LF-Net: Learning Local Features from Images

Yuki Ono[1], Eduard Trulls[2], Pascal Fua[2], and Kwang Moo Yi[3]

[1] Sony Imaging Products & Solutions Inc. [2] École Polytechnique Fédérale de Lausanne [3] University of Victoria

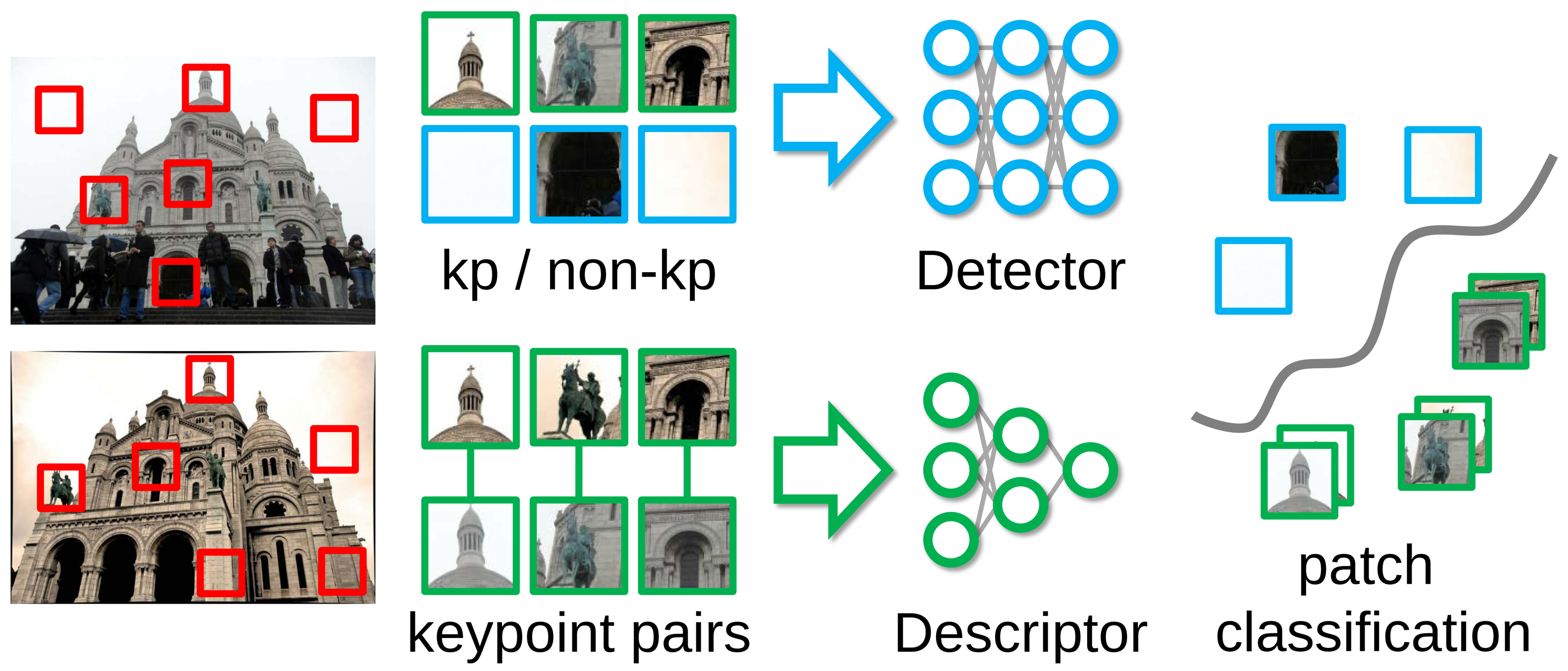
Motivation & Contributions

- We present LF-Net: a novel strategy to learn a deep network for **end-to-end local feature extraction pipeline** (keypoints and descriptors) from raw images, from scratch.
- To do so, we propose to break the **differentiability constraint** present in Siamese networks, using the outputs of one network to paint a virtual target for the other.
- Ground truth (camera calibration, depth) from noisy depth sensors or off-the-shelf SfM, **without human intervention.**
- State of the art on wide-baseline stereo, running at **60+ fps** for QVGA images. Code and models available.

