

Article

Enhanced Image Retrieval Using Multiscale Deep Feature Fusion in Supervised Hashing

Amina Belalia ¹, Kamel Belloulata ^{2,*} and Adil Redaoui ²¹ High School of Computer Sciences, Sidi Bel Abbes 22000, Algeria; a.belalia@esi-sba.dz² RCAM Laboratory, Telecommunications Department, Sidi Bel Abbes University, Sidi Bel Abbes 22000, Algeria; kamel.belloulata@univ-sba.dz

* Correspondence: kamellbelloulata99@gmail.com; Tel.: +213-7-7371-5910

Abstract: In recent years, deep-network-based hashing has gained prominence in image retrieval for its ability to generate compact and efficient binary representations. However, most existing methods predominantly focus on high-level semantic features extracted from the final layers of networks, often neglecting structural details that are crucial for capturing spatial relationships within images. Achieving a balance between preserving structural information and maximizing retrieval accuracy is the key to effective image hashing and retrieval. To address this challenge, we introduce Multiscale Deep Feature Fusion for Supervised Hashing (MDFF-SH), a novel approach that integrates multiscale feature fusion into the hashing process. The hallmark of MDFF-SH lies in its ability to combine low-level structural features with high-level semantic context, synthesizing robust and compact hash codes. By leveraging multiscale features from multiple convolutional layers, MDFF-SH ensures the preservation of fine-grained image details while maintaining global semantic integrity, achieving a harmonious balance that enhances retrieval precision and recall. Our approach demonstrated a superior performance on benchmark datasets, achieving significant gains in the Mean Average Precision (MAP) compared with the state-of-the-art methods: 9.5% on CIFAR-10, 5% on NUS-WIDE, and 11.5% on MS-COCO. These results highlight the effectiveness of MDFF-SH in bridging structural and semantic information, setting a new standard for high-precision image retrieval through multiscale feature fusion.



Academic Editors: Anastasios L. Kesidis and Giorgos Sfikas

Received: 17 November 2024

Revised: 9 December 2024

Accepted: 12 December 2024

Published: 12 January 2025

Citation: Belalia, A.; Belloulata, K.; Redaoui, A. Enhanced Image Retrieval Using Multiscale Deep Feature Fusion in Supervised Hashing. *J. Imaging* **2025**, *11*, 20. <https://doi.org/10.3390/jimaging11010020>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

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 [1–5]. Approximate Nearest Neighbor (ANN) searching [6] has emerged as a pivotal area of study in computer vision and information retrieval since it is a technique essential for reducing storage requirements and improving search efficiency in high-dimensional spaces. Hashing [7] has garnered considerable attention as a potent strategy within ANN searching, which transforms high-dimensional data into compact binary codes while preserving spatial relationships between data points. Deep hashing techniques [8–11] have been devised to concurrently learn visual features and binary hash codes, enriching encoded information with a 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 [12–16], there remains a need for further improvements, particularly in retaining local structural details within images. Most deep hashing approaches emphasize high-level features from fully connected (FC) layers [17–19], 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 [20–22]. Despite attempting 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). By leveraging ResNet50 [23], 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 [24], this fusion process incorporates top-down pathways and lateral connections, enabling the exploration of both top-layer semantic and bottom-layer spatial features. 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 the semantic and spatial details. Moreover, this 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 demonstrated that MDFF-SH surpassed 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 [25,26]. 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 [27,28] and unsupervised [29–33] approaches based on the use of labeled data. Researchers developed unsupervised hashing methods [34–37] to learn hash functions using unlabeled training samples, transforming input images into binary codes. Locality-Sensitive Hashing (LSH) [38] is one of the most well-known unsupervised methods, followed by significant approaches, such as Spectral Hashing (SH) [35] and Iterative Quantization (ITQ) [36].

In contrast, supervised hashing techniques leveraged labeled data to learn hash functions, often yielding higher accuracy compared with unsupervised methods. Supervised Hashing with Kernels (KSH) [39] employs kernel methods to create nonlinear hash functions. Minimal Loss Hashing (MLH) [40] uses structured SVMs to define an objective for learning hash functions. Supervised Discrete Hashing (SDH) [30] refines the objective function to produce high-quality hash codes without relaxation.

The emergence of deep neural networks propelled the development of deep hashing algorithms [14,18,41–46], which outperformed traditional methods by using rich feature representations. Researchers proposed pairwise and triplet-based similarity preservation strategies to utilize label information effectively. CNN-based Hashing (CNNH) [18] extracted features using CNNs but separated feature learning from hash function training, which limited feedback integration. Deep Pairwise-Supervised hashing (DPSH) [14] employed a Bayesian approach to model relationships between pairwise labels and hash codes, optimizing this relationship for better outcomes. HashGAN [41] uses a Wasserstein GAN to generate hash codes while leveraging pairwise similarities within a Bayesian framework. Zhuang et al. [42] developed a binary CNN classifier that leveraged triplet loss to maintain semantic relationships. Deep Triplet Quantization (DTQ) [43] integrated triplet-based quantization into a supervised learning framework, enabling the joint optimization of quantization and feature learning. In Supervised Semantics-Preserving Hashing (SSDH) [44], researchers embedded hash functions as a fully connected layer, focusing on minimizing the classification error during training. Wang et al. [45] offered a comprehensive framework for distance-preserving linear hashing extended to deep learning, where the fully connected layer's features supported hashing. Shen et al.'s Similarity-Adaptive Deep Hashing (SADH) [46] utilized outputs from fully connected layers to refine a similarity graph matrix for enhanced hash code learning.

