Deep Supervised Hashing for Image Retrieval

Nguyen Ba Cong
22127046
nbcong22@clc.fitus.edu.vn

Dang Tran Anh Khoa
22127024
dtakhoa22@clc.fitus.edu.vn

Abstract

This project presents Deep Supervised Hashing (DSH), a image retrieval method integrated by deep learning, enabling fast retrieval with minimal storage. However, DSH may suffer from quantization loss and suboptimal similarity preservation. Therefore, this project integrates DSH with a two-stage retrieval approach, where an initial fast search using hashing is followed by a refinement stage to improve ranking precision.

Contents

1 General

1.1 Motivation

With the exponential growth of image datasets, traditional image retrieval methods have become increasingly inefficient in terms of both accuracy and scalability. Conventional approaches struggle with the high-dimensional nature of image data, leading to slow retrieval speeds and excessive storage requirements. Modern databases contain millions of images, making retrieval computationally expensive in both time and memory.

To address these challenges, hashing-based techniques have been widely adopted. By encoding images into compact binary codes, they allow for fast similarity searches with reduced memory consumption. Among these methods, Deep Supervised Hashing (DSH) has emerged

as a powerful approach, leveraging deep learning to learn optimal hash functions that map high-dimensional image features into binary representations.

For DSH to be effective at scale, it must efficiently compute hash codes for each image and ensure that retrieval operations remain computationally feasible. Ideally, a well-trained hashing model enables near constant-time $O(1)$ lookups in hash tables for exact matches, while approximate nearest neighbor search in Hamming space remains significantly faster than exhaustive comparisons.

However, despite its advantages, existing DSH methods still face several challenges, including:

- Suboptimal hash code learning, leading to a loss of semantic information.
- Difficulty in preserving visual similarities, affecting retrieval accuracy.
- Sensitivity to noise, variations, and domain shifts, reducing robustness.
- Computational overhead, particularly during training and large-scale inference.
- Limited generalization across datasets, restricting real-world applicability.

Therefore, there is a need to explore and develop improved Deep Supervised Hashing techniques that enhance retrieval accuracy, efficiency, and robustness.

1.2 Problem Statement

This project aims to develop an efficient and accurate image retrieval system using deep learning. The system takes a query image as input and retrieves a ranked list of images that contain objects of the same type.

To be practical for large-scale databases, the retrieval process must operate under strict time and memory constraints while maintaining high accuracy. The model will be trained on a specific dataset and evaluated on its ability to generalize to unseen queries.

Key challenges include:

- Ensuring fast and scalable retrieval for large image datasets.
- Learning robust representations that preserve semantic similarity.
- Handling variations in object appearance, background clutter, and noise.

1.3 Contribution

Efficient image retrieval relies on encoding high-dimensional visual features into lower-dimensional representations while preserving semantic relationships. Hashing techniques have been widely adopted due to their ability to enable fast comparisons using Hamming distance. However, challenges such as semantic loss during hashing and sensitivity to data distribution limit their real-world effectiveness.

Along the general retrieval system implemented with deep hashing, we propose a two-stage retrieval approach can mitigate the aforementioned issues by leveraging hashing for an initial fast search, followed by a refinement stage that improves ranking precision. This study explores:

- The strengths and weaknesses of various DSH methods under this two-stage framework.
- The impact of different ranking refinement strategies on retrieval accuracy.
- A fair and standardized evaluation of these techniques across diverse datasets.

2 Related Work

Efficient image retrieval has become an essential task in Computer Vision, particularly for large-scale datasets like CIFAR-10. Traditional nearest neighbor search suffer from high computational cost, leading to the adoption of hashing techniques for approximate nearest neighbor retrieval. ANN methods facilitate faster search times by approximating the near-

est neighbors, making them suitable for large-scale applications where exact searches are impractical.