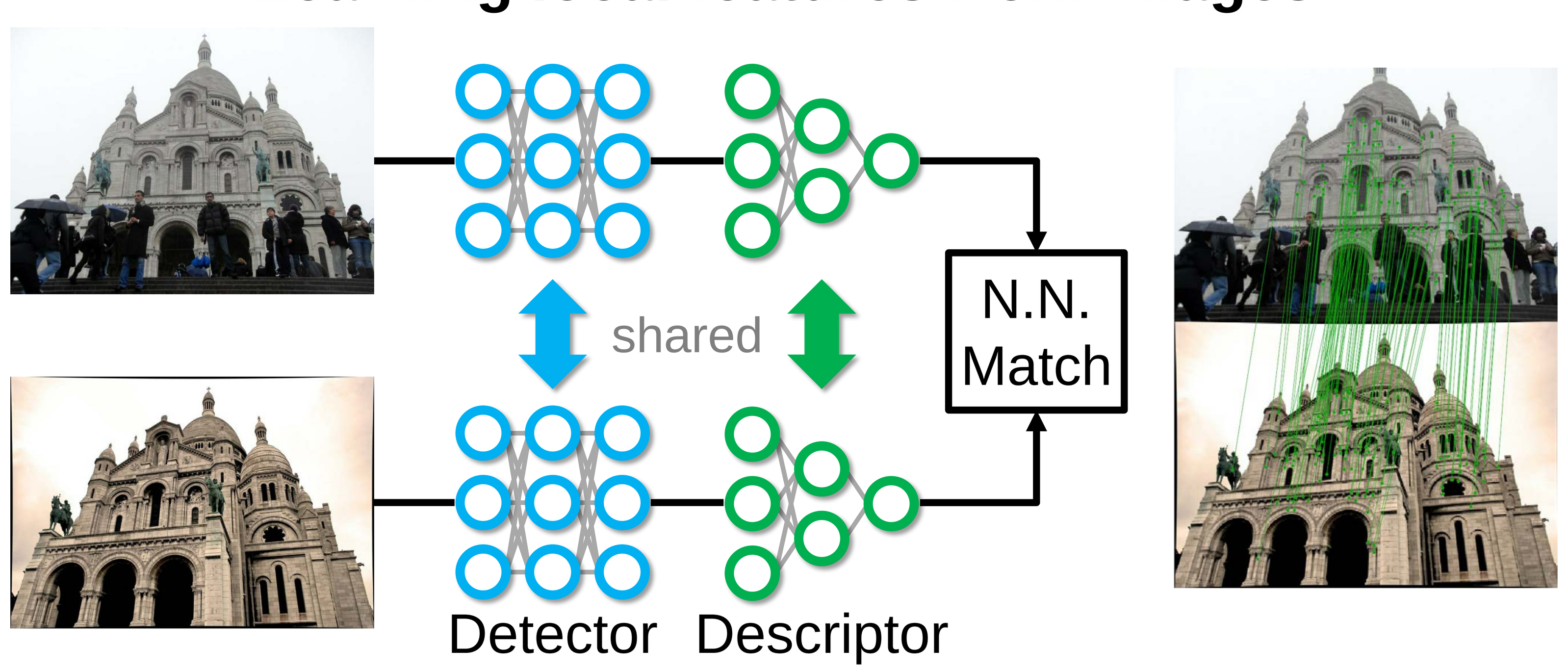
Training from whole images

- Previous end-to-end methods (LIFT) learn from SIFT matches, which upper-bounds the performance of the keypoint detector.
- We leverage the whole image canvas and learn the optimal keypoints, along with their associated descriptors.
- The challenge: we need **positive matches** to train! We show how to do this by enforcing a match for each keypoint, in a non-differentiable way.

