electronics

Article

Study on an Image Candidate Generation Method Based on PCA-Guided Projection

Yang Sun [1,2], Tianqi Wu [1,2], Ma Li [1,3] and Yubo Song [1,2,*]

[1] School of Cyber Science and Engineering, Frontiers Science Center for Mobile Information Communication and Security, Southeast University, Nanjing 211189, China; sunyangsy@seu.edu.cn (Y.S.); wutianqi1997@seu.edu.cn (T.W.); 230250288@seu.edu.cn (M.L.)
[2] Purple Mountain Laboratories, Nanjing 211111, China
[3] Nanjing Medical Security Data Center, Nanjing 210000, China
* Correspondence: songyubo@seu.edu.cn

Abstract

Image retrieval often begins with candidate generation, where a small subset of database images is selected for subsequent reranking or inspection. Standard multi-table p-stable locality-sensitive hashing (LSH) is a classical solution for this stage, but its isotropically sampled projection vectors do not explicitly exploit the principal structure of deep feature distributions. To address this limitation, this paper proposes a PCA-guided candidate generation method using the multi-table bucketization and query architecture of p-stable LSH. Unlike conventional PCA-based feature extraction, PCA is not used here to transform the image features themselves, but to bias the sampling distribution of projection vectors during indexing. Experiments on CALTECH101, CIFAR-10, and Tiny-ImageNet, performed using VGG19, ConvNeXt-Tiny, and ViT-B/16 features, show that the proposed method generally improves on the standard p-stable baseline across different settings. Additional comparisons with HNSW and IVF further position the method relative to modern ANN baselines. The results indicate that PCA-guided projection is an effective lightweight enhancement for systems based on the multi-table p-stable LSH architecture, improving candidate coverage and robustness while remaining compatible with the original online query process.

Keywords: image candidate generation; locality-sensitive hashing; principal component analysis; deep features; data-dependent projection

Check for updates

Academic Editors: Young-Ho Seo and George A. Tsihrintzis

Received: 26 July 2026
Revised: 26 August 2026
Accepted: 1 September 2026
Published: 3 September 2026

Copyright: © 2026 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.

1. Introduction

Image retrieval systems typically follow a staged processing pipeline, in which the candidate generation stage is responsible for quickly screening a background image database to obtain a small candidate set that may contain the target sample, thereby providing input for subsequent reranking or manual inspection [1]. At this stage, the primary objective is not to make a final decision directly, but rather to maximize coverage of the target sample under a limited candidate budget. Accordingly, the effectiveness of candidate generation depends largely on how well the indexing structure preserves neighborhood relationships in the feature space.

Locality-Sensitive Hashing (LSH) has long been used for candidate generation in nearest-neighbor retrieval because of its simple structure, relatively high query efficiency, and suitability for high-dimensional feature spaces [2,3]. Under Euclidean distance settings, p-stable distribution-based LSH preserves neighborhood relationships through random

projections and a multi-hash-table mechanism, making it one of the classical approaches to candidate generation [4]. Although this class of methods offers advantages such as low implementation cost and a straightforward indexing procedure, its projection vectors are typically generated by isotropic random sampling and therefore do not explicitly leverage the intrinsic structure of the data distribution.

For deep visual features, sample distributions are often not uniformly distributed throughout the entire high-dimensional space, but instead exhibit a principal-direction structure. Deep feature-based image representation methods have become a major technical approach in image retrieval [5,6]. When the projection directions are misaligned with the dominant directions of variation in the data, hash partitioning may fail to adequately preserve meaningful neighborhood relationships. As a result, target coverage under a fixed candidate budget may be reduced, and candidate generation may become more sensitive to parameter settings and content-preserving perturbations [7,8]. Motivated by this observation, this paper investigates a specific problem in candidate generation: while preserving the standard multi-table LSH framework and the basic query procedure, can the principal-direction information of image-database features be used to lightly guide the projection generation process, thereby improving the coverage of true target neighbors by the candidate set and enhancing result stability under different perturbation conditions?

To this end, this paper proposes an image candidate generation method based on PCA-guided projection. The proposed method uses the principal component information of image-database features to guide projection generation, thereby introducing data-distribution awareness into random projections while preserving the standard hash bucketization scheme and multi-table retrieval framework. Unlike conventional PCA-based feature extraction, PCA is not used here to transform or compress the image features themselves, but only to bias the sampling distribution of projection vectors in the indexing stage. The objective of this work is not to extend the classical collision-probability theory of p-stable LSH, but to investigate an empirical data-dependent modification of its offline projection sampling while retaining the original hash-function form, multi-table bucket organization, and online candidate-collection procedure.

To evaluate the above idea, this paper retains CALTECH101 as the main controlled setting for parameter sensitivity analysis, sensitivity analysis of the number of principal components, and perturbation-stability evaluation, and further includes CIFAR-10 and Tiny-ImageNet for a multi-dataset comparison. In addition to the original VGG19-based setting, ConvNeXt-Tiny and ViT-B/16 are introduced as representative modern CNN-based and Transformer-based feature extractors, respectively. Moreover, HNSW and IVF are included to position the proposed method relative to modern ANN baselines. The main contributions of this paper are summarized as follows:

1. This paper proposes a PCA-guided probabilistic projection-sampling strategy for image candidate generation using the hash-function form and multi-table retrieval architecture of p-stable LSH. Unlike PCA-based hashing methods that directly transform or compress feature vectors and subsequently produce compact binary codes, the proposed method leaves the database and query descriptors unchanged and uses PCA to parameterize the offline sampling distribution of projection vectors. Unlike learned hashing and more general data-dependent hashing methods, it does not optimize binary codes, similarity objectives, or data-dependent partitions. The methodological innovation therefore lies in introducing principal-direction information at the level of the projection-sampling distribution, thereby establishing a data-dependent projection mechanism that retains randomized projection diversity as well as the original bucketization scheme and online candidate-collection procedure.

2. This paper develops an experimental evaluation scheme focused on the candidate generation stage. CALTECH101 is used for detailed parameter sensitivity analysis, sensitivity analysis of the number of principal components, and perturbation-stability evaluation, while CIFAR-10 and Tiny-ImageNet are included in a multi-dataset and cross-backbone comparison. Additional experiments with ConvNeXt-Tiny, ViT-B/16, HNSW, and IVF place the proposed method in a more current retrieval setting. The results show that the proposed PCA-guided strategy generally improves over the standard p-stable baseline across different settings. While modern ANN baselines such as HNSW and IVF still achieve stronger overall recall, the proposed method provides a compatible data-dependent projection design for the multi-table p-stable LSH architecture without modifying its bucketization scheme or online query process.

The remainder of this paper is organized as follows: Section 2 reviews recent image retrieval, approximate nearest-neighbor indexing, data-dependent hashing, and PCA-based hashing methods. Section 3 defines the candidate generation problem and introduces the p-stable LSH baseline. Section 4 presents the proposed PCA-guided projection-sampling mechanism, index construction procedure, and complexity analysis. Section 5 describes the experimental settings and reports the parameter, perturbation-stability, and multi-dataset comparisons. Section 6 concludes the paper and discusses its scope and limitations.

2. Related Work

Image retrieval has progressively evolved from systems based on hand-crafted descriptors to pipelines built on deep visual representations [9,10]. Recent research has further investigated verification and reranking mechanisms, stronger retrieval benchmarks, and composed or cross-modal retrieval settings [11–15]. More recent instance-level retrieval methods have also explored patch-level representations and product quantization to improve spatial interpretability and large-scale efficiency [16]. Beyond retrieval-specific research, recent application-oriented visual systems such as JassidNet and ToRLNet demonstrate the growing importance of quantization-aware computation, lightweight representation, and deployment under constrained computational resources [17,18]. Although these studies address visual detection and phenotyping rather than candidate indexing, their emphasis on lightweight integration reflects a broader practical demand for improving individual processing modules without redesigning an entire visual system. In this context, the present work isolates the front-end candidate generation problem for fixed global image descriptors and investigates a lightweight modification of projection sampling under a limited candidate budget.

Alongside advances in visual representation, approximate nearest neighbor (ANN) indexing methods have become strong baselines for large-scale retrieval. Graph-based ANN search represented by HNSW has shown strong empirical performance in high-recall nearest-neighbor search [19,20]. Quantization-based large-scale similarity search systems, such as FAISS/IVF-style indexing, are also widely adopted in practice because of their favorable efficiency–accuracy trade-off [21]. These ANN methods are included to position the proposed approach relative to stronger contemporary indexing baselines.

Within the classical hashing family, locality-sensitive hashing (LSH) remains a representative approach for fast candidate generation in Euclidean feature spaces. In particular, p-stable LSH uses random projections derived from stable distributions and has long served as a standard baseline for approximate search [4]. Spectral Hashing introduced a representative data-dependent hashing formulation from the perspective of similarity-preserving binary coding [22]. Iterative Quantization further improved the quantization quality of learned binary embeddings through rotation optimization [23]. Kernelized LSH extended locality-sensitive hashing to more flexible nonlinear similarity settings [24]. More

recent hashing-based retrieval studies have further incorporated deep representations and graph-structured learning [25,26].

PCA has also been used directly in image hashing and retrieval. PCA-based hashing methods generally project descriptors into a lower-dimensional principal subspace and then quantize the transformed representations into compact binary codes [27]. Recent PCA-assisted image hashing similarly uses PCA for feature compression before binary-code generation [28]. In these methods, PCA directly changes the representation on which similarity search is performed. In the proposed method, neither the database descriptor nor the query descriptor is PCA-transformed or binarized. PCA is used only to construct a guidance direction for the offline sampling of projection vectors, after which the original real-valued hash-function form, bucketization scheme, and online candidate-collection procedure are retained.

Accordingly, the proposed method constitutes a distinct data-dependent projection-sampling design within the multi-table p-stable LSH architecture, rather than a binary representation-learning method or a replacement for modern ANN indices. Its methodological distinction lies in the interface at which the data distribution is introduced: standard p-stable LSH uses isotropic data-independent sampling; PCA hashing transforms and quantizes the feature representation; learned hashing optimizes codes, projections, or similarity objectives; and the proposed method uses principal-component statistics to parameterize the probability distribution from which randomized projection vectors are sampled. This design introduces data awareness without transforming the original descriptors or redesigning the online candidate-generation process.

3. Problem Definition

3.1. Image Representation

The goal of the candidate generation stage is to quickly screen a background image database under a limited online computational budget in order to obtain a candidate set likely to contain the target sample, thereby providing input for subsequent reranking or manual inspection. Accordingly, this stage is primarily concerned with the coverage of the target sample by the candidate set and the recall performance achieved at a given return size, rather than the accuracy of the final decision.

Image representation directly determines the form of the input to the subsequent indexing structure. When low-level features such as pixels, color histograms, or local textures are used, images may exhibit substantial appearance variations under content-preserving perturbations such as noise, compression, and geometric transformations, thereby weakening neighborhood relationships between similar samples in the feature space. In contrast, high-level visual features extracted by deep convolutional networks can retain semantic and global structural information to some extent and provide more stable representations under common content-preserving perturbations. They are therefore better suited as indexing inputs for the candidate generation stage.

Based on the above considerations, this paper uses deep features to represent image content. In the main controlled setting, VGG19 is adopted as the default feature extraction network [29]. VGG19 has a well-established architecture and mature public implementations, which facilitates reproduction of the experimental pipeline in a unified experimental setting. In addition, its high-level global representation has a fixed dimensionality, allowing consistent integration with the subsequent indexing module. In this work, VGG19 weights pretrained on ImageNet are used, and the network parameters are frozen during feature extraction. After standardized preprocessing, each input image is passed through the network to obtain a fixed-dimensional global feature vector, which serves as a unified representation for both offline database construction and online query processing. Specifically, each input

image is resized to a color tensor of $256 \times 256 \times 3$, and the final output is a global feature vector of dimension 1×4096.

To examine whether the proposed candidate generation strategy depends on a specific visual backbone, additional pretrained feature extractors are also considered in the comparative experiments. Specifically, ConvNeXt-Tiny is used as a representative modern CNN-based feature extractor [30], and ViT-B/16 is used as a representative Transformer-based vision model [31]. Unless otherwise stated, all backbone parameters are frozen, and the extracted descriptors are only used as the input feature space for subsequent indexing and candidate generation.

To reduce the effect of feature scale differences on the subsequent random projection and bucketization process, L2 normalization is applied to the extracted feature vectors. This step does not change the semantic information carried by the features; rather, it places different images on a more consistent numerical scale before indexing, thereby improving the stability of indexing parameters across different query samples. The normalized feature vectors collectively form the feature set of the background image database and serve as the input to the subsequent candidate generation model.

