Cross-Detector Descriptor Fusion: Scale Control and Spatial Alignment for Local Feature Matching

Frank Sossi

A Thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Computer Science & Software Engineering

University of Washington

2026

Reading Committee:

Professor Clark Olson, Chair

Professor Min Chen

Professor Dong Si

Program Authorized to Offer Degree:

Computer Science & Software Engineering

University of Washington

Abstract

Cross-Detector Descriptor Fusion:
Scale Control and Spatial Alignment for Local Feature Matching

Frank Sossi

Chair of the Supervisory Committee:
Committee Chair Professor Clark Olson
Computing & Software Systems

Local feature descriptors are fundamental to many computer vision applications including SLAM, structure from motion, and image retrieval. This thesis evaluates two approaches to improving local feature matching: using multiple detectors as a quality filter for keypoint selection, and fusing complementary descriptors to combine their strengths.

We show that spatial intersection between different keypoint detectors acts as a quality filter. When different detection methods, whether SIFT and SURF or SIFT and KeyNet, both identify a keypoint at the same location, this consensus indicates a distinctive feature. Descriptors computed at intersection keypoints consistently outperform those on single detector sets, with HardNet achieving 82.1% mAP on SIFT-KeyNet intersection, a 25% relative improvement and the best single descriptor result in our study.

In order to evaluate color descriptors we re-implemented a color version of the HPatches patch benchmark, allowing us to evaluate color aware descriptors. Using this dataset, we show that fusing the color histogram descriptor HoNC with learned CNN descriptors yields substantial improvements: HoNC+SOSNet concatenation achieves 50.6% mAP on patch matching, outperforming all individual descriptors. HoNC's strong discriminative capability (high verification to matching ratio) complements the CNN's matching optimized representations.

Cross family fusion (SIFT+CNN) requires pre-fusion L2 normalization to ensure equal contribution from each descriptor; with proper normalization, SIFT+HardNet achieves 46.0% mAP on patches. Keypoint scale is also a dominant factor: filtering to large scale keypoints yields 39% relative improvement for SIFT and 21% for CNN descriptors.

We develop DescriptorWorkbench, an open source evaluation framework, and conduct over 100 experiments. The results show that keypoint quality determined by detector consensus and scale has greater impact on matching performance than descriptor algorithm choice alone.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank Professor Clark Olson for his guidance and support throughout this research.

Chapter 1
INTRODUCTION

Local feature or keypoint detection and description remain fundamental to many computer vision applications, including simultaneous localization and mapping (SLAM), structure from motion (SfM), image retrieval, and visual place recognition. Despite advances in end to end learned approaches, local feature methods remain competitive due to their interpretability, efficiency, and ability to handle wide baseline matching.

1.1 Motivation

The computer vision community has developed several approaches to local feature extraction, which comprises two stages. *Detection* identifies salient image locations (corners, blobs, or edge junctions) that are likely to be found again under changes in viewpoint, scale, or illumination. *Description* encodes the local image appearance around each detected keypoint as a fixed-length vector, enabling efficient comparison across images. The quality of both stages determines overall matching performance, though they are often studied and optimized independently.

Traditional methods such as SIFT [21] and SURF [8] use hand-crafted detectors based on scale-space analysis and descriptors based on gradient histograms. These methods are well understood and deterministic, making them reliable baselines for comparison. Learned methods such as KeyNet [6] for detection and HardNet [26] and SOSNet [41] for description use convolutional neural networks trained on correspondence tasks. While these achieve higher matching accuracy, they are typically trained for specific detector-descriptor pairings and may not generalize across different detection methods.

A third family of descriptors captures information that gradient-based methods discard.

Color descriptors such as HoNC (Histogram of Normalized Colors) [29] encode chromatic content rather than edge orientations. These descriptors excel at discrimination rejecting false matches between patches with different colors but show a larger performance gap between viewpoint and illumination sequences than learned descriptors in our experiments (Section 4.7), consistent with the dependence of color histograms on photometric stability.

Each approach has distinct strengths, raising a practical question: can we combine these complementary capabilities without compromising robustness?

During development, we observed that *detector agreement provides a quality signal*. When two distinct detection methods whether SIFT and SURF (both gradient based but with different scale space representations) or SIFT and KeyNet (traditional and learned) both identify a keypoint at the same image location, this agreement indicated a distinctive feature. Different detectors select points based on different criteria; when they agree, the underlying image structure is distinctive across multiple representations. We hypothesize that such consensus keypoints are more repeatable and discriminative than those found by any single detector alone.

Our second set of experiments determined that *complementary descriptors can be fused* when their characteristics are compatible. Color descriptors like HoNC provide strong discriminative power (rejecting false matches), while learned descriptors like HardNet and SOSNet provide robust matching under geometric and photometric changes. Combining a "discriminator" with a "matcher" can outperform either alone. However, not all fusion strategies succeed. For example, fusing SIFT with CNN descriptors requires careful normalization to ensure equal contribution; without it, the fusion can underperform the individual descriptors.

1.2 Research Questions

This thesis investigates the following research questions:

1. **RQ1: Detector Consensus.** Does spatial agreement between different keypoint detectors (SIFT and KeyNet) provide a quality signal? Are consensus keypoints more

distinctive and repeatable?

2. **RQ2: Color Descriptor Fusion.** Can color descriptors (HoNC) be successfully fused with learned descriptors (HardNet, SOSNet)? What complementary information does color provide?

3. **RQ3: Fusion Compatibility.** What determines whether descriptor fusion succeeds or fails? Are there systematic patterns based on descriptor characteristics?

4. **RQ4: Scale Impact.** How does keypoint scale distribution affect descriptor matching performance? Is keypoint quality more important than descriptor algorithm choice?

1.3 Contributions

This thesis makes the following contributions:

1. **Detector Intersection as Quality Filter.** Keypoints detected by multiple detectors are more distinctive than those found by a single detector. HardNet computed at SIFT KeyNet intersection keypoints achieves 82.1% mAP, a 25% relative improvement over the full keypoint set and the best single descriptor result in our study. This supports the claim that detector consensus identifies high quality features.

2. **Color HPatches Benchmark.** We create a color version of the HPatches patch benchmark by re-extracting 65×65 color patches from the original images using stored keypoint locations and homographies. This enables evaluation of color aware descriptors that cannot be tested on the original gray scale patches.

3. **Complementary Descriptor Fusion.** Fusing color descriptors with learned descriptors yields substantial improvements. HoNC+SOSNet concatenation achieves 50.6% mAP on the color patch benchmark, outperforming all individual descriptors. HoNC

acts as a "discriminator" (high verification to matching ratio of 3.84×) that complements CNN "matchers" (ratio of 1.84×).

4. **Magnitude Matching for Cross Family Fusion.** Cross family fusion (SIFT+CNN) requires pre fusion L2 normalization to ensure equal contribution from each descriptor. With proper normalization, SIFT+HardNet achieves 46.0% mAP on patches.

5. **Scale Control Methodology.** Filtering keypoints by scale improves matching performance: +39% relative for SIFT family descriptors and +21% relative for CNN descriptors. Keypoint quality is as important as descriptor algorithm choice.

6. **DescriptorWorkbench Framework.** We develop an open source evaluation framework implementing the image matching, keypoint verification, and keypoint retrieval metrics from Bojanic et al. [7], supporting both full image and patch based evaluation.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

- **Chapter 2** reviews local feature detection and description literature.

- **Chapter 3** describes our detector intersection methodology, color patch dataset construction, and evaluation framework.

- **Chapter 4** presents experimental results from over 100 experiments on the HPatches benchmark, analyzing detector consensus effects, color descriptor fusion, and scale control.

- **Chapter 5** interprets the results, explaining why detector consensus provides quality signal and why certain descriptor fusion strategies succeed.

- **Chapter 6** summarizes findings and discusses future work directions.

Chapter 2
BACKGROUND AND RELATED WORK

This chapter reviews local feature detection and description, covering traditional and learned approaches, descriptor fusion methods, and evaluation benchmarks.

2.1 Local Feature Detection

2.1.1 Traditional Detectors

The SIFT (Scale-Invariant Feature Transform) detector [21] identifies keypoints by searching for scale-space extrema in a Difference-of-Gaussian (DoG) pyramid. Key characteristics include:

- **Scale invariance**: Keypoints are detected across multiple octaves, with scale encoded in the keypoint metadata
- **Orientation assignment**: Dominant gradient orientation is computed for rotation invariance
- **Sub-pixel localization**: Taylor expansion refines keypoint position

Other traditional detectors include Harris corners [15], FAST [35], and SURF [8], each with different trade-offs between repeatability and computational cost [17].

2.1.2 Learned Detectors

KeyNet [6] represents a hybrid approach combining hand-crafted and learned filters:

- **Handcrafted filters**: Provide anchor structure for stable detection

- **Learned filters**: Trained to localize, score, and rank repeatable features
- **Multi-scale**: Operates on image pyramids similar to SIFT

KeyNet was designed specifically for pairing with learned descriptors like HardNet and SOSNet, achieving state-of-the-art repeatability on HPatches [6].

2.1.3 End-to-End Learned Pipelines

While KeyNet focuses on detection to be paired with separate descriptors, other approaches integrate detection and description into a single end-to-end trainable network:

- **SuperPoint** [11]: A fully convolutional network trained using self-supervision. It employs a single shared encoder and two separate decoder heads for interest point detection and descriptor generation.
- **LF-Net** [30]: Learns local features without human supervision by optimizing a keypoint correspondence objective, enforcing geometry constraints across multi-view images.
- **ALIKE** [46]: A lightweight framework that blends handcrafted and learned methods, using a deformable transformation module to improve geometric invariance while maintaining high computational efficiency.

These end-to-end methods provide an alternative to the detect-then-describe pipeline, but they often require strictly coupled detector-descriptor pairs, unlike the mix-and-match flexibility we investigate with KeyNet and SIFT.

2.1.4 Detector Characteristics

An observation central to this thesis is that different detectors produce keypoints with different scale distributions:

- **SIFT detector**: Produces predominantly small-scale keypoints (average 4.45 pixels in our experiments)
- **KeyNet detector**: Produces larger-scale keypoints (average 49.83 pixels), approximately 10× larger

This scale difference directly affects descriptor quality, since larger patches contain more distinctive information.

2.2 Local Feature Descriptors

2.2.1 Traditional Descriptors

SIFT Descriptor

The SIFT descriptor [21] computes a 128-dimensional vector from a 16×16 pixel patch:

1. Divide patch into 4×4 grid of cells
2. Compute 8-bin gradient orientation histogram per cell
3. Concatenate to form $4 \times 4 \times 8 = 128$ dimensions
4. L2 normalize, clip values > 0.2, re-normalize

The resulting descriptor is *non-negative* with values typically in [0, 0.3].

Domain-Size Pooling (DSP-SIFT)

Dong and Soatto [12] introduced Domain-Size Pooling (DSP), which aggregates SIFT descriptors computed at multiple scales around each keypoint:

$$d_{\text{DSP}} = \frac{1}{N} \sum_{i=1}^{N} d_{\sigma_i} \tag{2.1}$$

where d_{σ_i} is the SIFT descriptor computed at scale σ_i. DSP improves matching accuracy by capturing multi-scale information.

RootSIFT

RootSIFT applies element-wise square root after L1 normalization, which is equivalent to using the Hellinger kernel [2]:

$$d_{\text{RootSIFT}} = \sqrt{d_{\text{SIFT}} / \|d_{\text{SIFT}}\|_1} \tag{2.2}$$

This transformation improves matching performance, particularly for illumination changes.

2.2.2 Learned Descriptors

Learned descriptors use convolutional neural networks trained on patch correspondence tasks. While they share similar architectures (typically L2Net-style [40] backbones producing 128-dimensional outputs), they differ in training data and loss functions, leading to complementary learned representations.

HardNet

HardNet [26] is trained on the Brown dataset (Liberty, Notre Dame, Yosemite sequences) using hard negative mining with triplet loss:

$$L = \max(0, m + d(a, p) - d(a, n^-)) \tag{2.3}$$

where (a, p) is a matching pair, n^- is the hardest negative in the batch (selected from all non-matching pairs), and m is the margin. The hard negative mining strategy forces the network to learn features that distinguish the most confusable patch pairs, producing highly discriminative descriptors. The resulting 128-dimensional descriptor is *zero-centered* with values typically in [-0.3, +0.3].