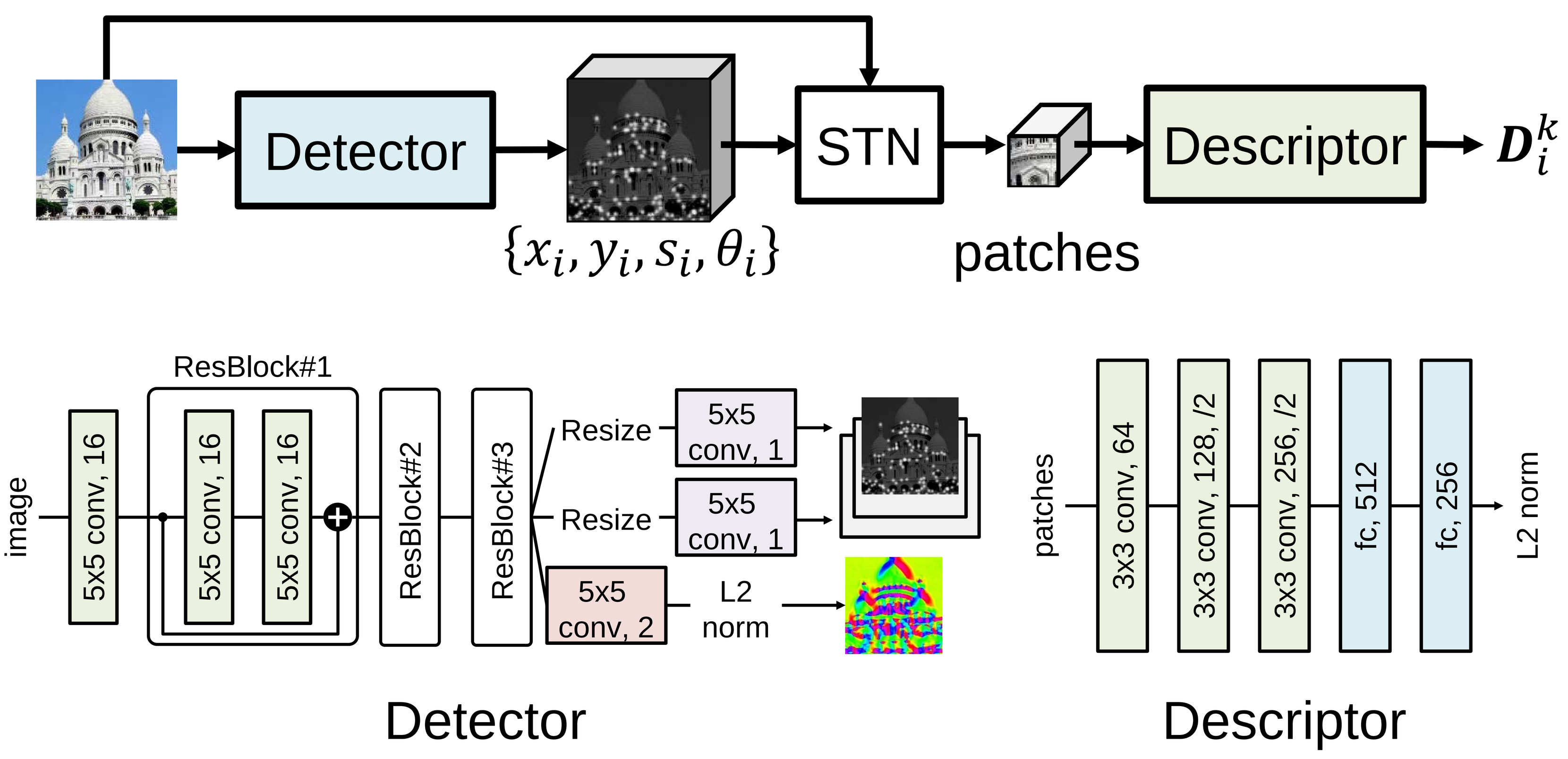
Learning local features from patches



Learning local features from images

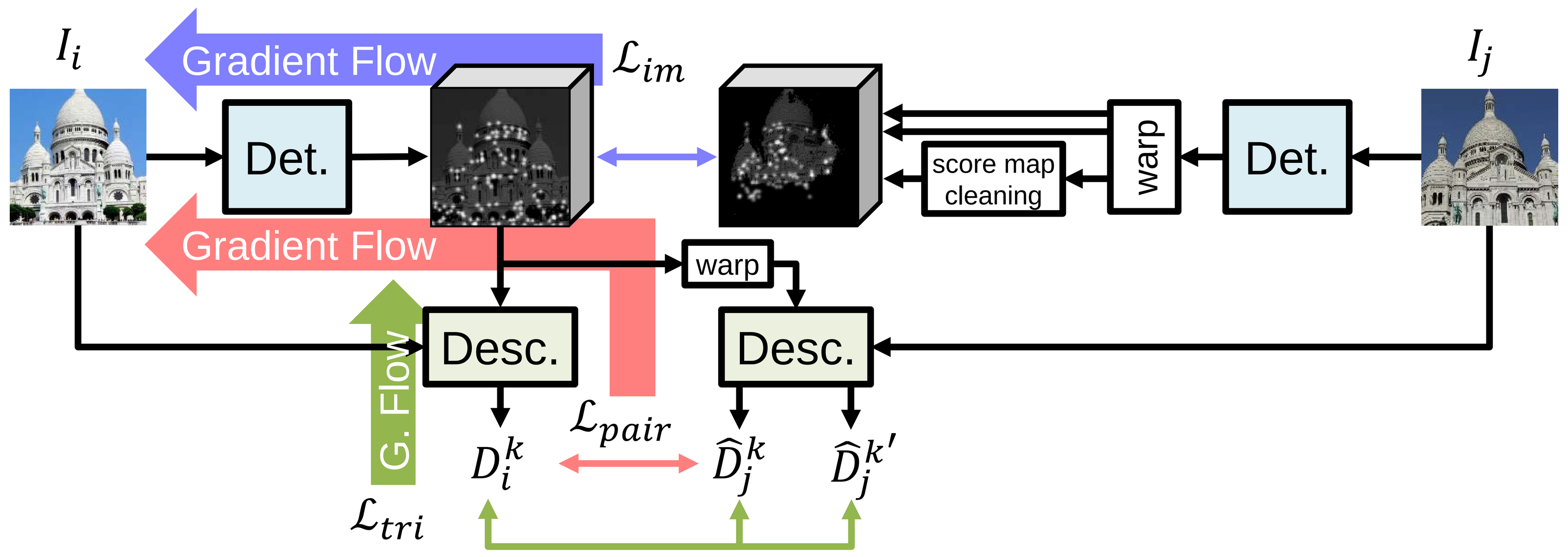


The Local Feature Network: LF-Net



- A fully-convolutional ***detector*** network, which outputs scale-space score maps with an orientation for every pixel.
- Spatial Transformers to sample patches around keypoints.
- Patches are fed to the ***descriptor*** network, which outputs a 256-D feature vector for every keypoint.

Training with two LF-Nets



- **Two copies** of the network, processing two different views.
- Branch *j* (right) is used to generate a **supervision signal** for branch *i* (left), in a **non-differentiable manner.** We optimize over branch *i* and re-use weights for branch *j*.
- Detector: select *K* points from score-map, build **sharp map for *j*, enforce *i* to be similar**. Descriptor: warp selected keypoint locations to **guarantee correspondences *i* to *j***.