On this basis, this paper formulates the candidate generation stage as an approximate nearest-neighbor search problem in feature space. Let the background image database, after feature extraction, be represented as a set of feature vectors

$$\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^{N} \tag{1}$$

where N is the number of images in the database and d is the feature dimensionality. For a given query image, its corresponding feature vector is denoted by $\mathbf{q} \in \mathbb{R}^d$. The objective of the candidate generation stage is to rapidly construct a size-controlled candidate set without exhaustively comparing the query against the entire database,

$$C(\mathbf{q}) \subseteq \mathcal{X}, \quad |C(\mathbf{q})| \leq N_c \tag{2}$$

so as to maximize the likelihood that the target sample is contained in $C(\mathbf{q})$.

3.2. p-Stable LSH Baseline

As the size of the background image database increases, directly computing distances between the query feature and all database features incurs a time complexity that grows linearly with database size, making it difficult to meet the response-time requirements of the online candidate generation stage. To reduce retrieval cost, candidate generation typically employs approximate indexing structures that confine the search to local candidate sets more likely to be relevant to the query. Locality-Sensitive Hashing (LSH) is a representative approximate indexing method. Its core idea is to map high-dimensional features into discrete buckets through random mappings, such that samples that are close in the original space fall into the same bucket with high probability, thereby enabling approximate nearest-neighbor search by examining samples in the same bucket.

In the Euclidean distance setting, this paper adopts p-stable distribution-based LSH as the baseline method for candidate generation. For any feature vector $\mathbf{v} \in \mathbb{R}^d$, the hash function is defined as

$$h_{\mathbf{a},b}(\mathbf{v}) = \lfloor \frac{\mathbf{a}^\top \mathbf{v} + b}{r} \rfloor \tag{3}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation, b is an offset sampled uniformly at random from $(0, r)$, and r is the bucket width. In standard p-stable LSH, $\mathbf{a}$ is a d-dimensional random vector drawn i.i.d. from a p-stable distribution. This formulation first projects the feature

vector onto a one-dimensional subspace defined by the random direction $\mathbf{a}$, and then quantizes the projection value using the bucket width r to obtain a discrete bucket index.

For two feature vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, denote their p-norm distance by $c = \|\mathbf{v}_1 - \mathbf{v}_2\|_p$. Under a single hash function, whether the two vectors fall into the same hash bucket depends on the relative magnitude of their projection difference along the random direction $\mathbf{a}$ with respect to the bucket width r. When $|\mathbf{a}^\top(\mathbf{v}_1 - \mathbf{v}_2)| < r$, the quantized values of the two vectors along that projection direction are more likely to coincide, causing them to fall into the same hash bucket. This indicates that the original distance between samples is reflected in bucket-collision behavior through random projection differences: samples that are closer are more likely to collide, whereas those that are farther apart have a relatively lower probability of collision. Therefore, hash bucket collisions can be used as a coarse-grained similarity criterion in the candidate generation stage to rapidly narrow the search space without exhaustively comparing the query against the entire database.

To improve the discriminative power of candidate generation, practical retrieval systems typically employ composite hashing together with a multi-table mechanism. Specifically, within a single hash table, k basic hash functions are concatenated to form a composite hash key

$$g_\ell(\mathbf{v}) = \left(h_{\mathbf{a}_{\ell 1}, b_{\ell 1}}(\mathbf{v}), h_{\mathbf{a}_{\ell 2}, b_{\ell 2}}(\mathbf{v}), \ldots, h_{\mathbf{a}_{\ell k}, b_{\ell k}}(\mathbf{v})\right) \tag{4}$$

where ℓ denotes the ℓ-th hash table. Increasing k improves bucket selectivity within a single table and reduces the probability of collisions among irrelevant samples, but it may also increase the risk that neighboring samples are assigned to different buckets. To mitigate this issue, L independent hash tables are typically constructed, and each database vector is inserted into all L tables. During querying, the composite hash key of a given query vector $\mathbf{q}$ is computed for each table, and the corresponding matched buckets are accessed. The samples retrieved from all matched buckets are then aggregated to form the candidate set $C(\mathbf{q})$.

Within this framework, candidate generation behavior is jointly determined by the bucket width r, the composite key length k, and the number of hash tables L. Specifically, the bucket width r controls the granularity of projection-value quantization, k determines the discriminative power of the composite hash key within a single table, and L compensates for potential recall loss caused by single-table partitioning through parallel lookup across multiple tables. A smaller bucket width and a larger k often reduce the number of samples per bucket and improve selectivity, but they may also decrease the probability that the target sample collides with the query. Conversely, a larger bucket width or a smaller k can improve the hit rate, but may lead to a larger candidate set and weaker filtering capability. Therefore, p-stable LSH essentially trades off candidate set size against target coverage through parameter configuration.

It should also be noted that the projection vectors in standard p-stable LSH are generated by isotropic random sampling. Because this design does not assume any prior knowledge of the data distribution, it offers strong general applicability. However, when the deep feature distribution exhibits a pronounced principal-direction structure, a fully isotropic projection generation strategy may fail to fully exploit the geometric structure of the data. Accordingly, this paper adopts standard p-stable LSH as the candidate generation baseline and further examines how projection vector design affects candidate generation performance while preserving its bucketization scheme and multi-table retrieval mechanism.

3.3. Candidate Generation Problem

Given the normalized database feature set $\mathcal{X}$, a query feature $\mathbf{q}$, and a candidate budget N_c, this paper considers projection-vector design under a fixed multi-table bucketization and candidate-collection procedure. Standard p-stable LSH samples projection vectors independently from an isotropic distribution. Although this sampling rule is independent of the database distribution, it does not explicitly use directional statistics of the observed feature space.

The projection direction can affect the relative separation of samples after one-dimensional projection and consequently alter their bucket-collision behavior, as illustrated in Figure 1.

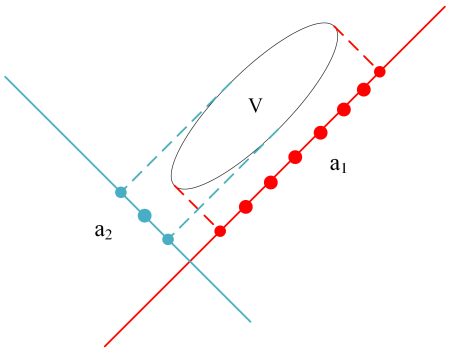


Figure 1. Effect of Projection Direction on the Discriminative Power of Bucket Partitioning. The gray ellipse represents the feature distribution; the red and blue solid lines indicate the two projection directions; the red and blue points denote the projected sample positions; and the dashed lines show the corresponding orthogonal projection paths.

The problem addressed in this paper is therefore to introduce database-derived directional information into projection-vector generation while keeping the feature descriptors, hash-function form, multi-table bucket organization, and online candidate-collection procedure unchanged. Section 4 presents the corresponding PCA-guided probabilistic sampling mechanism.

4. Method

4.1. Overall Framework

The proposed method builds on the hash-function form, bucket organization, and candidate-collection procedure of multi-table p-stable LSH. Its core idea is to use the principal-direction information of deep features in the image database to lightly guide the generation of projection vectors while preserving the original hash bucketization scheme and retrieval procedure, thereby improving the alignment between candidate partitioning and the feature distribution. The overall pipeline is illustrated in Figure 2.

In terms of the processing pipeline, the proposed method can be divided into two stages: offline database construction and online query processing. In the offline stage, feature extraction and L2 normalization are first performed on the background image database to obtain unified deep feature representations. PCA is then applied to the database feature set to extract the principal component directions and their corresponding weights, which are used to construct the guiding information for projection generation. Based on this information, projection vectors and random offset parameters are generated for each hash table, composite hash functions are constructed, and all database features are assigned to the corresponding hash buckets, thereby completing the multi-table index construction.

The online stage follows the standard multi-table LSH query procedure: for a given query image, its normalized feature representation is first extracted, after which the same

projection parameters as those used in the offline stage are employed to compute the composite hash key for each table, probe the matched buckets, and aggregate candidate samples, ultimately forming a size-controlled candidate set $C(\mathbf{q})$.

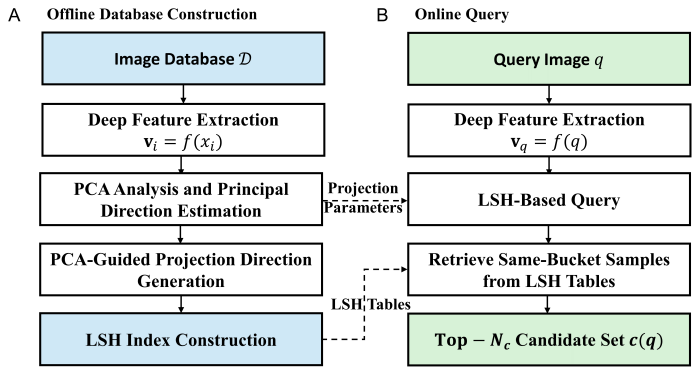


Figure 2. Image Candidate Generation Pipeline with PCA-Guided Projection.

Compared with standard p-stable LSH, the proposed method introduces a specific data-dependent change at the projection-sampling stage. In the standard approach, projection vectors are generated by isotropic random sampling without assuming any directional prior. By contrast, the proposed method incorporates the principal component information of image-database features into the projection vector generation process while preserving the bucket width r, composite key length k, number of hash tables L, and multi-table aggregation strategy. Consequently, standard p-stable LSH and the proposed method remain consistent in indexing structure and online query procedure, with the main difference lying in the projection distribution itself. This design makes it easier to attribute performance differences directly to changes in the projection-direction mechanism, rather than to alterations in the indexing framework or query process.

The empirical feature-distribution analysis supporting this motivation is reported separately in Section 5.1.3, while the present section focuses on the methodological formulation.

For an anisotropic feature distribution, when the projection directions are more closely aligned with the dominant directions of variation in the data, hash partitioning may better preserve local neighborhood relationships in the feature space. Conversely, when the projection directions deviate substantially from the principal structure of the data, the discriminative power of hash partitioning between relevant and irrelevant samples may be weakened. Motivated by this observation, this paper extends projection vector generation from purely random sampling to random sampling with a principal-direction bias. It should be emphasized that the proposed method does not seek to redesign the overall retrieval framework of standard p-stable LSH, nor does it alter the organization of the multi-table hash index. Rather, its goal is to introduce awareness of the data distribution while keeping the original candidate generation procedure essentially unchanged, thereby increasing the likelihood that the candidate set $C(\mathbf{q})$ covers the target sample under a fixed candidate budget and improving result stability under different perturbation conditions.

4.2. PCA-Guided Projection Vector Generation

To enable the random projection process to better capture the principal structure of deep features in the image database, this paper incorporates PCA-based guidance into the projection generation mechanism of standard p-stable LSH [32]. Let the background image database, after feature extraction and normalization, be represented as a sample set $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$. The sample set is first centered, and PCA is then applied to obtain the

top k_{pc} principal component directions $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{k_{\mathrm{pc}}}\}$ together with their corresponding eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_{k_{\mathrm{pc}}}\}$. The principal component directions characterize the dominant directions of variation in the feature distribution, while the eigenvalues quantify the relative importance of these directions. Because each PCA eigenvector is mathematically defined only up to sign, a deterministic orientation convention is required before the principal components are aggregated. The experiments in this work use the same deterministic sign-flipping convention as the adopted PCA implementation. Specifically, for each principal component $\mathbf{p}_i$, let $j_i^* = \arg\max_j |p_{ij}|$, where the smallest coordinate index is selected in the event of a tie. The component is then oriented as $\mathbf{p}_i \leftarrow \mathrm{sign}(p_{ij_i^*})\mathbf{p}_i$, so that its loading with the largest absolute magnitude is positive. This rule was already applied in all reported experiments and is stated here explicitly to ensure reproducibility across repeated runs and equivalent PCA implementations.

On this basis, this paper constructs a normalized guidance vector using the principal component directions together with their associated energy weights

$$\boldsymbol{\mu}_{k_{\mathrm{pc}}} = \frac{\sum_{i=1}^{k_{\mathrm{pc}}} \lambda_i \mathbf{p}_i}{\left\| \sum_{i=1}^{k_{\mathrm{pc}}} \lambda_i \mathbf{p}_i \right\|_2} \tag{5}$$

This vector is a compact aggregation of the deterministically oriented leading principal components. It combines the variance information represented by the eigenvalues with the corresponding oriented component directions. It should therefore be understood as a reproducible guidance vector constructed from a deterministically oriented PCA basis, rather than as an orientation-free representation of the principal subspace itself. The use of several components avoids relying exclusively on the first principal direction, while eigenvalue weighting retains the unequal variance contributions of the selected components that would be discarded by equal weighting. Alternative designs, such as randomly selecting individual principal directions, sampling directly within the principal subspace, or modifying the projection covariance, would produce different projection distributions and introduce additional design choices. The present formulation instead uses one normalized guidance vector to introduce a bounded mean shift while retaining isotropic random variation around it. The effects of the number of components and of equal versus eigenvalue-weighted aggregation are compared empirically in Section 5.3.

