

Proposal: A Comprehensive Survey on Learning Hash for Image Retrieval

1st Ba Cong Nguyen

*Faculty of Information Technology
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam
nbcong22@clc.fitus.edu.vn*

2nd Tran Anh Khoa Dang

*Faculty of Information Technology
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam
dtakhoa22@clc.fitus.edu.vn*

Abstract—The rapid growth of image datasets has made traditional image retrieval methods inefficient, both in terms of accuracy and scalability. Hashing-based techniques provide a promising solution by encoding high-dimensional image data into compact binary codes, enabling fast retrieval with minimal storage requirements. In this paper, we introduce Deep Supervised Hashing, an advanced approach for retrieving images from large-scale datasets. This method leverages deep learning to generate highly discriminative hash codes, improving retrieval accuracy while maintaining computational efficiency. Although Deep Supervised Hashing has demonstrated its effectiveness in large-scale image search tasks, there remains substantial room for further enhancement. Various factors, such as the optimization of hash function learning, the handling of semantic relationships in image data, and the robustness of hash codes against variations, present opportunities for improvement. To address these challenges, we explore several advancements in Deep Supervised Hashing methodologies. By analyzing existing techniques and identifying potential refinements, we aim to contribute to the ongoing development of more accurate, efficient, and scalable image retrieval systems.

Index Terms—hashing, image retrieval, deep learning

I. INTRODUCTION

Hashing is a cornerstone of efficient image retrieval due to its ability to encode high-dimensional image features into compact binary representations. These binary hash codes allow for quick comparisons using Hamming distance, making hashing an essential technique for real-time applications.

However, current approaches often fail to maximize the potential of hashing. Many hashing methods inadequately preserve semantic relationships between images, leading to reduced precision. Challenges in scalability limit the deployment of hashing techniques for massive datasets. The dynamic nature of visual content demands adaptive and robust hashing methods.

Deep learning offers an unprecedented opportunity to revolutionize hashing by enabling end-to-end learning frameworks where hash codes are directly optimized for retrieval tasks. This research seeks to push the boundaries of hashing methods, blending computational efficiency with semantic richness.

A. Problem Statement

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 to handle the high-dimensional nature of image data, leading to slow retrieval speeds and excessive storage requirements. Hashing-based techniques offer a promising alternative by encoding images into compact binary codes, allowing for faster search operations with reduced memory consumption.

Despite the advancements in hashing-based retrieval, existing methods, particularly Deep Supervised Hashing, still face challenges. These include suboptimal hash code learning, difficulty in preserving semantic similarities, sensitivity to noise and variations in images, and computational overhead during training. Furthermore, many current models lack the ability to generalize well across diverse datasets, limiting their applicability in real-world scenarios.

Therefore, there is a need to explore and develop improved Deep Supervised Hashing techniques that enhance retrieval accuracy, efficiency, and robustness. This research aims to investigate potential improvements to hashing methodologies, addressing key limitations and contributing to the development of more effective large-scale image retrieval systems.

II. RELATED WORKS

The field of image retrieval has seen significant advancements in hashing techniques. Some notable contributions include:

Deep Supervised Hashing (DSH):

Training/Learning methodology:

The training methodology initially considers the use of a Siamese structure for generating image pairs offline. However, this approach faces limitations: for n images, it would produce only $n/2$ valid pairs, and storing the image pairs would require significant storage space. To overcome these challenges, we propose generating image pairs online by utilizing all unique pairs within each mini-batch. During training, images are randomly sampled from the entire dataset in each iteration, ensuring that pairs across batches are effectively covered. This approach eliminates the need to store a complete pair-wise similarity matrix, making it highly scalable for large datasets.

Additionally, training separate models from scratch for varying code lengths would be resource-intensive and inefficient. Instead, preceding layers of the network can be shared across

models to save computational resources. However, as code length increases, the output layer grows in complexity with additional parameters, increasing the risk of overfitting. This necessitates careful design and training strategies to ensure performance and scalability. Feature Learning-based Deep Supervised Hashing with Pairwise Labels (A.k.a. deep pairwise-supervised hashing (DPSH)) [3]

