JCTA Journal of Computing Theories and Applications ISSN:3024-9104

*Research/Review Article*

# Frozen DINOv2–CLIP Fusion and Binary Hashing for Remote Sensing Image Retrieval: A Controlled Evaluation

**Jumi [1,*], Tedjo Mulyono [2], Achmad Zaenuddin [3], and Suwardi [4]**

[1] Department of Business Administration, Politeknik Negeri Semarang, Semarang 50275, Indonesia; e-mail : jumimail06@gmail.com; zaeyenne@yahoo.co.id; cakshoe@gmail.com
[2] Department of Civil Engineering, Politeknik Negeri Semarang, Semarang 50275, Indonesia; e-mail : teted64@gmail.com
* Corresponding Author : Jumi

**Abstract:** This study evaluates a controlled content-based remote sensing image retrieval pipeline that combines frozen vision foundation models with binary hashing. Across five stratified seeds on PatternNet and NWPU-RESISC45, calibrated DINOv2–CLIP score fusion achieved mAP values of 0.8143 ± 0.0007 and 0.5693 ± 0.0020, respectively, improving over the stronger single backbone by 0.0582 and 0.0514. The two similarity spaces were related but not redundant, with Spearman correlations of ρ = 0.510 and 0.515 across 200,000 sampled pairs. The contribution of texture features was dataset- and backbone-dependent: DINOv2–texture received a weight of 0.1 on PatternNet, whereas CLIP–texture and three-stream fusion assigned zero weight to texture; forcing texture into a 32-bit representation reduced mAP by 0.0147 and 0.0510. Among the unsupervised hashing methods, 64-bit iterative quantization (ITQ) exceeded the full-precision fused baseline by 0.0533 and 0.0445 mAP on PatternNet and NWPU-RESISC45, respectively. The corresponding paired parametric tests were significant, although the exact five-pair sign-flip test remained resolution-limited (p = 0.0625). A supervised CSQ-style head achieved mAP values of 0.9886 and 0.9087 at 64 bits; however, this improvement cannot be attributed to compression alone because gallery labels were used during training. In a controlled 200,000-vector benchmark constructed through deterministic repetition, 64-bit flat Hamming search required 1.6 MB and 0.674 ms/query, compared with 1.02 GB and 5.343 ms/query for 1,280-dimensional float vectors. Overall, the results support DINOv2–CLIP fusion and 64–128-bit ITQ as practical label-free choices, while restricting the scalability claim to the tested flat-search setting.

**Keywords:** Binary hashing; CLIP; Content-based image retrieval; DINOv2; Feature fusion; Iterative quantization; Remote sensing; Vision foundation models.

**Received: June, 24th 2026**
**Revised: July, 25th 2026**
**Accepted: August, 2nd 2026**
**Published: August, 13th 2026**



## 1. Introduction

Earth-observation programs generate image collections that are increasingly difficult to inspect manually. Content-based remote sensing image retrieval (CBRSIR) addresses this challenge by ranking archive images according to their visual or semantic similarity to a query [1], [2]. The task is challenging because aerial scenes exhibit variations in scale, orientation, illumination, and season, while visually similar structures may belong to different semantic classes [3], [4].

Deep global descriptors have largely replaced handcrafted features in image retrieval [5]–[7]. More recently, vision transformers and large-scale pre-training have produced reusable feature spaces [8]–[14]. DINOv2 learns visual structure without language supervision [8], whereas CLIP organizes image features through image–text alignment [9]. Because their pre-training objectives differ, their similarity signals may capture complementary aspects of aerial scenes. If this complementarity is meaningful, combining the two similarity spaces should provide more consistent retrieval than either space alone. Conversely, a high correlation between their similarity scores would suggest substantial redundancy and would limit the expected benefit of fusion. Therefore, mAP improvement alone does not establish non-redundancy; the relationship between the two spaces should also be measured directly.

Remote-sensing retrieval has concurrently moved toward self-supervised and transformer-based hashing. Deep contrastive self-supervised hashing uses augmentation consistency [15], soft-pseudo-label hashing learns unlabeled neighborhood structure [16], and deep attention hashing introduces distance-adaptive ranking [17]. Domain-specific foundation models, including RemoteCLIP [18], SatMAE [19], and continual geospatial pre-training [20], demonstrate that Earth-observation pre-training can differ materially from general web-image pre-training. These studies primarily learn task-specific hash functions or domain-specific encoders. In contrast, the present work keeps two general-purpose backbones frozen and asks a narrower systems question: how much can complementary pretrained features, transparent calibration, and established hash functions deliver without remote-sensing fine-tuning? This positioning avoids claiming a new backbone or loss.

On the indexing side, full-precision exhaustive search incurs storage and arithmetic costs [21], [22]. Binary hashing reduces descriptors to short codes that support Hamming-distance search [23]–[27]. However, supervised and unsupervised hashing consume different types of information. A supervised head may outperform a float baseline because labels reshape the representation, rather than because binarization is intrinsically beneficial. Comparisons should therefore be made within the same information regime and against the same fused float representation. This distinction is particularly important when the objective is to assess the value of binary compression itself rather than the additional information introduced by supervision. This controlled study addresses three research questions (RQs):

- RQ1: Are frozen DINOv2 and CLIP similarity spaces complementary for CBRSIR, and does handcrafted texture add information once these foundation embeddings are available?
- RQ2: How do data-independent, unsupervised data-dependent, and supervised binary codes trade retrieval quality against code length when fitted to the same fused representation?
- RQ3: What storage and flat-search latency differences are observed at the tested 200,000-vector scale?

The contributions are an explicit leakage-controlled protocol with exact checkpoints and preprocessing; a correlation and class-wise analysis of DINOv2–CLIP complementarity; a nuanced texture result rather than a universal zero-weight claim; within-regime hashing comparisons with five-seed uncertainty and paired tests; and a narrowly scoped 200,000-vector efficiency benchmark. The study is analytical rather than algorithmic, and all numerical claims are linked to the supplied seed-level results and source notebook.

## 2. Related Work

### 2.1. Foundation Representations for Retrieval

Global convolutional descriptors [5], [6], learned pooling [7], and attentive local features [28] established strong baselines for general image retrieval. More recently, vision transformers [12], [13] and self-supervised pre-training [10], [14] have improved the transferability of learned visual representations, while DINOv2 provides strong frozen features without requiring task-specific fine-tuning [8]. CLIP provides a complementary visual space learned through image–text alignment [9], [11]. These differences in pre-training objectives motivate the possibility that DINOv2 and CLIP may encode partially different similarity cues, particularly for scenes with both structural and semantic variation.

For Earth observation, domain-specific foundation models such as RemoteCLIP [18] and SatMAE [19], as well as continual domain pre-training [20], demonstrate the value of adapting representation learning to remote-sensing data and sensing modalities. However, these studies primarily focus on developing or adapting domain-specific representations. The present study takes a different and narrower perspective: it keeps two accessible general-purpose foundation models frozen and evaluates whether their similarity spaces provide complementary retrieval information when combined through calibrated score fusion. This distinction is important because the study aims to isolate the value of complementary pretrained representations rather than the effect of additional remote-sensing pre-training or backbone fine-tuning.

### 2.2. Remote-Sensing Retrieval and Hashing

Remote-sensing image hashing has evolved from supervised deep networks [29] and metric-learning approaches [30] toward self-supervised, pseudo-label, and attention-based methods [15]–[17]. These approaches demonstrate that binary representations can improve retrieval efficiency while preserving semantic similarity. However, their learning objectives and supervision settings differ substantially, making direct comparisons between supervised and unsupervised hashing potentially misleading.