In standard p-stable LSH, projection vectors are drawn independently from an isotropic random distribution. In contrast, this paper extends the process to a principal-direction-biased random sampling scheme. Specifically, for each hash function, the random offset is generated in the same way as in the standard setting:

$$b \sim U(0, r) \tag{6}$$

The projection vectors are sampled according to the following distribution:

$$\mathbf{a} \sim \mathcal{N}(\alpha \boldsymbol{\mu}_{k_{\mathrm{pc}}}, \sigma^2 \mathbf{I}) \tag{7}$$

where α denotes the guidance strength and controls how strongly the mean of the projection distribution is biased toward the guidance direction; σ controls the sampling randomness, thereby preserving the diversity of projection directions; and $\mathbf{I}$ is the identity matrix. Under this formulation, the zero-mean isotropic sampling used to generate projection vectors in standard p-stable LSH is replaced by Gaussian sampling centered at $\boldsymbol{\mu}_{k_{\mathrm{pc}}}$ with a directional bias. Unlike conventional PCA-based feature extraction, PCA is not used here to transform the image features themselves, but only to bias the sampling distribution of projection vectors in the indexing stage, as illustrated in Figure 3.

The zero-mean Gaussian residual is retained because standard p-stable LSH in the Euclidean ($p = 2$) setting uses the Gaussian distribution as its corresponding stable projection family. The complete mean-shifted distribution in Equation (7) is no longer the standard zero-mean p-stable distribution, as discussed below, but retaining Gaussian random variation preserves the connection to the original Euclidean projection setting. Replacing the residual with a different distribution family would correspond to a different metric assumption or to a separate non-p-stable extension rather than an interchangeable implementation choice. The sensitivity of the proposed method to the Gaussian scale parameter σ is evaluated in Section 5.4.

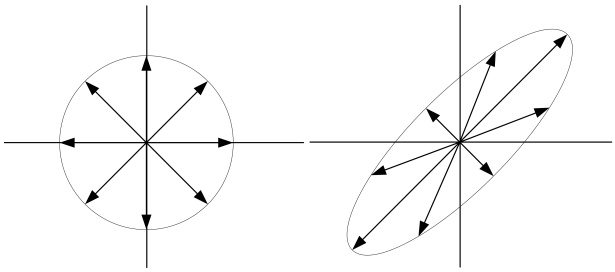


Figure 3. Illustration of isotropic projection sampling in standard p-stable LSH and PCA-guided projection sampling in the proposed method. The arrows indicate the sampled projection vectors and their corresponding directions.

Under the above formulation, the final hash function still retains the standard bucketization form given in Equation (3).

Therefore, the proposed method does not modify the hash bucket generation mechanism or the indexing structure of multi-table LSH, but introduces data-dependent information only during the sampling of projection parameters. This design allows the proposed method to incorporate the principal-direction structure of image-database features into the hash partitioning process while preserving the original retrieval procedure.

It is important to distinguish architectural compatibility with p-stable LSH from inheritance of its original theoretical guarantee. Let $\mathbf{z} = \mathbf{x} - \mathbf{y}$ denote the difference between two feature vectors. Under the projection sampling rule in Equation (7), the projected difference follows

$$\mathbf{a}^\top \mathbf{z} \sim \mathcal{N}\left(\alpha \boldsymbol{\mu}_{k_{\mathrm{pc}}}^\top \mathbf{z},\, \sigma^2 \|\mathbf{z}\|_2^2\right) \tag{8}$$

For a fixed projection vector $\mathbf{a}$ and a random offset $b \sim U(0, r)$, the conditional probability that $\mathbf{x}$ and $\mathbf{y}$ are assigned to the same bucket is determined by their projected separation. Averaging over the projection distribution gives

$$\begin{aligned} \Pr_b[h_{\mathbf{a},b}(\mathbf{x}) = h_{\mathbf{a},b}(\mathbf{y}) \mid \mathbf{a}] &= \left(1 - \frac{|\mathbf{a}^\top \mathbf{z}|}{r}\right)_+ \\ \Pr_{\mathbf{a},b}[h_{\mathbf{a},b}(\mathbf{x}) = h_{\mathbf{a},b}(\mathbf{y})] &= \mathbb{E}_{\mathbf{a}}\left[\left(1 - \frac{|\mathbf{a}^\top \mathbf{z}|}{r}\right)_+\right] \end{aligned} \tag{9}$$

where $(u)_+ = \max\{u, 0\}$. According to Equations (8) and (9), when $\alpha > 0$, the collision behavior depends on both the Euclidean magnitude $\|\mathbf{z}\|_2$ and the directional alignment $\boldsymbol{\mu}_{k_{\mathrm{pc}}}^\top \mathbf{z}$. Therefore, the mean-shifted projection distribution in Equation (7) is not the standard zero-mean isotropic p-stable distribution and does not directly inherit the classical Euclidean LSH collision guarantee. In this paper, compatibility with p-stable LSH refers specifically to retaining the hash-function form in Equation (3), the multi-table bucket

organization, and the online candidate-collection procedure, while modifying the offline projection distribution in a data-dependent manner.

The proposed projection generation strategy can be understood in terms of second-order projection response. Let $\mathbf{z}$ denote an arbitrary difference vector. If the projection vector satisfies

$$\mathbf{a} = \alpha \boldsymbol{\mu}_{k_{\mathrm{pc}}} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \tag{10}$$

then it follows that

$$\mathbb{E}\left[(\mathbf{a}^\top \mathbf{z})^2\right] = \alpha^2 \left(\boldsymbol{\mu}_{k_{\mathrm{pc}}}^\top \mathbf{z}\right)^2 + \sigma^2 \|\mathbf{z}\|_2^2 \tag{11}$$

The practical magnitude of the directional preference can also be quantified directly. Because $\boldsymbol{\mu}_{k_{\mathrm{pc}}}$ is normalized, the norm of the mean shift is $\|\mathbb{E}[\mathbf{a}]\|_2 = \alpha$, whereas the root-mean-square norm of the isotropic random component is $\sqrt{\mathbb{E}\|\boldsymbol{\varepsilon}\|_2^2} = \sigma\sqrt{d}$. Their global magnitude ratio is therefore $\rho = \alpha/(\sigma\sqrt{d})$. For the 4096-dimensional VGG19 descriptor with the default setting $\alpha = \sigma = 1$, this ratio is $1/64 \approx 0.0156$, confirming that the global angular bias is relatively weak. However, along the normalized guidance direction, $\mathbf{a}^\top \boldsymbol{\mu}_{k_{\mathrm{pc}}} \sim \mathcal{N}(\alpha, \sigma^2)$, and the standardized displacement is $\alpha/\sigma = 1$ under the default setting. The proposed mechanism should therefore be interpreted as introducing a light global directional preference together with a nonzero shift along the selected guidance axis, rather than forcing the projection vectors to align strongly with that axis.

The second-order expression shows that the proposed sampling rule introduces an additional term associated with the alignment between the difference vector and the guidance direction. However, this expression alone does not establish improved collision probability or nearest-neighbor preservation. Whether the additional directional response is useful depends on whether the selected guidance direction is relevant to the local query–target neighborhood structure. The analysis is therefore intended as geometric motivation for the proposed sampling strategy, while its practical effect is evaluated empirically in Section 5.

From the perspective of candidate generation, changes in the distribution of projection directions further influence the bucket collision patterns of samples across multiple hash tables. When the dominant variations are concentrated mainly in the principal subspace, projections with a principal-direction bias are more likely to produce discriminative responses along these meaningful directions of variation, thereby altering the proportion of relevant and irrelevant samples included in the candidate set. In other words, the potential benefit of the proposed method does not arise from any change in the hash formulation itself, but primarily from the improved alignment between the projection-direction distribution and the principal structure of the data.

The main comparative experiments use $\alpha = 1$ and $\sigma = 1$ as a unified default configuration. To verify that the reported behavior is not tied to an unexamined parameter choice, the experiments additionally evaluate the guidance strength α, the Gaussian scale parameter σ, and the number of hash tables L through a one-factor-at-a-time sensitivity analysis. The number and aggregation of principal components are examined separately. These controlled analyses are used to justify the default configuration while avoiding a large Cartesian-product parameter search.

4.3. Index Construction

After obtaining the PCA-guided projection parameters, this paper builds the candidate index using the retained multi-table bucket organization of p-stable LSH. For the ℓ-th hash table, the composite hash key remains as defined in Equation (4), where $h_{\mathbf{a}_{\ell j}, b_{\ell j}}(\cdot)$ denotes the j-th basic hash function in the ℓ-th hash table. In contrast to standard p-stable LSH,

the projection vector $\mathbf{a}_{\ell j}$ is generated using the PCA-guided mechanism introduced in the previous subsection, while the random offset $b_{\ell j}$ continues to be sampled independently in the standard way.

During offline index construction, for each database feature $\mathbf{v}_i$, the composite hash key is computed for all L hash tables, and the corresponding sample index is inserted into the associated buckets. As a result, each sample in the background image database is assigned a bucket location in every hash table, thereby forming a multi-table indexing structure. The corresponding pseudocode is provided in Algorithm 1.

At query time, for a given query feature $\mathbf{q}$, its composite hash key is computed in each hash table, and the corresponding matched buckets are accessed to collect candidate sample indices. The results returned from different hash tables are then aggregated and deduplicated to form the candidate set $C(\mathbf{q})$. If the number of candidates exceeds the predefined budget, each candidate is first ranked in descending order according to the number of hash tables in which it occurs in a query-matched bucket. Candidates with the same occurrence frequency are further ranked in ascending order according to their Euclidean distance from the query in the L2-normalized feature space. Any remaining exact ties are resolved according to the database sample index in ascending order to ensure deterministic output. The first N_c candidates in the resulting order are retained as the final candidate set.

Algorithm 1: PCA-guided Multi-Table LSH Index Construction

Input: Feature set $V = \{\mathbf{v}_i\}_{i=1}^{N}$, guidance vector $\boldsymbol{\mu}_{k_{\text{pc}}}$, bucket width r, number of hash tables L, number of hash functions in each hash table k, sampling parameters α, σ

Output: Multi-table composite hash index $\{T_\ell\}_{\ell=1}^{L}$

foreach *hash table* $\ell \in \{1, 2, \ldots, L\}$ **do**
  Initialize an empty hash table T_ℓ;
  for $j = 1$ ***to*** k **do**
    Sample $b_{\ell j} \sim U(0, r)$;
    Sample $\mathbf{a}_{\ell j} \sim \mathcal{N}(\alpha \boldsymbol{\mu}_{k_{\text{pc}}}, \sigma^2 \mathbf{I})$;
  end
end
foreach *feature vector* $\mathbf{v}_i \in \mathbf{V}$ **do**
  for $\ell = 1$ ***to*** L **do**
    Compute the composite hash key $g_\ell(\mathbf{v}_i) = \left(h_{\mathbf{a}_{\ell 1}, b_{\ell 1}}(\mathbf{v}_i), \ldots, h_{\mathbf{a}_{\ell k}, b_{\ell k}}(\mathbf{v}_i)\right)$
    Insert sample index i into bucket $g_\ell(\mathbf{v}_i)$ of hash table T_ℓ;
  end
end
return $\{T_\ell\}_{\ell=1}^{L}$;

4.4. Complexity

Let N denote the size of the image database, d the feature dimensionality, L the number of hash tables, k the composite key length per table, and N_c the candidate budget. In this work, the complexity analysis focuses on three aspects: offline index construction, online query processing, and index storage.

During offline index construction, each database feature must be mapped into all L hash tables by computing its composite hash key in each table. Because each composite key consists of k basic hash functions, and each projection computation has a cost proportional to the feature dimension d, the dominant time complexity of index construction can be expressed as

$$T_{\text{index}} = O(N \cdot Lk \cdot d) \tag{12}$$

Compared with standard p-stable LSH, the proposed method introduces PCA fitting and guidance-vector construction before the hash index is built. These operations are performed once during offline preprocessing. After the projection vectors have been generated, database insertion follows the same multi-table hash-index construction procedure as standard p-stable LSH. The additional PCA preprocessing therefore does not enter the per-query online procedure.