Hashing-based methods have emerged as a prominent solution for ANN retrieval by mapping high-dimensional image features into compact binary codes. These binary representations enable rapid and memory-efficient similarity computations. Early hashing approaches such as Locality-Sensitive Hashing (LSH) [1] use random projections to preserve similarity, but they require long hash codes for accurate retrieval, which increase storage costs. Spectral Hashing (SH) [2] and Iterative Quantization (ITQ) [3] introduced data-dependent projections, improving code efficiency but still lacking supervised learning capabilities.

To address the limitations of unsupervised hashing methods, Deep Supervised Hashing (DSH) techniques have been developed, integrating deep learning models to learn optimal hash codes from labeled data. A seminal work in this area is the Deep Supervised Hashing (DSH) method. [4] In this approach, a convolutional neural network (CNN) is trained with pairs of images labeled as similar or dissimilar. The network is designed to produce binary-like outputs that preserve semantic similarities, effectively capturing complex image variations. The loss function encourages the network's outputs to approximate discrete binary values (e.g., $+1$ or -1), facilitating the generation of compact and discriminative hash codes. This method demonstrated significant improvements in retrieval performance on large-scale datasets. Methods such as Deep Pairwise Supervised Hashing (DPSH) [5] and HashNet [6] incorporate pairwise similarity or triplet-based loss functions to generate more discriminative binary codes. However, these approaches often require complex training strategies and large amount of labeled data.

Recent work explores auto-encoder-based hashing, which utilizes deep auto-encoders to jointly learn feature representations and binary codes. Deep Supervised Auto-Encoder Hashing (SAEH) extends this approach by integrating supervised constraints within an auto-encoder framework, ensuring that the learned hash codes preserve semantic similarities while maintaining compactness and retrieval efficiency. This approach effectively balances reconstruction quality and hash discriminability.

A critical challenge in hash-based retrieval is balancing speed and accuracy. To address this, two-stage retrieval approaches have been proposed. In the first stage, an initial hash-based search performs coarse candidate selection, rapidly narrowing down the search space. The second stage involves a more refined similarity ranking using deep feature embeddings. For example, methods like deep feature re-ranking [7] utilize deep neural networks to extract rich feature representations, which are then used to re-rank the initially retrieved candidates, enhancing retrieval precision. By implementing a two-stage retrieval pipeline that combines the efficiency of hashing with the precision of deep feature-based ranking, retrieval performance on datasets like CIFAR-10 can be significantly optimized.

3 Methodology

Our goal is to create an efficient image hashing system, in which the binary codes are compact, similar images should be encoded to similar binary codes in Hamming space, and the binary codes should be computed efficiently.

In this project, we implement Deep Supervised Auto-Encoder Hashing, followed by a two-stage retrieval pipeline, where SAEH-generated hash codes provide fast approximate search.

Retrieving images is a process of two steps. Step 1 is fast approximate search by SAEH Hash Codes, in which images are encoded into compact binary hash codes using SAEH. A hash-based search quickly retrieves a coarse set of candidate images that are similar to the query.

3.1 Data Preparation

We utilize the CIFAR-10 dataset, comprising 60k 32×32 color images across 10 classes, with 6k images per class. Each image is normalized to have zero mean and unit variance.

3.2 Stage 1: SAEH

The SAEH model integrates supervised learning within an auto-encoder architecture to generate compact binary hash codes for images. This model consists of three primary components: an encoder, a supervisory network, and a decoder.

The encoder employs a convolutional neural network (CNN) to extract high-level features from input images and map them to a lower-dimensional space, producing initial hash codes. Given an input image x, the encoder function $E(x;\theta_e)$ maps the image into a feature space h, where $h = E(x;\theta_e)$. The feature space representation is then binarized into hash codes b, where $b = \text{sign}(h)$.