Among established hashing approaches, LSH is data-independent and does not require learning from the retrieval gallery [26], [27]. PCA-sign uses gallery covariance information without class labels, whereas ITQ learns an orthogonal rotation after PCA to reduce binary quantization error [23]. In contrast, CSQ-style learning associates semantic classes with binary centers and therefore uses label information [31]. The latter regime should consequently be evaluated separately from label-free hashing when the objective is to assess the benefit of binary representation itself. Approximate-nearest-neighbor structures such as product quantization and HNSW provide complementary alternatives to the flat indexes considered in this study [21], [22].

**Table 1.** Positioning of the present study against representative remote-sensing retrieval studies.

| Study | Representation | Supervision | Primary focus | Distinction from the present study |
|---|---|---|---|---|
| Li et al. [29] | CNN hash | Labels | Deep remote-sensing hashing | Frozen foundation features and separated supervision regimes |
| Roy et al. [30] | Metric hash | Labels | Metric-learning binary codes | No backbone fine-tuning |
| Tan et al. [15] | Contrastive hash | Self-supervised | Augmentation consistency | Established hashing methods applied to fixed fused features |
| Sun et al. [16] | Deep hash | Unsupervised | Soft pseudo-labels | Calibration isolates the effect of representation fusion |
| Zhang et al. [17] | Attention hash | Labels | Distance-adaptive ranking | No new loss or attention module |
| RemoteCLIP [18] | Domain-specific VLM | Image–text | Remote-sensing foundation representation | Evaluates general-purpose DINOv2–CLIP complementarity rather than a domain-specific encoder |
| This study | Frozen DINOv2 + CLIP | None for fusion; labels only for CSQ | Controlled fusion, texture, hashing, and efficiency | Explicit splits, uncertainty analysis, representation correlation, and bounded scale evaluation |

Table 1 highlights the main methodological gap addressed by this study. Existing remote-sensing retrieval research has largely focused on developing new retrieval representations, learning task-specific hash functions, or adapting foundation models to the remote-sensing domain. In contrast, the present work evaluates how far frozen, general-purpose foundation features can be exploited through transparent fusion and established hashing methods without introducing a new backbone, loss function, or fine-tuning procedure. This controlled setting enables a more direct analysis of representation complementarity, the residual value of handcrafted texture, and the effect of supervision on hashing performance.

## 3. Proposed Method

The proposed methodology follows a controlled evaluation pipeline that separates feature extraction, fusion, hashing, and retrieval. Figure 1 summarizes the overall CBRSIR pipeline.

### 3.1. Data Partitioning and Relevance

PatternNet contains 30,400 images from 38 balanced classes [3], while NWPU-RESISC45 contains 31,500 images from 45 balanced classes [4]. Table 2 summarizes

the dataset partition sizes used for each seed. For both datasets, the partitioning is performed independently for each seed while preserving the class distribution.

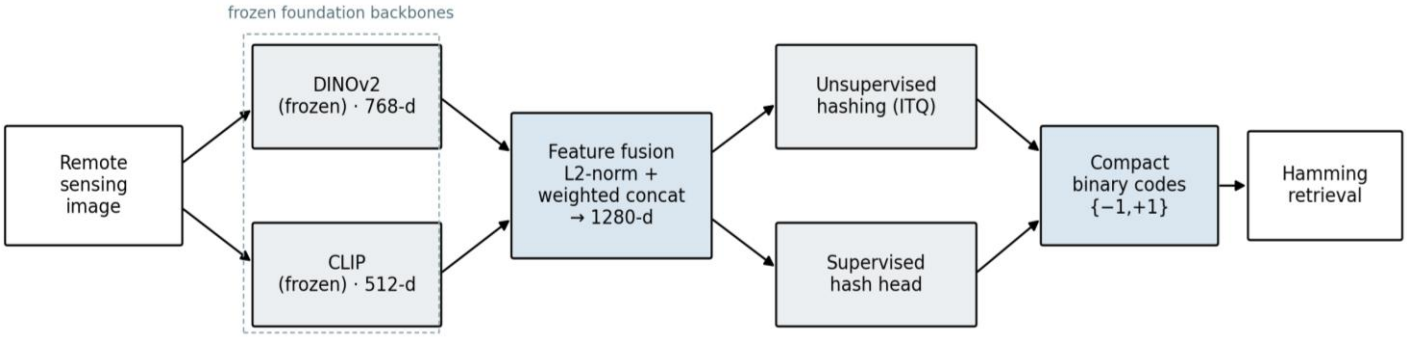


**Figure 1.** Controlled CBRSIR pipeline.

**Table 2.** Dataset partition sizes per seed.

| Dataset | Classes | Images per class | Gallery | Calibration | Test queries | Relevant gallery/ query |
|---|---|---|---|---|---|---|
| PatternNet | 38 | 800 | 18,240 | 6,080 | 6,080 | 480 |
| NWPU-RESISC45 | 45 | 700 | 18,900 | 6,300 | 6,300 | 420 |

For each seed $s \in \{0,1,2,3,4\}$, image indices are shuffled independently within each class using NumPy default_rng(s). The first 60% of each class are assigned to the gallery, the next 20% to the calibration set, and the final 20% to the test-query set. The resulting index arrays are sorted to fix the evaluation order. The gallery, calibration, and test sets are mutually disjoint. An item is considered relevant to a query when its class label matches the query label.

### 3.2. Frozen Feature Extraction and Texture

DINOv2 and CLIP are used as frozen feature extractors to isolate the effect of their pretrained representations without introducing task-specific fine-tuning. Table 3 summarizes the backbone checkpoints, input preprocessing, output dimensions, and normalization procedures used for both foundation-model streams and the handcrafted texture stream.

DINOv2 is instantiated through timm as vit_base_patch14_dinov2.lvd142m with pretrained weights and num_classes=0. The resulting 768-dimensional global class-token output is extracted at a resolution of 518×518 pixels. The timm data configuration uses bicubic resizing, center cropping with crop_pct = 1.0, and ImageNet mean (0.485,0.456,0.406) and standard deviation (0.229,0.224,0.225).

CLIP is instantiated using open_clip.create_model_and_transforms ('ViT-B-16-quickgelu', pretrained='openai'). Its 512-dimensional image embedding is extracted using the model's default transformation at 224×224 pixels, with bicubic resizing and center cropping. The normalization uses mean (0.48145466,0.45782750,0.40821073) and standard deviation (0.26862954,0.26130258,0.27577711). The QuickGELU checkpoint is used consistently throughout the experiments; an earlier incompatible cache was deleted before the recorded feature extraction.

All images are converted to RGB. Both encoders operate in evaluation mode with gradients disabled, using a batch size of 64, two data-loader workers, and CUDA automatic mixed precision. The resulting embeddings are converted to float32 and L2-normalized. The exact OpenCLIP package version was not logged; therefore, the checkpoint identifier, preprocessing constants, extraction code, canonical array dimensions, and SHA-256 hashes are provided to bound this reproducibility limitation.

The handcrafted texture descriptor is computed after converting each image to grayscale and resizing it to 128×128 pixels. Sixteen complex Gabor filters are constructed using frequencies {0.1, 0.2, 0.3, 0.4}, orientations {0, $\pi/4$, $\pi/2$, $3\pi/4$}, 39-pixel kernels, $\gamma$=0.5, $\psi$=0, and $\sigma$=0.56/f. The real and imaginary responses are combined by magnitude, and their mean and standard deviation produce 32 values. Uniform LBP histograms with (P,R)=(8,1)

and (16,2) contribute 10 and 18 bins, respectively. The resulting 60-dimensional texture vector is standardized using gallery-only means and standard deviations and then L2-normalized.