In terms of storage, both standard p-stable LSH and PCA-guided LSH store each database sample in L hash tables, resulting in a dominant inverted-index storage cost of approximately $O(NL)$. Both methods also store Lk projection vectors of dimensionality d. PCA-guided LSH additionally stores the selected principal-component information and the normalized guidance vector, requiring approximately $O(k_{\mathrm{pc}}d)$ additional values. Because k_{pc} is small and the inverted-bucket entries remain the dominant index component, the proposed method preserves the principal memory characteristics of the original multi-table p-stable LSH architecture.

For comparison, HNSW stores the feature vectors together with graph-neighbor links, whose graph-storage component is related to the database size and connectivity parameter M. Its preprocessing stage constructs the navigable graph incrementally. IVFFlat stores the feature vectors together with coarse-centroid information and inverted-list assignments, and its preprocessing stage includes coarse-quantizer training and database assignment. Thus, HNSW, IVF, and PCA-guided LSH allocate their offline cost and memory to different index components. The proposed method is distinguished by retaining the existing multi-table bucket structure and adding only PCA guidance information, rather than introducing a graph or a separately trained coarse-quantization structure.

During online query processing, given a query feature $\mathbf{q}$, the composite hash key must first be computed for all L hash tables, resulting in a time complexity of $O(Lk \cdot d)$. Afterward, the corresponding buckets in each table are probed and the retrieved candidates are aggregated. Let $B_\ell(\mathbf{q})$ denote the matched bucket for query $\mathbf{q}$ in the ℓ-th hash table. Then, the total scan cost can be expressed as

$$S_{\mathbf{q}} = \sum_{\ell=1}^{L} |B_\ell(\mathbf{q})| \tag{13}$$

Accordingly, the cost of candidate aggregation, deduplication, and ranking can be approximated as $O(S_{\mathbf{q}} + M \log M)$, where $M \leq S_{\mathbf{q}}$ is the number of unique candidates after deduplication.

Overall, the proposed method does not alter the asymptotic online complexity of standard multi-table LSH. Its main additional cost is incurred in the offline stage, primarily due to distributional analysis and guidance parameter generation, while the online stage still avoids the $O(N \cdot d)$ linear growth associated with exhaustive distance computation over the entire database. Therefore, while retaining a similar complexity profile, any performance differences are more likely to stem from changes in the projection-direction generation mechanism rather than from additional structural overhead in the online retrieval process.

5. Experiments and Analysis

5.1. Experimental Setup

The experiments in this section focus on the candidate generation stage and examine three aspects. The first concerns the ability of the candidate set $C(\mathbf{q})$ to cover the target sample under a fixed candidate budget. The second concerns the sensitivity of this behavior to parameter changes within the standard p-stable LSH framework. The third concerns the consistency of candidate generation behavior across different datasets, feature extractors,

and representative ANN baselines. The evaluated methods include standard p-stable LSH, the proposed PCA-guided LSH, HNSW, and IVF. Among them, standard p-stable LSH and the proposed PCA-guided LSH share the same indexing structure and query procedure, differing only in how the projection vectors are generated. This design allows their performance differences to be attributed primarily to changes in the projection distribution.

5.1.1. Dataset

The experiments in this paper use three image datasets. CALTECH101 is adopted as the main controlled setting for parameter studies, sensitivity analysis of the number of principal components, and perturbation-stability evaluation [33]. The dataset contains 101 object categories and a total of 9144 images, with each category typically containing approximately 40–800 images. Owing to its relatively clear intra-class and inter-class differences, it is well suited for controlled image retrieval experiments.

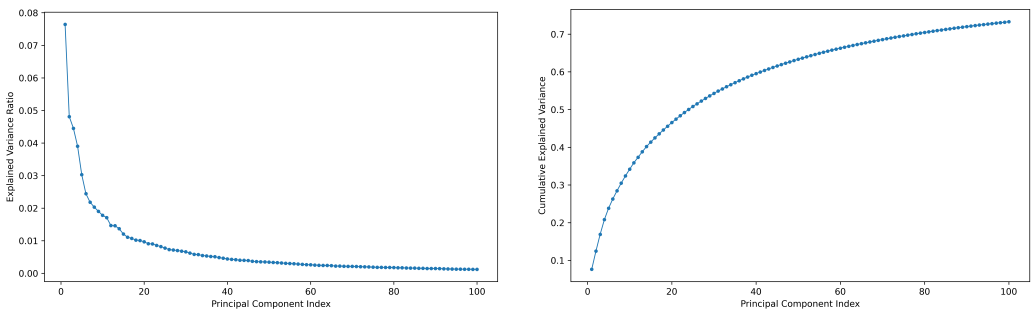
To extend validation beyond a single benchmark, CIFAR-10 and Tiny-ImageNet are further introduced for multi-dataset comparison. CIFAR-10 contains 10 object categories and provides a visually different distribution from CALTECH101 [34]. Tiny-ImageNet contains 200 categories and serves as a larger and more diverse ImageNet-style subset [35]. In the experiments, these additional datasets are used to compare the behavior of the proposed candidate generation strategy across independently fitted dataset-specific settings under a unified evaluation protocol.

5.1.2. Feature Extractors

Deep visual features are used as the image representation for subsequent indexing and candidate generation. In the main controlled setting, VGG19 is adopted as the default feature extractor [29]. In addition, ConvNeXt-Tiny is included as a representative modern CNN-based feature extractor [30], and ViT-B/16 is included as a representative Transformer-based vision model [31]. Unless otherwise stated, all backbones are used in a pretrained and frozen manner, and the extracted descriptors are only treated as the input feature space for subsequent candidate generation.

5.1.3. Feature-Space Anisotropy Check

Before evaluating retrieval performance, a descriptive PCA analysis is performed on the normalized VGG19 features of CALTECH101 to examine the feature-distribution premise underlying the proposed sampling design. Figure 4a reports the explained variance ratios of the leading principal components, while Figure 4b reports the corresponding cumulative explained variance.



(**a**) (**b**)

Figure 4. PCA Analysis Results for the Normalized VGG19 Features of CALTECH101. (**a**) Distribution of Explained Variance Ratios Across Principal Components. (**b**) Cumulative Explained Variance Curve.

The leading components explain more variance than the subsequent components, and the cumulative explained variance increases as additional components are included. This descriptive result indicates that the evaluated feature distribution is anisotropic rather than uniformly distributed across all directions. It provides empirical motivation for examining a projection distribution with a light principal-direction preference. The result is used only to characterize the feature space; the PCA model and the candidate index are fitted independently within each evaluated dataset–backbone combination.

5.1.4. Compared Methods

The experimental evaluation involves several categories of candidate generation methods. Standard p-stable LSH serves as the primary controlled baseline because it shares the same hash-function form, bucket organization, multi-table structure, and online candidate-collection procedure as the proposed method. The comparison between standard p-stable LSH and PCA-guided LSH therefore isolates the effect of modifying the projection-sampling distribution. Representative modern ANN methods, including HNSW [19] and IVF [21], are additionally included to position the proposed method relative to stronger graph-based and quantization-based indexing systems. These methods operate through different index structures and are treated as system-level reference methods rather than mechanism-matched variants. For the perturbation-stability evaluation, SimHash LSH [36] and multi-table SimHash (MTSH) [37] are also included as additional classical hashing-based baselines.

5.1.5. Query Construction

Although the proposed indexing mechanism can be used as a general candidate generation approach, the evaluation protocol adopted in this work focuses on exact-instance and near-duplicate recovery rather than category-level semantic retrieval. Each query is generated as a content-preserving perturbation of a reference image whose unperturbed version is registered in the database. Retrieval is considered successful when this registered target instance is included in the returned candidate set; other images belonging to the same semantic category are not treated as target matches. Two query construction settings are considered. The first is used for detailed parameter analysis and perturbation-stability evaluation, while the second is used for multi-dataset and cross-backbone comparison under a unified default setting.

The primary challenge considered in the candidate generation stage comes from content-preserving perturbations, under which the semantic content of an image remains unchanged while its pixel statistics or spatial arrangement may be altered. To examine several representative perturbation conditions under a controlled evaluation protocol, this paper considers additive white Gaussian noise (AWGN), JPEG lossy compression, speckle noise, and cyclic vertical translation. These operations represent acquisition noise, compression distortion, multiplicative noise, and controlled cyclic spatial rearrangement, respectively. They are used as representative query conditions for evaluating the stability of candidate generation.

Among these perturbations, AWGN is generated using the standard additive model. For a pixel-domain image $I \in [0, 255]$, zero-mean Gaussian noise $n \sim \mathcal{N}(0, \sigma^2)$ is added, and the resulting image is clipped to ensure that all pixel values remain within the valid range:

$$\tilde{I} = \mathrm{clip}(I + n,\ 0,\ 255) \tag{14}$$

Speckle noise is generated using a multiplicative noise model. In the pixel domain, let

$$\tilde{I} = \mathrm{clip}(I + I \odot n_s,\ 0,\ 255), \qquad n_s \sim \mathcal{N}(0, \sigma_s^2) \tag{15}$$

where $\odot$ denotes element-wise multiplication, and σ_s controls the strength of the speckle noise.

JPEG compression produces distorted samples by encoding and decoding the image with a specified quality factor. The translation perturbation is implemented as a cyclic vertical shift: pixels leaving one image boundary re-enter from the opposite boundary, without introducing cropping, padding, or missing regions. This controlled perturbation preserves the complete pixel content while changing its spatial arrangement and is used to examine the sensitivity of the feature-indexing pipeline to cyclic positional shifts.

For the parameter study, 200 original images are randomly selected from the query set, and 8 typical content-preserving perturbations are applied to each image, resulting in a total of 1600 query samples. These samples are used in a grid-based analysis of 54 parameter combinations, obtained by crossing 6 different hash signature lengths k with 9 different bucket widths r, to evaluate their effects on recall and efficiency. The same query set is also used to compare the effects of different projection strategies on recall and efficiency.

For the stability experiments, multiple intensity levels are specified for each perturbation type, and corresponding groups of distorted samples are constructed from 1000 randomly selected original images. Specifically, for each perturbation type, 1000 query samples are generated at every perturbation intensity level. In total, 4 perturbation types and 19 intensity levels are considered, resulting in 19,000 query samples for the stability evaluation. This setup is intended to examine the degradation trends and performance fluctuations of different methods as perturbation strength gradually increases.

For the multi-dataset and cross-backbone comparison, 1000 reference images are sampled for each dataset–backbone combination. One JPEG-compressed query image is then generated from each reference image with a fixed quality factor of $q = 10$, and the corresponding original image is treated as the registered target. This design provides a unified query setting for comparing different datasets, feature extractors, and candidate generation methods.

5.1.6. Parameter Settings

In the main controlled setting, the indexing input is given by 4096-dimensional global vectors extracted using VGG19, followed by L2 normalization before indexing to ensure that retrieval is conducted in a unified feature space. For the multi-dataset and cross-backbone comparison, the corresponding feature descriptors extracted by each backbone are used in the same way. Candidate generation adopts the multi-table p-stable LSH indexing framework, whose key parameters include the bucket width r, hash signature length k, and number of hash tables L. Unless otherwise stated, the number of hash tables is set to $L = 2$, and the PCA-guided projection distribution uses $\alpha = 1$ and $\sigma = 1$. These default values are examined in the sensitivity analyses reported in Sections 5.3 and 5.4.

To avoid notational ambiguity, k is used throughout the manuscript exclusively to denote the hash signature length, equivalent to the number of hash functions in each composite hash key, whereas k_{pc} exclusively denotes the number of principal components used to construct the guidance vector.

In the parameter study, a grid-based scan over (k, r) is performed to characterize the trade-off between selectivity and recall and to compare the proposed PCA-guided LSH with standard p-stable LSH. Unless otherwise noted, the proposed PCA-guided LSH uses the first 5 principal components by default to construct the guidance vector, i.e., $k_{\mathrm{pc}} = 5$. In the stability experiments, the candidate budget and indexing parameters are fixed to reduce the coupling between index settings and perturbation strength: the number of returned candidates is set to $N_c = 10$, the hash signature length to $k = 16$, and the bucket

width to $r = 50$. Under this unified configuration, candidate generation performance is compared across different perturbation types and perturbation strengths.

For the multi-dataset and cross-backbone comparison, the same indexing configuration is used for standard p-stable LSH and PCA-guided LSH. The HNSW and IVF configurations were selected through bounded preliminary calibration around commonly used practical settings, with preference given to configurations that provided relatively high recall while maintaining reasonable indexing and query costs. For HNSW, cosine similarity is used with $M = 16$, ef_construction $= 200$, and ef_search $= 50$. For IVF, L2-normalized features are indexed using an IVFFlat structure with inner-product search, where $nprobe = 10$ and the number of coarse clusters is set as $nlist = \min\{4\sqrt{n}, n\}$ according to the database size n. After this preliminary calibration, the selected configurations were held fixed across all dataset–backbone combinations. This avoids introducing dataset-specific tuning advantages and provides a consistent and reproducible comparison protocol, while still using competitive rather than arbitrary ANN configurations.