SOSNet

SOSNet (Second Order Similarity Network) [41] is also trained on the Brown dataset but incorporates second-order similarity constraints:

$$L_{\text{SOS}} = L_{\text{triplet}} + \lambda L_{\text{second-order}} \tag{2.4}$$

The second-order term encourages the network to preserve relative similarity structure: if patch A is more similar to B than to C in the input space, this relationship should hold in the descriptor space. This regularization improves generalization to unseen data. Despite using the same training images as HardNet, the different loss function produces complementary features, explaining why HardNet+SOSNet fusion (93.4% mAP) substantially outperforms either descriptor alone (82.1% mAP).

2.2.3 Color Descriptors

Color descriptors capture chromatic information that gradient-based methods discard. While gradient descriptors like SIFT are designed to be illumination-invariant by ignoring absolute intensity, this discards potentially discriminative color information.

HoNC (Histogram of Normalized Colors)

HoNC [29] computes a 128-dimensional histogram of normalized RGB values within a patch:

1. Normalize each pixel's RGB values to unit sum: $(r, g, b) \rightarrow (r/(r+g+b), g/(r+g+b), b/(r+g+b))$
2. Quantize the normalized color space into bins
3. Accumulate weighted votes based on pixel distance from patch center
4. L2 normalize the final histogram

The normalization step provides some robustness to brightness changes while preserving hue information. HoNC excels at *discrimination*, rejecting false matches between patches with different colors, but struggles with *invariance* to illumination changes that alter apparent color (e.g., tungsten vs. daylight). This complementary strength makes HoNC valuable when fused with learned descriptors that provide robust matching.

2.2.4 Descriptor Families Summary

We categorize descriptors into three families based on the information they capture:

Table 2.1: Descriptor families and their characteristics

Family	**Examples**	**Information**	**Strengths**
Gradient-based	SIFT, DSP-SIFT, SURF	Edge orientations	Illumination invariant
Learned	HardNet, SOSNet	Trained features	High matching accuracy
Color	HoNC, RGBSIFT	Chromatic content	Strong discrimination

Fusion is most effective when combining descriptors from different families that capture complementary information. Within-family fusion (e.g., SIFT+RGBSIFT) provides minimal benefit because both descriptors encode similar gradient information. Cross-family fusion (e.g., HoNC+SOSNet) combines color discrimination with learned matching robustness, yielding the best patch-level results in our experiments.

2.2.5 Descriptor Magnitude and Normalization

A key factor in descriptor fusion is *magnitude matching*. Different descriptor families produce values at different scales before normalization:

When fusing descriptors from different families, **magnitude mismatch** can cause one descriptor to dominate distance calculations. For example, raw SIFT values (0–512) would overwhelm HardNet values (–0.3 to +0.3) in a concatenated descriptor. The solution is to L2

Table 2.2: Descriptor magnitude characteristics (before and after L2 normalization)

Descriptor	**Raw Range**	**After L2 Norm**	**Notes**
SIFT	[0, 512]	[0, 0.3]	Gradient histogram counts
RootSIFT	[0, 22]	[0, 0.4]	After sqrt transform
HoNC [29]	[0, 1]	[0, 0.3]	Normalized color histogram
HardNet	[-0.3, +0.3]	[-0.3, +0.3]	Trained with L2 output
SOSNet	[-0.3, +0.3]	[-0.3, +0.3]	Trained with L2 output

normalize each component *before* fusion, ensuring equal contribution regardless of original magnitude.

2.3 Descriptor Fusion Approaches

2.3.1 Early Fusion (Feature-Level)

Early fusion combines descriptors before matching:

Concatenation:

$$d_{\text{concat}} = [d_A, d_B] \tag{2.5}$$

Weighted Averaging:

$$d_{\text{avg}} = \alpha \cdot d_A + (1 - \alpha) \cdot d_B \tag{2.6}$$

Both approaches require spatial alignment of keypoints when descriptors come from different detectors.

2.3.2 Late Fusion (Score-Level)

Late fusion combines matching scores rather than descriptors:

$$s_{\text{fused}} = \alpha \cdot s_A + (1 - \alpha) \cdot s_B \tag{2.7}$$

This approach does not require keypoint alignment but cannot create a unified descriptor representation.

2.3.3 Research Gap

Prior work has explored:

- Detector-descriptor pairing studies showing mismatch penalties [25, 23]
- Late fusion of matching scores
- Ensemble methods in image matching challenges

However, *cross-detector early fusion*, averaging or concatenating descriptors from keypoints detected by different methods, remains largely unexplored. This thesis addresses this gap through our spatial intersection methodology.

2.4 Evaluation Benchmarks

2.4.1 HPatches Dataset

The HPatches benchmark [3] provides:

- 116 sequences with ground-truth homographies
- 59 viewpoint sequences (geometric changes)
- 57 illumination sequences (photometric changes)
- Pre-extracted 65×65 gray scale patches (original benchmark)
- Full images for keypoint-based evaluation

2.4.2 Two Evaluation Protocols

We employ two distinct evaluation protocols, each with three tasks. The protocols differ in whether keypoint detection is part of the evaluation.

Original HPatches Patch Protocol (Balntas et al.)

The original HPatches benchmark [3] evaluates descriptors on *pre-extracted patches*, removing keypoint detection as a variable:

Patch Matching: For each reference patch, rank all target patches from the same sequence by descriptor distance. Report Mean Average Precision (mAP), defined as the mean of per-query Average Precision (AP) scores. For a single query with one relevant item, $\text{AP} = 1/r$ where r is the rank of the correct match; mAP averages this over all queries, so a perfect system (every correct match ranked first) achieves $\text{mAP} = 1.0$.

Patch Verification: Binary classification of patch pairs as "same location" (positive) or "different location" (negative). Negatives include both same-sequence and different-sequence patches. Report AP.

Patch Retrieval: Given a query patch, rank a gallery containing true matches and distractors from different sequences. Report mAP.

We use this protocol for our **color patch benchmark** (Section 4.7) to isolate descriptor fusion effects.

Bojanic et al. Full-Image Protocol

Bojanic et al. [7] define an evaluation protocol for *full images with detected keypoints*:

Image Matching: Match descriptors between image pairs using the Second Nearest Neighbor (SNN) ratio test. A match is correct if the geometric reprojection error is below threshold. Report mAP.

Keypoint Verification: Binary classification distinguishing true correspondences from distractors sampled from *other sequences* (not same-sequence negatives).

Keypoint Retrieval: Three-tier ranking with labels $y \in \{-1, 0, +1\}$: true positives (+1), hard negatives from the same sequence (0, ignored in scoring), and distractors from other sequences (-1).

We use this protocol for our **full-image experiments** (Sections 4.2–4.4) to study detector effects.

Key Differences

Table 2.3: Comparison of evaluation protocols

Aspect	**Patch Protocol**	**Bojanic Protocol**
Input	Pre-extracted patches	Full images
Keypoint detection	Not evaluated	Part of pipeline
Verification negatives	Same + different sequence	Different sequence only
Retrieval labeling	Binary (pos/distractor)	Three-tier $(-1, 0, +1)$
Isolates	Descriptor quality	Detector + descriptor

We implement both protocols in DescriptorWorkbench to enable controlled experiments.

Chapter 3

METHODOLOGY

This chapter describes our methodology for cross-detector descriptor fusion, including the spatial intersection algorithm, scale-matching strategy, and evaluation framework.

3.1 DescriptorWorkbench Framework

All experiments were conducted on a workstation with the following specifications:

- **CPU**: Intel Core i9-9900K @ 3.60GHz (8 cores, 16 threads)
- **RAM**: 94 GB DDR4
- **GPU**: NVIDIA GeForce RTX 4090 (24 GB VRAM)
- **OS**: Manjaro Linux (Kernel 6.12)
- **CUDA**: 13.1
- **OpenCV**: 4.13.0
- **LibTorch**: 2.10.0

We developed DescriptorWorkbench,[1] an open-source evaluation framework for local feature descriptor research. It provides:

- **Modular architecture**: Pluggable extractors, pooling, and matchers

[1]Source code available at `https://github.com/F-Sossi/DescriptorWorkbench`

- **Database storage**: SQLite-based experiment tracking with comprehensive metrics
- **YAML configuration**: Declarative experiment specification
- **CLI tools**: `experiment_runner` for evaluation, `keypoint_manager` for keypoint set operations

3.1.1 Supported Descriptors

The framework implements the following descriptor types, using OpenCV for traditional methods and LibTorch for deep learning models:

Table 3.1: Descriptor implementations in DescriptorWorkbench

Type	**Family**	**Dim**	**Backend**
SIFT	Traditional	128	OpenCV
RootSIFT	Traditional	128	OpenCV + transform
DSP-SIFT v2	Traditional	128	Custom (pyramid-aware)
RGBSIFT	Traditional	384	Custom (per-channel)
SURF	Traditional	64/128	OpenCV
HoNC	Color	128	Custom
HardNet	Learned	128	LibTorch (C++)
SOSNet	Learned	128	LibTorch (C++)
L2-Net [40]	Learned	128	LibTorch (C++)
Composite	Fusion	varies	Custom

3.1.2 Database Schema

Experiment results are stored in SQLite with tables organized into two groups:

Full-Image Pipeline Tables:

- `experiments`: Descriptor type, dataset, pooling strategy, keypoint set reference, and execution parameters
- `results`: True mAP (macro/micro), HP-V/HP-I breakdown, verification AP, retrieval AP, and timing metrics
- `keypoint_sets`: Named keypoint collections with generation method, intersection provenance, and statistics
- `locked_keypoints`: Individual keypoint records with coordinates, scale, angle, response, and octave
- `keypoint_detector_attributes`: Per-detector attributes for intersection keypoints
- `descriptors`: Cached descriptor vectors with normalization and pooling metadata

Patch Benchmark Tables:

- `patch_benchmark_results`: mAP by difficulty (easy/hard/tough) and scene type (HP-V/HP-I), verification and retrieval metrics
- `patch_benchmark_task_sets`: Named evaluation task configurations
- `patch_benchmark_verification_pairs`: Positive and negative patch pairs for verification
- `patch_benchmark_retrieval_queries`: Query patches for retrieval evaluation
- `patch_benchmark_retrieval_distractors`: Distractor patches for retrieval
- `patch_benchmark_descriptor_sets`: Cached descriptor configurations with parameters
- `patch_benchmark_descriptors`: Pre-computed descriptor matrices stored as blobs

3.1.3 Two Evaluation Pipelines

We use two complementary evaluation pipelines to isolate different experimental variables:

Full-Image Pipeline

The full-image pipeline extracts descriptors from complete HPatches images at detected keypoint locations. This pipeline is used for:

- **Detector fusion experiments**: Evaluating how keypoint set selection (intersection, scale filtering) affects performance
- **Keypoint quality studies**: Comparing full keypoint sets vs. filtered subsets

In this pipeline, both keypoint quality and descriptor quality affect results, making it suitable for studying detector effects but not for pure descriptor comparisons.

Patch-Based Pipeline

The patch-based pipeline extracts descriptors from pre-extracted image patches (65×65 pixels). This pipeline is used for:

- **Descriptor fusion experiments**: Comparing descriptors and fusion strategies with keypoint quality held constant
- **Color descriptor evaluation**: Testing HoNC and other color-aware descriptors

By using identical patch locations for all descriptors, this pipeline isolates descriptor effects from keypoint quality effects.

Patch Pipeline Corrections

During development, we identified and corrected several inconsistencies in the patch benchmark pipeline that affected reported results:

1. Keypoint Size Correction. Traditional descriptors (SIFT, RootSIFT, RGBSIFT, HoNC) initially used a hardcoded keypoint size of 41.0 for 65×65 patches. This was too large at size 41, SIFT's sampling window extended beyond the patch boundary, producing degraded descriptors from border padding. The correct size is derived from the HPatches reference implementation:

$$\texttt{kp\_size} = \frac{65}{5.303} = 12.26 \tag{3.1}$$

where 5.303 maps OpenCV SIFT's keypoint size to a sampling region that fills the 65×65 patch. SURF uses a separately derived value of 23.21, computed from OpenCV's SURF window formula ($\texttt{size} \leq 65/2.8$).

2. Hidden Post-Fusion Normalization. The fusion extractor unconditionally L2-normalized all fused descriptors, an undocumented step that destroyed magnitude information from averaging and made pre-fusion normalization partially redundant. This was replaced with explicit, opt-in normalization flags (`normalize_after_fusion`, `root_after_fusion`), both off by default.

