Supplementary Material: Discriminative Keypoint Detection via eCNN and SSIM-Based Triplet Loss

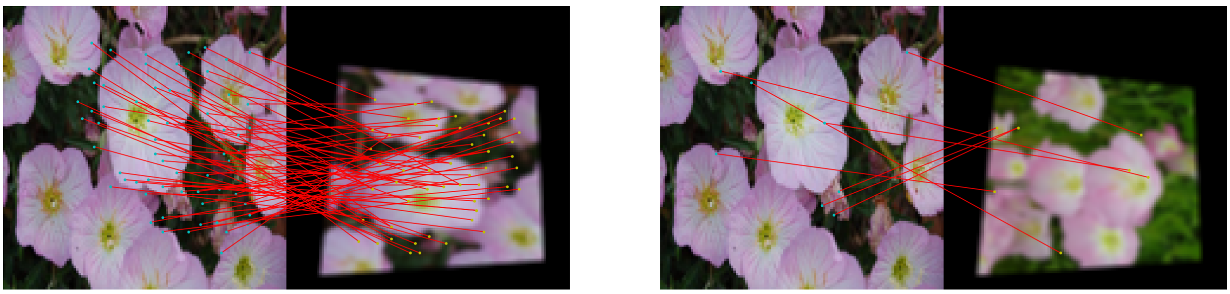
This supplementary material provides extended insights into the paper titled *"Optimizing Image Identification: Discriminative Keypoint Detection with Equivariant CNN and SSIM-Based Triplet Loss."* In addition to presenting further experimental results that illustrate the method's robustness under diverse photometric and geometric transformations, we include visual examples of challenging test cases to better contextualize the identification problem. We also expand on the theoretical background, focusing on the role of equivariance and keypoint repeatability in the proposed pipeline. Furthermore, we delve deeper into the *Keypoint Matching and Descriptor Evaluation* step, which is crucial for validating the discriminative power of our method and quantifying performance improvements over traditional local techniques.

The source code and supplementary materials for this work are publicly available at `https://github.com/binarycode11/singular-points`. The repository contains the full implementation, usage examples, and instructions for reproducing the experiments detailed in the paper. We encourage readers to explore the repository for a deeper understanding of the method and to facilitate reproducibility of the results.

1 Visual Evaluation of Keypoint Matching

To qualitatively evaluate the effectiveness of our keypoint matching strategy, we constructed three custom datasets based on distinct visual categories: flowers, wood textures, and textile fibers. Each dataset contains two types of image pairs:

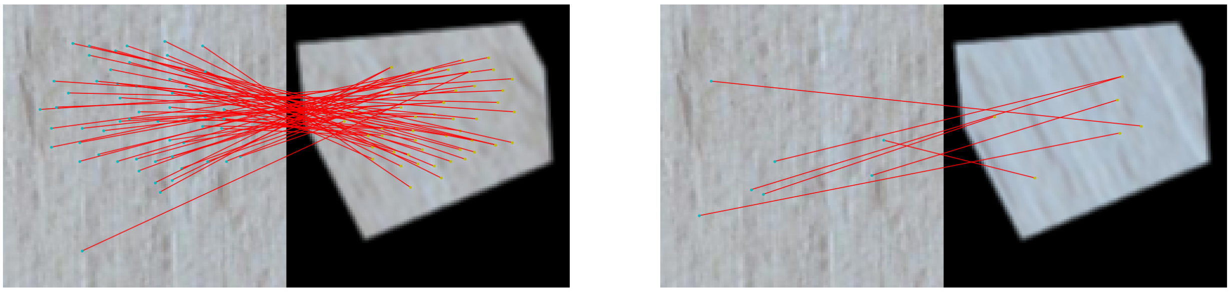
- **Positive Pair (Same)**: Both subimages are extracted from the same original image and depict the same visual content, with controlled geometric transformations such as rotation, scaling, or homography.
- **Negative Pair (Similar)**: Subimages come from different source images, selected for their high visual similarity (e.g., similar texture, color, or structure), but representing distinct content. These are used to test the method's robustness against false correspondences.



(a) Same (b) Similar

Figure 1: Qualitative matching examples from the Flowers Dataset.

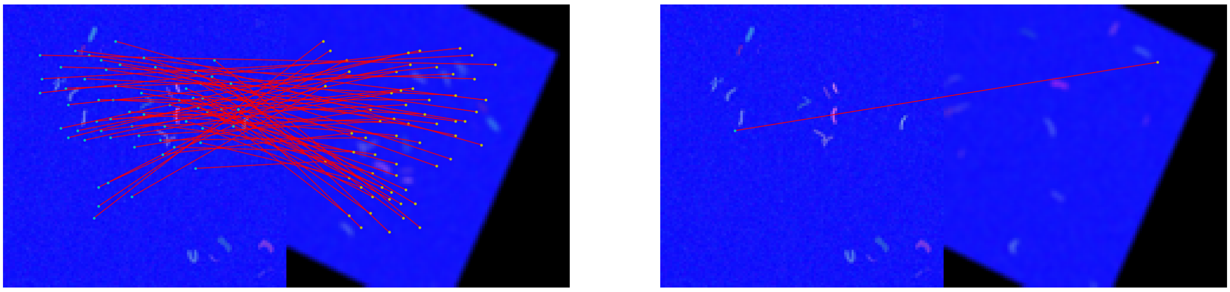
Figure 1 illustrates examples from the Flowers dataset, where each image juxtaposes subimages from a reference and a query, with overlaid keypoint correspondences. In subfigure (a), the pair originates from the same flower under a geometric transformation, resulting in a dense set of accurately aligned keypoints and a high similarity measure—**58 correct correspondences**. Conversely, subfigure (b) depicts a visually similar yet semantically different flower pair, with only **8 correspondences**, demonstrating the model's ability to prioritize local features with high discriminative power over mere visual resemblance.