All experiments were conducted in a 64-bit Windows environment on a computer equipped with an AMD Ryzen 7 5800H CPU, 16 GB of system memory, and an NVIDIA GeForce RTX 3060 Laptop GPU with 6 GB of graphics memory. The experimental pipeline used Python 3.9, and PCA was implemented using the SVD-based PCA implementation provided by scikit-learn 1.6.1. Five fixed random seeds were used consistently across the matched repeated experiments, with the same query set and corresponding seed assignment used for each compared method. To support reproducibility, the source code, complete parameter configurations, and exact random-seed list will be made publicly available after acceptance of the manuscript.

5.1.7. Evaluation Metrics

This paper evaluates candidate generation mainly in terms of recall and runtime. Under the evaluation protocol used in this work, the unperturbed reference image corresponding to each query is treated as the registered target instance. Recall therefore measures target-instance recovery: a query is counted as successfully retrieved when its registered target is included in the returned candidate set. In the multi-dataset and cross-backbone comparison, target-instance Recall@10 under the unified default setting is reported as the primary quantitative metric.

Let $\mathbf{x}^*(\mathbf{q})$ denote the registered target corresponding to query $\mathbf{q}$. Then, recall under a fixed candidate budget is defined as

$$\text{Recall@}N_c = \frac{1}{|Q|} \sum_{\mathbf{q} \in Q} 1\{\mathbf{x}^*(\mathbf{q}) \in C(\mathbf{q})\} \tag{16}$$

Let $t(\mathbf{q})$ denote the end-to-end query time from feature extraction to the output of the candidate set for query $\mathbf{q}$. Then, the average response time is

$$\bar{t} = \frac{1}{|Q|} \sum_{\mathbf{q} \in Q} t(\mathbf{q}) \tag{17}$$

Here, recall measures the ability of the candidate generation stage to cover the corresponding target under a fixed candidate budget, and $\bar{t}$ reflects the end-to-end query cost under different parameter settings. Offline build time is reported separately and includes PCA guidance construction, where applicable, together with hash-index construction.

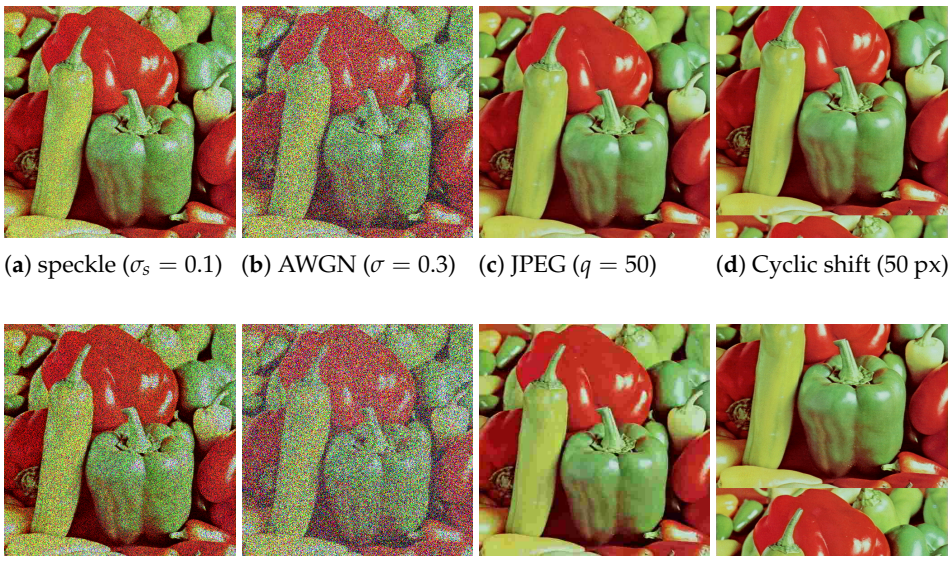
For statistical comparison under the fixed controlled configuration, the two methods are evaluated using the same query set and five matched random seeds. Statistical significance is assessed using a two-sided paired t-test on the seed-level Recall@10 values,

with a significance level of 0.05. A 95% confidence interval is also reported for the mean paired difference.

In the stability experiments, Recall@10 is adopted to evaluate candidate generation performance under different image perturbation conditions. Specifically, if the original registered image $\mathbf{x}^*(\mathbf{q})$ appears in the Top-10 candidate set of a perturbed query image $\mathbf{q}$, the query is regarded as successfully recalled. In this part, the mean Recall@10, denoted by μ, is reported, and the standard deviation σ_r is used to quantify its variability. A higher mean Recall@10 indicates better target coverage under perturbation, whereas a smaller standard deviation indicates more stable candidate generation performance across different samples and perturbation conditions.

5.2. Parameter Sensitivity

This subsection investigates candidate generation performance and the associated computational cost under different parameter configurations by varying the hash signature length k and bucket width r, and compares standard p-stable LSH with PCA-guided LSH under the same indexing framework. To this end, 200 original images are randomly selected from the query image set, and 8 typical content-preserving perturbations are applied to each image, yielding a total of 1600 query images. These images form the query set, with representative examples shown in Figure 5. The background image database remains CALTECH101, which introduces cross-category distractors and competing neighbors, thereby making the evaluation scenario closer to the retrieval load encountered in practical candidate generation. For each parameter configuration, the average recall, offline index construction time, and average end-to-end query time are reported to analyze the effect of projection strategy on candidate generation behavior. Unless otherwise stated, the auxiliary steering-related parameters are kept fixed in the following experiments, so that the present analysis focuses on the main indexing parameters and the role of principal-direction information itself. Similar fixed-parameter or unified-protocol evaluation strategies have also been used in recent ANN studies and benchmarking work. For example, Elliott and Clark adopt a fixed parameterization in their HNSW analysis to study the effects of data and intrinsic dimensionality rather than extensive retuning [38]. Benchmark-oriented ANN comparisons likewise emphasize standardized and reproducible evaluation under unified settings [39].

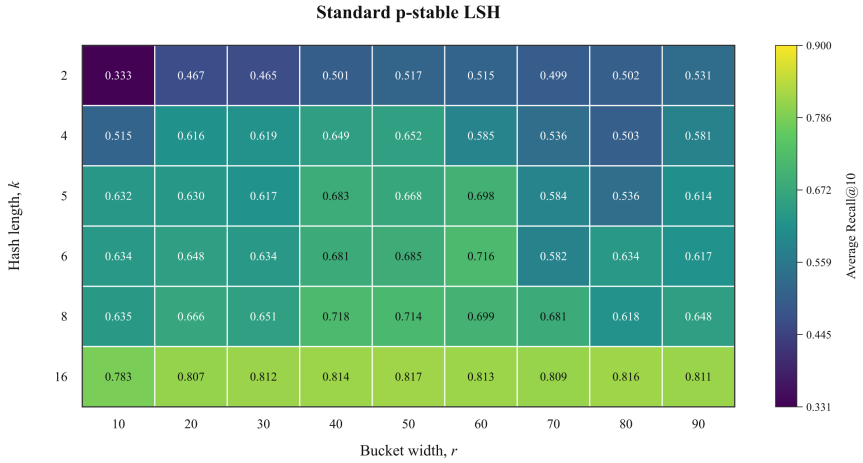


(**a**) speckle ($\sigma_s = 0.1$) (**b**) AWGN ($\sigma = 0.3$) (**c**) JPEG ($q = 50$) (**d**) Cyclic shift (50 px)

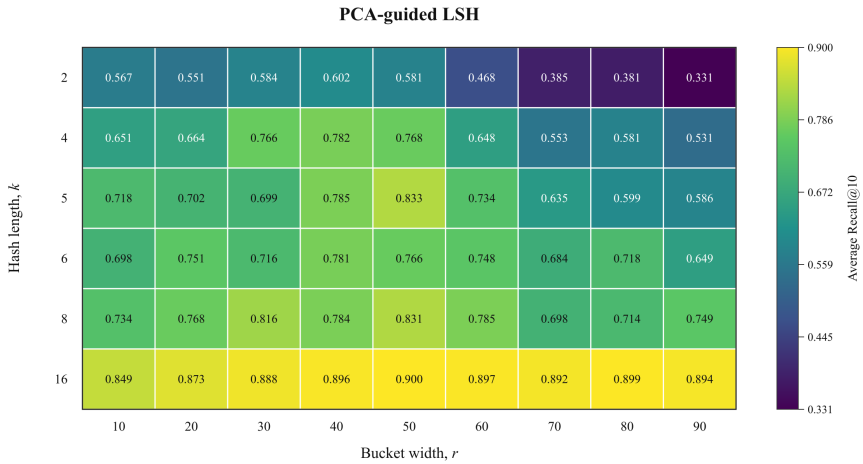
(**e**) speckle ($\sigma_s = 0.4$) (**f**) AWGN ($\sigma = 0.5$) (**g**) JPEG ($q = 10$) (**h**) Cyclic shift (150 px)

Figure 5. Example Query Samples Used in the Performance Tests.

Figure 6 presents the Recall@10 results as annotated heatmaps, making the interaction between the hash signature length k and bucket width r more explicit. PCA-guided LSH obtains higher Recall@10 than standard p-stable LSH in 48 of the 54 tested parameter combinations. The improvements are particularly consistent at $k = 16$, where PCA-guided LSH performs better for every tested bucket width. However, the proposed method is not uniformly superior. The configurations in which standard p-stable LSH performs better are concentrated mainly at $k = 2$ with large bucket widths ($r \geq 60$), together with a small number of other low-k, large-r settings. These results indicate that PCA guidance is more beneficial when the hash signature provides sufficient selectivity, whereas overly short signatures combined with coarse bucketization may weaken the benefit of the directional bias.



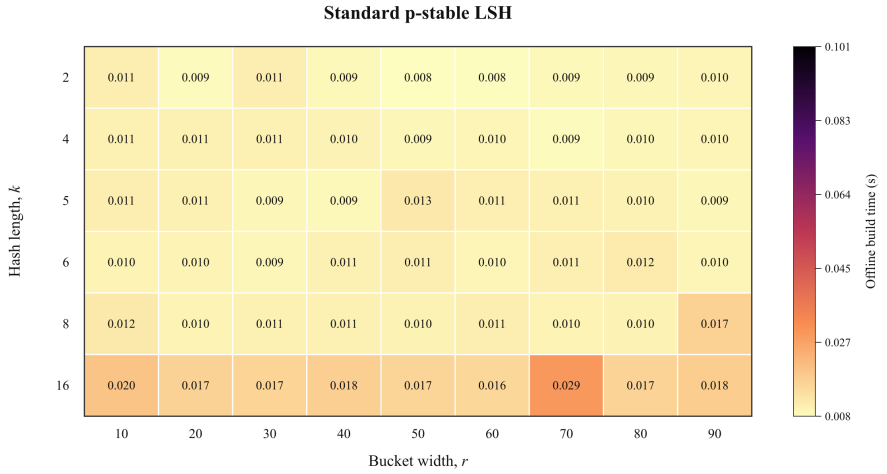
(**a**) Standard p-stable LSH



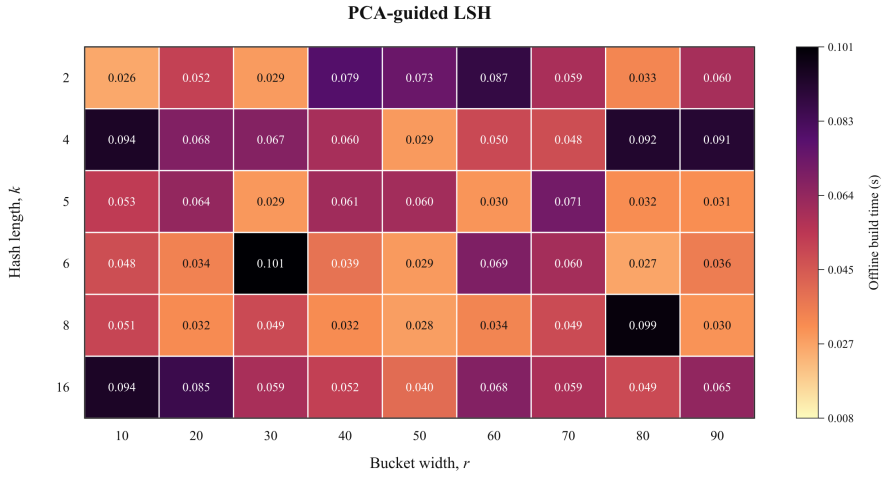
(**b**) PCA-guided LSH

Figure 6. Comparison of Average Candidate Recall across Different Bucket Widths r and Hash Lengths k.