**Table 3.** Feature-extraction and texture configuration.

| Stream | Checkpoint / Descriptor | Input and preprocessing | Output | Normalization |
|---|---|---|---|---|
| DINOv2 | timm vit_base_patch14_dinov2.lvd142m | (518\times518); bicubic resize; center crop | 768-D class token | ImageNet mean/SD; L2 |
| CLIP | OpenCLIP ViT-B-16-QuickGELU, OpenAI weights | (224\times224); bicubic resize; center crop | 512-D image embedding | OpenAI CLIP mean/SD; L2 |
| Texture | 16 Gabor mean/SD features + 2 uniform LBP histograms | (128\times128) gray-scale | 60-D | Gallery z-score; L2 |

### 3.3. Calibrated Fusion

For L2-normalized streams $x_1$ and $x_2$, the fused vector is defined as $L2\left(\left[\sqrt{w}x_1, \sqrt{1-w}x_2\right]\right)$. Its inner product is exactly $w\langle x_1, x_1'\rangle + (1-w)\langle x_2, x_2'\rangle$, so weighted concatenation implements score fusion. Two-stream weights are searched over {0.0, 0.1, … ,1.0}, while three-stream weights are searched over the non-negative simplex in increments of 0.2. Calibration mAP is used as the optimization objective. A candidate replaces the incumbent only when its score is strictly higher; therefore, exact ties retain the first lexicographic grid point. The selected weights are then fixed for each seed and evaluated once on the test queries. Table 4 reports the calibration-selected weights for the two- and three-stream configurations. The same selections were obtained across all five seeds.

**Table 4.** Calibration-selected fusion weights.

| Dataset | DINOv2 + CLIP | CLIP + Texture | DINOv2 + Texture | DINOv2 + CLIP + Texture |
|---|---|---|---|---|
| PatternNet | (0.3, 0.7) | (1.0, 0.0) | (0.9, 0.1) | (0.2, 0.8, 0.0) |
| NWPU-RESISC45 | (0.3, 0.7) | (1.0, 0.0) | (1.0, 0.0) | (0.2, 0.8, 0.0) |

All hashing experiments use the fixed 1,280-dimensional L2-normalized concatenation $\left[\sqrt{0.3}\mathrm{DINOv2}, \sqrt{0.7}\mathrm{CLIP}\right]$. This representation is determined by the stable calibration outcome rather than by test-set performance.

### 3.4. Binary Encoders

Four binary encoding approaches are evaluated to distinguish data-independent, unsupervised data-dependent, and supervised regimes. LSH uses b Gaussian random projections generated with seed $1000+s$. PCA-sign centers the gallery vectors, estimates the top-$b$ covariance eigenvectors, and thresholds the resulting projections at zero. ITQ first applies the same $b$-dimensional PCA projection, initializes an orthogonal rotation using a QR factorization with seed $s$, and then alternates binary sign assignment and SVD-based Procrustes rotation for 50 iterations [23]. Thus, PCA-sign provides a dimensionality-reduction baseline, while ITQ additionally optimizes an orthogonal rotation to reduce sign-quantization error.

The supervised comparator is a CSQ-style head rather than a claim to reproduce the entire original CSQ system [31]. Class centers are selected by max–min Hamming separation from 4,000 Rademacher candidates. A 1,280→1,024→b multilayer perceptron with ReLU is trained using gallery labels with Adam (learning rate $10^{-3}$, weight decay $10^{-5}$), a batch size of 512, and 20 epochs. The loss consists of binary cross-entropy between the logits and the assigned class center, plus 0.1 times the mean squared term $(1-|\tanh(\mathrm{logit})|)^2$. Encoding is performed in batches of 2,048. LSH, PCA-sign, and ITQ are evaluated at 16, 32, 64, and 128 bits, while the supervised head is evaluated at 16, 32, and 64 bits. Table 5 summarizes the hashing configurations and implementation hyperparameters used in the experiments.

**Table 5.** Hashing and implementation hyperparameters.

| Component | Configuration |
|---|---|
| LSH | Gaussian projections; random seed (1000+$s$); sign threshold |
| PCA-sign | Gallery-centered top-$b$ PCA coordinates; sign threshold |
| ITQ | Top-$b$ PCA; QR initialization with seed $s$; 50 sign/SVD iterations |
| CSQ-style | 1280–1024–$b$; Adam, $10^{-3}$; weight decay $10^{-5}$; batch size 512; 20 epochs; $\lambda_q$=0.1 |
| Software | Python 3.12.13; PyTorch 2.11.0+cu128; torchvision 0.26.0+cu128; timm 1.0.27; FAISS 1.14.3; NumPy 2.0.2; SciPy 1.16.3; scikit-learn 1.6.1; scikit-image 0.25.2; Pillow 11.3.0 |
| Hardware | NVIDIA Tesla T4, 15.6 GB; CUDA 12.8; CPU model not recorded |

The complete leakage-controlled procedure for one dataset seed, from data partitioning and feature extraction to fusion, hashing, and retrieval evaluation, is summarized in Algorithm 1.

**Algorithm 1.** Leakage-Controlled Fusion, Hashing, and Retrieval Evaluation

INPUT: Images and labels; seed $s$; code length $b$; admissible fusion-weight grids
OUTPUT: Per-seed retrieval metrics, selected fusion weights, class-wise metrics, and efficiency measurements

1: Within each class, shuffle the image indices using NumPy default_rng(s) and allocate 60% to the gallery, 20% to calibration, and 20% to the test-query set.
2: Extract frozen DINOv2 and CLIP embeddings, compute gallery-standardized texture features, and L2-normalize all feature streams.
3: Evaluate each admissible two- and three-stream fusion weight on the calibration queries and select the weight that maximizes calibration mAP; resolve exact ties using the first lexicographic grid point.
4: Fix the selected fusion weights and construct the corresponding 1,280-dimensional L2-normalized fused representation for the hashing experiments.
5: Fit LSH, PCA-sign, ITQ, or the CSQ-style supervised head using gallery data only; exclude calibration and test labels from the fitting process.
6: Encode the gallery and test images, then rank the results using cosine similarity for full-precision representations or Hamming distance for binary codes.
7: Compute AP, mAP, P@K, and R@100 for each seed; repeat the procedure for $s \in \{0,1,2,3,4\}$ and report the mean and sample SD.

### 3.5. Metrics and Statistical Analysis

For each query, average precision (AP) is computed over the complete gallery ranking, and mean average precision (mAP) is obtained by averaging AP across all test queries [32]. Precision at K (P@K) is defined as the fraction of relevant items among the top-K retrieved results. Recall@100 (R@100) is computed as the number of relevant items retrieved within the first 100 ranks divided by the total number of relevant gallery items, which is 480 for PatternNet and 420 for NWPU-RESISC45. All retrieval results are reported as the mean ± sample SD across five seeds.

Paired seed-wise differences are used to assess DINOv2–CLIP fusion against the stronger single-backbone baseline and ITQ against the fused full-precision baseline. For each comparison, a two-sided paired t-test, a 95% t-interval for the mean difference, and an exact two-sided sign-flip test are reported. Because only five paired seeds are available, the smallest attainable two-sided exact p-value when all differences have the same sign is $2/2^5$=0.0625. Thus, the parametric test provides an estimate of the consistency and magnitude of the observed differences, whereas the exact sign-flip test is inherently resolution-limited and cannot reach the conventional 0.05 threshold with five pairs.