The supervisory network introduces supervised constraints by enforcing similarity preservation. It employs a classification loss to ensure that images from the same class have similar hash codes. Given class label y, a cross-entropy loss L_{cls} is defined as:

$$L_{\text{cls}} = -\Sigma y_i \log(\hat{y}_i) \tag{1}$$

where $\hat{y}_i$ is the predicted class probability for image x_i. Additionally, a pairwise similar loss is applied to minimize the Hamming distance between similar images:

$$L_{\text{sim}} = \Sigma_{i,j} S_{ij} \parallel b_i - b_j \parallel^2 \tag{2}$$

where S_{ij} is a similarity matrix where $S_{ij} = 1$ if x_i and x_j belong to the same class, and $S_{ij} = 0$ otherwise.

The decoder reconstructs the original image from the hash codes to maintain essential image features. The reconstruction loss L_{rec} is given by:

$$L_{\text{rec}} = \Sigma_i \parallel x_i - D(b_i;\theta_D) \parallel^2 \tag{3}$$

where $D(b;\theta_D)$ is the decoder function that reconstructs the image from hash codes.

The final loss function combines classification, similarity, and reconstruction losses with weighting factors λ_1 and λ_2:

$$L = L_{\text{cls}} + \lambda_1 L_{\text{sim}} + \lambda_2 L_{\text{rec}} \tag{4}$$

The training process involves in optimizing L to jointly minimize classification error, ensure similar images have similar hash codes, and preserve essential structure for image retrieval.

3.3 Stage 2: Deep Feature Re-ranking

After retrieving an initial candidate set using SAEH hash codes, we apply deep feature re-ranking to refine the hash results. The goal of this step is to improve the ranking accuracy by leveraging richer image representations. Instead of relying solely on binary hash codes, deep feature re-ranking computes feature distances in a continuous space for improved similarity estimation.

1. Feature Extraction: For each retrieved image, we extract deep feature embeddings from an intermediate layer of the SAEH encoder. Let $f_1 = F(x_i, \theta_F)$ represent the feature vector extracted from image x_i, where θ_F denotes the learned parameters of the encoder.
2. Similarity Computation: Instead of using Hamming distance from hash codes, we compute pairwise distances between deep features.

$$S_{ij} = \frac{f_i \cdot f_j}{\|f_i\| \, \|f_j\|} \tag{5}$$

where S_{ij} measures the similarity between query image x_i and retrieved image x_j.

3. Re-ranking: The initial retrieval list is re-ranked based on the refined similarity scores. Given an initial list R_0 from the hash-based search, the re-ranked list R_1 is computed as:

$$R_1 = \text{Sort}\big(R_0, S_{ij}\big) \tag{6}$$

4. Final Retrieval: The top-ranked images from R_1 form the final retrieval set, improving ranking precision over the initial hash-based results.

3.4 Expected Outcomes

- A clear comparative analysis of DSH methods under the two-stage retrieval framework.
- Identification of the best-performing refinement techniques for improving accuracy.
- Practical insights into balancing retrieval speed and ranking precision.
- Comprehensive benchmarking results that position the proposed method as a leader in learning-based hashing for image retrieval.

4 Experiment

Code: https://github.com/KhoaEzh/CV-Hashing-Project

4.1 Dataset

CIFAR-10 consists of 60k 32×32 color images of ten color objects, including 50k train images and 10k test images.

Bibliography

[1] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Commun. ACM*, vol. 51, no. 1, pp. 117–122, Jan. 2008, doi: 10.1145/1327452.1327494.

[2] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, in Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference. Neural Information Processing Systems, 2009, pp. 1753–1760.

[3] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013, doi: 10.1109/TPAMI.2012.193.

[4] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep Supervised Hashing for Fast Image Retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[5] W.-J. Li, S. Wang, and W.-C. Kang, "Feature Learning based Deep Supervised Hashing with Pairwise Labels." [Online]. Available: https://arxiv.org/abs/1511.03855

[6] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep Learning to Hash by Continuation." [Online]. Available: https://arxiv.org/abs/1702.00758

[7] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end Learning of Deep Visual Representations for Image Retrieval." [Online]. Available: https://arxiv.org/abs/1610.07940

A Appendix