Figure 7 presents the offline index construction time using annotated heatmaps. PCA-guided LSH incurs additional offline cost because PCA analysis and guidance-vector construction are performed before the hash index is built. Nevertheless, the construction times of the two methods remain within the same practical order of magnitude across the evaluated parameter grid. This result is consistent with the complexity analysis, because the additional computation is confined to the offline indexing stage.



(**a**) Standard p-stable LSH



(**b**) PCA-guided LSH

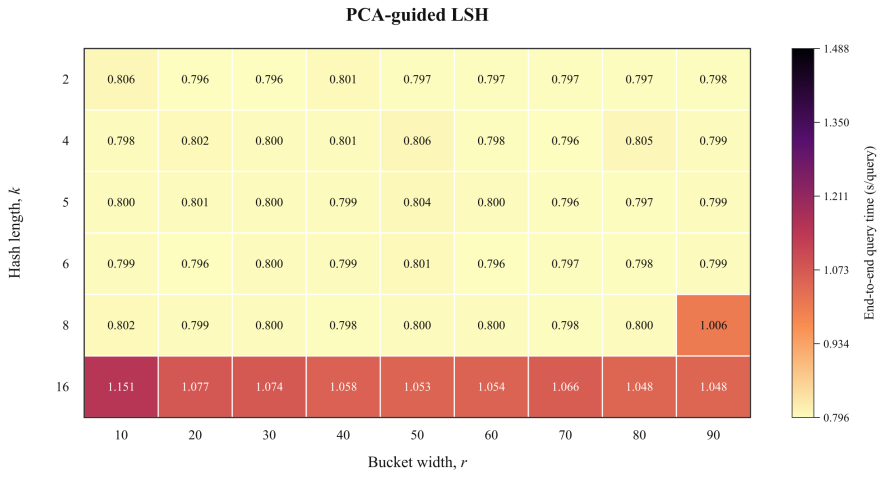
Figure 7. Comparison of Offline Indexing Time across Different Bucket Widths r and Hash Lengths k.

Figure 8 presents the average end-to-end query time, including feature extraction and candidate generation. The two methods have similar query times across most parameter combinations, which is consistent with the fact that PCA guidance modifies only the offline projection-sampling procedure and does not alter the online hash lookup process. Some isolated high-latency configurations are visible in the heatmaps, indicating that query cost can still vary with the bucket occupancy and candidate-scanning behavior induced by particular combinations of k and r. Overall, the proposed method does not introduce a systematic increase in online query time.

Overall, the parameter study shows that candidate generation performance depends jointly on the bucket width r and hash signature length k. PCA-guided LSH achieves higher Recall@10 in most of the tested configurations, while its weaker cases are concentrated in short-signature and large-bucket settings. The method introduces additional offline construction cost but maintains an online query-time scale comparable to that of standard p-stable LSH. These results support PCA guidance as a lightweight empirical refinement under appropriate indexing configurations, rather than implying uniform improvement for every parameter combination.



(**a**) Standard p-stable LSH



(**b**) PCA-guided LSH

Figure 8. Comparison of Average End-to-End Query Time across Different Bucket Widths r and Hash Lengths k.

To further analyze the source of the observed performance gains, this paper compares standard p-stable LSH and PCA-guided LSH in terms of candidate scan cost and candidate set size under a fixed parameter setting. The experimental settings are as follows: the number of returned candidates $N_c = 10$, the number of hash tables $L = 2$, hash signature length $k = 16$, and bucket width $r = 50$. Using the same query set, five randomized repeated trials are conducted to report the average Recall@10, average scan volume $S_{\mathbf{q}}$, average deduplicated candidate count, and average query time. The corresponding results are presented in Table 1.

Table 1. Analysis of Candidate Set Size for Standard p-stable LSH and PCA-guided LSH (5 Randomized Repeated Trials).

Method	Recall@10	Mean Scan Size $S_{\mathbf{q}}$	Mean Unique Candidates	Mean End-to-End Query Time (s)
Standard p-stable LSH	0.8163 ± 0.0173	698.76	458.02	0.7958
PCA-guided LSH	$\mathbf{0.9000 \pm 0.0069}$	711.20	461.53	0.7979

Note: Bold values indicate the best Recall@10 result among the compared methods.

As shown in Table 1, the average Recall@10 of PCA-guided LSH improves from 0.8163 ± 0.0173 for standard p-stable LSH to 0.9000 ± 0.0069. This indicates that, under the current setting, the proposed method achieves not only higher average target coverage but also smaller performance fluctuations. Meanwhile, the two methods differ only slightly in average scan volume, average deduplicated candidate count, and average query time. Specifically, PCA-guided LSH increases the average scan volume by about 1.8% and the average deduplicated candidate count by about 0.8%, while leaving the average query time almost unchanged.

Because both methods were evaluated using the same query set and the same five random seeds, a two-sided paired t-test was applied to the seed-level Recall@10 values. The mean paired improvement obtained by PCA-guided LSH was 0.0837, corresponding to 8.37 percentage points. The 95% confidence interval of the mean paired difference was $[0.0548, 0.1126]$. The test result was statistically significant, $t(4) = 8.053$, $p = 0.0013$. Therefore, under this controlled configuration, the observed improvement is unlikely to be explained solely by random variation across the matched runs.

These results suggest that the performance gains of PCA-guided LSH under a fixed candidate budget do not come from substantially enlarging the candidate set or increasing online computation, but more likely from better alignment between projection partitioning and the feature distribution. In other words, under similar candidate set size and query cost, PCA-based principal-direction bias helps improve both the average performance and the stability of candidate generation.

5.3. Sensitivity to the Number and Aggregation of Principal Components

To further examine the practical role of PCA guidance, this subsection evaluates both the number of principal components k_{pc} and the rule used to aggregate them. All indexing parameters and the candidate budget are kept fixed. Standard p-stable LSH is used as the unguided reference. For eigenvalue-weighted guidance, $k_{\mathrm{pc}} \in \{1, 3, 5, 7, 10\}$ is tested, extending the previous range beyond five components. An equal-weighted aggregation of the first five deterministically oriented components is also included to examine whether the observed behavior depends specifically on the eigenvalue weights.

For each configuration, index construction and query evaluation are repeated using the same query set and five random seeds. Recall@10 is reported as mean $\pm$ standard deviation across the repeated runs, while the timing columns report the corresponding averages. The end-to-end query time includes feature extraction and candidate generation. The results are presented in Table 2.

Table 2. Sensitivity of PCA-Guided LSH to the Number and Aggregation of Principal Components.

Method	Aggregation	k_{pc}	Average Recall@10	Offline Build Time (s)	End-to-End Query Time (s/Query)
Standard p-stable LSH	–	–	0.900 ± 0.031	0.016	0.821
PCA-guided LSH	Single component	1	0.915 ± 0.023	0.035	0.824
PCA-guided LSH	Eigenvalue-weighted	3	0.910 ± 0.026	0.056	0.826
PCA-guided LSH	Equal-weighted	5	0.907 ± 0.027	0.048	0.824
PCA-guided LSH	Eigenvalue-weighted	5	$\mathbf{0.918 \pm 0.008}$	0.050	0.827
PCA-guided LSH	Eigenvalue-weighted	7	0.912 ± 0.018	0.052	0.826
PCA-guided LSH	Eigenvalue-weighted	10	0.908 ± 0.013	0.061	0.830

Note: Bold values indicate the best Recall@10 result among the tested configurations.

The results show a non-monotonic dependence on k_{pc}. Eigenvalue-weighted guidance with $k_{\mathrm{pc}} = 5$ achieves the highest mean Recall@10, 0.918, and the lowest standard deviation, 0.008, among the tested configurations. Extending the range to $k_{\mathrm{pc}} = 7$ and $k_{\mathrm{pc}} = 10$ does

not yield further improvement; the corresponding mean recalls decrease to 0.912 and 0.908, respectively. Thus, the choice of five components is not an artifact of the previous upper boundary. Under the current setting, adding lower-variance components beyond this point appears to dilute rather than strengthen the useful directional preference.

The aggregation comparison provides additional evidence for the chosen formulation. With the same five components, equal weighting obtains 0.907 ± 0.027, whereas eigenvalue weighting obtains 0.918 ± 0.008. The eigenvalue-weighted scheme therefore has a higher mean and lower variability in this experiment. This result is consistent with retaining the unequal variance contributions of the selected components, although it is interpreted as an empirical comparison under the present setting rather than as a universal superiority claim over all possible subspace-sampling designs.

The timing differences remain small. Increasing k_{pc} mainly affects offline PCA guidance construction, while the end-to-end query times remain between 0.824 and 0.830 s/query for all PCA-guided configurations. Based on the combined recall, variability, and cost results, $k_{\mathrm{pc}} = 5$ with eigenvalue weighting is used as the default configuration in the subsequent experiments. The optimal choice may nevertheless vary with the dataset and feature representation.

5.4. Sensitivity to Guidance and Index Parameters

The preceding experiments analyze the hash signature length k, bucket width r, number of principal components k_{pc}, and aggregation rule. This subsection examines the remaining scalar parameters in Algorithm 1: the guidance strength α, Gaussian scale parameter σ, and number of hash tables L. A one-factor-at-a-time protocol is used so that the effect of each parameter can be interpreted without a large Cartesian-product search. When α is varied, $\sigma = 1$ and $L = 2$ are fixed; when σ is varied, $\alpha = 1$ and $L = 2$ are fixed; and when L is varied, $\alpha = \sigma = 1$ is used. In all cases, $k_{\mathrm{pc}} = 5$, $k = 16$, $r = 50$, and $N_c = 10$ are kept unchanged. Each configuration uses the same query set and five random seeds. Recall@10 is reported as mean $\pm$ standard deviation, while the timing columns report averages and include feature extraction in the end-to-end query time. The corresponding results are summarized in Table 3.

Table 3. One-Factor-at-a-Time Sensitivity Analysis of Guidance and Index Parameters.

Varied Parameter	Value	Average Recall@10	Offline Build Time (s)	End-to-End Query Time (s/Query)
α	0.0	0.900 ± 0.031	0.016	0.821
α	0.5	0.906 ± 0.017	0.048	0.824
α	1.0	0.918 ± 0.008	0.050	0.827
α	1.5	0.908 ± 0.014	0.051	0.822
α	2.0	0.892 ± 0.015	0.051	0.823
σ	0.5	0.904 ± 0.039	0.043	0.818
σ	1.0	0.918 ± 0.008	0.050	0.827
σ	1.5	0.873 ± 0.033	0.052	0.820
L	1	0.911 ± 0.011	0.024	0.818
L	2	0.918 ± 0.008	0.050	0.827
L	4	0.901 ± 0.036	0.082	0.839

The three default rows $\alpha = 1$, $\sigma = 1$, and $L = 2$ refer to the same configuration and therefore share the same measured result. The $\alpha = 0$ setting uses the standard p-stable branch because its projection distribution reduces to the zero-mean Gaussian baseline; consequently, its lower build time does not represent a computational advantage caused by changing α alone.

The guidance-strength results are non-monotonic. Increasing α from zero to one raises the mean Recall@10 from 0.900 to 0.918, whereas stronger shifts reduce recall to 0.908 at $\alpha = 1.5$ and 0.892 at $\alpha = 2$. This behavior supports the use of a moderate mean shift: insufficient guidance approaches the unguided baseline, while an overly strong shift can reduce projection diversity.

The Gaussian-scale analysis likewise favors the intermediate setting $\sigma = 1$, which achieves higher mean recall and lower variability than the tested values 0.5 and 1.5. Because σ also changes the projection scale relative to the fixed bucket width r, this result should be interpreted as empirical sensitivity under the stated bucketization setting rather than as an isolated distribution-scale effect.

For the number of hash tables, $L = 2$ achieves the highest mean recall and the lowest variability among the tested settings. Increasing L to four increases the offline build time from 0.050 to 0.082 s and the end-to-end query time from 0.827 to 0.839 s/query without improving recall. Conversely, $L = 1$ has lower cost but slightly lower recall. The default $L = 2$ therefore provides a practical balance between target coverage, stability, and computational cost in the controlled setting. Together with the preceding analyses of k, r, and k_{pc}, these results cover the scalar indexing and guidance parameters used by the proposed method.

5.5. Stability Test

To further examine candidate generation performance under content-preserving perturbations, this subsection compares standard p-stable LSH and PCA-guided LSH under four conditions: AWGN, JPEG compression, speckle noise, and translation. In addition, two baseline methods are introduced for benchmarking. SimHash (SRP) LSH [36] uses random hyperplane projections with sign-bit hashing to approximately preserve angular similarity between samples. Multi-table SimHash [37], a variant of SimHash, further combines random-hyperplane sign hashing with multi-table subcode indexing for candidate retrieval. For brevity, these two methods are referred to below as SimHash LSH and MTSH, respectively. For each perturbation type, multiple intensity levels are considered. By measuring recall under a fixed candidate budget, the experiments examine how the target coverage of the candidate set $C(\mathbf{q})$ degrades as perturbation strength increases.