To further assess representation complementarity, Pearson and Spearman correlations are computed between the DINOv2 and CLIP cosine similarities for 200,000 deterministically sampled image pairs using analysis seed 20260723. The analysis also compares similarity distributions for same-class and different-class pairs to determine whether the two encoders provide overlapping or distinct similarity information.

## 4. Results and Discussion

### 4.1. RQ1: Single Features, Fusion, and Complementarity

#### *4.1.1. Single-Feature and Fusion Performance*

RQ1 examines whether frozen DINOv2 and CLIP provide complementary retrieval information and whether handcrafted texture features remain useful once foundation-model embeddings are available. The analysis first compares individual feature streams and calibrated combinations, followed by precision–recall behavior and direct analysis of the two similarity spaces. Table 6 summarizes the retrieval performance of the individual and fused representations across the two datasets.

**Table 6.** Retrieval mAP (mean ± sample SD over five seeds). Fusion weights are reported in the order of the corresponding representation streams.

| Representation | PatternNet | NWPU-RESISC45 | Selected weight |
|---|---|---|---|
| DINOv2 | 0.7356 ± 0.0005 | 0.4843 ± 0.0036 | — |
| CLIP | 0.7561 ± 0.0009 | 0.5178 ± 0.0015 | — |
| Texture | 0.3450 ± 0.0023 | 0.1127 ± 0.0008 | — |
| DINOv2 + CLIP | **0.8143 ± 0.0007** | **0.5693 ± 0.0020** | (0.3, 0.7) |
| CLIP + texture | 0.7561 ± 0.0009 | 0.5178 ± 0.0015 | (1.0, 0.0) |
| DINOv2 + texture | 0.7583 ± 0.0009 | 0.4843 ± 0.0036 | (0.9, 0.1) / (1.0, 0.0) |
| DINOv2 + CLIP + texture | 0.8111 ± 0.0008 | 0.5666 ± 0.0017 | (0.2, 0.8, 0.0) |

CLIP is the stronger single frozen backbone on both datasets, achieving mAP values of 0.7561 on PatternNet and 0.5178 on NWPU-RESISC45. However, the calibrated DINOv2–CLIP fusion achieves 0.8143 and 0.5693, respectively, corresponding to improvements of 0.0582 and 0.0514 over the stronger single-backbone baseline. The mean paired differences have 95% t-intervals of [0.0577, 0.0586] and [0.0495, 0.0534], with paired t-test p-values of $2.77\times10^{-10}$ and $2.22\times10^{-7}$. The exact two-sided sign-flip test gives $p=0.0625$ for both datasets because only five paired seeds are available. Thus, the fusion improvement is consistent across the observed splits, although the exact nonparametric test is inherently resolution-limited at this sample size.

The retrieval quality of the fused representation is also reflected in its precision and recall. Table 7 reports P@K and R@100 for the calibrated DINOv2–CLIP representation. Precision remains high at small retrieval depths and gradually decreases as more results are retrieved, while R@100 captures the proportion of relevant gallery images recovered within the first 100 ranks.

**Table 7.** Precision and recall of the calibrated DINOv2–CLIP fusion (mean ± sample SD over five seeds).

| Dataset | P@5 | P@10 | P@20 | P@50 | P@100 |
|---|---|---|---|---|---|
| PatternNet | 0.9830 ± 0.0019 | 0.9793 ± 0.0018 | 0.9730 ± 0.0016 | 0.9592 ± 0.0014 | 0.9393 ± 0.0016 |
| NWPU-RESISC45 | 0.8998 ± 0.0013 | 0.8807 ± 0.0007 | 0.8555 ± 0.0006 | 0.8111 ± 0.0012 | 0.7622 ± 0.0015 |

The relatively high P@5 and P@10 values indicate that the fused representation places relevant images near the top of the ranking, particularly on PatternNet. The lower R@100 values are expected given the relatively large number of relevant gallery images per query and the limited retrieval depth of 100. Thus, the precision and recall results provide a complementary view of the mAP improvements rather than suggesting that the fused representation retrieves all relevant images within the top 100 ranks.

Figure 2 provides representative seed-0 retrieval traces for two NWPU-RESISC45 queries. In both examples, the calibrated DINOv2–CLIP fusion increases the number of same-class gallery items within the top five compared with either single backbone.

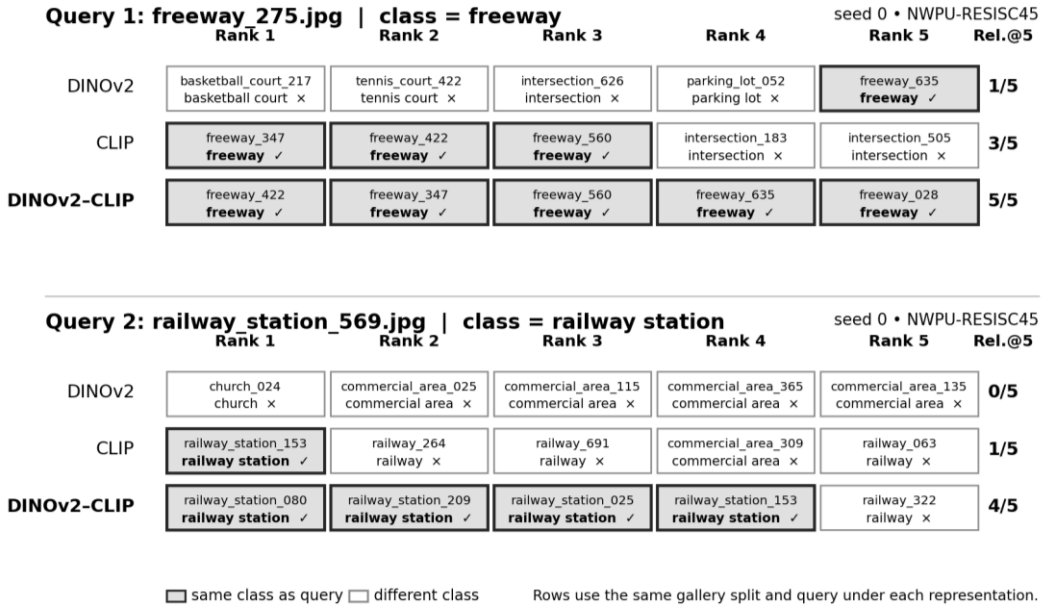


**Figure 2**. Representative seed-0 retrieval traces on NWPU-RESISC45. Shaded entries share the query class; Rel.@5 denotes the number of relevant items among the top five results.

### *4.1.2. Representation Complementarity*

The representation analysis provides further evidence beyond mAP. Figure 3 shows the relationship between DINOv2 and CLIP cosine similarities over 200,000 deterministically sampled image pairs for each dataset. The similarity scores have Pearson correlations of r=0.559 and r=0.537, and Spearman correlations of $\rho$=0.510 and $\rho$=0.515 on PatternNet and NWPU-RESISC45, respectively. These moderate correlations indicate that the two similarity spaces are related but not identical.