Product Quantization (PQ) techniques [34] have played a pivotal role in large-scale image retrieval due to their ability to compress high-dimensional data into compact representations while maintaining similarity-preserving properties. For instance, the study by Ma et al. [47] introduced a novel framework that leverages progressive quantization strategies to enhance fine-grained retrieval tasks. By integrating causal intervention into the quantization process, this approach achieves robust encoding and improved semantic preservation, making it highly effective for large-scale datasets. While vector quantization methods primarily optimize data compression and efficiency, our proposed MDFF-SH approach focuses on multiscale feature fusion to enhance the semantic representation and retrieval accuracy. These two paradigms are complementary, as future extensions of MDFF-SH could benefit from incorporating advanced quantization strategies to further improve the scalability in massive image databases.

Traditional approaches often relied on high-level features, typically from the final fully connected layers. However, capturing diverse features for a more comprehensive representation became a focus of multi-level image-retrieval methods. Lin et al. introduced Discriminative Deep Hashing (DDH) [16], which integrates end-to-end learning and multi-scale feature extraction from convolution-pooling layers. Yang et al. [48] 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) [49] to leverage multi-level visual and semantic data, and Deep Supervised Hashing with Multiscale Feature Fusion (DSHFMDF) [50], which extracts and combines multiscale features from various convolutional layers for robust image retrieval. Ng et al. [51] introduced Multi-Level Supervised Hashing (MLSH), which separately trains tables at different feature levels to enhance both the structural and semantic representations.

3. Proposed Methodology

This section outlines the proposed method in detail. First, we define the primary objective of our network: converting image information into hash codes for efficient retrieval. Next, we describe the structure of the proposed model, illustrated in Figure 1. Finally, we present the objective function that guides the optimization of the method.

3.1. 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 used 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 $f : x \mapsto B \in \{-1, 1\}^L$ that 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.

3.2. 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.

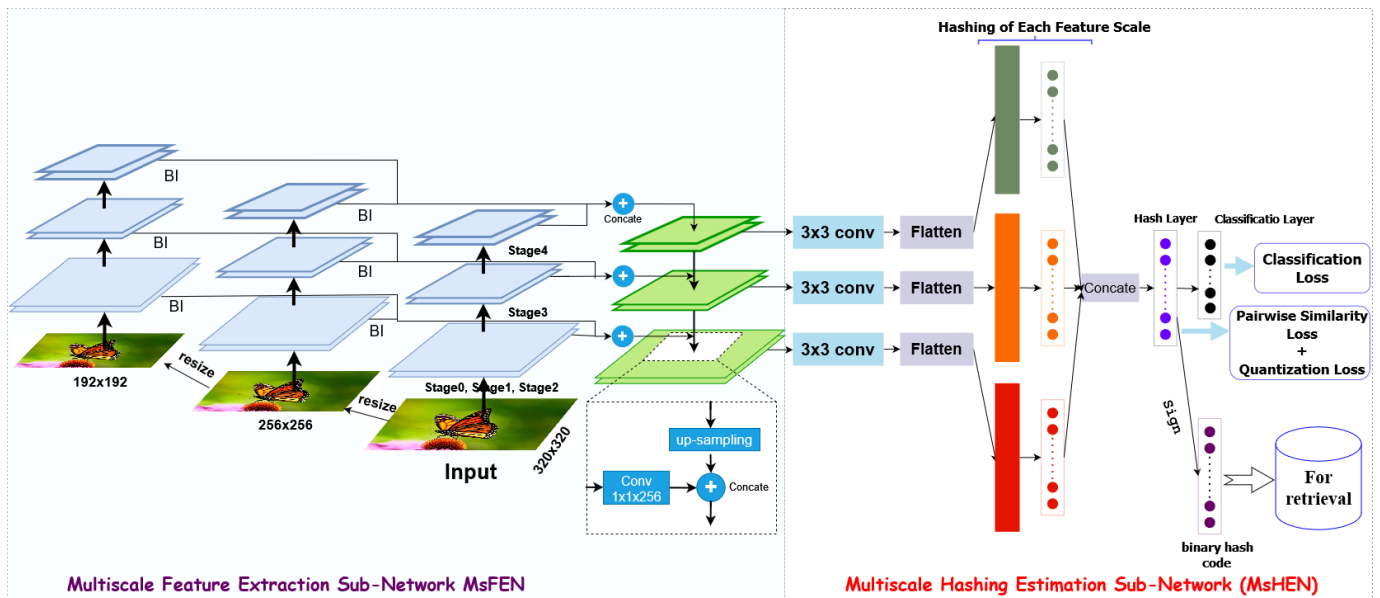


Figure 1. Enhanced image retrieval using Multiscale Deep Feature Fusion in Supervised Hashing (MDFF-SH).