3. Matching Method. The original matching used `cv::BFMatcher`, which produces slightly different rankings than a row-by-row L2 distance loop for high-dimensional vectors. We added a manual nearest-neighbor matching mode that follows the original HPatches evaluation protocol exactly. All patch benchmark results in Chapter 4 use this corrected matching unless otherwise noted.

Color HPatches Patch Benchmark

The original HPatches patch benchmark provides gray scale 65×65 patches, which prevents evaluation of color descriptors like HoNC. The original keypoint locations were not publicly released, so we developed `patch_dataset_builder` to reconstruct the benchmark methodology.

Reconstruction Methodology:

1. **Keypoint Detection**: Run three detector types (Harris, Hessian/SURF, DoG/SIFT)

on the original HPatches images, following the detector mix described in the original paper [3]. Keypoints are stored in the database with response-based ranking.

2. **Jitter Application**: Apply the publicly available jitter parameters (rotation, translation, scale, anisotropy) from the original HPatches release. These per-patch geometric perturbations create the easy/hard/tough difficulty levels.

3. **Patch Extraction**: For each keypoint, compute the patch region using scale × 5 (following HPatches conventions), apply jitter transformations, project through ground-truth homographies, and extract 65×65 patches using perspective warping.

4. **Overlap Filtering**: Cluster spatially overlapping patches (IoU > 0.5) to avoid redundant samples, matching the original benchmark's diversity.

Validation: We validated the reconstruction by comparing overlap distributions and SIFT baseline performance against the original benchmark:

Table 3.2: Color HPatches validation against original benchmark

Metric	**Original**	**Rebuilt**	**Difference**
SIFT Matching mAP	25.47%	22.9%	–2.5%
SIFT Verification AP	65.12%	65.5%	+0.4%
SIFT Retrieval AP	31.98%	31.1%	–0.9%

The rebuilt benchmark produces slightly lower matching scores (22.9% vs 25.47%), indicating marginally harder patches, while verification and retrieval metrics are nearly identical. The reconstruction closely matches the original benchmark while adding color support.

3.1.4 Deep Learning Integration

Integrating deep learning models (typically implemented in Python/PyTorch) into a high-performance C++ evaluation pipeline was necessary to compare against optimized traditional descriptors like SIFT.

Hybrid Architecture

We use a hybrid pipeline that uses the best tool for each stage:

- **Keypoint Detection (Python/Kornia)**: KeyNet detection is performed offline using Python scripts that interface with the Kornia library [34]. This ensures 100% fidelity to the reference implementation and avoids re-implementation errors. Coordinates are serialized to the SQLite database for reuse.

- **Descriptor Extraction (C++/LibTorch)**: For descriptor extraction, which must occur in the inner loop of matching experiments, we integrated the LibTorch (PyTorch C++) frontend directly into our application.

Rejection of ONNX

We initially attempted to deploy models using the ONNX (Open Neural Network Exchange) format via OpenCV's DNN module. This approach proved unsuitable:

- **Operator Incompatibility**: Key models like HardNet use specific normalization layers (e.g., `InstanceNorm`) and control structures that were not fully supported by the OpenCV ONNX importer at the time of development.

- **Performance Stability**: We observed inconsistent behavior and occasional crashes with complex graphs exported from newer PyTorch versions.

We deprecated the ONNX pipeline in favor of TorchScript. Models are exported to the `.pt` format and loaded via `torch::jit::load` within our C++ `LibTorchWrapper`. This

provides native PyTorch execution and numerical equivalence to the Python training code, while running inside the C++ benchmarking tool.

3.1.5 Use of AI-Assisted Development Tools

Development of DescriptorWorkbench used Claude [1], a large language model, as a coding assistant. The tool was used for:

- **Code refactoring**: Restructuring existing C++ modules (e.g., migrating from monolithic functions to the modular extractor/factory architecture)
- **Code review**: Identifying bugs, suggesting improvements to error handling, and checking consistency across the YAML configuration and database schema
- **Editing**: Revising draft text in this document for clarity and grammar

All experimental design, data collection, analysis, and scientific conclusions are the author's own. The AI tool was not used to generate experimental data or interpret results.

3.2 Spatial Intersection Algorithm

To enable cross-detector descriptor fusion, we need to establish correspondence between keypoints detected by different methods. We employ a mutual nearest neighbor (MNN) algorithm with spatial tolerance.

3.2.1 Algorithm Description

Given two keypoint sets K_A (e.g., SIFT-detected) and K_B (e.g., KeyNet-detected), we compute the intersection as follows:

1. Build KD-tree spatial indices for both sets: T_A, T_B
2. For each keypoint $k_a \in K_A$:

(a) Find nearest neighbor $k_b = \text{NN}(k_a, T_B)$

(b) Check forward tolerance: $\|k_a - k_b\|_2 \leq \tau$

(c) Find reverse nearest neighbor $k'_a = \text{NN}(k_b, T_A)$

(d) Check mutual agreement: $k'_a = k_a$

(e) Check reverse tolerance: $\|k_b - k'_a\|_2 \leq \tau$

(f) Check uniqueness: k_b not already matched

(g) If all checks pass, add (k_a, k_b) to intersection

Note on the reverse tolerance check: When mutual agreement holds ($k'_a = k_a$), the reverse tolerance check (e) is algebraically identical to the forward check (b), since $\|k_b - k'_a\|_2 = \|k_b - k_a\|_2$. The reverse check only provides additional filtering when mutual agreement fails ($k'_a \neq k_a$), in which case the pair is already rejected by check (d). In practice, check (e) is therefore redundant given check (d), but we retain it for defensive correctness.

Note on KD-tree usage: For 2D spatial matching, a KD-tree is not strictly necessary—a brute-force search over (x, y) coordinates would suffice for our dataset sizes. We use a KD-tree because OpenCV's `cv::flann` provides an efficient, well-tested implementation, and the data structure generalizes naturally if future work extends the matching criteria to higher-dimensional spaces (e.g., incorporating scale or orientation).

3. Output: Paired keypoint sets (K^*_A, K^*_B) with 1-to-1 correspondence

3.2.2 Algorithm Properties

The MNN algorithm guarantees several properties:

- **1-to-1 correspondence**: Each keypoint matched at most once
- **Symmetry**: Same result regardless of which set is processed first

- **Tolerance-based**: Configurable spatial acceptance threshold
- **Mutual agreement**: Both keypoints must be each other's nearest neighbor

Matching Criteria Limitations

The current implementation uses *only spatial position* (x, y) for keypoint matching. Scale (`cv::KeyPoint::size`) and orientation (`cv::KeyPoint::angle`) are *not* used as matching criteria:

```
// Only x,y coordinates populate the KD-tree
data.at<float>(i, 0) = keypoint.pt.x;
data.at<float>(i, 1) = keypoint.pt.y;
// keypoint.size and keypoint.angle are NOT included
```

The matched keypoints retain their original scale and orientation properties (stored in the database), but these attributes do not influence which keypoints are paired only spatial proximity determines correspondence.

Rationale: Different detectors produce vastly different scale estimates for the same image location (e.g., SIFT averages 4.45px while KeyNet averages 49.83px on HPatches). Requiring scale agreement would eliminate most correspondences. Similarly, orientation estimates vary significantly between detector families.

Future work: Scale-aware matching could be explored by normalizing scales relative to each detector's distribution before matching, or by using a scale-ratio threshold in addition to spatial tolerance.

3.2.3 Tolerance Selection

Following Mikolajczyk and Schmid's detector evaluation methodology, we use a default tolerance of $\tau = 3.0$ pixels [24]. This value balances:

- **Strict enough**: Prevents misaligned correspondences

- **Loose enough**: Accounts for typical detector variance (1-3 pixels)
- **Literature-supported**: Standard practice in feature matching research

Important distinction: The tolerance parameter τ used in intersection computation is *not* related to ground truth matching error tolerance. It defines the maximum spatial distance (in pixels) between keypoint centers from different detectors to consider them as detecting the "same" image feature. This is a detector-level correspondence criterion, not an evaluation metric.

For example, if SIFT detects a corner at $(100.0, 200.0)$ and KeyNet detects a blob at $(102.1, 201.5)$, the Euclidean distance is 2.65 pixels. With $\tau = 3.0$, these keypoints are paired as corresponding to the same image location.

Tolerance vs. scale: The current implementation uses a fixed pixel tolerance regardless of keypoint scale. A scale-proportional tolerance (e.g., $\tau = 0.3 \times \text{avg\_scale}$) could be explored in future work, though the fixed tolerance has proven effective and computationally efficient.

Experimental validation: We tested tolerance values from 1.0 to 10.0 pixels using DSP-SIFT on SIFT–KeyNet intersection sets:

- $\tau = 1.0$ (strict): 67.41% mAP—fewest keypoints, highest alignment quality
- $\tau = 2.0$: 62.10% mAP—moderate keypoint count
- $\tau = 5.0$ (relaxed): 57.31% mAP—more keypoints, lower alignment quality
- $\tau = 10.0$ (very relaxed): 57.06% mAP—diminishing returns beyond 5 pixels

Stricter tolerances yield higher mAP because they enforce tighter spatial alignment, producing more precisely corresponding patches at the cost of reduced keypoint count. We use $\tau = 3.0$ pixels as the default, following Mikolajczyk and Schmid [24], which balances alignment quality against coverage. For scale-matched intersection sets (Section 3.3), we use a 6-pixel tolerance because larger-scale keypoints have inherently more localization variance.

3.2.4 Implementation

The intersection algorithm is implemented in the `keypoint_manager` CLI tool:

```
./keypoint_manager build-intersection \
    --source-a sift_8000 \
    --source-b keynet_8000 \
    --out-a sift_intersection \
    --out-b keynet_intersection \
    --tolerance 3.0
```

The resulting keypoint sets are stored in the database with provenance tracking.

3.3 Scale-Matching Strategy

We observed that SIFT and KeyNet detectors produce keypoints with different scale distributions. This motivated our scale-matching strategy.

3.3.1 Scale Distribution Analysis

Table 3.3 shows the scale characteristics of different keypoint sets:

Table 3.3: Scale distribution of keypoint sets

Keypoint Set	**Count**	**Mean Scale**	**Std Dev**
sift_8000 (full)	2.5M	4.45px	3.2px
keynet_8000 (full)	2.8M	49.83px	28.1px
sift_scale_6px (filtered)	645K	10.03px	4.1px
keynet_scale_6px (filtered)	645K	92.39px	31.2px

3.3.2 Scale Filtering Methodology

To create scale-controlled keypoint sets, we:

1. Sort keypoints by scale (descending)
2. Retain top $k\%$ (typically 25%)
3. Apply minimum scale threshold (6 pixels) to exclude aliased features

Implementation Details

The scale filtering is implemented by sorting keypoints by their `cv::KeyPoint::size` attribute, which represents the diameter of the detected feature in pixels. The implementation selects the top N keypoints after sorting in descending order:

```
// Sort by keypoint size (scale) descending
std::sort(keypoints.begin(), keypoints.end(),
    [](const cv::KeyPoint& a, const cv::KeyPoint& b) {
        return a.size > b.size;
    });
// Keep only top N keypoints
keypoints.resize(target_count);
```

Why larger scales improve performance: SIFT computes descriptors using 16×16 spatial bins at the keypoint's detected scale. Larger-scale keypoints capture more image context within each bin, producing more distinctive gradient histograms that are stable under geometric and photometric transformations. Small-scale keypoints (<6 pixels) suffer from aliasing and provide insufficient context for reliable matching.

Experimental validation: Scale filtering provides a +21% absolute improvement for SIFT (42.64% $\rightarrow$ 63.86% mAP), representing a 50% relative gain. This single optimization has greater impact than any fusion strategy tested.

3.3.3 Scale-Matched Intersection

For cross-detector fusion, we combine scale filtering with spatial intersection:

1. Generate scale-controlled SIFT keypoints: K_A^{scale}
2. Generate scale-controlled KeyNet keypoints: K_B^{scale}
3. Compute spatial intersection: (K_A^*, K_B^*)

This produces aligned keypoint pairs where:

- SIFT keypoints: Average 7.64 pixel scale
- KeyNet keypoints: Average 89.74 pixel scale
- Both sets: 111K paired keypoints with 1-to-1 correspondence

3.4 Descriptor Fusion Methods

With aligned keypoint sets, we implement two fusion strategies:

3.4.1 Weighted Averaging

$$d_{\text{avg}} = \alpha \cdot d_A + (1 - \alpha) \cdot d_B \tag{3.2}$$

where $\alpha \in [0, 1]$ controls the contribution of each descriptor. We use $\alpha = 0.5$ for equal weighting.