Feature Learning-based Deep Supervised Hashing with Pairwise Labels

Training/Learning methodology

The training data consists of three key components: a labeled set of samples (initially small, known as the "seed"), a large pool of unlabeled samples, and a query set containing selected samples for which labels are obtained from an Oracle. Initially, only a small number of samples are labeled to start the active deep hashing process. In each cycle of active learning, the steps are as follows: (1) train the hashing model using pairs sampled from the labeled set; (2) identify and select the most informative samples from the unlabeled pool using a selection function and add these to the query set; and (3) request the Oracle to label the query samples and update the labeled set, the unlabeled pool, and the query set accordingly. The hash model generates binary codes by projecting the data through the network, with binary values assigned through a thresholding step (e.g., values may be set to either 0 or 1 based on their sign). Using a supervised hashing approach, a loss function is applied to train the model. The goal of the loss function is to minimize the distance in the binary code space for similar samples while maximizing the distance for dissimilar samples. This ensures that the semantic relationships among the data points are captured effectively in the learned binary codes. As directly optimizing binary codes is computationally difficult and non-differentiable, the process is relaxed by approximating the binary constraints with continuous values. The optimization process also balances multiple objectives, including pairwise supervision and reducing quantization error, through adjustable hyperparameters. To train the hash model, pairs of labeled samples are used to learn the binary codes effectively. This method can also be combined with semi-supervised techniques to further enhance performance, benefiting from the inclusion of both labeled and unlabeled data while leveraging active learning for supervision. [2]

Deep Supervised Discrete Hashing (DSDH or DDSH)

The main contributions of this work are as follows:

The last layer of the model is explicitly designed to output binary codes. These codes are optimized to preserve similarity relationships while ensuring consistency with label information. To the best of our knowledge, this is the first deep hashing framework that integrates both pairwise labels and classification information into a single-stream architecture for hash code learning.

To minimize quantization errors, we maintain the discrete properties of the hash codes throughout the optimization process. This is achieved using an alternating minimization

strategy, where the objective function is optimized via the discrete cyclic coordinate descent method.

Comprehensive experiments demonstrate that our method surpasses existing state-of-the-art approaches on benchmark datasets for image retrieval, validating the effectiveness of the proposed framework.

This approach highlights the innovative combination of end-to-end learning with discrete optimization to achieve high-performance hash code generation.

Training/Learning methodology

The k -th bit of all points is determined by minimizing an objective function, with constraints ensuring that the bit values remain binary (either -1 or +1). The formulation includes terms for pairwise similarities and previously computed bits, ensuring optimal clustering for the binary hash codes.

For the hash function, the non-differentiable nature of the sign function is addressed by relaxing it to a differentiable alternative, like the hyperbolic tangent. This adjustment enables the use of backpropagation for updating the neural network's parameters. The process involves sampling mini-batches, calculating gradients, and refining parameters iteratively. The overall learning algorithm for this discrete hashing model is summarized in an outlined step-by-step procedure. The algorithm adopts an alternating optimization strategy to iteratively refine both the hash codes and the neural network parameters. When optimizing the binary hash codes, each bit is updated individually by considering its impact on the overall objective function. This ensures that the resulting hash codes maintain their discrete binary nature while minimizing errors. For the neural network parameters, backpropagation is used to adjust the model based on the fixed binary codes. This involves computing gradients for all layers, with adjustments propagating backward to improve alignment between the network's outputs and the desired hash codes. The optimization alternates between updating the binary codes and fine-tuning the network, converging on a solution that balances both components effectively. [1]

III. METHODOLOGY

In this project, we aim to compare various deep learning based hashing models (or methods) in a multitude of aspects (such as training, runtime and general efficiency of resources) to hopefully find out the most effective model in general. The models will all be fed the same dataset and operated on the same device to ensure that the comparison is as impartial as possible, as different machines' processing power and datasets certainly will make or break some parts of the algorithms. If our goal is to be accomplished, then there are certain steps to follow:

- 1) Finding at least one suitable dataset to feed into the models. Some proposed datasets are MNIST digits dataset, MNIST Kuzushiji dataset and Cifar 10 dataset.
- 2) Implementing, evaluating and making inferences to assess each individual hashing model using Python (more specifically, Pytorch).

- 3) Draw conclusions based on the performance of each model using the datasets, while comparing certain aspects to see if any model exceeds or falls short in one key point or another.

IV. EXPECTED OUTCOMES

- 1) Clear comparisons and analysis on various techniques.
- 2) High-quality binary hash codes that achieve a balance between semantic richness and compactness.
- 3) Scalability of the system to handle large datasets efficiently while maintaining high retrieval accuracy.
- 4) Comprehensive benchmarking results that position the proposed method as a leader in learning-based hashing for image retrieval.

ACKNOWLEDGMENT

REFERENCES

- [1] Qi Li, Zhenan Sun, Ran He, and Tieniu Tan. Deep supervised discrete hashing. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 2479–2488, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [2] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. Feature learning based deep supervised hashing with pairwise labels, 2016.
- [3] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.