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 the ‘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 the extraction, the dimensionality of the multiscale features is reduced to retain discriminative power without excessive computational overhead. Using a 1×1 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 the 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 the 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 between classes based on learned binary representations, reinforcing the network's retrieval effectiveness.

Through this structured architecture presented in Table 1, MDFF-SH captures both the local and 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 employed a 1×1 convolutional kernel to reduce the dimensionality while preserving the discriminative information. This process enhances the 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 applied 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 rep-

resentation 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 the local details and global context, leading to improved retrieval performance and accurate image classification.

3.3. 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.

3.3.1. 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 the hash codes b_i and b_j , defined as $\text{dist}_{\text{H}}(b_i, b_j) = \frac{1}{2}b_i^T b_j$. 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

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

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 $\text{dist}_{\text{H}}(b_i, b_j)$ and a higher value of $p(1|b_i, b_j)$. When $s_{ij} = 1$, the binary codes b_i and b_j are considered similar.

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

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

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

3.3.2. 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 employed a continuous approximation for hash coding. Let u_i denote the output of the hash layer, with b_i defined as $b_i = \text{sgn}(u_i)$. To minimize the gap between continuous and discrete representations, we introduced quantization loss as a secondary objective:

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

where Q represents the mini-batch size.

3.3.3. Classification Loss

To support the 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

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

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:

$$J = J_1 + \beta J_2 + \gamma J_3 \quad (5)$$

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

4. 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 was to demonstrate the effectiveness of the proposed method compared with several leading hashing approaches. We begin with an overview of these datasets and follow with the experimental setup. Section 4.3 details the evaluation metrics and baseline methods. Finally, we present the results, including a comparative analysis with state-of-the-art hashing techniques.

4.1. Datasets

CIFAR-10 [52]: This dataset consists of 60,000 color images across ten object classes, with each class containing 6000 images sized 32×32 pixels. Following the protocol in [53], we randomly selected 1000 images (100 per class) as the query set, with the remaining images serving as the database. From the database, we sampled 5000 images (500 per class) as the training set.

NUS-WIDE [54]: 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 experiments, we randomly selected 5000 images as the query set, 10,000 as the training set, and used the remaining images as the database.

MS-COCO [55]: This is a dataset consisting of 123,287 color images (40,504 validation and 82,783 training). Each image is labeled by one or more of 80 categories. We randomly selected 5000 images as query points and 10,000 images as the training dataset. The rest of the images were used as the database.

4.2. Experimental Settings

The MDFF-SH method was implemented using the PyTorch 2.0 deep learning framework, and we initialized the network parameters with the ResNet50 convolutional model pretrained on the ImageNet dataset [56]. All experiments were conducted using the RMSProp optimizer [57] with a learning rate of 1×10^{-5} and a batch size of 32. The hyperparameters were set as follows: $\gamma = 0.01$ and $\beta = 0.1$.

4.3. Evaluation Metrics and Baselines

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

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

The MDFF-SH method was compared with a selection of traditional and state-of-the-art methods, including five unsupervised methods: LSH [6], SH [35], SGH [58], ITQ [36], and PCAH [59], as well as two supervised hashing methods: SDH [30] and KSH [39]. Additionally, we included nine deep supervised hashing methods: CNNH [18], DNNH [10], DCH [60], DHN [9], HashNet [61], DHDW [62], DPH [63], LRH [53], and MFLH [64]. For multi-label datasets, such as MS-COCO and NUS-WIDE, two samples were considered similar if they shared one or more semantic label.

4.4. Results

Table 2 presents a comparison of the 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 outperformed all the other methods. Specifically, compared with the best traditional hashing method SDH [30], MDFF-SH achieved average MAP improvements 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 delivered average MAP increases of 9.58% and 2.95%, respectively, over the second-best method MFLH [64] across all hash code lengths. For example, the MAP values of MDFF-SH at different lengths were enhanced by 52.6%, 52.5%, 53.3%, and 52.4%, respectively, compared with the SDH method. The MAP of the MDFF-SH method was also significantly improved compared with the deep hash method. These results indicate the capability of MDFF-SH to produce high-quality hash codes for efficient image retrieval.