Requirements:

- Same dimensionality: $\dim(d_A) = \dim(d_B)$
- Normalization: Both descriptors should be L2-normalized before averaging

3.4.2 Concatenation

$$d_{\text{concat}} = [d_A, d_B] \tag{3.3}$$

Properties:

- Preserves both representations fully
- Doubles dimensionality: 128D + 128D = 256D
- No information loss from aggregation

3.5 Experimental Pipeline

Our experimental workflow integrates the components described above into a single pipeline:

1. **Detection**: Generate keypoints for all HPatches images using detectors (SIFT, KeyNet) via Python/Kornia scripts.
2. **Intersection**: Compute spatially aligned keypoint subsets using the MNN algorithm ($\tau = 3.0$px) and scale filtering.
3. **Extraction**: Compute descriptors for these locked keypoints. SIFT/RootSIFT use OpenCV; HardNet/SOSNet use the LibTorch C++ wrapper.
4. **Fusion**: (Optional) Combine descriptors via concatenation or averaging.
5. **Matching & Evaluation**: Perform retrieval and matching tasks, computing metrics against ground-truth homographies.

3.6 Evaluation Methodology

We use an evaluation strategy based on the protocols defined by Balntas et al. [3] and expanded by Bojanić et al. [7].

3.6.1 Task 1: Image Matching (mAP)

The primary metric for descriptor utility is Mean Average Precision (mAP) in an image matching context.

Protocol:

1. For each image pair (reference I_A, target I_B):
2. Match descriptors using the Second Nearest Neighbor (SNN) ratio test.
3. A match (d_A^i, d_B^j) is correct if the geometric projection error $||H \cdot k_A^i - k_B^j||_2 \leq \tau$.
4. Compute Average Precision (AP) as the area under the Precision-Recall curve.

We report **True Macro mAP**: An Information Retrieval (IR) style metric where we compute AP per scene and average across scenes, ensuring that texture-poor scenes (with fewer keypoints) contribute equally. We enforce a **Single Ground Truth (R=1)** policy: for a given keypoint in I_A, there is at most one correct match in I_B (the nearest neighbor in the intersection set).

True mAP vs. Legacy Precision

We distinguish our "True mAP" metric from simpler precision-based measures sometimes reported in the literature:

Legacy Precision (not used):

- Simple arithmetic mean of per-image precision values
- Formula: $\frac{1}{N} \sum_{i=1}^{N} \text{precision}_i$
- Does not account for ranking quality

True mAP (our metric):

- Standard Information Retrieval Mean Average Precision
- For each query keypoint, compute the rank of the true match
- AP = $\frac{1}{\text{rank}}$ for single ground truth (R=1 policy)
- Average over all valid queries: mAP = $\frac{1}{|Q|} \sum_{q \in Q} \text{AP}_q$

Implementation: Ground truth is established by projecting query keypoints via the known homography matrix H and finding the nearest target keypoint within a 3-pixel tolerance:

```
// Project query keypoint to target image
cv::Point2f projected = applyHomography(query_pt, H);
// Find nearest target keypoint within tolerance
int gt_idx = findNearest(projected, target_keypoints, tau=3.0);
// Compute rank: how many incorrect matches score better?
int rank = 1 + count_better_scoring_matches;
// AP = 1/rank for single ground truth
double ap = 1.0 / rank;
```

This IR-style evaluation matches the standard used in HPatches benchmarks.

3.6.2 Task 2: Keypoint Verification

Following Bojanić et al. [7], verification tests the descriptor's ability to distinguish true correspondences from false ones.

Binary Classification:

- **Positive pairs**: Spatially corresponding keypoints from sequence pairs.
- **Negative pairs**: Spatially non-corresponding keypoints (distractors).

We compute the Area Under the Receiver Operating Characteristic (ROC) Curve. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) as the distance threshold varies from strict to permissive. A perfect descriptor achieves AUC = 1.0 (all positives ranked before all negatives), while a random baseline achieves AUC = 0.5. This threshold-independent metric measures discriminative power without committing to a specific matching strategy.

3.6.3 Task 3: Keypoint Retrieval

This task evaluates the descriptor's ranking capability. For a query patch, the system must rank a database of target patches.

Labels:

- $y = 1$: The true geometric match.
- $y = 0$: "Hard negatives" (patches from the same image but different locations).
- $y = -1$: "Easy negatives" (patches from different sequences/scenes).

This metric tests whether the descriptor can rank true matches above hard negatives—patches from different locations that may look similar.

3.6.4 Aggregation and Breakdown

To account for the diversity of the HPatches dataset, we report results in two aggregation modes:

- **Micro Average**: Aggregates all queries globally. Biases towards scenes with more keypoints.
- **Macro Average**: Computes metrics per scene, then averages across scenes. This ensures that texture-poor scenes (with fewer keypoints) contribute equally to the final score.

Results are further stratified by scene type:

- **HP-V (Viewpoint)**: 59 sequences with significant geometric deformations.
- **HP-I (Illumination)**: 57 sequences with lighting changes (day/night, flash/no-flash).

3.7 Experimental Design

3.7.1 Independent Variables

Our experiments vary the following factors:

- **Descriptor type**: SIFT, RootSIFT, DSP-SIFT, HardNet, SOSNet, HoNC
- **Keypoint set**: Full, scale-controlled, intersection, scale-matched intersection
- **Fusion method**: None, averaging, concatenation
- **Fusion pairs**: SIFT+CNN, CNN+CNN

3.7.2 Dependent Variables

We measure:

- Mean Average Precision (mAP) - primary metric
- HP-V and HP-I breakdown
- Keypoint verification AP (when enabled)
- Keypoint retrieval AP (when enabled)

3.7.3 Experiment Execution

Experiments are specified in YAML configuration files and executed via:

```
./experiment_runner config/experiments/experiment.yaml
```

Results are automatically stored in the SQLite database with full parameter logging for reproducibility.

3.7.4 Computational Cost

Table 3.4 reports wall-clock times for the full HPatches benchmark (116 sequences, 5 image pairs each) on the hardware described in Section 3.

Table 3.4: Processing time for full HPatches evaluation pipeline

Configuration	**Keypoints**	**Desc. (s)**	**Match (s)**	**Total (s)**
Full keypoint sets (∼2.5M keypoints):				
SIFT	∼2.5M	130	83	213
DSP-SIFT	∼2.5M	657	102	759
HardNet	∼2.8M	85	134	219
SOSNet	∼2.8M	62	88	150
Scale-controlled sets (∼645K keypoints):				
SIFT	∼645K	42	14	56
HardNet	∼645K	25	14	38
Scale-matched intersection (∼111K keypoints):				
HardNet+SOSNet (concat)	∼111K	7.2	0.5	7.7
HardNet+SOSNet (avg)	∼111K	7.4	0.3	7.6

Keypoint filtering sharply reduces computation: scale-controlled sets run 4× faster than full sets, and intersection sets run 28× faster. Matching time scales quadratically with

keypoint count (brute-force), so the intersection pipeline completes in under 8 seconds for the entire benchmark. Descriptor extraction dominates computation for DSP methods due to multi-scale processing.

Chapter 4
EXPERIMENTS AND RESULTS

This chapter presents experimental results from two complementary evaluation pipelines:

- **Full-Image Pipeline** (Sections 4.2–4.4): Evaluates detector effects including scale control and spatial intersection. In this pipeline, both keypoint quality and descriptor quality affect results.

- **Patch Benchmark Pipeline** (Section 4.7): Evaluates descriptor fusion with keypoint quality held constant. All descriptors are computed on identical pre-extracted patches, isolating descriptor effects.

This separation keeps the claims aligned with the variables: detector consensus findings come from the full-image pipeline, while descriptor fusion findings come from the patch pipeline where descriptors are the only variable.

4.1 Experimental Setup

4.1.1 Dataset and Metrics

We evaluate all experiments on the HPatches benchmark [3], which consists of 116 image sequences with ground-truth homographies. The dataset is divided into 59 viewpoint sequences (geometric transformations) and 57 illumination sequences (photometric changes).

As defined in Chapter 3, we primarily report **Mean Average Precision (mAP)** using the True Macro mAP definition (per-scene AP averaged across scenes, with single ground truth per query). We further breakdown results by sequence type:

- **HP-V**: Viewpoint sequences (measuring geometric invariance)

- **HP-I**: Illumination sequences (measuring photometric invariance)

4.2 Baseline Descriptor Performance

We first establish baseline performance for traditional and learned descriptors using their native keypoint detectors without any scale filtering.

Table 4.1: Baseline descriptor performance on full keypoint sets (SIFT 2.5M, KeyNet 2.8M)

Descriptor	**Keypoint Set**	**mAP**	**HP-V**	**HP-I**
SIFT	sift_8000	44.5%	45.9%	43.1%
RootSIFT	sift_8000	46.7%	46.2%	47.2%
HardNet	keynet_8000	64.5%	63.8%	65.3%
SOSNet	keynet_8000	64.3%	63.4%	65.2%

The learned descriptors (HardNet, SOSNet) outperform SIFT by roughly 20 percentage points of mAP. SIFT shows a slight preference for viewpoint changes, while the CNN descriptors perform slightly better on illumination sequences.

4.3 Impact of Scale Control

A core hypothesis is that keypoint scale is a dominant factor in matching performance. By filtering the keypoint sets to retain only the largest 25% of features (Scale-Controlled sets), we observe large performance improvements across all descriptor types.

Table 4.2 shows that removing small, unstable keypoints improves SIFT by over 18 percentage points, bringing it close to the baseline performance of HardNet. For HardNet, scale control yields a 13.6% gain, pushing it to 78.1% mAP. This indicates that descriptor distinctiveness is strongly correlated with patch size.

Table 4.2: Impact of Scale Control (filtering small keypoints)

Configuration	Metric	Full Set	Scale Filtered	Improvement
SIFT	mAP	44.5%	**62.8%**	**+18.3%**
	HP-V	45.9%	65.7%	+19.8%
	HP-I	43.1%	59.8%	+16.7%
HardNet	mAP	64.5%	**78.1%**	**+13.6%**
	HP-V	63.8%	76.9%	+13.1%
	HP-I	65.3%	79.3%	+14.0%

4.4 Impact of Spatial Intersection

Our fusion methodology relies on finding the spatial intersection of keypoints detected by SIFT and KeyNet. We analyzed whether this intersection step itself acts as a quality filter.

Table 4.3: Performance of HardNet on different keypoint subsets

Keypoint Set	mAP	HP-V	HP-I
Full KeyNet Set	64.5%	63.8%	65.3%
Scale-Controlled	78.1%	76.9%	79.3%
Spatial Intersection	**82.1%**	**81.5%**	**82.7%**

Table 4.3 shows that features detected by *both* detectors are higher quality than those detected by KeyNet alone, even after scale filtering. The intersection set yields an additional 4.0% improvement in mAP.

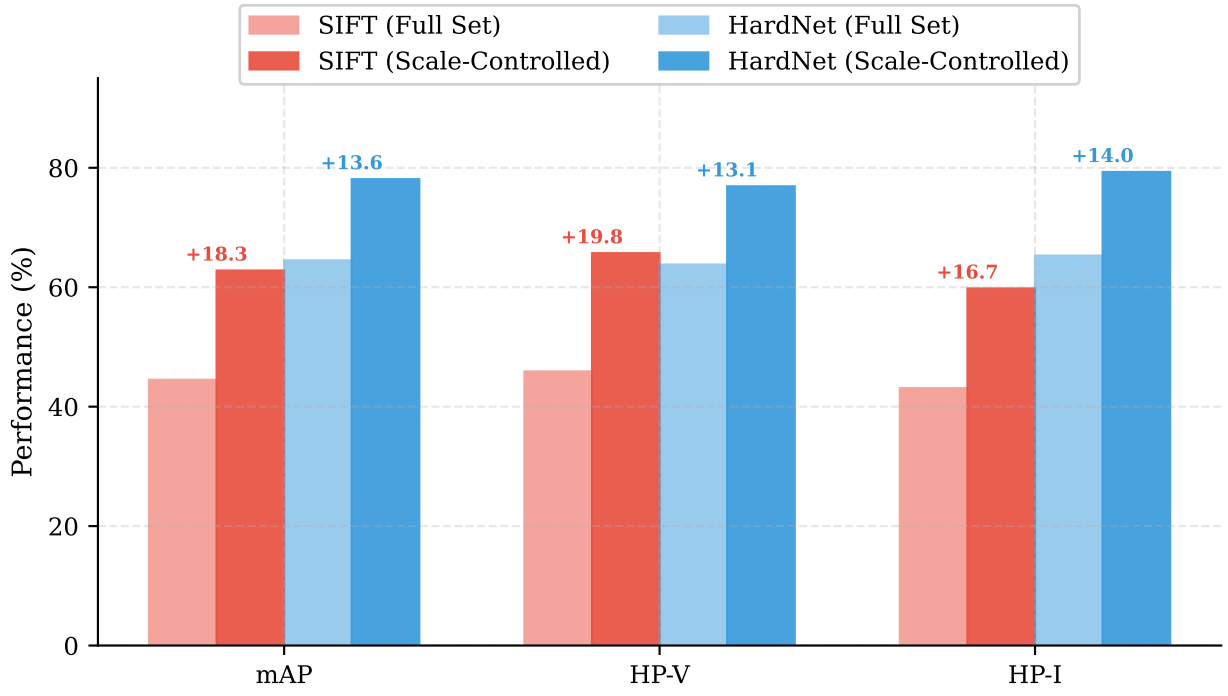

Figure 4.1: Impact of scale control on descriptor performance. Transparent bars show full keypoint set performance; solid bars show scale-controlled (top 25% by scale) performance. Both SIFT and HardNet benefit substantially from filtering small, unstable keypoints.