5.5.1. Additive White Gaussian Noise

Additive White Gaussian Noise (AWGN) is a common form of random perturbation used to simulate additive noise encountered during image acquisition and transmission. To evaluate the stability of the candidate generation module in the presence of AWGN, zero-mean Gaussian noise $\bar{n} \sim \mathcal{N}(0, \sigma^2)$ is added to the query images, and the recall of each method is measured at different noise levels $\sigma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, as reported in Table 4.

The results show that under weak noise conditions ($\sigma \leq 0.2$), all four methods achieve a recall of 1.000, indicating that low-level AWGN has almost no effect on candidate generation. When the noise level increases to $\sigma = 0.3$, recall begins to decline for all methods, although the overall performance remains high. In particular, PCA-guided LSH achieves a recall of 0.994, exceeding SimHash LSH (0.988), p-stable LSH (0.986), and MTSH (0.976). As the noise level further increases to $\sigma = 0.4$ and $\sigma = 0.5$, the performance differences among the methods become more pronounced. PCA-guided LSH attains recalls of 0.947 and 0.861, respectively, consistently outperforming p-stable LSH (0.882 and 0.757), SimHash LSH (0.880 and 0.737), and MTSH (0.855 and 0.706).

The standard deviation results further show that recall variability increases for all methods as the noise level rises. By contrast, under strong noise conditions, PCA-guided LSH

not only achieves higher average recall but also exhibits smaller fluctuations, demonstrating greater stability. Overall, although the recall of all methods decreases as noise intensity increases, PCA-guided LSH maintains higher average recall across most noise levels and shows a smoother degradation trend. Compared with standard p-stable LSH, PCA-guided LSH retains better candidate coverage under moderate and high noise conditions, suggesting that the introduction of principal-direction bias makes the projection-based partitioning more robust to feature perturbations caused by additive noise.

Table 4. Recall under AWGN with noise standard deviation σ.

Method	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
p-stable LSH	1.000 ± 0.000	1.000 ± 0.000	0.986 ± 0.051	0.882 ± 0.167	0.757 ± 0.223
PCA-guided LSH	1.000 ± 0.000	1.000 ± 0.000	0.994 ± 0.034	0.947 ± 0.108	0.861 ± 0.181
SimHash LSH	1.000 ± 0.000	1.000 ± 0.000	0.988 ± 0.047	0.880 ± 0.164	0.737 ± 0.224
MTSH	1.000 ± 0.000	1.000 ± 0.000	0.976 ± 0.070	0.855 ± 0.184	0.706 ± 0.239

5.5.2. JPEG Compression

JPEG (Joint Photographic Experts Group) compression is used to simulate the lossy distortion introduced during image storage, transmission, and platform re-encoding. To evaluate the stability of the candidate generation module under JPEG compression, JPEG compression with different quality factors q is applied to the query images, and the recall of each method is measured at $q \in \{10, 30, 50, 70, 90\}$, as reported in Table 5.

The results show that, as the quality factor increases, the recall of all methods rapidly approaches 1.000. When the compression quality is high ($q \geq 50$), all four methods achieve 1.000, indicating that mild JPEG compression has almost no effect on candidate generation. However, when the quality factor is reduced to $q = 10$, the performance differences among the methods become pronounced. In this case, PCA-guided LSH achieves a recall of 0.984, outperforming SimHash LSH (0.935), p-stable LSH (0.933), and MTSH (0.927).

The standard deviation results further show that PCA-guided LSH exhibits smaller fluctuations under strong compression, indicating that it not only achieves higher average recall but also shows greater consistency across different categories of samples. Overall, candidate generation performance degrades gradually as the quality factor decreases, but PCA-guided LSH maintains recall that is at least comparable to that of standard p-stable LSH under most compression conditions, and achieves higher recall under some settings. This suggests that even when images are affected by local frequency-domain distortion introduced by compression, the principal-direction-guided projection mechanism can still preserve good target coverage.

Table 5. Recall under JPEG compression with quality factor q.

Method	$q = 10$	$q = 30$	$q = 50$	$q = 70$	$q = 90$
p-stable LSH	0.933 ± 0.113	0.994 ± 0.034	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
PCA-guided LSH	0.984 ± 0.067	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
SimHash LSH	0.935 ± 0.129	0.998 ± 0.020	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
MTSH	0.927 ± 0.125	0.998 ± 0.020	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000

5.5.3. Speckle Noise

Speckle noise is a typical form of multiplicative noise commonly encountered in coherent imaging systems, where the noise intensity depends on the original signal. To evaluate the stability of the candidate generation module in the presence of speckle noise, the pertur-

bation level is controlled by the noise strength σ_s, and the recall of each method is measured at $\sigma_s \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, as reported in Table 6.

The results show that under low-level speckle noise ($\sigma_s \leq 0.2$), all four methods achieve a recall of 1.000, indicating that mild multiplicative perturbation has almost no effect on candidate generation. When the noise strength increases to $\sigma_s = 0.3$, recall begins to decline for all methods, although overall performance remains high. In particular, PCA-guided LSH achieves 0.998, outperforming SimHash LSH (0.996), MTSH (0.992), and p-stable LSH (0.978). As the noise strength further increases to $\sigma_s = 0.4$ and $\sigma_s = 0.5$, the performance differences among the methods become more pronounced. PCA-guided LSH attains 0.949 and 0.878, respectively, consistently outperforming p-stable LSH (0.890 and 0.824), SimHash LSH (0.855 and 0.759), and MTSH (0.800 and 0.700).

The standard deviation results further show that recall variability increases for all methods as the speckle noise becomes stronger. By contrast, under high noise conditions, PCA-guided LSH not only achieves higher average recall but also exhibits smaller fluctuations, demonstrating greater stability. Overall, all four methods show gradual performance degradation as speckle noise increases, but PCA-guided LSH consistently maintains the best or jointly best performance across the entire test range, with its advantage becoming more pronounced under moderate and high perturbation levels.

Table 6. Recall under speckle noise with standard deviation σ_s.

Method	$\sigma_s = 0.1$	$\sigma_s = 0.2$	$\sigma_s = 0.3$	$\sigma_s = 0.4$	$\sigma_s = 0.5$
p-stable LSH	1.000 ± 0.000	1.000 ± 0.000	0.978 ± 0.084	0.890 ± 0.147	0.824 ± 0.180
PCA-guided LSH	1.000 ± 0.000	1.000 ± 0.000	0.998 ± 0.020	0.949 ± 0.104	0.878 ± 0.149
SimHash LSH	1.000 ± 0.000	1.000 ± 0.000	0.996 ± 0.028	0.855 ± 0.156	0.759 ± 0.205
MTSH	1.000 ± 0.000	1.000 ± 0.000	0.992 ± 0.039	0.800 ± 0.210	0.700 ± 0.236

5.5.4. Cyclic Vertical Translation

Cyclic vertical translation is used to evaluate the sensitivity of the candidate generation module to a controlled spatial rearrangement. In the experiments, query images are cyclically shifted along the vertical direction without cropping, padding, or removing image content. Specifically, a cyclic vertical translation operator $\mathcal{T}_{\delta_y}$ is applied to the query images, where $\delta_y \in \{50, 100, 150, 200\}$ px denotes the displacement magnitude along the y direction. The corresponding recall results are reported in Table 7.

The results show that after a cyclic vertical shift of 50 px, all methods still maintain relatively high recall. Among them, PCA-guided LSH achieves 0.967, outperforming SimHash LSH (0.927), p-stable LSH (0.886), and MTSH (0.882). As the cyclic displacement magnitude increases to 100 px and 150 px, the performance of all methods declines further, and the performance gap widens. PCA-guided LSH attains 0.924 and 0.925, respectively, consistently exceeding SimHash LSH (0.814 and 0.841), p-stable LSH (0.802 and 0.804), and MTSH (0.757 and 0.784).

When the displacement magnitude reaches 200 px, the recall of all methods partially rebounds. PCA-guided LSH reaches 0.975 and remains the best-performing method, while SimHash LSH, MTSH, and p-stable LSH achieve 0.908, 0.906, and 0.855, respectively. This non-monotonic behavior may be associated with the cyclic nature of the perturbation and the response of the deep feature extractor to the resulting spatial rearrangement. The standard deviation results further show that PCA-guided LSH achieves both higher average recall and smaller fluctuations under the tested cyclic-shift settings. Because this operation preserves all pixels through wrap-around, these results should be interpreted specifically as robustness to controlled cyclic vertical shifts rather than to real image displacements involving cropping, padding, or missing content.

Table 7. Recall under cyclic vertical translation with displacement magnitude δ_y (pixels).

Method	$\delta_y = 50$ px	$\delta_y = 100$ px	$\delta_y = 150$ px	$\delta_y = 200$ px
p-stable LSH	0.886 ± 0.155	0.802 ± 0.183	0.804 ± 0.170	0.855 ± 0.182
PCA-guided LSH	0.967 ± 0.080	0.924 ± 0.119	0.925 ± 0.115	0.975 ± 0.072
SimHash LSH	0.927 ± 0.118	0.814 ± 0.175	0.841 ± 0.157	0.908 ± 0.133
MTSH	0.882 ± 0.146	0.757 ± 0.232	0.784 ± 0.191	0.906 ± 0.136

5.6. Multi-Dataset and Cross-Backbone Comparison

To compare the behavior of the proposed candidate generation strategy across different evaluation settings, this subsection reports results for three datasets and three feature extractors under a unified evaluation protocol. CALTECH101, CIFAR-10, and Tiny-ImageNet are considered together with VGG19, ConvNeXt-Tiny, and ViT-B/16. The compared methods include standard p-stable LSH, the proposed PCA-guided LSH, HNSW, and IVF. For each dataset–backbone combination, PCA and the corresponding candidate index are fitted independently using the database features from that setting. Therefore, this experiment evaluates the consistency of the method across multiple independently fitted settings rather than transfering of one fitted PCA model to an unseen dataset.

For each dataset–backbone combination, 1000 reference images are sampled, and one JPEG-compressed query image is generated from each reference image with a fixed quality factor of $q = 10$. The corresponding original image is treated as the registered target. Recall@10 is reported as the primary quantitative metric under this unified query setting. The results are summarized in Table 8.

The multi-dataset experiment is designed as a controlled comparison of projection-sampling behavior across different feature representations rather than as a million-scale ANN benchmark. This design makes it possible to compare the proposed projection modification with its mechanism-matched p-stable baseline under consistent query and candidate-budget conditions. Because the proposed method preserves the original multi-table index organization, its database-size dependence follows the same index-construction and bucket-lookup structure as standard p-stable LSH, with PCA guidance contributing an additional one-time offline preprocessing step.

As shown in Table 8, HNSW achieves the highest Recall@10 in all evaluated settings, while IVF also provides strong retrieval performance across most dataset–backbone combinations. Compared with standard p-stable LSH, the proposed PCA-guided LSH improves candidate recall in most settings, indicating that principal-direction-guided projection sampling is beneficial within the standard p-stable LSH framework.

Table 8. Recall@10 (%) of different candidate generation methods across datasets and feature extractors.

Dataset	VGG19				ConvNeXt-Tiny				ViT-B/16			
	p-Stable	PCA-Guided	HNSW	IVF	p-Stable	PCA-Guided	HNSW	IVF	p-Stable	PCA-Guided	HNSW	IVF
Caltech101	93.3	98.4	**99.8**	99.4	95.2	98.8	**99.9**	99.8	94.1	98.8	**100.0**	99.7
CIFAR-10	66.8	67.7	**78.6**	71.4	71.6	72.2	**87.3**	79.4	70.5	70.9	**79.4**	73.2
Tiny-ImageNet	70.7	72.6	**80.9**	73.4	76.7	79.1	**91.9**	83.4	71.2	73.7	**86.9**	82.0

Note: Bold values indicate the highest Recall@10 result for each dataset–backbone combination.

The gain of the proposed method is particularly clear on CALTECH101 and Tiny-ImageNet. On CALTECH101, PCA-guided LSH improves Recall@10 from 93.3% to 98.4% with VGG19, from 95.2% to 98.8% with ConvNeXt-Tiny, and from 94.1% to 98.8% with ViT-B/16. On Tiny-ImageNet, the corresponding improvements are from 70.7% to 72.6%, from 76.7% to 79.1%, and from 71.2% to 73.7%, respectively. On CIFAR-10, the improvements are smaller, increasing from 66.8% to 67.7% with VGG19, from 71.6% to 72.2% with ConvNeXt-Tiny, and from 70.5% to 70.9% with ViT-B/16. These results suggest that PCA-guided projection is effective as a lightweight refinement of the standard p-stable baseline under different feature representations.