Table 2. Mean Average Precision (MAP) of the Hamming ranking for different numbers of bits on CIFAR-10 and NUS-WIDE. The MAP values were calculated for the top 5000 retrieval images from the NUS-WIDE dataset.

| Method | CIFAR-10 (MAP) | | | | NUS-WIDE (MAP) | | | |
|----------------|----------------|--------------|--------------|--------------|----------------|--------------|--------------|--------------|
| | 12 Bits | 24 Bits | 32 Bits | 48 Bits | 12 Bits | 24 Bits | 32 Bits | 48 Bits |
| SH [35] | 0.127 | 0.128 | 0.126 | 0.129 | 0.454 | 0.406 | 0.405 | 0.400 |
| ITQ [36] | 0.162 | 0.169 | 0.172 | 0.175 | 0.452 | 0.468 | 0.472 | 0.477 |
| KSH [39] | 0.303 | 0.337 | 0.346 | 0.356 | 0.556 | 0.572 | 0.581 | 0.588 |
| SDH [30] | 0.285 | 0.329 | 0.341 | 0.356 | 0.568 | 0.600 | 0.608 | 0.637 |
| CNNH [18] | 0.439 | 0.511 | 0.509 | 0.522 | 0.611 | 0.618 | 0.625 | 0.608 |
| DNNH [10] | 0.552 | 0.566 | 0.558 | 0.581 | 0.674 | 0.697 | 0.713 | 0.715 |
| DHN [9] | 0.555 | 0.594 | 0.603 | 0.621 | 0.708 | 0.735 | 0.748 | 0.758 |
| HashNet [61] | 0.609 | 0.644 | 0.632 | 0.646 | 0.643 | 0.694 | 0.737 | 0.750 |
| DPH [63] | 0.698 | 0.729 | 0.749 | 0.755 | 0.770 | 0.784 | 0.790 | 0.786 |
| LRH [53] | 0.684 | 0.700 | 0.727 | 0.730 | 0.726 | 0.775 | 0.774 | 0.780 |
| MFLH [64] | 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 shows the performance of MDFF-SH on the MS-COCO dataset. MDFF-SH achieved a superior retrieval performance at all code lengths compared with all the baseline methods. As a multi-label dataset, MS-COCO presented a more complex semantic structure than the single-label datasets, which posed a greater challenge for maintaining semantic integrity in hash codes. For example, the MDFF-SH method improved 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 achieved the best results, underscoring the

effectiveness and robustness of the proposed approach for high-precision image retrieval in complex datasets.

Table 3. Mean Average Precision (MAP) of the Hamming ranking for different numbers of bits on MS-COCO. The MAP values were calculated for the top 5000 retrieval images.

| Method | MS-COCO (MAP) | | | |
|----------------|---------------|--------------|--------------|--------------|
| | 16 Bits | 32 Bits | 48 Bits | 64 Bits |
| SGH [58] | 0.362 | 0.368 | 0.375 | 0.384 |
| SH [35] | 0.494 | 0.525 | 0.539 | 0.547 |
| PCAH [59] | 0.559 | 0.573 | 0.582 | 0.588 |
| LSH [6] | 0.406 | 0.440 | 0.486 | 0.517 |
| ITQ [36] | 0.613 | 0.649 | 0.671 | 0.680 |
| DHN [9] | 0.608 | 0.640 | 0.661 | 0.678 |
| HashNet [61] | 0.642 | 0.671 | 0.683 | 0.689 |
| DCH [60] | 0.652 | 0.680 | 0.689 | 0.690 |
| DHDW [62] | 0.655 | 0.681 | 0.695 | 0.702 |
| MDFF-SH | 0.726 | 0.797 | 0.830 | 0.842 |

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

Additionally, Figures 2b,c and 3b,c compare the precision–recall and precision at top results performance of MDFF-SH with the other methods. In particular, Figures 2c and 3c show that MDFF-SH achieved the highest precision with 48-bit codes across varying numbers of returned samples, especially in the range of 100 to 1000. Furthermore, Figures 2b and 3b illustrate that MDFF-SH achieved 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 with the other methods evaluated.

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

In summary, our MDFF-SH method consistently outperformed 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. This figure features query images in the first column, while the subsequent columns display images retrieved using MDFF-SH. This example reinforced our approach’s capability to precisely retrieve pertinent images, further substantiating its practical utility.

In summary, our MDFF-SH method consistently surpassed 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.