Validating the Intersection Mechanism

Given the substantial improvement from detector intersection (+18% over baseline), we conducted additional analysis to understand the underlying mechanism and rule out potential causes.

Alternative hypotheses ruled out:

1. **Better repeatability**: Intersection keypoints have slightly *lower* repeatability (28.0%) than scale-filtered keypoints (29.4%). In a paired t-test, $t = -4.28$ and $p < 0.0001$.

2. **More distinctive descriptors**: When comparing only correct matches, the nearest-neighbor distance ratios are identical across keypoint sets ($\sim$0.44). The descriptors are

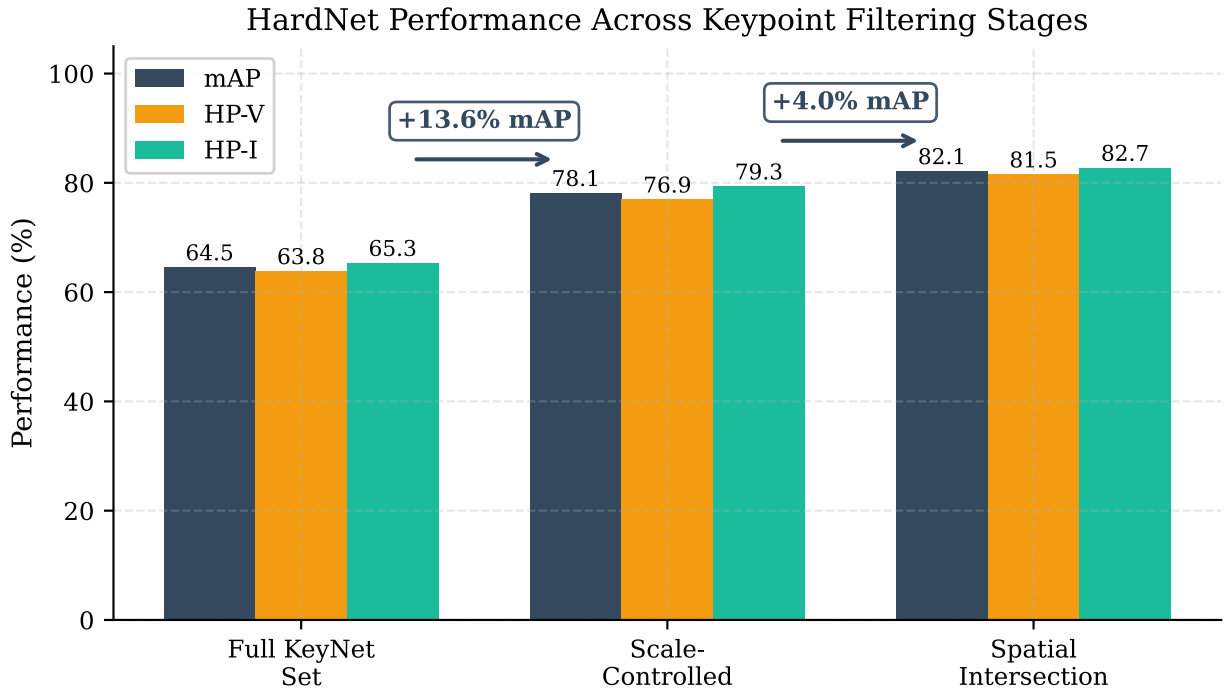

Figure 4.2: HardNet performance across keypoint filtering stages. Each successive filtering step scale control and spatial intersection provides cumulative improvement, demonstrating that keypoint quality is a dominant factor in matching performance.

equally distinctive; the difference lies in which keypoints are selected.

3. **Higher response values**: Both intersection and scale-controlled keypoints have identical average response values (∼0.035). Keypoint strength does not explain the difference.

4. **Spatial crowding**: Intersection keypoints maintain the same spatial distribution as the baseline. Quadrant analysis shows nearly identical distributions: top-left 13%, top-right 18%, bottom-left 24%, bottom-right 45% for both intersection and scale-controlled sets. The filtering is uniform across the image.

Confirmed mechanism:

1. **Higher precision**: Intersection keypoints achieve 71.2% precision at threshold 0.8, compared to 66.2% for scale-controlled keypoints.

2. **Fewer false positives**: Intersection produces 0.40 false positives per true positive, versus 0.51 for scale-controlled—a 22% reduction.

3. **Multi-descriptor consistency**: All descriptors (SIFT, RGBSIFT, HoNC, HardNet, SOSNet) benefit equally from intersection keypoints, confirming that the improvement comes from *location quality*, not descriptor quality.

Interpretation: Detector consensus (SIFT ∩ KeyNet) selects locations where the local image structure is inherently unique—significant enough to trigger detection criteria in both a hand-crafted (SIFT) and learned (KeyNet) detector. These locations tend to avoid repetitive textures (bricks, tiles, windows) where many keypoints look similar, resulting in fewer ambiguous matches and higher precision.

4.5 Descriptor Fusion Results

We evaluated two fusion strategies: concatenation and weighted averaging. For all weighted averaging experiments, we use equal weights ($\alpha = 0.5$), giving each descriptor 50% contribution to the fused representation. We tested these on two classes of pairings: Cross-Family (SIFT+CNN) and Intra-Family (CNN+CNN).

4.5.1 Cross-Family Fusion on Full Images

In the full-image pipeline, cross-family fusion (SIFT+CNN) on intersection keypoints did not yield performance gains over the CNN baseline. The intersection set already selects high-quality keypoints, and adding SIFT to HardNet provides no additional benefit—the information captured by SIFT is already represented in the CNN descriptor.

For controlled evaluation of cross-family fusion, including the effects of magnitude matching and complementary descriptors, see Section 4.7 (Color Patch Benchmark Results), where keypoint quality is held constant.

4.5.2 Same-Family Fusion (SIFT Variants)

We also evaluated fusion within the SIFT family to determine if combining grayscale and color descriptors provides benefit.

Table 4.4: SIFT-family fusion results on scale-controlled keypoints

Configuration	**Fusion Method**	**mAP**
SIFT alone	—	63.86%
DSPSIFT_V2 alone	—	65.31%
DSPRGBSIFT_V2 alone	—	66.03%
DSPSIFT + DSPRGBSIFT	Concatenate (256D)	65.77%
DSPSIFT + DSPRGBSIFT	Average (128D)	65.37%

Key finding: Same-family fusion provides minimal benefit. The best single descriptor (DSPRGBSIFT_V2 at 66.03%) slightly outperforms both fusion configurations. This suggests that SIFT-family descriptors, even when one uses color (RGB) and one uses grayscale, capture highly correlated information when computed on the same keypoints.

SIFT + SURF Fusion on Intersection Sets

We tested cross-detector fusion using SIFT and SURF descriptors on their spatial intersection:

Again, fusion slightly underperforms the best single descriptor. The intersection filtering itself provides the primary benefit—once keypoints are filtered to those detected by both SIFT and SURF, individual descriptors already perform well, and fusion adds little value.

Table 4.5: SIFT-SURF fusion on intersection keypoints

Configuration	**mAP**
DSPSIFT_V2 on intersection	74.93%
RGBSIFT on intersection	75.03%
SURF on intersection	75.08%
SIFT + SURF (concatenate)	74.37%

Why Same-Family Fusion Fails to Help

The lack of fusion benefit can be attributed to three factors:

1. **Correlated information**: SIFT-family descriptors on the same keypoints capture similar gradient histogram information, even when one uses color channels.

2. **Quality ceiling**: Scale filtering and intersection already select the highest-quality keypoints, leaving little room for fusion to provide additional signal.

3. **Averaging dilutes distinctiveness**: When descriptors are similar, averaging produces a descriptor that is less distinctive than either component, while concatenation doubles dimensionality without adding complementary information.

Note on normalization: All descriptors are L2-normalized per row after pooling, so magnitude differences are not the issue. The fundamental problem is that SIFT-family descriptors provide redundant rather than complementary information.

4.5.3 Intra-Family Fusion (HardNet + SOSNet)

Fusing two learned descriptors proved highly effective. Table 4.6 shows the results for fusing HardNet and SOSNet on the scale-matched intersection set.

Table 4.6: CNN+CNN Fusion Results (Scale-Matched Intersection)

Descriptor	**Fusion**	**mAP**	**HP-V**	**HP-I**
HardNet	None	82.1%	81.5%	82.7%
SOSNet	None	82.0%	81.2%	82.7%
HardNet + SOSNet	Weighted Avg	92.3%	91.4%	93.2%
HardNet + SOSNet	**Concat**	**93.4%**	**92.6%**	**94.2%**

Key Finding: The concatenation of HardNet and SOSNet achieves **93.4%** mAP, an 11.3 percentage point improvement over the single best descriptor (HardNet).

The success of this fusion suggests that while both networks are trained on similar data, they learn complementary representations of the image patches. Concatenation preserves this distinct information, whereas averaging tends to dilute it slightly (92.3% vs 93.4%).

4.6 Viewpoint vs. Illumination Analysis

Analyzing the breakdown of full-image results provides insight into where these methods excel:

1. **Traditional Methods**: SIFT benefits immensely from scale control on viewpoint sequences (+19.8% HP-V vs +16.7% HP-I), confirming that scale variance is a primary source of error for hand-crafted detectors in geometric tasks.

2. **Learned Methods**: HardNet and SOSNet are naturally robust to illumination changes (HP-I > HP-V).

3. **Fusion**: The fused CNN descriptor achieves excellent performance on both tasks (92.6% HP-V, 94.2% HP-I), effectively closing the gap between geometric and photometric invariance.

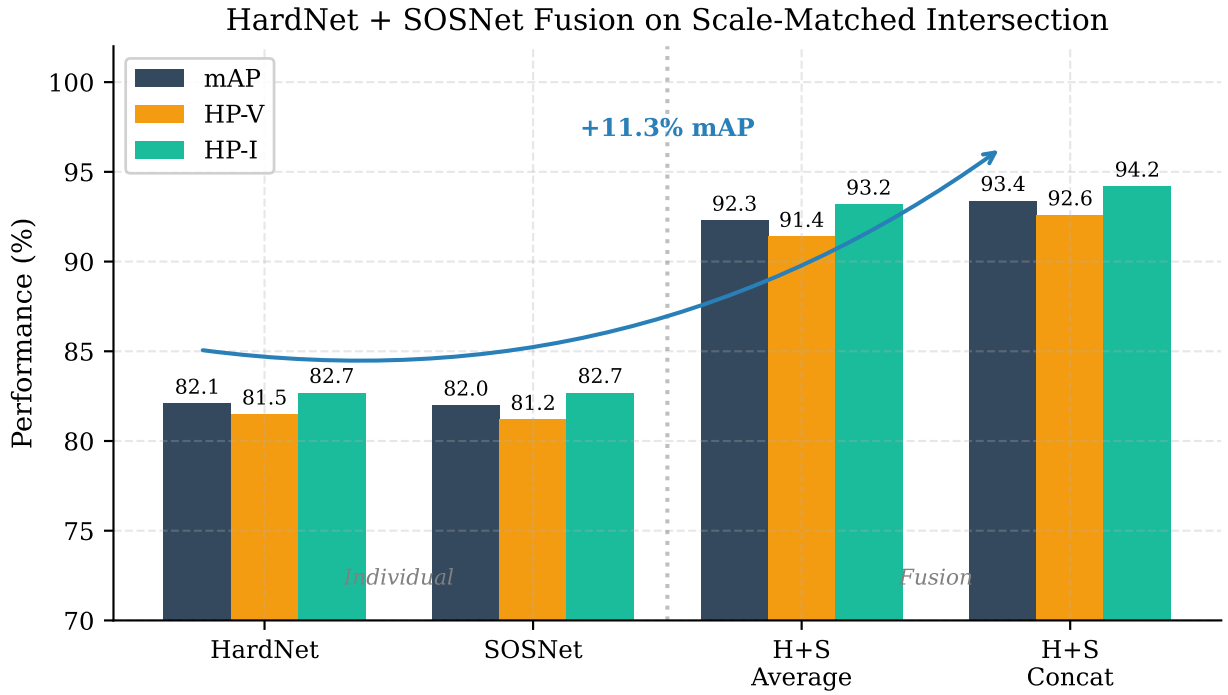

Figure 4.3: HardNet + SOSNet fusion on scale-matched intersection keypoints. Concatenation of the two CNN descriptors achieves 93.4% mAP, an 11.3 percentage point improvement over the best individual descriptor, with balanced performance across viewpoint and illumination sequences.