(a) Same (b) Different

Figure 2: Qualitative matching examples from the Wood Texture Dataset.

Extending this evaluation to the Wood Texture dataset in Figure 2, we observe a similar trend. The positive pair in subfigure (a), derived from the same wood sample with geometric variation, yields **59 correct matches**, indicating strong agreement at the local feature level. Meanwhile, subfigure (b) contrasts this with a visually alike but distinct wood sample pair, where correspondences sharply decline to **7**, confirming the method's effectiveness in discerning subtle local differences despite comparable texture and grain patterns.



(a) Same (b) Different

Figure 3: Qualitative matching examples from the Fiber Texture Dataset.

Finally, Figure 3 presents matching results on the Paper Security Fiber Dataset, which features highly intricate fiber patterns critical for authentication. Here, the positive pair in subfigure (a) achieves a notable similarity score of **60**, while the visually similar but different fiber pair in subfigure (b) scores only **1**. This highlights the method's robustness in capturing subtle yet essential local variations within highly repetitive and complex textures, where traditional descriptors tend to fail.

These visual analyses indicate that our method cannot be categorized as a simple similar image retrieval system based solely on pattern recognition. By effectively leveraging local features with high discriminative capacity alongside equivariant representations, the approach excels at capturing subtle and distinctive details. These discriminative characteristics enable more precise and unambiguous identification of individual images.

2 Keypoint Matching

This section details the process of **keypoint matching** which is an important aspect of the whole solution for image identification.

Each selected keypoint must be associated with a **compact descriptor** that characterizes its region discriminatively, i.e., maintaining the discriminative nature of the features around it, effectively encapsulating the most important visual features. These descriptors are essential for calculating the best **local descriptor correspondence** between two images based on their Euclidean distance.

2.1 Descriptor Pre-processing

The **descriptor pre-processing** stage is part of the process necessary to assess the keypoint detectors.To make the matching process, we need to represent the features around the detected keypoints in a compact way using an appropriate descriptor. However, we must apply some preliminary steps before computing such a descriptor. The first is retrieving bounding boxes $B(K_i)$ containing each keypoint K_i. For each bounding box $B(K_i)$, we define an inner circular mask C that yields a symmetrical content $C(B(K_i))$ feeding a descriptor capable of accommodating rotation-symmetry. This process operates on the original image domain.

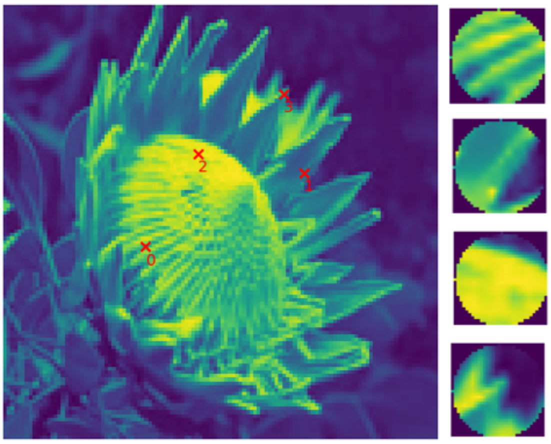


Figure 4: In the left image, the location of 4 keypoints (K_i) are highlighted. In the right image, it is possible to visualize the pre-processing, with bounding boxes and internal circular masks ($C(B(K_i))$) associated with each identified keypoint (K_i).

2.2 Descriptor Computation

After the descriptor pre-processing step we compute a descriptor $D(C_i)$ for each symmetrical masked bounding box $C_i = C(B(K_i))$. Any descriptor method can be used to compute the feature descriptor. One crucial point is that the way we do this process is independent of the chosen descriptor. Consequently, we can plug different descriptors into the pipeline, including SIFT, HardNet, SOSNet, and HyNet. By incorporating these descriptors, we can obtain insights on how each descriptor behaves when applied to the keypoints detected in the keypoint detection stage. From now on, we denote $D(K_i(I)) = D(C_i)$ and use $D(K(I))$ to represent the set of descriptors computed for all keypoints detected on a given image I.

2.3 Local Descriptor Matching

Given two images I and I', the **Local Descriptor Matching** stage, compares the set of descriptors $D(K(I))$ and $D(K(I'))$ where $K(I)$ and $K(I')$ are the detected keypoints respectively in I and I', using some metric dm. Here, for simplicity, dm is defined as the Euclidean distance. We opted for this simplistic choice because our main purpose is not to evaluate the descriptors but the detectors.

A point-to-point search is performed between the artifacts, trying to match the $n = |K(I)|$ point descriptors of I with the $m = |K(I')|$ point descriptors belonging to I'. A match between two descriptors $D(K_i(I))$ and $D(K_j(I'))$ happens when $dm(D(K_i(I)), D(K_j(I'))) < \kappa$ where κ is the **matching similarity threshold**. The matching process utilizes a bidirectional brute-force approach, identifying the optimal pairs of points in both directions.

Image similarity is measured based on the matching score $MS(U, V)$ (equation 1), which counts the number of correspondences between the set of descriptors $U = D(K(I))$ and $V = D(K(I'))$ computed for the keypoints detected on two images I and I'. If the number of valid correspondences is below a given threshold η, the matching score is considered zero:

$$
\begin{aligned}
&MS(U,V) = \begin{cases} |S|, |S| >= \eta \\ \quad 0, |S| < \eta \end{cases}, \\
&S = \{s = (u,v) | u \in U, v \in V \wedge dm(u,v) < \kappa\}
\end{aligned} \tag{1}
$$