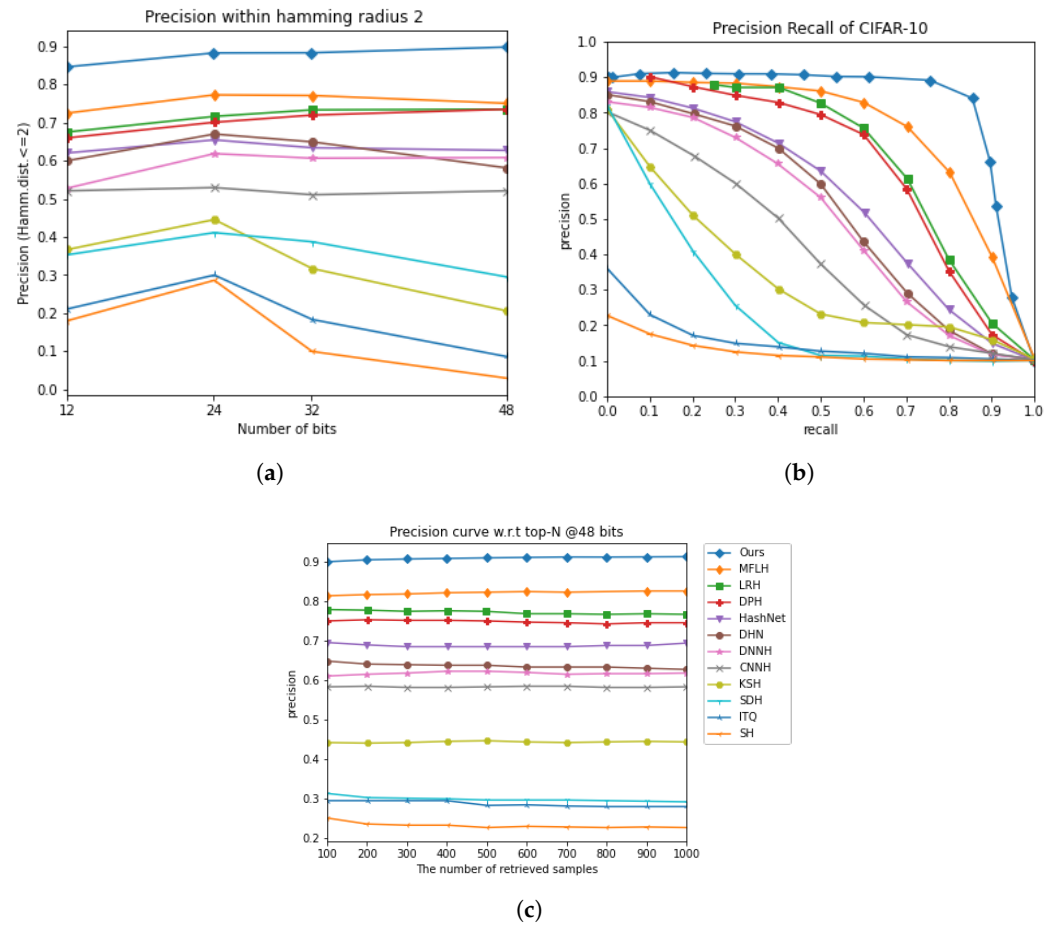


Figure 2. The comparison results on the CIFAR-10 dataset under three evaluation metrics. (a) Precision within Hamming radius 2. (b) Precision recall curve on 48 bits. (c) Precision curve with respect to top-N @48 bits.

4.5. 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 convs 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, where it achieved an average MAP of 85.5%, and thus, demonstrated the effectiveness of combining multi-scale features for enhanced retrieval performance.
- (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 the quantization loss, which minimizes the quantization error, and the classification loss, which preserves semantic information, for generating high-quality hash codes.