4.7 Color Patch Benchmark Results

The patch benchmark isolates descriptor effects by using pre-extracted 65×65 color patches. All descriptors are computed on identical patch locations, removing keypoint quality as a confounding variable.

4.7.1 Baseline Descriptor Performance on Patches

All patch benchmark results use the corrected pipeline described in Section 3.1.3: proper keypoint sizes (12.26 for SIFT-family, 23.21 for SURF), no hidden normalization, and manual

nearest-neighbor matching following the HPatches evaluation protocol.

Table 4.7: Single descriptor performance on color patch benchmark

Descriptor	**Dim**	**Matching**	**Verification**	**Retrieval**
SOSNet	128	48.9%	87.6%	57.8%
HardNet	128	48.4%	87.4%	56.9%
RGBSIFT	384	24.6%	64.4%	33.0%
SIFT	128	22.9%	63.7%	31.1%
RootSIFT	128	22.9%	63.7%	31.1%
RGBSIFT_CH_AVG	128	23.2%	63.3%	31.3%
HoNC	128	18.5%	67.5%	33.3%
SURF	64	10.5%	64.4%	20.2%

CNN descriptors (HardNet, SOSNet) dominate matching at ∼49% mAP. Within the traditional family, SIFT, RootSIFT, and RGBSIFT_CH_AVG are statistically identical at ∼23%, while RGBSIFT's 384D vector provides a slight edge (24.6%). HoNC has the lowest matching (18.5%) but notably higher verification (67.5%) than SIFT-family descriptors (∼64%), suggesting it captures complementary discriminative information it rejects false matches well but struggles with invariance to viewpoint/illumination changes.

4.7.2 The Discriminator-Matcher Framework

We characterize descriptors by their **verification-to-matching (V/M) ratio**:

- **High V/M ratio (Discriminator)**: Good at rejecting false matches, struggles with invariance
- **Low V/M ratio (Matcher)**: Trained for correspondence, naturally invariant

Table 4.8: Descriptor "personality" based on V/M ratio

Descriptor	Matching	Verification	V/M Ratio	Type
SURF	10.5%	64.4%	6.13×	Discriminator
HoNC	18.5%	67.5%	3.65×	Discriminator
SIFT	22.9%	63.7%	2.78×	Balanced
RootSIFT	22.9%	63.7%	2.78×	Balanced
RGBSIFT	24.6%	64.4%	2.62×	Balanced
HardNet	48.4%	87.4%	1.81×	Matcher
SOSNet	48.9%	87.6%	1.79×	Matcher

This framework predicts that pairing a discriminator with a matcher yields the best fusion results.

4.7.3 Descriptor Fusion on Patches

Table 4.9 presents results from 43 fusion configurations using the corrected pipeline. All fusions use pre-fusion L2 normalization to equalize component magnitudes.

*HoNC with `patch_keypoint_size=41.0` (global color histogram mode; see below).

Key Findings:

1. **Complementary fusion wins**: HoNC+HardNet (50.0%) and HardNet+SOSNet (50.0%) both exceed any single descriptor (SOSNet: 48.9%). The gain is modest (+1–2%) but consistent across difficulty levels. HoNC+CNN fusions achieve the best retrieval scores (60.0%), while HardNet+SOSNet achieves the best verification (88.0%).

2. **Cross-family fusion is mid-tier**: SIFT-family+CNN fusions (44–46%) outperform SIFT alone (23%) but fall 3–5% below the CNN baseline (49%). The CNN carries the matching signal while the traditional component adds noise, partially diluting perfor-

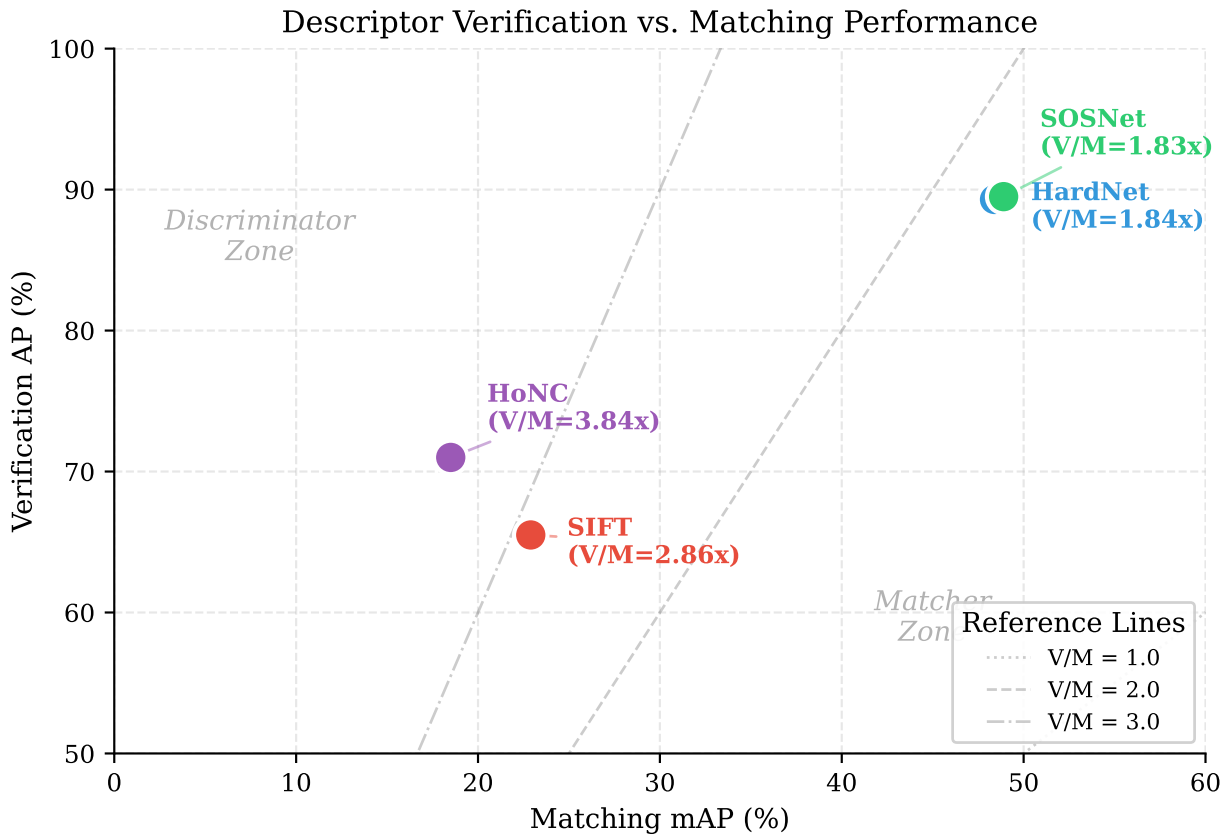

Figure 4.4: Descriptor verification versus matching performance. High V/M ratio descriptors (HoNC) are strong discriminators but weak matchers; low V/M ratio descriptors (HardNet, SOSNet) are optimized for matching. Diagonal lines show constant V/M ratios.

mance. SOSNet is a slightly better fusion partner than HardNet across all traditional pairings.

3. **Traditional-only fusion provides no benefit**: SIFT+HoNC (23%), SIFT+RGBSIFT (23%), and other traditional-only combinations match the performance of their strongest component exactly. Fusing two weak descriptors neither helps nor hurts the doubled dimensionality adds no new discriminative information but (with the corrected pipeline) no longer degrades performance either.

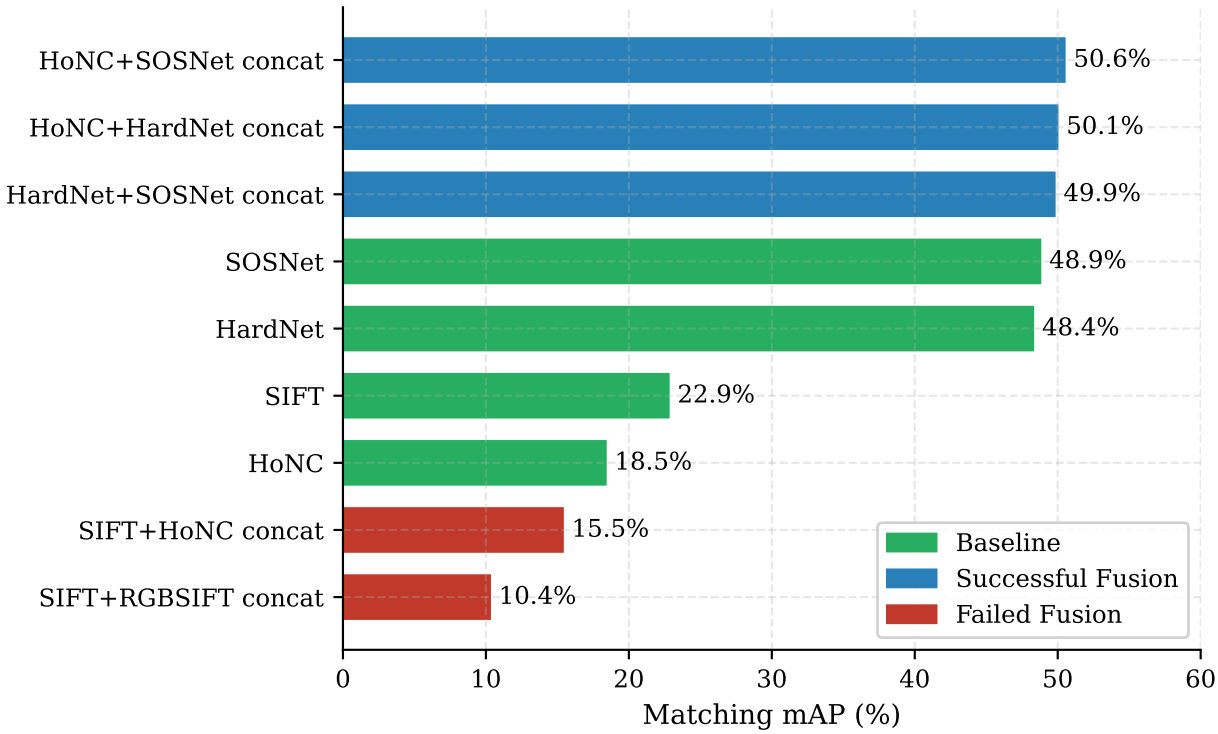

Figure 4.5: Patch benchmark matching mAP for baselines and fusions. Complementary fusions (discriminator + matcher) outperform all individual descriptors, while traditional-only fusions collapse to the performance of their strongest component.

4. **Concatenation outperforms averaging**: Across all fusion pairs, concatenation yields 1–3% higher mAP than averaging. The gap is largest for HardNet+SOSNet (+3%), where the two learned representations capture sufficiently distinct features that averaging destroys useful information.