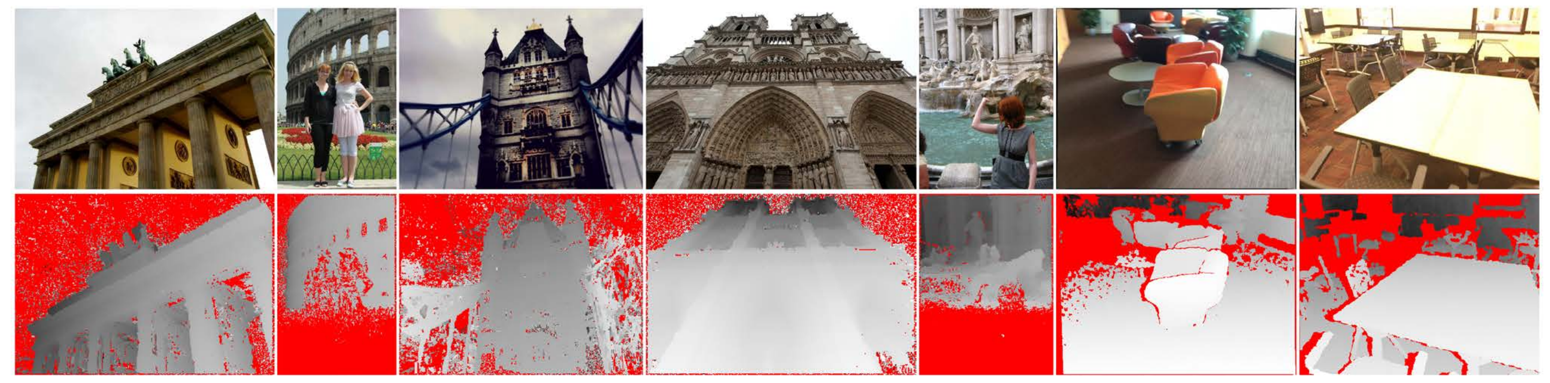
Loss Functions

- Detector: $\mathcal{L}_{det} = \underbrace{\mathcal{L}_{im} + \lambda_{ori}\mathcal{L}_{ori} + \lambda_s \mathcal{L}_s}_{\text{L2 loss between 2 views}} + \underbrace{\lambda_{pair}\mathcal{L}_{pair}}_{\text{Matching patch distances}}$
- Descriptor:

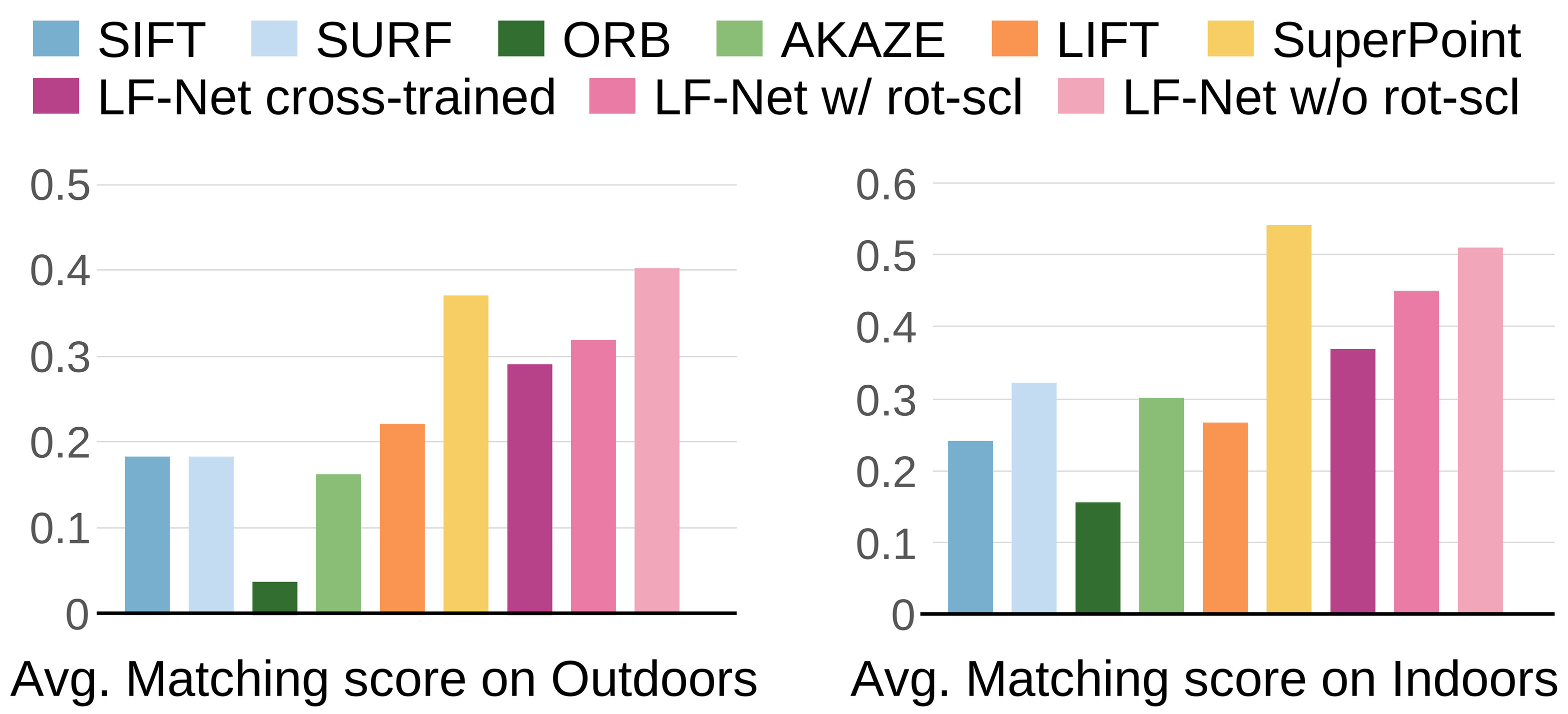
$$\mathcal{L}_{desc} = \mathcal{L}_{tri} = \sum_k \max\left(0, \left|D_i^k - \hat{D}_j^k\right|^2 - \left|D_i^k - \hat{D}_j^{k'}\right|^2 + C\right)$$