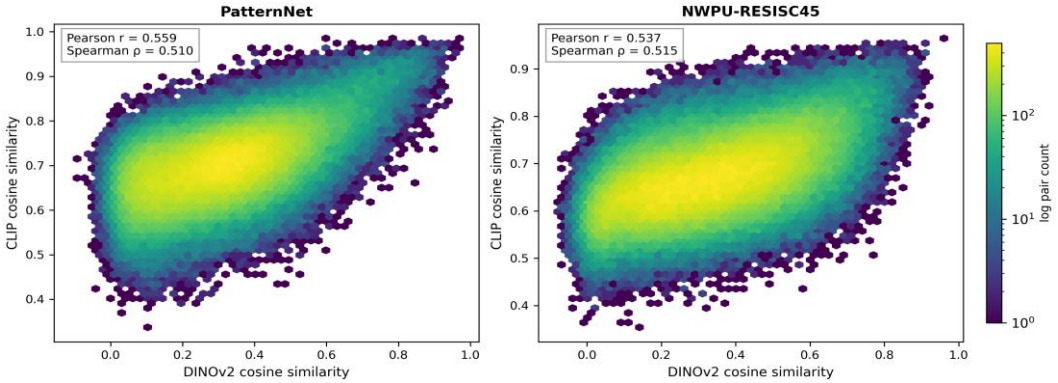


**Figure 3.** Relationship between frozen DINOv2 and CLIP cosine similarities over 200,000 sampled image pairs per dataset.

Same-class and different-class mean similarities provide additional context. On PatternNet, the corresponding means are 0.663 versus 0.306 for DINOv2 and 0.866 versus 0.698 for CLIP. On NWPU-RESISC45, they are 0.632 versus 0.337 for DINOv2 and 0.822 versus 0.670 for CLIP. Both representations therefore distinguish same-class from different-class pairs, but their only moderate rank correlation suggests that they do not make identical similarity judgments. This provides evidence of practical complementarity under the tested protocol and helps explain why score fusion improves retrieval performance. It does not, however, imply statistical independence between the two representations.

### *4.1.3. Texture Contribution*

The contribution of texture requires a more nuanced interpretation. Texture is not assigned zero weight in every configuration: the DINOv2–texture combination selects a weight of (0.9, 0.1) on PatternNet and improves over DINOv2 alone by 0.0227 mAP, whereas the

selected texture weight is zero on NWPU-RESISC45. In contrast, the CLIP–texture combination assigns all weight to CLIP on both datasets, and the three-stream fusion also assigns zero weight to texture. Adding texture to the DINOv2–CLIP representation consequently reduces mAP by 0.0032 on PatternNet and 0.0027 on NWPU-RESISC45. Moreover, when texture is forced into a 32-bit compact representation, mAP decreases by 0.0147 and 0.0510, respectively.

Taken together, these results do not support a universal claim that handcrafted texture is useless. Instead, they indicate that its contribution is dataset- and representation-dependent: texture can provide a modest complement to the weaker DINOv2 representation on PatternNet, but becomes redundant once CLIP is included and can become detrimental under the tested 32-bit compression setting. The result therefore concerns the incremental value of texture in combination with foundation-model representations rather than the general usefulness of handcrafted texture descriptors.

#### *4.1.4. Class-Wise Behavior and Retrieval Traces*

While the aggregate metrics demonstrate the overall effectiveness of DINOv2–CLIP fusion, class-level analysis provides additional insight into where the representation succeeds and where systematic retrieval errors remain. Table 8 summarizes the class-wise performance range and representative seed-0 retrieval traces for both datasets.

**Table 8. Class-wise DINOv2–CLIP performance and representative seed-0 retrieval traces.**

| Dataset / Case | Summary or Top-10 Retrieved Labels |
|---|---|
| PatternNet class range | Bridge: 0.5580 ± 0.0094 mAP; median class: 0.8544; airplane: 0.9958 ± 0.0022 |
| PatternNet weak classes | Bridge, ferry terminal, parking space, dense residential, sparse residential |
| Bridge query | golf course; golf course; golf course; bridge; golf course; golf course; golf course; ferry terminal; golf course; golf course (1/10) |
| Parking-space query | parking space; parking space; runway marking; runway marking; parking space; parking space; runway marking; parking space; runway marking; parking space (6/10) |
| NWPU-RESISC45 class range | Palace: 0.2911 ± 0.0152 mAP; median class: 0.5423; sea ice: 0.9431 ± 0.0055 |
| NWPU-RESISC45 weak classes | Palace, freeway, church, intersection, railway station |
| Palace query | railway station; railway station; dense residential ×5; bridge; dense residential ×2 (0/10) |

The class-wise range is substantial, particularly on NWPU-RESISC45. Some of the observed errors are interpretable: bridges, ferry terminals, and palaces can share elongated structures, water context, or dense built environments with visually competing classes, whereas airplanes and sea ice are more visually distinctive. These differences indicate that strong aggregate retrieval performance does not necessarily translate into uniform performance across all scene categories.

The retrieval traces also provide concrete examples of this behavior. For the bridge query, only one of the top ten retrieved labels belongs to the correct class, whereas the sea-ice query retrieves the correct class in all ten positions. The parking-space query performs better but still shows confusion with runway marking. These examples illustrate how visually similar scene structures can produce systematic class-level confusion despite strong overall mAP.

The label traces are provided as an auditable summary because source image pixels are not embedded in the manuscript package. The supplied notebook can render the corresponding queries and gallery images when the public datasets are mounted. Thus, the class-wise analysis complements the aggregate metrics by identifying both the strengths and failure modes of the fused representation.

### 4.2. RQ2: Binary Encoding and the Role of Rotation

RQ2 examines how different binary encoding regimes trade retrieval quality against code length when applied to the same fused representation. The comparison includes da-

ta-independent LSH, unsupervised PCA-sign and ITQ, and a supervised CSQ-style head. Table 9 summarizes the retrieval performance across code lengths, while Figure 4 highlights the corresponding performance trends.

**Table 9.** Retrieval mAP for full-precision and binary representations (mean ± sample SD over five seeds).

| Regime / Method | Bits | PatternNet | NWPU-RESISC45 |
|---|---|---|---|
| **Float fused** | 40,960* | **0.8143 ± 0.0007** | **0.5693 ± 0.0020** |
| LSH | 16 | 0.1640 ± 0.0159 | 0.0915 ± 0.0098 |
| | 32 | 0.2661 ± 0.0257 | 0.1408 ± 0.0144 |
| | 64 | 0.4095 ± 0.0134 | 0.2348 ± 0.0078 |
| | 128 | 0.5741 ± 0.0169 | 0.3521 ± 0.0159 |
| PCA-sign | 16 | 0.5537 ± 0.0051 | 0.3247 ± 0.0029 |
| | 32 | 0.6856 ± 0.0022 | 0.4102 ± 0.0032 |
| | 64 | 0.5995 ± 0.0050 | 0.4102 ± 0.0021 |
| | 128 | 0.4283 ± 0.0036 | 0.3220 ± 0.0009 |
| ITQ | 16 | 0.6894 ± 0.0072 | 0.4042 ± 0.0083 |
| | 32 | 0.8206 ± 0.0058 | 0.5416 ± 0.0014 |
| | 64 | 0.8676 ± 0.0030 | 0.6137 ± 0.0035 |
| | 128 | 0.8765 ± 0.0022 | 0.6366 ± 0.0009 |
| CSQ-style† | 16 | 0.9808 ± 0.0019 | 0.8421 ± 0.0059 |
| | 32 | 0.9867 ± 0.0013 | 0.8882 ± 0.0024 |
| | 64 | 0.9886 ± 0.0014 | 0.9087 ± 0.0026 |

* The float representation contains 1,280 float32 coordinates, equivalent to 40,960 stored bits.
† The CSQ-style head uses gallery labels and is therefore not directly comparable to the label-free methods.