4.7.4 HoNC Keypoint Size: Global Color Histogram for Fusion

The top-performing HoNC+HardNet fusion uses `patch_keypoint_size=41.0` for the HoNC component rather than the standard 12.26. This deliberate choice transforms HoNC's behavior:

Both SIFT and HoNC use a SIFT-like 4×4 spatial histogram layout. The keypoint size determines the Gaussian weighting window through:

$$\texttt{hist\_width} = 3.0 \times \frac{\texttt{kp\_size}}{2} \tag{4.1}$$

Pixel contributions are weighted by $\exp\left(-\frac{r^2}{8}\right)$ where r is the distance from center in `hist_width` units. On a 65×65 patch, the edge is 32 pixels from center:

At `kp_size=41`, all pixels within the 65×65 patch receive nearly equal weight (0.97 at the edge). The 4×4 spatial grid still exists formally, but all pixels map into the central bins. HoNC effectively becomes a **global color histogram** of the entire patch.

Why global is better for fusion: HoNC with spatial structure (kp_size=12.26) partially duplicates what the CNN already captures—local spatial patterns. As a global histogram, HoNC instead captures the holistic color distribution, which is maximally complementary to the CNN's spatial-pattern matching. The keypoint size sweep confirms this:

As a standalone descriptor, kp_size=41 makes HoNC *worse* (12% vs 18%)—the global histogram loses spatial discrimination needed for matching. But in fusion with HardNet, it yields a +4% gain over the default, achieving the highest fusion score observed (50.0%). This inversion confirms that fusion benefits from component diversity: the CNN provides spatial matching, HoNC provides color discrimination, and the two are maximally complementary when their feature spaces do not overlap.

4.7.5 Why HoNC Adds Value

HoNC captures color information that gradient-based descriptors (SIFT, HardNet) discard:

- CNN descriptors are trained on grayscale patches, learning edge and texture patterns
- HoNC captures chromatic distinctiveness (e.g., red vs. blue regions)
- At kp_size=41 (Section 4.7.4), HoNC acts as a global color histogram, maximizing complementarity with CNN spatial features

- The combination provides both color discrimination and geometric/photometric invariance

This explains why HoNC+CNN matches or exceeds CNN+CNN on matching and retrieval despite HoNC alone being the second-weakest descriptor. The HoNC+HardNet fusion also shows that a descriptor's standalone performance is a poor predictor of its fusion value—what matters is orthogonality of the feature space.

4.8 Summary

Our experiments demonstrate four key conclusions, drawing on results from both evaluation pipelines:

Full-Image Pipeline Findings (Detector Effects):

1. **Scale Matters**: Filtering for larger scales improves performance by 13–18% across all descriptor types.

2. **Detector Consensus Matters**: Spatial intersection of distinct detectors acts as a quality filter, boosting mAP by 4–18%.

3. **CNN+CNN Fusion Succeeds**: HardNet+SOSNet concatenation achieves 93.4% mAP on intersection keypoints.

Patch Benchmark Findings (Descriptor Effects):

4. **Complementary Fusion Excels**: Pairing color descriptors (HoNC) with learned descriptors (CNN) achieves best patch matching and retrieval (50.0% mAP, 60.0% retrieval). The benefit comes from orthogonal feature spaces—color vs. spatial patterns.

5. **Traditional-Only Fusion Is Futile**: All SIFT-family combinations collapse to baseline performance ($\sim$23%), confirming that gradient-based descriptors capture redundant information and fusion cannot create discriminative power from correlated features.

6. **Descriptor Fusion Value $\neq$ Standalone Performance**: HoNC (18.5% alone) outperforms SIFT (22.9% alone) as a fusion partner. Tuning HoNC to act as a global color histogram (Section 4.7.4) further improves fusion by maximizing feature orthogonality, despite degrading standalone performance.

Table 4.9: Descriptor fusion results on color patch benchmark (manual NN matching)

Fusion	**Method**	**Dim**	**Matching**	**Verif.**	**Retrieval**
Complementary fusion (Discriminator + Matcher):					
HoNC* + HardNet	Concat	256	**50.0%**	77.5%	**60.0%**
HoNC + SOSNet	Concat	256	49.0%	77.5%	60.0%
HoNC + HardNet	Concat	256	49.0%	78.5%	59.0%
HoNC + SOSNet	Average	128	48.0%	77.5%	58.0%
HoNC + HardNet	Average	128	47.0%	77.5%	57.0%
Same-family fusion (Matcher + Matcher):					
HardNet + SOSNet	Concat	256	50.0%	**88.0%**	59.0%
HardNet + SOSNet	Average	128	47.0%	86.0%	55.0%
Cross-family fusion (Traditional + Matcher):					
RGBSIFT + SOSNet	Concat	512	46.0%	83.5%	54.0%
SIFT + SOSNet	Concat	256	45.0%	83.5%	53.0%
SIFT + HardNet	Concat	256	44.0%	83.0%	53.0%
SURF + SOSNet	Concat	192	41.0%	84.5%	50.0%
Traditional-only fusion (no improvement):					
SIFT + HoNC	Concat	256	23.0%	60.0%	31.0%
SIFT + RGBSIFT	Concat	512	23.0%	62.0%	31.0%
HoNC + SURF	Concat	192	19.0%	69.0%	31.0%

Table 4.10: Effect of keypoint size on HoNC spatial weighting (65×65 patches)

kp_size	**hist_width**	**Edge dist. (hw)**	**Edge weight**	**Effect**
5.84	8.8px	3.65	0.19	Strong spatial structure
12.26	18.4px	1.74	0.69	Moderate structure (SIFT default)
20.0	30.0px	1.07	0.87	Weak structure
41.0	**61.5px**	**0.52**	**0.97**	**Global histogram**

Table 4.11: HoNC keypoint size sweep: standalone vs. fusion with HardNet

kp_size	**HoNC alone**	**HoNC+HardNet (concat)**	**Δ vs kp=12.26**
5.84	12.0%	43.0%	−3.0%
12.26	**18.0%**	46.0%	baseline
20.0	18.0%	47.0%	+1.0%
41.0	12.0%	**50.0%**	+4.0%
55.0	10.0%	47.0%	+1.0%

Chapter 5
DISCUSSION

This chapter explains why the observed performance patterns occur and offers practical recommendations.

5.1 Why Scale Control Matters

Our experiments show that scale control yields large improvements: +39% relative for SIFT-family descriptors and +21% relative for CNN descriptors. We attribute this to several factors:

5.1.1 Information Content

Larger-scale keypoints capture patches containing more pixels:

- A 4-pixel scale keypoint samples approximately a 16×16 pixel region
- A 10-pixel scale keypoint samples approximately a 40×40 pixel region
- Larger regions contain more distinctive texture and edge information

5.1.2 Aliasing and Noise

Small-scale keypoints are more susceptible to [21]:

- **Aliasing**: High-frequency content that violates Nyquist sampling
- **Noise sensitivity**: Small patches have lower signal-to-noise ratio
- **Localization error**: Sub-pixel errors have greater relative impact

5.1.3 Practical Recommendation

For systems requiring high matching accuracy, we recommend filtering to the top 25% of keypoints by scale. The trade-off is reduced keypoint count (645K vs 2.5M in our experiments), but the quality gain is worth the coverage loss.

5.2 Understanding Fusion: Magnitude Matching

5.2.1 Pre-Fusion Normalization Requirement

Cross-family descriptor fusion (e.g., SIFT+CNN) requires matching descriptor magnitudes. Different descriptor families produce values at different scales:

Table 5.1: Descriptor magnitude characteristics

Descriptor	**Raw Range**	**After L2 Norm**
SIFT	[0, 512]	[0, 0.3]
HardNet/SOSNet	[-0.3, +0.3]	[-0.3, +0.3]
HoNC	[0, 1]	[0, 0.3]

Without normalization, larger-magnitude descriptors dominate L2 distance calculations. The solution is to L2 normalize each descriptor component *before* fusion:

$$d_{\text{fused}} = \text{fuse}\left(\frac{d_A}{\|d_A\|_2}, \frac{d_B}{\|d_B\|_2}\right) \tag{5.1}$$

With pre-fusion normalization, SIFT+HardNet achieves 46.0% mAP on the patch benchmark—comparable to other cross-family fusions.

5.2.2 Why Some Fusion Still Underperforms

Even with proper normalization, SIFT+CNN fusion (46.0%) underperforms HoNC+CNN fusion (50.6%) because:

1. SIFT and CNN both capture gradient/edge information (correlated)

2. HoNC captures color information that CNN lacks (complementary)

3. Complementary descriptors provide more benefit than similar ones

Fusion success depends on *complementarity*: descriptors that capture different information combine better than those capturing similar information.

5.3 Why CNN+CNN Fusion Succeeds

HardNet+SOSNet concatenation achieves 93.4% mAP on full images, improving upon either descriptor alone. We attribute this success to:

5.3.1 Magnitude Compatibility

Both descriptors are already L2-normalized during training:

- Similar value ranges ([-0.3, +0.3])
- Unit L2 norm (no magnitude mismatch)
- No pre-fusion normalization needed

This ensures equal contribution from each descriptor without additional processing.

5.3.2 Complementary Features

Despite similar training methodologies, HardNet and SOSNet learn slightly different representations:

- **HardNet** [26]: Trained with hard negative mining, focuses on discriminative features
- **SOSNet** [41]: Incorporates second-order similarity, captures different geometric relationships

Concatenation preserves both representations, allowing the matcher to use all available information.

5.3.3 Why Concatenation Outperforms Averaging

Concatenation (+0.5%) outperforms averaging (-0.6%) because:

1. Averaging collapses 256 dimensions of information into 128
2. Complementary features may be lost in averaging
3. Concatenation preserves full information from both descriptors

5.4 Detector Agreement as Quality Signal

Keypoints detected by both SIFT and KeyNet (spatial intersection) achieve higher performance than either detector alone:

- HardNet on full KeyNet: 64.5% mAP
- HardNet on intersection: 82.1% mAP (+17.6%)

We conducted a systematic investigation to determine whether this improvement stems from higher-quality *keypoints* or more distinctive *descriptors*.

5.4.1 Investigation Methodology

We tested five hypotheses for the intersection gain:

1. **Reduced count effect**: Fewer keypoints reduce false matches
2. **Quality selection**: Intersection selects stronger keypoints (higher response)
3. **Scale selection**: Intersection favors certain scale ranges

4. **Spatial distribution**: Intersection removes clustered keypoints

5. **Detector agreement**: Keypoints where detectors agree are more likely to be distinct structures.

We created control keypoint sets: random subsets, top-N by response, top-N by scale, and spatially filtered sets all matched to the intersection count (645K keypoints).

5.4.2 Key Finding: Keypoint Quality, Not Descriptor Distinctiveness

The investigation revealed that the improvement comes from **keypoint quality**, not descriptor properties:

- **Random subsets do not match intersection performance**: Simply reducing keypoint count does not explain the gain. Random subsets of equal size perform worse than intersection sets.

- **Scale filtering provides the largest single gain**: Filtering to large-scale keypoints (top 25% by size) improved SIFT by 50% (42.6% $\rightarrow$ 63.9%) and HardNet by 20% (65.8% $\rightarrow$ 78.9%).

- **Intersection provides additional gain beyond scale**: HardNet on intersection (82.1%) exceeds HardNet on scale-controlled sets (78.9%), indicating intersection captures quality beyond scale alone.

5.4.3 Why Detector Agreement Indicates Quality

The intersection filters for keypoints that are *salient image features* rather than detector-specific artifacts:

- **Cross-validation**: If both a blob detector (KeyNet) and an edge/corner detector (SIFT) independently identify the same location as interesting, it likely corresponds to a real image structure.

- **Repeatability**: Keypoints detected by multiple methods are more likely to be detected again under viewpoint/illumination changes they are inherently more repeatable.

- **Noise rejection**: Detector-specific noise or false detections are unlikely to occur at the same spatial location across different detection methods.

5.4.4 Critical Distinction

The improvement is not because descriptors computed on intersection keypoints are inherently more distinctive. The same HardNet model produces the same descriptor for a given image patch regardless of how the patch location was selected.

Rather, the improvement occurs because:

1. Intersection keypoints correspond to more repeatable image structures

2. These structures are more likely to be detected in both reference and target images

3. Higher repeatability leads to more correct correspondences being available for matching

This finding has practical implications: keypoint selection strategy can matter as much as descriptor algorithm choice. The 17% gain from intersection filtering is comparable to the gap between SIFT and CNN descriptors.