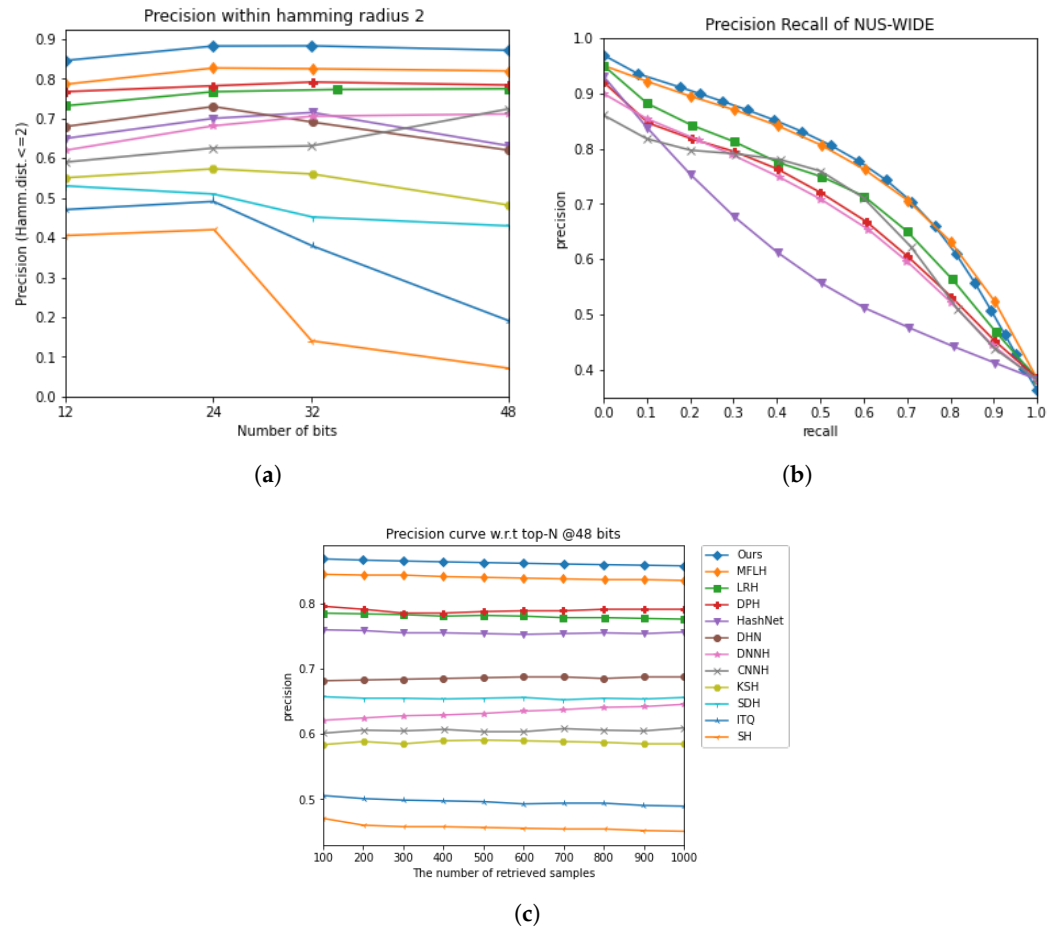


Figure 3. The comparison results on the NUS-WIDE dataset under three evaluation metrics. (a) Precision within Hamming radius 2. (b) Precision recall curve on 48 bits. (c) Precision curve with respect to top-N @48 bits.

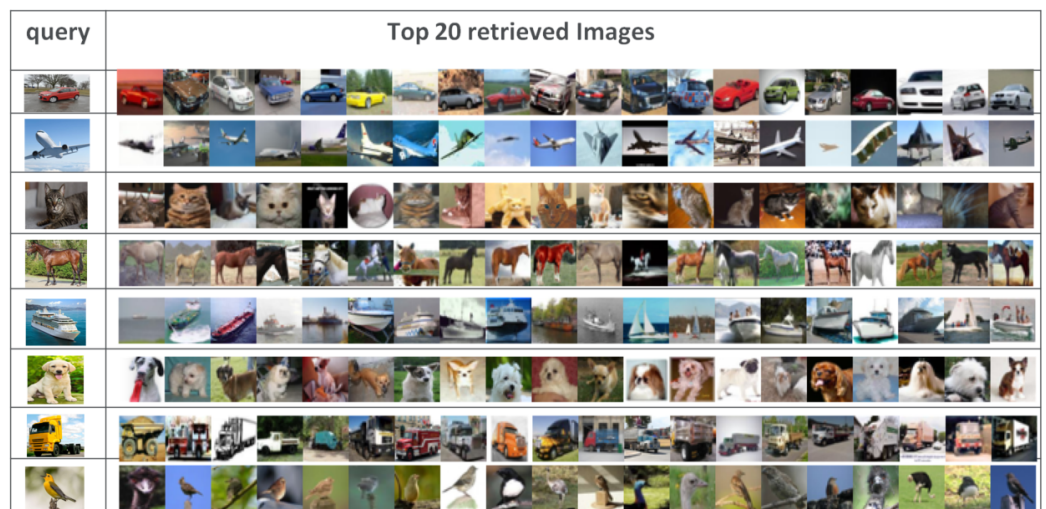


Figure 4. Presented are the top 20 retrieved results from the CIFAR-10 dataset, which utilized MDFF-SH with 48-bit hash codes. The first column showcases the query images, while the subsequent columns display the retrieval results generated by MDFF-SH.

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 |
| <i>fc1</i> | 0.710 | 0.761 | 0.775 | 0.788 |
| <i>Convs 3–5</i> | 0.580 | 0.595 | 0.639 | 0.688 |
| MDFP-SH | 0.811 | 0.854 | 0.874 | 0.880 |

Table 5. Mean Average Precision (MAP) results for different variants of the objective function on CIFAR-10.

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

5. 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 (MDFP-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, MDFP-SH achieves a balance between the structural detail and retrieval accuracy, leading to improved recall and precision.

Extensive experiments on standard image retrieval datasets demonstrate the superior performance of MDFP-SH compared with 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 A.R.; 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 read and agreed to the published version of this 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:

| | |
|----------|--|
| MDF-F-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 |