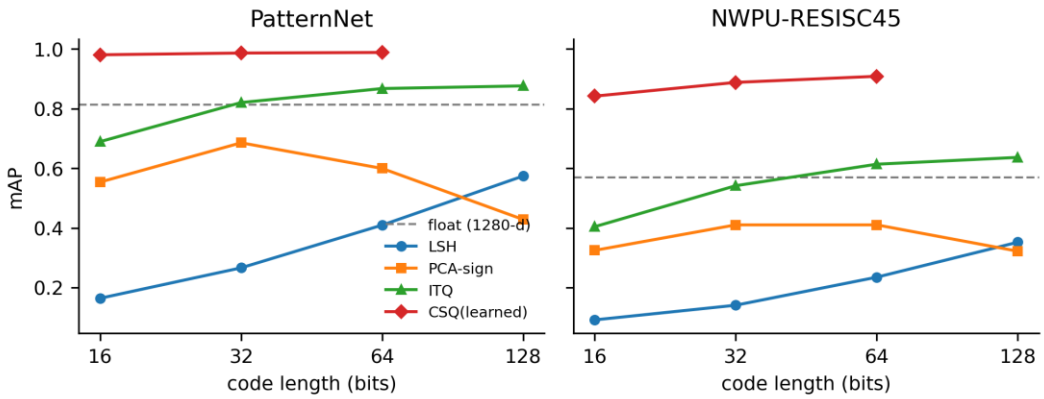


**Figure 4. Retrieval mAP as a function of binary code length.**

ITQ provides the strongest label-free binary performance across all tested code lengths. At 64 bits, ITQ reaches 0.8676 mAP on PatternNet and 0.6137 on NWPU-RESISC45, exceeding the full-precision fused baseline by 0.0533 and 0.0445, respectively. At 128 bits, the corresponding gains increase to 0.0622 and 0.0674. Thus, in this controlled setting, increasing the ITQ code length does not merely recover the performance lost through binarization; it produces a higher retrieval score than the original full-precision fused representation.

The behavior of the other label-free methods is substantially different. As shown in Figure 4, LSH improves gradually with code length but remains below the fused float baseline even at 128 bits. PCA-sign performs better at short codes, reaching its highest mAP at 32 bits, but its performance decreases at 64 and 128 bits. In contrast, ITQ improves consistently from 16 to 128 bits on both datasets. This pattern indicates that retaining leading PCA variance alone is not sufficient for robust binary retrieval at longer code lengths; the additional or-

thogonal rotation learned by ITQ provides an important advantage. Table 10 provides the paired statistical comparison between ITQ and the fused full-precision baseline.

**Table 10.** Paired validation of ITQ against the fused full-precision baseline.

| Contrast | Dataset | Mean ΔmAP | 95% CI | Paired (t) (p) | Exact sign-flip (p) |
|---|---|---|---|---|---|
| ITQ-64 − Float | PatternNet | +0.0533 | [0.0497, 0.0569] | $2.10 \times 10^{-6}$ | 0.0625 |
| ITQ-64 − Float | NWPU-RESISC45 | +0.0445 | [0.0412, 0.0477] | $3.00 \times 10^{-6}$ | 0.0625 |
| ITQ-128 − Float | PatternNet | +0.0622 | [0.0601, 0.0643] | $1.32 \times 10^{-7}$ | 0.0625 |
| ITQ-128 − Float | NWPU-RESISC45 | +0.0674 | [0.0649, 0.0698] | $1.88 \times 10^{-7}$ | 0.0625 |

ITQ exceeds the float baseline at both 64 and 128 bits for every observed seed. The paired t-tests indicate large and consistent mean differences, while the exact sign-flip test remains resolution-limited at p=0.0625 because only five paired seeds are available. The PCA-sign control is particularly informative: both PCA-sign and ITQ start from the same top-b PCA coordinates, but their performance trends diverge substantially. This supports the interpretation that ITQ's learned orthogonal rotation, rather than dimensionality reduction alone, is important for reducing sign-quantization error in this setting. However, the experiment does not isolate a causal "denoising" mechanism, nor does it establish that binarization universally improves retrieval. The result is more narrowly interpreted as a consistent ranking improvement for the specific fused representation, datasets, code lengths, and evaluation protocol used here.

The supervised CSQ-style head produces substantially higher mAP, reaching 0.9886 on PatternNet and 0.9087 on NWPU-RESISC45 at 64 bits. As indicated in Table 9, however, this result belongs to a different information regime because gallery labels are used during training. Its performance therefore demonstrates the potential of label-shaped binary representations rather than the intrinsic benefit of compression. It should not be interpreted as evidence that supervised hashing is inherently superior to label-free hashing. For label-free deployment, 64-bit ITQ provides the most attractive quality–storage trade-off among the tested operating points, while 128-bit ITQ achieves the highest retrieval quality at the cost of doubling the code length. This distinction is practically relevant because the appropriate code length depends on whether retrieval quality or compact storage is the primary constraint.

### 4.3. RQ3: Controlled 200,000-Vector Efficiency

RQ3 evaluates the storage and flat-search efficiency of the fused representation under a controlled gallery size. The analysis focuses on the cost of storing and searching full-precision versus binary representations, while keeping the retrieval setting fixed. Latency is measured using FAISS CPU IndexFlatIP for full-precision vectors and IndexBinaryFlat for packed binary codes. The native PatternNet seed-0 gallery contains 18,240 vectors. To evaluate a fixed larger gallery size without introducing unverified imagery, the descriptors are repeated in deterministic gallery order and truncated to exactly 200,000 entries. The first 200 test queries are used for timing. This setup measures representation storage and flat-scan cost under controlled repetition; it does not evaluate image diversity, approximate-nearest-neighbor indexes, ingestion, feature extraction, or a real 200,000-image archive. Table 11 summarizes the measured storage requirements and query latency for the native and 200,000-vector settings.

**Table 11.** Flat-search storage and query latency. Latency is environment-specific.

| Gallery / Representation | Bytes/vector | Index memory | ms/query | Speed-up vs. float |
|---|---|---|---|---|
| 18,240 / float-1280 | 5,120 | 93.4 MB | 0.469 | 1.0× |
| 18,240 / binary-32 | 4 | 0.073 MB | 0.065 | 7.2× |
| 18,240 / binary-64 | 8 | 0.146 MB | 0.095 | 4.9× |
| 18,240 / binary-128 | 16 | 0.292 MB | 0.105 | 4.4× |
| 200,000 / tiled / float-1280 | 5,120 | 1.02 GB | 5.343 | 1.0× |
| 200,000 / tiled / binary-64 | 8 | 1.60 MB | 0.674 | 7.9× |

Figure 5 illustrates the corresponding latency and memory differences between full-precision and binary representations.