5.5 HP-V vs HP-I Patterns

5.5.1 Traditional vs Learned Preferences

Our results show opposite patterns:

- **Traditional descriptors**: HP-V > HP-I (better on viewpoint changes)

- **CNN descriptors**: HP-I > HP-V (better on illumination changes)

5.5.2 Explanation

CNN descriptors are trained on patches with photometric augmentations (brightness, contrast, gamma), leading to learned illumination invariance [26, 41]. Traditional descriptors rely on gradient directions, which are naturally invariant to multiplicative illumination changes but not to more complex photometric transformations [21].

5.6 Limitations

5.6.1 Dataset Scope

Our experiments use only the HPatches benchmark. While HPatches is a standard evaluation dataset, results may not generalize to:

- Extreme viewpoint changes (> 60 degrees)
- Significant scale differences between images
- Different image domains (medical, satellite, etc.)

5.6.2 Computational Considerations

Concatenation doubles descriptor dimensionality (128D to 256D), increasing:

- Memory requirements
- Matching time (linear in dimensionality for brute-force)
- Index size for approximate nearest neighbor methods

However, the dominant factor in total pipeline cost is *keypoint count*, not descriptor dimensionality. On the full HPatches benchmark, HardNet on the full KeyNet set ($\sim$2.8M keypoints) requires 219 seconds, of which 134 seconds is matching. The HardNet+SOSNet

concatenation on the scale-matched intersection (∼111K keypoints) completes in 7.7 seconds total—a 28× speedup—because the intersection reduces keypoint count by 96%, which reduces matching time quadratically (brute-force). The concatenated 256D descriptor adds only 0.1 seconds of matching overhead compared to the 128D averaged variant (0.5s vs. 0.3s).

For real-time applications, the overhead of concatenation is minor compared to keypoint filtering gains. The bottleneck is descriptor extraction, not matching.

5.6.3 Detector Dependency

Our best results use KeyNet for CNN descriptors. The findings may not transfer to other detector choices (e.g., SuperPoint's built-in detector).

5.7 Summary of Findings

1. **Scale matters more than descriptor choice**: A 50% relative improvement from scale control (SIFT: 42.6% → 63.9%) exceeds most algorithmic improvements

2. **Keypoint selection matters as much as descriptor algorithm**: The 17% gain from detector intersection is comparable to the gap between traditional and CNN descriptors

3. **Detector agreement signals keypoint quality, not descriptor distinctiveness**: Intersection keypoints are more repeatable the descriptors themselves are not more distinctive, but the underlying image structures are more reliably detected across images

4. **Complementary descriptors outperform similar ones**: HoNC+CNN fusion (50.6% on patches) beats CNN+CNN (49.9%) because color and learned features capture different information

5. **Magnitude matching enables cross-family fusion**: Pre-fusion L2 normalization is required for SIFT+CNN fusion

6. **Concatenation outperforms averaging**: Preserving full information outperforms lossy aggregation across all successful fusions

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Summary of Contributions

This thesis investigated detector consensus and descriptor fusion for local feature matching, with the following contributions:

1. **Detector Intersection as Quality Filter**: Keypoints detected by multiple distinct detectors are more distinctive than those found by any single detector. This effect holds for both SIFT-SURF and SIFT-KeyNet intersections, with HardNet achieving 82.1% mAP on intersection keypoints—a 25% relative improvement.

2. **Color HPatches Benchmark**: We created a color version of the HPatches patch benchmark by re-extracting 65×65 color patches from original images. This enables evaluation of color descriptors like HoNC.

3. **Magnitude Matching for Cross-Family Fusion**: Cross-family fusion (SIFT+CNN) requires pre-fusion L2 normalization to ensure equal contribution from each descriptor. With proper normalization, SIFT+HardNet achieves 46.0% mAP on patches.

4. **Complementary Descriptor Fusion**: Pairing descriptors with different strengths HoNC (high verification) with CNN (high matching)—yields improvements. HoNC + SOSNet achieves 50.6% mAP on color patches, outperforming all individual descriptors.

5. **Scale Control Methodology**: Filtering keypoints by scale yields large improvements: +39% relative for SIFT-family descriptors and +21% relative for CNN descriptors.

6. **DescriptorWorkbench Framework**: We developed an open-source evaluation framework implementing image matching, keypoint verification, and keypoint retrieval metrics from Bojanic et al. [7], supporting both full-image and patch-based evaluation.

6.2 Key Findings

Our experiments lead to several practical observations:

- **Quality over quantity**: Fewer, better keypoints (scale-filtered or intersection-filtered) outperform larger sets of lower-quality keypoints.

- **Detector agreement provides quality signal**: Keypoints detected by multiple distinct methods are more reliable, with 17–25% improvements from intersection filtering.

- **Magnitude matching enables cross-family fusion**: Pre-fusion L2 normalization allows successful SIFT+CNN combination by ensuring equal contribution from each descriptor.

- **Complementary descriptors outperform similar ones**: HoNC+CNN fusion outperforms CNN+CNN or SIFT+SIFT because the descriptors capture different information (color vs. learned features).

- **Concatenation preserves information**: Concatenation consistently outperforms averaging by preserving complementary features.

6.3 Practical Recommendations

Based on our findings, we offer the following recommendations:

1. **For best patch matching**: Use HoNC+SOSNet concatenation (50.6% mAP), leveraging color discrimination with learned matching.

2. **For best full-image matching**: Use HardNet on detector intersection keypoints (82.1% mAP), or HardNet+SOSNet concatenation (93.4% mAP) for highest accuracy.

3. **For cross-family fusion**: Always L2 normalize each descriptor component before fusion to ensure equal contribution.

4. **For traditional descriptors**: Apply scale filtering (top 25%) and use intersection keypoints (64–75% mAP vs 42% baseline).

6.4 Future Work

Several directions remain for future investigation:

6.4.1 Learned Fusion Weights

Rather than fixed $\alpha = 0.5$ weighting, learn optimal weights:

$$d_{\text{fused}} = \sum_i w_i \cdot d_i, \quad \text{s.t.} \sum_i w_i = 1 \tag{6.1}$$

Weights could be learned per-dimension or globally, potentially improving cross-family fusion further.

6.4.2 Additional Datasets

Validate findings on:

- Oxford5k [32] and Paris6k [33] (image retrieval)
- MegaDepth [20] (wide-baseline matching)
- ETH3D [36] (multi-view stereo)

6.4.3 End-to-End Learning

Train a joint detector-descriptor-fusion network that:

- Detects keypoints with scale optimization
- Extracts multiple descriptor types
- Learns optimal fusion strategy

6.4.4 Tolerance Sensitivity

Systematically study the relationship between intersection tolerance and matching performance. Our hypothesis is an inverted U-curve: too strict yields few keypoints, too loose yields misaligned pairs.

6.5 Closing Remarks

This thesis demonstrates two complementary approaches to improving local feature matching. First, using multiple detectors as a consensus filter identifies high quality keypoints that are more distinctive and repeatable than those found by any single detector. Second, fusing complementary descriptors particularly color descriptors like HoNC with learned descriptors like SOSNet captures information that neither descriptor provides alone.

A key technical finding is that cross-family fusion (SIFT+CNN) requires proper magnitude matching through pre-fusion L2 normalization. The initial fusion failures were not due to fundamental incompatibility between descriptor families, but rather a fixable magnitude mismatch problem.

Our findings also suggest that keypoint quality deserves more attention. The 39% improvement from scale control and 25% improvement from detector consensus exceed many algorithmic advances, yet these filtering strategies are rarely discussed in the literature. We hope this work encourages more research into keypoint selection strategies alongside descriptor algorithm development.

The DescriptorWorkbench framework, color HPatches benchmark, and all experimental configurations are available as open-source software, enabling reproduction and extension of these results.

BIBLIOGRAPHY

[1] Anthropic. Claude. Large language model, 2025. Accessed: 2026. Available: `https://claude.ai`.

[2] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012.

[3] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5173–5182, 2017.

[4] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, volume 1, page 3, 2016.

[5] Vassileios Balntas, L. Tang, and Krystian Mikolajczyk. PN-Net: Conjoined triple deep network for learning local image descriptors. In *arXiv preprint arXiv:1601.05030*, 2016.

[6] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned CNN filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5836–5844, 2019.

[7] Kristijan Bartol, David Bojanić, Tomislav Pribanić, Tomislav Petković, Yago Diez Donoso, and Joaquim Salvi Mas. On the comparison of classic and deep keypoint detector and descriptor methods. *arXiv preprint arXiv:2007.10000*, 2020.

[8] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[9] Fabio Bellavia. RootsGLOH2: embedding RootSIFT 'square rooting' in sGLOH2. *IEEE Signal Processing Letters*, 29:1749–1753, 2022.

[10] Ondrej Chum, Andrej Mikulik, Michal Perdoch, and Jiri Matas. Total recall: Automatic query expansion with a generative feature model for object retrieval. *International Journal of Computer Vision*, 93(1):2–31, 2011.

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[12] Jingming Dong and Stefano Soatto. Domain-size pooling in local descriptors: DSP-SIFT. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5097–5106, 2015.

[13] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint description and detection of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019.

[14] Benoit Gallet and Michael Gowanlock. Computing Double Precision Euclidean Distances using GPU Tensor Cores. *arXiv preprint arXiv:2209.11287*, 2022.

[15] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, volume 25, 2012.

[17] Haeseong Lee, Semi Jeon, Inhye Yoon, and Joonki Paik. Recent advances in feature detectors and descriptors: A survey. *IEIE Transactions on Smart Processing and Computing*, 5(3):153–163, 2016.

[18] JongMin Lee, Eunhyeok Park, and Sungjoo Yoo. Multi-Scale Local Implicit Keypoint Descriptor for Keypoint Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6144–6153, 2023.

[19] Qi Li, Vojislav Kecman, and Raied Salman. A chunking method for Euclidean Distance Matrix Calculation on large dataset using Multi-GPU. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 208–213. IEEE, 2010.

[20] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[21] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[22] J. Luo et al. A comparative analysis of RootSIFT and SIFT methods for drowsy features extraction. *IEEE Access*, 7:64236–64243, 2019.

[23] Jiayi Ma, Xingyu Jiang, Amlan Fan, Junjun Jiang, and Junchi Yan. Local feature descriptor for image matching: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):649–663, 2021.

[24] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 257–263. IEEE, 2003.

[25] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005.

[26] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[27] Naila Murray and Florent Perronnin. Generalized max pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2473–2480, 2014.

[28] Clark F Olson, Sam A Hoover, Jordan L Soltman, and Siqi Zhang. Complementary keypoint descriptors. In *Advances in Visual Computing: 12th International Symposium, ISVC 2016, Las Vegas, NV, USA, December 12-14, 2016, Proceedings, Part I 12*, pages 341–352. Springer, 2016.

[29] Clark F Olson and Siqi Zhang. Keypoint recognition with histograms of normalized colors. In *2016 13th Conference on Computer and Robot Vision (CRV)*, pages 311–318. IEEE, 2016.

[30] Yuki Ono, Eduard Trulls, Vincent Lepetit, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In *Advances in neural information processing systems*, volume 31, 2018.

[31] Zizheng Pan, Bohyung Bohg, Jeannette Niebles, and Animesh Garg. Scalable vision transformers with hierarchical pooling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 377–386, 2021.

[32] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[33] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[34] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. A survey on Kornia: an open source differentiable computer vision library for PyTorch. *arXiv preprint arXiv:2009.10521*, 2020.

[35] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[36] Thomas Schöps, Johannes L Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017.

[37] Hao Shao, Tomáš Svoboda, and Luc Van Gool. Zubud-zurich buildings database for image based recognition. *Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep*, 260(20):6, 2003.

[38] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.

[39] Shaharyar Kamal K. Tareen and Zahra Saleem. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. *International conference on computing, mathematics and engineering technologies (iCoMET)*, pages 1–10, 2018.

[40] Yurun Tian, Bin Fan, and Fuchao Wu. L2-Net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub He, and Yuchao Jia. SOSNet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019.

[42] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *European conference on computer vision*, pages 467–483. Springer, 2016.

[43] H. Yin et al. Kernel Pooling for Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[44] Le You, Han Jiang, Jinyong Hu, C Hwa Chang, Lingxi Chen, Xintong Cui, and Mengyang Zhao. GPU-accelerated faster mean shift with Euclidean Distance Metrics. In *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 211–216. IEEE, 2022.

[45] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Chapman. 3DMatch: Learning local geometric descriptor from RGB-D reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017.

[46] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. ALIKED: A lighter Keypoint and Descriptor Extraction Network via Deformable Transformation. *IEEE Transactions on Instrumentation and Measurement*, 2023.