P.S. Hashnet: Deep learning to hash by continuation. Proceedings of the IEEE international conference on computer vision, 2017, pp. 5608–5617.
59. Sun, Y.; Yu, S. Deep Supervised Hashing with Dynamic Weighting Scheme. 2020 5th IEEE International Conference on Big Data Analytics (ICBDA). IEEE, 2020, pp. 57–62.
60. Bai, J.; Ni, B.; Wang, M.; Li, Z.; Cheng, S.; Yang, X.; Hu, C.; Gao, W. Deep progressive hashing for image retrieval. *IEEE Transactions on Multimedia* **2019**, *21*, 3178–3193.
61. Feng, H.; Wang, N.; Tang, J.; Chen, J.; Chen, F. Multi-granularity feature learning network for deep hashing. *Neurocomputing* **2021**, *423*, 274–283.