References

1. Yan, C.; Shao, B.; Zhao, H.; Ning, R.; Zhang, Y.; Xu, F. 3D room layout estimation from a single RGB image. *IEEE Trans. Multimed.* **2020**, *22*, 3014–3024. [CrossRef]
2. Yan, C.; Li, Z.; Zhang, Y.; Liu, Y.; Ji, X.; Zhang, Y. Depth image denoising using nuclear norm and learning graph model. *ACM Trans. Multimed. Comput. Commun. Appl. TOMM* **2020**, *16*, 1–17. [CrossRef]
3. Li, S.; Chen, Z.; Li, X.; Lu, J.; Zhou, J. Unsupervised variational video hashing with 1d-cnn-lstm networks. *IEEE Trans. Multimed.* **2019**, *22*, 1542–1554. [CrossRef]
4. Belloulata, K.; Belhallouche, L.; Belalia, A.; Kpalma, K. Region Based Image Retrieval using Shape-Adaptive DCT. In Proceedings of the Proceedings ChinaSIP-14 (2nd IEEE China Summit and International Conference on Signal and Information Processing), Xi'an, China, 9–13 July 2014; pp. 470–474.
5. Belalia, A.; Belloulata, K.; Kpalma, K. Region-based image retrieval in the compressed domain using shape-adaptive DCT. *Multimed. Tools Appl.* **2016**, *75*, 10175–10199. [CrossRef]
6. Gionis, A.; Indyk, P.; Motwani, R. Similarity search in high dimensions via hashing. In Proceedings of the VLDB, San Francisco, CA, USA, 7–10 September 1999; Volume 99, pp. 518–529.
7. Wang, J.; Zhang, T.; Sebe, N.; Shen, H.T.S. A survey on learning to hash. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 769–790. [CrossRef]
8. Erin Liong, V.; Lu, J.; Wang, G.; Moulin, P.; Zhou, J. Deep hashing for compact binary codes learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2475–2483.
9. Zhu, H.; Long, M.; Wang, J.; Cao, Y. Deep hashing network for efficient similarity retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
10. Lai, H.; Pan, Y.; Liu, Y.; Yan, S. Simultaneous feature learning and hash coding with deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3270–3278.
11. Cakir, F.; He, K.; Bargal, S.A.; Sclaroff, S. Hashing with mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2424–2437. [CrossRef] [PubMed]
12. Li, Q.; Sun, Z.; He, R.; Tan, T. Deep supervised discrete hashing. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2479–2488. Available online: <https://proceedings.neurips.cc/paper/2017/file/e94f63f579e05cb49c05c2d050ead9c0-Paper.pdf> (accessed on 31 July 2024).
13. Yue, C.; Long, M.; Wang, J.; Han, Z.; Wen, Q. Deep quantization network for efficient image retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 3457–3463.
14. Li, W.J.; Wang, S.; Kang, W.C. Feature learning based deep supervised hashing with pairwise labels. *arXiv* **2015**, arXiv:1511.03855.
15. Lu, J.; Liong, V.E.; Zhou, J. Deep hashing for scalable image search. *IEEE Trans. Image Process.* **2017**, *26*, 2352–2367. [CrossRef]
16. Lin, J.; Li, Z.; Tang, J. Discriminative Deep Hashing for Scalable Face Image Retrieval. In Proceedings of the IJCAI, Melbourne, VIC, Australia, 19–25 August 2017; pp. 2266–2272.
17. Jiang, Q.Y.; Li, W.J. Asymmetric deep supervised hashing. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
18. Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised hashing for image retrieval via image representation learning. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014.
19. Shen, F.; Gao, X.; Liu, L.; Yang, Y.; Shen, H.T. Deep asymmetric pairwise hashing. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 1522–1530.
20. Li, Y.; Xu, Y.; Wang, J.; Miao, Z.; Zhang, Y. Ms-rmac: Multiscale regional maximum activation of convolutions for image retrieval. *IEEE Signal Process. Lett.* **2017**, *24*, 609–613. [CrossRef]
21. Toliás, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2015**, arXiv:1511.05879.
22. Seddati, O.; Dupont, S.; Mahmoudi, S.; Parian, M. Towards good practices for image retrieval based on CNN features. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 1246–1255.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]