The variation in improvement across datasets is consistent with the geometric motivation of the proposed method. PCA guidance is expected to provide a larger benefit when the fixed feature representation contains a sufficiently concentrated directional structure and when the leading variance directions are informative for separating the registered query–target neighborhoods. Under such conditions, the mean-shifted projection distribution can exploit directional information that is not used by isotropic sampling. In contrast, when the feature covariance is more diffuse, or when the locally useful neighborhood structure is not well represented by a single globally aggregated guidance direction, the difference from isotropic sampling is expected to be smaller. The relatively limited gains on CIFAR-10 suggest that less exploitable global directional information is available under the evaluated feature and query settings. Feature dimensionality alone does not explain the observed pattern, because the same dataset-dependent tendency is observed across VGG19, ConvNeXt-Tiny, and ViT-B/16 representations.

A practical boundary arises when the leading eigenvalues are only weakly separated and the feature covariance is close to isotropic. In this situation, the estimated principal direction provides limited additional information beyond isotropic sampling, and a strong directional shift may unnecessarily reduce projection diversity. The eigenvalue distribution can therefore serve as a preliminary diagnostic for the usefulness of PCA guidance. When no pronounced directional structure is observed, a smaller guidance strength may be selected; in the limiting case $\alpha = 0$, the projection distribution returns to the zero-mean Gaussian form used by standard p-stable LSH.

HNSW and IVF achieve higher overall Recall@10 in most evaluated settings, reflecting the stronger search capability of graph navigation and coarse-quantization-based probing. The proposed method is therefore positioned as a compatible refinement for systems that already use the multi-table p-stable LSH architecture and wish to introduce data-dependent guidance without replacing the existing bucket organization or online query procedure. Its practical advantages lie in limited integration changes, retention of the original hash-based candidate-generation pipeline, and the confinement of PCA-related computation to offline index preparation. Accordingly, PCA-guided LSH is not intended as a universal replacement for HNSW or IVF, but as a lightweight option when architectural compatibility, implementation simplicity, or preservation of an existing LSH-based deployment is important.

Figure 9 further provides qualitative retrieval examples under the unified JPEG query setting ($q = 10$). The figure shows that, for representative query cases, the proposed PCA-guided LSH can return more relevant candidates than standard p-stable LSH and can place the registered target in a more favorable candidate position.

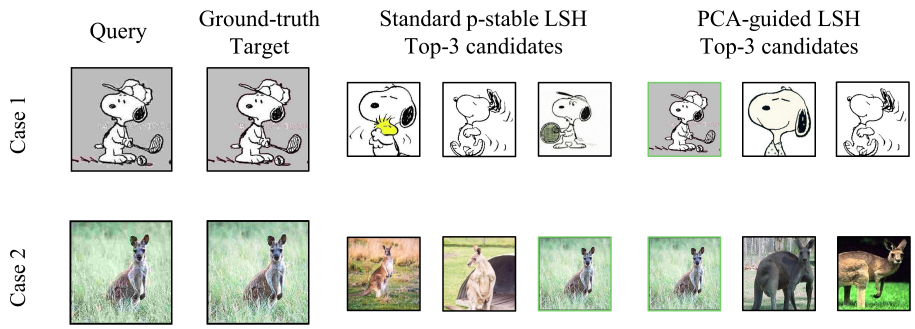


Figure 9. Qualitative retrieval examples of standard p-stable LSH and PCA-guided LSH under the unified JPEG query setting ($q = 10$).

6. Conclusions

This paper studies the problem of projection-direction design in image candidate generation and proposes a PCA-guided candidate generation method based on the hash-function form and multi-table retrieval architecture of p-stable LSH. Without modifying the original bucketization mechanism or online query procedure, the proposed method incorporates principal-direction information of deep image features into projection generation, so as to improve the alignment between candidate partitioning and the underlying feature distribution. Unlike conventional PCA-based feature extraction, PCA is not used here to transform the image representation itself, but only to bias the sampling distribution of projection vectors in the indexing stage.

A deep feature-based candidate generation pipeline is constructed to evaluate the proposed method. Experimental results from parameter sensitivity analysis, sensitivity analysis of the number of principal components, perturbation-stability evaluation, and multi-dataset comparison show that the proposed PCA-guided LSH generally improves on the standard p-stable baseline in most settings. The gain is particularly clear on CALTECH101 and Tiny-ImageNet, and the method also exhibits more stable retrieval behavior under multiple content-preserving perturbations. Additional experiments with ConvNeXt-Tiny, ViT-B/16, HNSW, and IVF further place the proposed method in a broader and more current retrieval setting.

Overall, the proposed method can be regarded as a lightweight data-dependent projection-sampling refinement for systems based on the multi-table p-stable LSH architecture. Its main value lies in improving candidate coverage and robustness without changing the basic retrieval pipeline. At the same time, the results also indicate that modern ANN baselines remain stronger in overall recall, suggesting that the proposed method is better understood as a compatible enhancement of classical p-stable LSH rather than as a replacement for more advanced ANN indexing structures.

Author Contributions: Conceptualization, Y.S. (Yubo Song); methodology, Y.S. (Yang Sun); software, Y.S. (Yang Sun) and T.W.; formal analysis, M.L.; investigation, Y.S. (Yang Sun) and T.W.; data curation, Y.S. (Yang Sun); writing—original draft preparation, Y.S. (Yang Sun); writing—review and editing, T.W. and Y.S. (Yubo Song); visualization, Y.S. (Yang Sun) and M.L.; supervision, Y.S. (Yubo Song); project administration, Y.S. (Yubo Song); funding acquisition, Y.S. (Yubo Song). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Funds for the Central Universities (No. 2242022K60005), China.

Data Availability Statement: Data is contained within the article.

Acknowledgments: The authors would like to acknowledge the support of the Fundamental Research Funds for the Central Universities.

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

References

1. Gusev, A.; Xu, J. Evolution of a web-scale near duplicate image detection system. In Proceedings of the Web Conference 2020, Taipei, China, 20–24 April 2020; pp. 2733–2739.
2. Indyk, P.; Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Dallas, TX, USA, 23–26 May 1998; pp. 604–613.
3. Gionis, A.; Indyk, P.; Motwani, R. Similarity search in high dimensions via hashing. *Proc. VLDB* **1999**, *99*, 518–529.
4. 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.
5. Song, X.; Lin, H.; Wen, H.; Hou, B.; Xu, M.; Nie, L. A comprehensive survey on composed image retrieval. *ACM Trans. Inf. Syst.* **2025**, *44*, 1–54. [CrossRef]
6. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
7. Chum, O.; Philbin, J.; Isard, M.; Zisserman, A. Scalable near identical image and shot detection. In Proceedings of the 6th ACM International Conference on Image and Video Retrieval, Amsterdam, The Netherlands, 9–11 July 2007; pp. 549–556.
8. McKeown, S.; Buchanan, W.J. Hamming distributions of popular perceptual hashing techniques. *Forensic Sci. Int. Digit. Investig.* **2023**, *44*, 301509. [CrossRef]
9. Chen, W.; Liu, Y.; Wang, W.; Bakker, E.M.; Georgiou, T.; Fieguth, P.; Liu, L.; Lew, M.S. Deep learning for instance retrieval: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 7270–7292. [CrossRef] [PubMed]
10. Zhang, C.; Liu, J. Content based deep learning image retrieval: A survey. In Proceedings of the 2023 9th International Conference on Communication and Information Processing, Lingshui, China, 14–16 December 2023; pp. 158–163.
11. Lee, S.; Seong, H.; Lee, S.; Kim, E. Correlation verification for image retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 5374–5384.
12. Kordopatis-Zilos, G.; Stojnić, V.; Manko, A.; Suma, P.; Ypsilantis, N.A.; Efthymiadis, N.; Laskar, Z.; Matas, J.; Chum, O.; Tolias, G. Ilias: Instance-level image retrieval at scale. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 11–15 June 2025; pp. 14777–14787.
13. Song, C.H.; Yoon, J.; Hwang, T.; Choi, S.; Gu, Y.H.; Avrithis, Y. On train-test class overlap and detection for image retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 17375–17384.
14. Vendrow, E.; Pantazis, O.; Shepard, A.; Brostow, G.; Jones, K.E.; Mac Aodha, O.; Beery, S.; Van Horn, G. INQUIRE: A natural world text-to-image retrieval benchmark. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 126500–126514. [CrossRef]
15. Wan, Y.; Wang, W.; Zou, G.; Zhang, B. Cross-modal feature alignment and fusion for composed image retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 8384–8388.
16. Choi, W.; Lim, S.; Hyeon-Woo, N.; Ye-Bin, M.; Jeong, D.J.; Hwang, J.; Oh, T.H. Patch-wise retrieval: A bag of practical techniques for instance-level matching. In *Proceedings of the 2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*; IEEE: New York, NY, USA, 2026; pp. 4922–4931.
17. Wang, R.F.; Cui, K.; Schardong, I.B.; Bauer, M.C.; Somala, R.V.; Xu, M.; West, D.; Jones, D.C.; Taylor, S.V.; Roberts, P.M.; et al. JassidNet as a quantization-aware lightweight phenotyping framework for high-throughput cotton jassid (*Amrasca biguttula*) detection and counting toward objective resistance screening. *Ind. Crop. Prod.* **2026**, *249*, 123716. [CrossRef]
18. Sun, H.; Xi, X.; Wu, A.Q.; Wang, R.F. ToRLNet: A lightweight deep learning model for tomato detection and quality assessment across ripeness stages. *Horticulturae* **2025**, *11*, 1334. [CrossRef]
19. Malkov, Y.A.; Yashunin, D.A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 824–836. [CrossRef] [PubMed]
20. Pandit, G.; Röder, M.; Ngonga Ngomo, A.C. Evaluating Approximate Nearest Neighbour Search Systems on Knowledge Graph Embeddings. In *Proceedings of the European Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 59–76.
21. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* **2019**, *7*, 535–547. [CrossRef]
22. Weiss, Y.; Torralba, A.; Fergus, R. Spectral hashing. *Adv. Neural Inf. Process. Syst.* **2008**, *21*, 1753–1760.
23. 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]
24. Kulis, B.; Grauman, K. Kernelized locality-sensitive hashing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 1092–1104. [CrossRef] [PubMed]

25. Zhang, Z.; Wang, J.; Zhu, L.; Luo, Y.; Lu, G. Deep collaborative graph hashing for discriminative image retrieval. *Pattern Recognit.* **2023**, *139*, 109462. [CrossRef]
26. Yang, W.; Wang, L.; Cheng, S. Deep parameter-free attention hashing for image retrieval. *Sci. Rep.* **2022**, *12*, 7082. [CrossRef] [PubMed]
27. Yu, X.; Zhang, S.; Liu, B.; Zhong, L.; Metaxas, D. Large scale medical image search via unsupervised PCA hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, OR, USA, 23–28 June 2013; pp. 393–398.
28. Yilmaz, A. Bilateral cross hashing image retrieval based on principal component analysis. *Arab. J. Sci. Eng.* **2025**, *50*, 12495–12512. [CrossRef]
29. Mascarenhas, S.; Agarwal, M. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In *Proceedings of the 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*; IEEE: New York, NY, USA, 2021; Volume 1, pp. 96–99.
30. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 11976–11986.
31. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
32. Groth, D.; Hartmann, S.; Klie, S.; Selbig, J. Principal components analysis. In *Computational Toxicology: Volume II*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 527–547.
33. Fei-Fei, L.; Fergus, R.; Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*; IEEE: New York, NY, USA, 2004; p. 178.
34. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, Department of Computer Science, University of Toronto, Toronto, ON, Canada, 2009. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 31 August 2026).
35. Le, Y.; Yang, X. Tiny imagenet visual recognition challenge. *CS 231N* **2015**, *7*, 3.
36. Andoni, A.; Indyk, P.; Laarhoven, T.; Razenshteyn, I.; Schmidt, L. Practical and optimal LSH for angular distance. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1225–1233.
37. Norouzi, M.; Punjani, A.; Fleet, D.J. Fast search in hamming space with multi-index hashing. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2012; pp. 3108–3115.
38. Elliott, O.P.; Clark, J. The impacts of data, ordering, and intrinsic dimensionality on recall in hierarchical navigable small worlds. In Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval, Washington, DC, USA, 13 July 2024; pp. 25–33.
39. Aumüller, M.; Bernhardsson, E.; Faithfull, A. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Inf. Syst.* **2020**, *87*, 101374. [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.