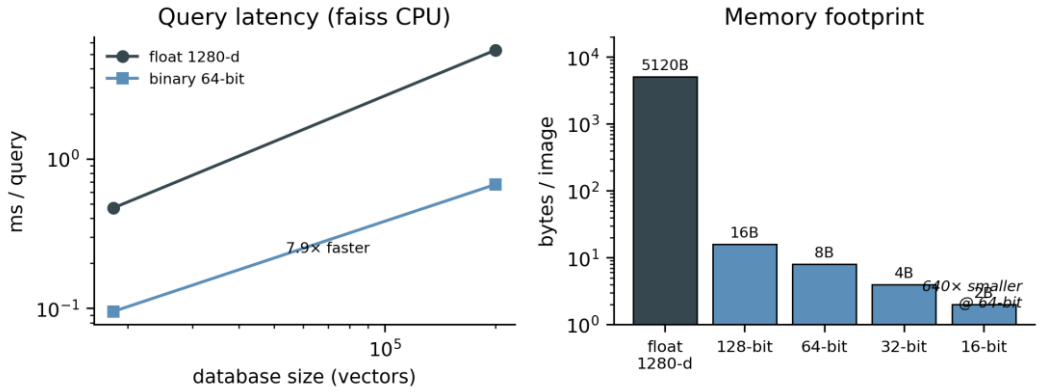


**Figure 5.** Flat-search latency and memory footprint for the tested representations.

At 200,000 vectors, the 64-bit representation requires only 1.60 MB compared with 1.02 GB for the 1,280-dimensional float representation. This corresponds to a 640× reduction in storage per gallery representation. The measured query latency also decreases from 5.343 ms to 0.674 ms, corresponding to a 7.9× speed-up under the tested FAISS CPU configuration.

The efficiency benefit therefore extends beyond storage compression: the shorter binary representation also reduces the measured cost of flat Hamming search. However, the magnitude of the latency improvement is hardware- and implementation-dependent. The CPU model was not recorded, and only one runtime environment was evaluated; consequently, these latency values should be interpreted as an illustrative system measurement rather than a hardware-independent benchmark.

The result also has an important scope limitation. The 200,000-vector gallery is constructed by deterministic repetition of the native PatternNet gallery rather than by adding 200,000 distinct remote-sensing images. Therefore, the experiment demonstrates the representation and flat-search cost at the tested vector count, but it does not establish performance on a genuinely diverse 200,000-image archive or on million-scale operational archives. Within these limits, RQ3 is supported for the evaluated flat-search setting.

### 4.4. Contextual Comparison with Prior Work

Table 12 provides numerical context by comparing the present results with representative published studies on PatternNet and NWPU-RESISC45. Because the studies use different data splits, training protocols, relevance definitions, and evaluation metrics, these values should not be interpreted as a controlled leaderboard.

**Table 12.** Published numerical context on the evaluated remote-sensing benchmarks. Metrics and protocols differ across studies and are not directly comparable.

| Method | Training / Protocol | PatternNet | NWPU-RESISC45 |
|---|---|---|---|
| DCL [33] | Fine-tuned ResNet50; multi-scale/SPoC; paper-specific split | mAP 99.43% | mAP 99.44% |
| Proxy-Anchor + generation [34] | Supervised metric learning; 10% train / 90% test; top-10 metrics | R@1 99.10%; mAP@10 98.10% | R@1 95.75%; mAP@10 94.45% |
| This study: frozen fusion | No backbone tuning; 60/20/20 split; full-ranking mAP | mAP 81.43% | mAP 56.93% |
| This study: CSQ-style 64-bit | Gallery labels; same 60/20/20 split; full-ranking mAP | mAP 98.86% | mAP 90.87% |

The supervised fine-tuning and metric-learning approaches reported in DCL [33] and Proxy-Anchor + generation [34] achieve higher numerical results under their respective experimental settings. However, these values are obtained using different splits, training access,

and evaluation protocols. They therefore provide useful numerical context but do not establish superiority or inferiority relative to the present study.

The comparison also reinforces the distinction between the study's label-free frozen-feature setting and its supervised CSQ-style setting. The latter approaches the performance reported by supervised methods on PatternNet but remains lower on NWPU-RESISC45. The main contribution of the present study is therefore not a claim of state-of-the-art supervised retrieval accuracy, but a controlled analysis of frozen foundation-model fusion, texture contribution, binary hashing, and efficiency under a consistent evaluation protocol. A rigorous leaderboard comparison would require re-evaluating competing methods under identical splits, relevance definitions, and full-ranking metrics.

### 4.5. Practical Recommendations

The findings provide several practical considerations for CBRSIR deployment under the evaluated setting.

- When labels and fine-tuning resources are unavailable, a frozen CLIP representation provides a strong starting point, while combining it with DINOv2 using the tested 0.3/0.7 score weighting can further improve retrieval. The fusion weights should be recalibrated when the target domain or data distribution changes.
- Handcrafted texture should not be assumed to be universally ineffective. Its contribution should be evaluated against the strongest foundation-model representation available. In the present experiments, texture provided a modest benefit when combined with DINOv2 alone on PatternNet but did not provide additional benefit once CLIP was included.
- For label-free compact retrieval, 64-bit ITQ provides a practical operating point among the tested configurations, offering higher retrieval quality than the fused full-precision baseline while substantially reducing representation size. A 128-bit code can be considered when the additional storage cost is acceptable and retrieval quality is prioritized.
- Supervised hashing should be reported separately from label-free methods. Its higher retrieval performance reflects the use of class labels during training and therefore cannot be attributed to binary representation alone.
- Flat-search latency should be treated as a deployment-specific baseline. Before operational deployment, the target hardware, gallery diversity, and indexing strategy should be evaluated because the reported measurements were obtained under a single controlled FAISS CPU configuration.

These recommendations are intended as practical guidance from the tested configurations rather than universal prescriptions. Their applicability should be validated when the dataset, backbone, or deployment environment differs from those evaluated here.

### 4.6. Limitations

Several limitations should be considered when interpreting the findings. First, the study uses two balanced RGB scene-classification datasets, so the conclusions may not directly transfer to class-imbalanced, multi-label, multispectral, temporal, or cross-sensor archives. Second, DINOv2 and CLIP are used as frozen general-purpose backbones; domain-specific encoders such as RemoteCLIP or fine-tuned models may produce different complementarity patterns and retrieval performance. The evaluation also uses a global class-equality relevance definition and therefore does not capture graded semantic similarity.

The statistical analysis is based on five seeds, which provides repeated estimates but limits the resolution of exact nonparametric testing. The precision/recall and class-wise analyses use cached float16-extracted features, while the exact OpenCLIP package version and CPU model were not recorded. These implementation details constrain complete reproducibility of the reported runtime environment.

The efficiency evaluation is also limited to flat cosine and Hamming indexes. The 200,000-vector gallery is constructed by deterministic repetition of PatternNet descriptors and therefore evaluates representation storage and scan cost rather than retrieval behavior on a naturally diverse large archive. No million-image experiment is performed. Finally, the CSQ-style head uses gallery labels and therefore represents a supervised information regime that should not be interpreted as a label-free compression result.

## 5. Conclusions

This study provides a controlled evaluation of frozen foundation-model representations and binary hashing for CBRSIR. The results show that combining DINOv2 and CLIP can provide complementary retrieval information, while the contribution of handcrafted texture depends on the representation and dataset. Among the evaluated label-free hashing methods, ITQ provides the strongest retrieval performance and remains effective at compact code lengths, whereas the supervised CSQ-style head represents a separate label-informed regime. The efficiency experiment further shows that binary representations can substantially reduce storage and flat-search cost under the tested 200,000-vector setting. Overall, the findings support frozen DINOv2–CLIP fusion with compact ITQ codes as a practical configuration when labels and backbone fine-tuning are unavailable, while emphasizing that fusion weights, code length, and indexing choices should be validated for the target dataset and deployment environment.