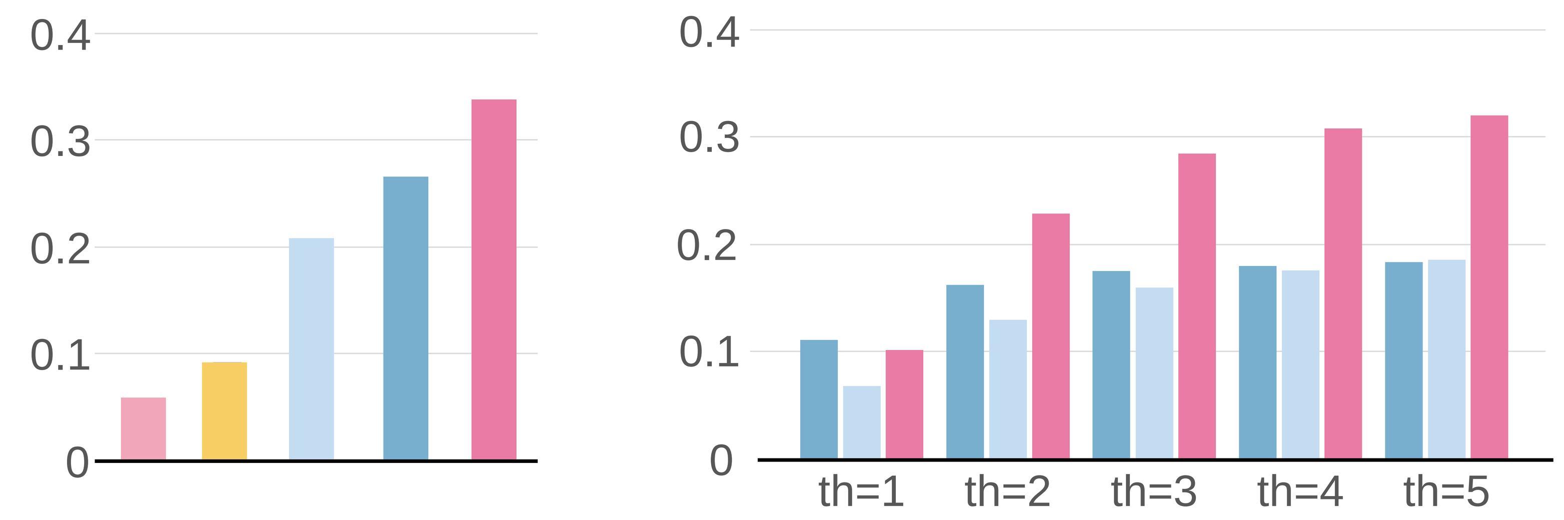
Evaluation

- **Datasets:** Outdoors (YFCC100M) and Indoors (ScanNet).