24. Zhao, Y.; Han, R.; Rao, Y. A new feature pyramid network for object detection. In Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Jishou, China, 14–15 September 2019; pp. 428–431.
25. Jin, Z.; Li, C.; Lin, Y.; Cai, D. Density sensitive hashing. *IEEE Trans. Cybern.* **2013**, *44*, 1362–1371. [[CrossRef](#)] [[PubMed](#)]
26. Andoni, A.; Indyk, P. Near-optimal hashing algorithms for near neighbor problem in high dimension. *Commun. ACM* **2008**, *51*, 117–122. [[CrossRef](#)]
27. Kulis, B.; Darrell, T. Learning to hash with binary reconstructive embeddings. *Adv. Neural Inf. Process. Syst.* **2009**, *22*. Available online: <https://proceedings.neurips.cc/paper/2009/file/6602294be910b1e3c4571bd98c4d5484-Paper.pdf> (accessed on 31 July 2024).
28. Liu, H.; Ji, R.; Wu, Y.; Liu, W. Towards optimal binary code learning via ordinal embedding. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
29. Wang, J.; Wang, J.; Yu, N.; Li, S. Order preserving hashing for approximate nearest neighbor search. In Proceedings of the 21st ACM International Conference on Multimedia, Barcelona, Spain, 21–25 October 2013; pp. 133–142.
30. Shen, F.; Shen, C.; Liu, W.; Shen, H.T. Supervised discrete hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 37–45.
31. Salakhutdinov, R.; Hinton, G. Semantic hashing. *Int. J. Approx. Reason.* **2009**, *50*, 969–978. [[CrossRef](#)]
32. Zhang, S.; Li, J.; Jiang, M.; Yuan, P.; Zhang, B. Scalable discrete supervised multimedia hash learning with clustering. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 2716–2729. [[CrossRef](#)]
33. Lin, M.; Ji, R.; Liu, H.; Sun, X.; Wu, Y.; Wu, Y. Towards optimal discrete online hashing with balanced similarity. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8722–8729.
34. Jégou, H.; Douze, M.; Schmid, C. Product Quantization for Nearest Neighbor Search. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 117–128. [[CrossRef](#)]
35. Weiss, Y.; Torralba, A.; Fergus, R. Spectral hashing. *Adv. Neural Inf. Process. Syst.* **2008**, *21*. Available online: <https://proceedings.neurips.cc/paper/2008/file/d58072be2820e8682c0a27c0518e805e-Paper.pdf> (accessed on 31 July 2024).
36. Gong, Y.; Lazebnik, S.; Gordo, A.; Perronnin, F. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 2916–2929. [[CrossRef](#)] [[PubMed](#)]
37. Liu, W.; Wang, J.; Kumar, S.; Chang, S.F. Hashing with graphs. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June 2011; pp. 1–8.
38. Datar, M.; Immorlica, N.; Indyk, P.; Mirrokni, V.S. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the Twentieth Annual Symposium on Computational Geometry, Brooklyn, NY, USA, 8–11 June 2004; pp. 253–262.
39. Liu, W.; Wang, J.; Ji, R.; Jiang, Y.G.; Chang, S.F. Supervised hashing with kernels. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2074–2081.
40. Norouzi, M.; Fleet, D.J. Minimal loss hashing for compact binary codes. In Proceedings of the ICML, Bellevue, WA, USA, 28 June 2011.
41. Cao, Y.; Liu, B.; Long, M.; Wang, J. Hashgan: Deep learning to hash with pair conditional wasserstein gan. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1287–1296.
42. Zhuang, B.; Lin, G.; Shen, C.; Reid, I. Fast training of triplet-based deep binary embedding networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5955–5964.
43. Liu, B.; Cao, Y.; Long, M.; Wang, J.; Wang, J. Deep triplet quantization. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Republic of Korea, 22–26 October 2018; pp. 755–763.
44. Yang, H.F.; Lin, K.; Chen, C.S. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 437–451. [[CrossRef](#)]
45. Wang, M.; Zhou, W.; Tian, Q.; Li, H. A general framework for linear distance preserving hashing. *IEEE Trans. Image Process.* **2017**, *27*, 907–922. [[CrossRef](#)]
46. Shen, F.; Xu, Y.; Liu, L.; Yang, Y.; Huang, Z.; Shen, H.T. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 3034–3044. [[CrossRef](#)] [[PubMed](#)]
47. Ma, L.; Luo, X.; Hong, H.; Meng, F.; Wu, Q. Logit Variated Product Quantization Based on Parts Interaction and Metric Learning With Knowledge Distillation for Fine-Grained Image Retrieval. *IEEE Trans. Multimed.* **2024**, *26*, 10406–10419. [[CrossRef](#)]
48. Yang, Y.; Geng, L.; Lai, H.; Pan, Y.; Yin, J. Feature pyramid hashing. In Proceedings of the 2019 International Conference on Multimedia Retrieval, Ottawa, ON, Canada, 10–13 June 2019; pp. 114–122.
49. Redaoui, A.; Belloulata, K. Deep Feature Pyramid Hashing for Efficient Image Retrieval. *Information* **2023**, *14*, 6. [[CrossRef](#)]
50. Redaoui, A.; Belalia, A.; Belloulata, K. Deep Supervised Hashing by Fusing Multiscale Deep Features for Image Retrieval. *Information* **2024**, *15*, 143. [[CrossRef](#)]
51. 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. [[CrossRef](#)]
52. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.

53. 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 Trans. Multimed.* **2019**, *22*, 215–228. [[CrossRef](#)]
54. 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. In Proceedings of the ACM International Conference on Image and Video Retrieval, Santorini Island, Greece, 8–10 July 2009; pp. 1–9.
55. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
56. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
57. Hinton, N.S.G.; Swersky, K. Overview of Mini Batch Gradient Descent. In *Proceedings of the Computer Science Department, University of Toronto*; N. 8; Coursera: Mountain View, CA, USA, 2012; Volume 575.
58. Jiang, Q.Y.; Li, W.J. Scalable graph hashing with feature transformation. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
59. Wang, J.; Kumar, S.; Chang, S.F. Semi-supervised hashing for large-scale search. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2393–2406. [[CrossRef](#)] [[PubMed](#)]
60. Cao, Y.; Long, M.; Liu, B.; Wang, J. Deep cauchy hashing for hamming space retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1229–1237.
61. Cao, Z.; Long, M.; Wang, J.; Yu, P.S. Hashnet: Deep learning to hash by continuation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5608–5617.
62. Sun, Y.; Yu, S. Deep Supervised Hashing with Dynamic Weighting Scheme. In Proceedings of the 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), Xiamen, China, 8–11 May 2020; pp. 57–62.
63. Bai, J.; Ni, B.; Wang, M.; Li, Z.; Cheng, S.; Yang, X.; Hu, C.; Gao, W. Deep progressive hashing for image retrieval. *IEEE Trans. Multimed.* **2019**, *21*, 3178–3193. [[CrossRef](#)]
64. Feng, H.; Wang, N.; Tang, J.; Chen, J.; Chen, F. Multi-granularity feature learning network for deep hashing. *Neurocomputing* **2021**, *423*, 274–283. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.