**Author Contributions:** Conceptualization: J.; T.M.; Methodology: J.; Software: J.; Validation: J.; T.M.; A.Z.; S.; Formal analysis: J.; Investigation: J.; Resources: T.M.; Data curation: J.; Writing—original draft preparation: J.; Writing—review and editing: T.M.; A.Z.; Visualization: J.; Supervision: A.Z.; Project administration: A.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** PatternNet is publicly available from the dataset authors' official page at https://sites.google.com/view/zhouwx/dataset. NWPU-RESISC45 is publicly available from the official Northwestern Polytechnical University dataset page at https://gcheng-nwpu.github.io/#Datasets, with the official download link provided there. The Supplementary Research Material accompanying this article contains the executable analysis notebook, configuration files, seed-level result CSV files, high-resolution manuscript figures, and SHA-256 hashes for the canonical cached arrays. A public copy of the source notebook is available at https://colab.research.google.com/drive/1xfEFARWRpwX6A1OVBzXVraoEGkCqhz_E. The notebook implements the stratified data partitioning, frozen DINOv2 and CLIP feature extraction, calibrated fusion, binary-hashing evaluation, statistical analyses, class-wise analyses, and the controlled efficiency benchmark reported in this study. The raw cached embeddings are not duplicated in the supplementary material because of their size; they can be regenerated using the supplied extraction code and verified against the provided SHA-256 hashes.

**Acknowledgments:** The authors acknowledge the dataset providers, the open-source software communities supporting computer vision and information retrieval research, and the use of AI-assisted tools for language editing and manuscript preparation.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

[1] B. Demir and L. Bruzzone, "Hashing-Based Scalable Remote Sensing Image Search and Retrieval in Large Archives," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 892–904, Feb. 2016, doi: 10.1109/TGRS.2015.2469138.

[2] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, "Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 3735–3756, 2020, doi: 10.1109/JSTARS.2020.3005403.

[3] W. Zhou, S. Newsam, C. Li, and Z. Shao, "PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 197–209, Nov. 2018, doi: 10.1016/j.isprsjprs.2018.01.004.

[4] G. Cheng, J. Han, and X. Lu, "Remote Sensing Image Scene Classification: Benchmark and State of the Art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017, doi: 10.1109/JPROC.2017.2675998.

[5] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural Codes for Image Retrieval," in *Lecture Notes in Computer Science*, 2014, pp. 584–599. doi: 10.1007/978-3-319-10590-1_38.

[6] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep Image Retrieval: Learning Global Representations for Image Search," in *Lecture Notes in Computer Science*, 2016, pp. 241–257. doi: 10.1007/978-3-319-46466-4_15.

[7] F. Radenovic, G. Tolias, and O. Chum, "Fine-Tuning CNN Image Retrieval with No Human Annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, Jul. 2019, doi: 10.1109/TPAMI.2018.2846566.

[8] M. Oquab *et al.*, "DINOv2: Learning Robust Visual Features without Supervision," *Trans. Mach. Learn. Res.*, Feb. 2024, doi: 10.48550/arxiv.2304.07193.
[9] A. Radford *et al.*, "Learning Transferable Visual Models from Natural Language Supervision," in *Proceedings of the International Conference on Machine Learning*, 2021, vol. 139, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html
[10] M. Caron *et al.*, "Emerging Properties in Self-Supervised Vision Transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 9630–9640. doi: 10.1109/ICCV48922.2021.00951.
[11] M. Cherti *et al.*, "Reproducible Scaling Laws for Contrastive Language-Image Learning," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 2818–2829. doi: 10.1109/CVPR52729.2023.00276.
[12] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *ICLR 2021*, Jun. 2021. [Online]. Available: http://arxiv.org/abs/2010.11929
[13] A. Vaswani *et al.*, "Attention Is All You Need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762
[14] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 15979–15988. doi: 10.1109/CVPR52688.2022.01553.
[15] X. Tan, Y. Zou, Z. Guo, K. Zhou, and Q. Yuan, "Deep Contrastive Self-Supervised Hashing for Remote Sensing Image Retrieval," *Remote Sens.*, vol. 14, no. 15, p. 3643, Jul. 2022, doi: 10.3390/rs14153643.
[16] Y. Sun *et al.*, "Unsupervised deep hashing through learning soft pseudo label for remote sensing image retrieval," *Knowledge-Based Syst.*, vol. 239, p. 107807, Mar. 2022, doi: 10.1016/j.knosys.2021.107807.
[17] Y. Zhang, X. Zheng, and X. Lu, "Remote Sensing Image Retrieval by Deep Attention Hashing With Distance-Adaptive Ranking," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 16, pp. 4301–4311, 2023, doi: 10.1109/JSTARS.2023.3271303.
[18] F. Liu *et al.*, "RemoteCLIP: A Vision Language Foundation Model for Remote Sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–16, 2024, doi: 10.1109/TGRS.2024.3390838.
[19] Y. Cong *et al.*, "SatMAE: Pre-training Transformers for Temporal and Multi-spectral Satellite Imagery," in *Advances in Neural Information Processing Systems*, 2022, vol. 35, pp. 197–211. [Online]. Available: https://doi.org/10.52202/068431-0015
[20] M. Mendieta, B. Han, X. Shi, Y. Zhu, and C. Chen, "Towards Geospatial Foundation Models via Continual Pretraining," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 16760–16770. doi: 10.1109/ICCV51070.2023.01541.
[21] J. Johnson, M. Douze, and H. Jegou, "Billion-Scale Similarity Search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021, doi: 10.1109/TBDATA.2019.2921572.
[22] Y. A. Malkov and D. A. Yashunin, "Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, Apr. 2020, doi: 10.1109/TPAMI.2018.2889473.
[23] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013, doi: 10.1109/TPAMI.2012.193.
[24] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A Survey on Learning to Hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, Apr. 2018, doi: 10.1109/TPAMI.2017.2699960.
[25] X. Luo *et al.*, "A Survey on Deep Hashing Methods," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 1, pp. 1–50, Feb. 2023, doi: 10.1145/3532624.
[26] A. Gionis, P. Indyk, and R. Motwani, "Similarity Search in High Dimensions via Hashing," in *Proceedings of the International Conference on Very Large Data Bases*, 1999, pp. 518–529. [Online]. Available: https://www.vldb.org/conf/1999/P49.pdf
[27] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, May 2002, pp. 380–388. doi: 10.1145/509907.509965.
[28] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-Scale Image Retrieval with Attentive Deep Local Features," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 3476–3485. doi: 10.1109/ICCV.2017.374.
[29] Y. Li, Y. Zhang, X. Huang, H. Zhu, and J. Ma, "Large-Scale Remote Sensing Image Retrieval by Deep Hashing Neural Networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 950–965, Feb. 2018, doi: 10.1109/TGRS.2017.2756911.
[30] S. Roy, E. Sangineto, B. Demir, and N. Sebe, "Metric-Learning-Based Deep Hashing Network for Content-Based Retrieval of Remote Sensing Images," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 2, pp. 226–230, Feb. 2021, doi: 10.1109/LGRS.2020.2974629.
[31] L. Yuan *et al.*, "Central Similarity Quantization for Efficient Image and Video Retrieval," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 3080–3089. doi: 10.1109/CVPR42600.2020.00315.
[32] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. doi: 10.1017/CBO9780511809071.
[33] L. Fan, H. Zhao, and H. Zhao, "Distribution Consistency Loss for Large-Scale Remote Sensing Image Retrieval," *Remote Sens.*, vol. 12, no. 1, p. 175, Jan. 2020, doi: 10.3390/rs12010175.
[34] P. Liu, X. Liu, Y. Wang, Z. Liu, Q. Zhou, and Q. Li, "An Intra-Class Ranking Metric for Remote Sensing Image Retrieval," *Remote Sens.*, vol. 15, no. 16, p. 3943, Aug. 2023, doi: 10.3390/rs15163943.