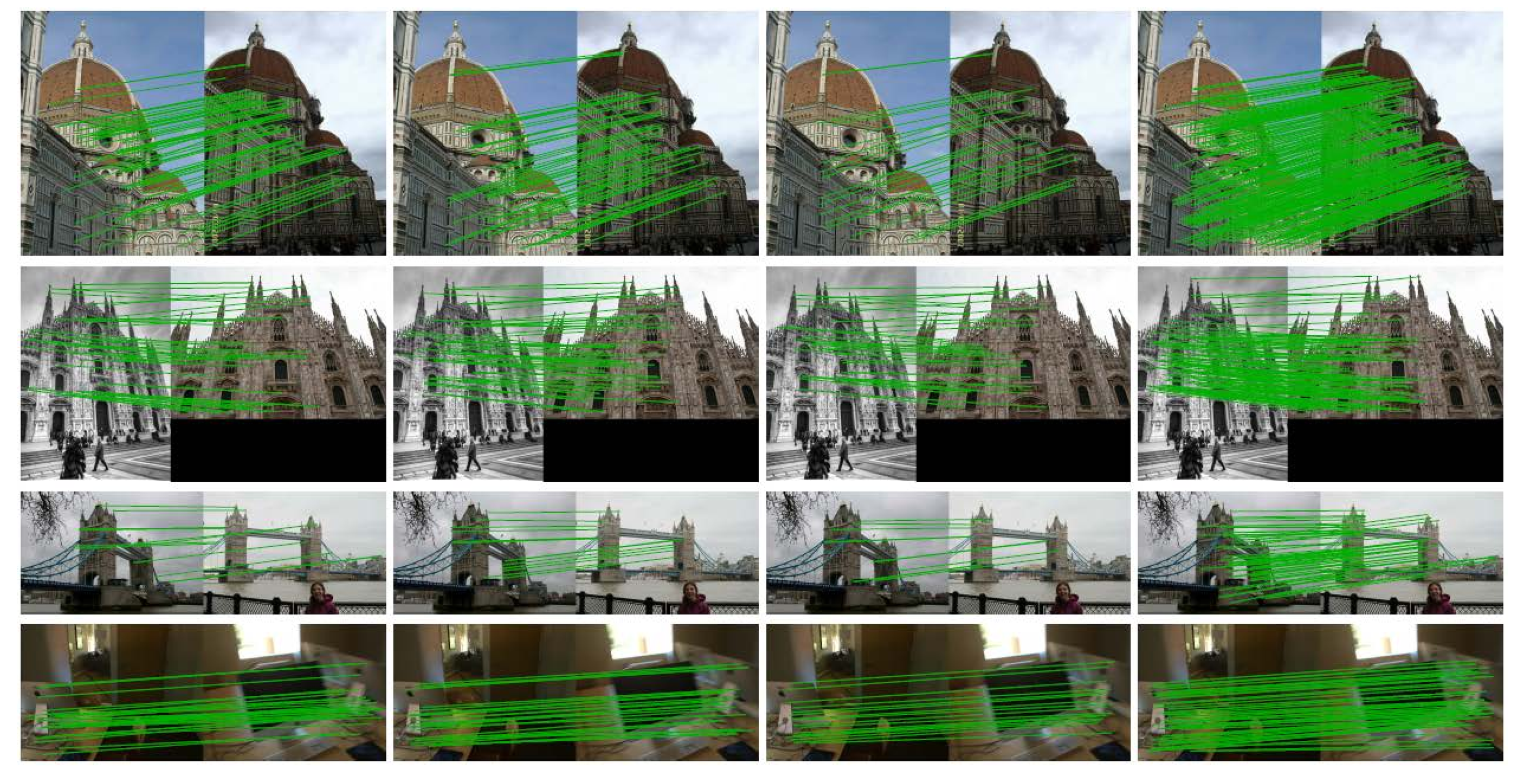
- **Ground truth:** Camera intrinsics / extrinsics from SfM and depth from SfM or sensors. Noisy, but sufficient for training!
- **Metrics:** matching score (% of correspondences we can recover with nearest-neighbor matching).
- **Results:** SoA outdoors, close to SoA indoors.





Avg. Matching score on rotated outdoor dataset

Avg. Matching score on Outdoors with different thresholds



(a) SIFT (b) SURF (c) A-KAZE (d) LF-Net (ours)