NEURAL OUTLIER REJECTION FOR SELF-SUPERVISED KEYPOINT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating reliable illumination and viewpoint invariant keypoints is critical for tasks such as feature-based SLAM and SfM. Recently, many learned keypoint methods have demonstrated improved performance on challenging benchmarks. However, it is extremely difficult to create consistent training samples for interest points in natural images, since they are hard to define clearly and consistently for a human annotator. In this work, we propose a novel end-to-end self-supervised learning scheme that can effectively exploit unlabeled data to provide more reliable keypoints under various scene conditions. Our key contributions are (i) a novel way of regressing keypoints, which avoids discretization errors introduced by related methods; (ii) a novel way of extracting associated descriptors by means of an upsampling step, which allows regressing the descriptors with a more fine-grained detail for the per-pixel level metric learning and (iii) a novel way of training the descriptor by using a proxy task, i.e. neural outlier rejection. By using this proxy task we can derive a fully self-supervised training loss for the descriptor, thus avoiding the need for manual annotation. We show that these three contributions greatly improve the quality of feature matching and homography estimation on challenging benchmarks over the state-of-the-art.

1 INTRODUCTION

Detecting interest points in RGB images and matching them across views is a fundamental capability of many robotic system. Tasks such Simultaneous Localization and Mapping (SLAM) (Cadena et al., 2016), Structure-from-Motion (SfM) (Agarwal et al., 2010) and object detection assume that salient keypoints can be detected and re-identified in a wide range of scenarios, which requires invariance properties to lighting effects, viewpoint changes, scale, time of day, etc. However, these tasks still mostly rely on handcrafted image features such as SIFT (Lowe et al., 1999) or ORB (Rublee et al., 2011), which have been shown to be limited in performance when compared to learned alternatives (Balntas et al., 2017).

Deep learning methods have revolutionized many computer vision applications including 2D/3D object detection (Lang et al., 2019; Tian et al., 2019), semantic segmentation (Li et al., 2018; Kirillov et al., 2019), human pose estimation (Sun et al., 2019), etc. However, most learning algorithms need supervision and rely on labels which are often expensive to acquire. Moreover, supervising interest point detection is unnatural, as a human annotator cannot readily identify salient regions in images as well as key signatures or descriptors, which would allow their re-identification. Self-supervised learning methods have gained in popularity recently, being used for tasks such as depth regression (Guizilini et al., 2019), tracking (Vondrick et al., 2018) representation learning (Wang et al., 2019; Kolesnikov et al., 2019). Following DeTone et al. (2018b) and Christiansen et al. (2019), we propose a self-supervised methodology for jointly training a keypoint detector as well as its associated descriptor. Furthermore, we introduce a novel neural outlier rejection scheme (Brachmann & Rother, 2019), in the form of an additional discriminative network that learns to generate optimal inlier sets from possible corresponding point-pairs. Even though trained in a self-supervised manner without any direct supervision, this network can effectively learn distinguishable features for homography estimation, leading to a better gradient flow between consistent point-pairs.

Our main contributions are: (i) a baseline architecture that achieves better performance than existing self-supervised/unsupervised keypoint detection and description methods; (ii) a novel self-

supervised learning scheme which cascades an additional outlier rejection network to improve training performance; and (iii) we show that the combination of both establishes a new state-of-the-art performance for the task of self-supervised keypoint detection and descriptor learning.

2 Related Work

The recent success of deep learning-based methods in many computer vision applications, especially feature descriptors, has motivated general research in the direction of image feature detection beyond handcrafted methods. Such state-of-the-art learned keypoint detectors and descriptors have recently demonstrated improved performance on challenging benchmarks (DeTone et al., 2018b; Christiansen et al., 2019; Sarlin et al., 2019).

Rosten & Drummond (2006) (extended in Rosten et al. (2010)) pioneered an attempt leveraging learning to detect corner type features in the image by creating a decision tree that encodes rules for correctly classifying corners present in training samples. Recent advances in Convolutional Neural Networks (CNNs) have also dramatically shifted the technical direction of learning-based keypoint methods. In TILDE (Verdie et al., 2015), the authors introduced multiple piece-wise linear regression models to detect features under severe changes in weather and lighting conditions. To train the regressors, they generate *pseudo* ground truth interest points by using a Difference-of-Gaussian (DoG) detector (Lowe, 2004) from an image sequence captured at different times of day and seasons. The limitation is, however, that the detector is not rotation and scale invariant, as it is trained on static viewpoint images. LIFT (Yi et al., 2016) is able to estimate features which are robust to significant viewpoint and illumination differences using an end-to-end learning pipeline consisting of three modules: interest point detection, orientation estimation and descriptor computation. They use an off-the-shelf SfM algorithm to generate more realistic training samples under different viewpoints and lighting conditions. However, this method is too slow for real-time applications as each module in the pipeline does not share the same computation, and the model does not train on whole images but small patches in multiple steps. In LF-Net (Ono et al., 2018a), the authors introduced an end-to-end differentiable network which estimates position, scale and orientation of features by jointly optimizing the detector and descriptor in a single module. This model has demonstrated state-of-the-art performance for SfM image matching, but it also does not share computations between the detector and descriptor, and its use of patches restricts the area from which the network can learn descriptors.

Quad-networks (Savinov et al., 2017) introduced an unsupervised learning scheme for training a shallow 2-layer network to predict feature points. The model is trained to learn a ranking of invariant interest points under various image transformations. Interest points are then extracted from top/bottom quantiles of this ranking. It is however still a patch-based network, and does not provide descriptors for each patch. SuperPoint (DeTone et al., 2018b) is a self-supervised framework that is trained on whole images and is able to predict both interest points and descriptors. Its architecture shares most of the computation in the detection and description modules, making it fast enough for real-time operation, but it requires multiple stages of training which is not desirable in practice. First, a base detector model is trained on online-generated synthetic images of simple geometrical shapes with *annotated* ground truth interest points defined as corners, junctions, blobs and line segments. The model is then trained on natural images to generate *pseudo* ground truth points by aggregating predicted points of different homography transformations per image. Finally, a siamese network model is trained to estimate both points and descriptors. Most recently, UnsuperPoint (Christiansen et al., 2019) presented a fast deep-learning based keypoint detector and descriptor which requires only one round of training in a self-supervised manner. Inspired by SuperPoint, it also shares most of the computation in the detection and description modules, and uses a siamese network to learn descriptors. They also employ simple homography adaptation along with non-spatial image augmentations to create the 2D synthetic views required to train their self-supervised keypoint estimation model, which is advantageous because it trivially solves data association between these views. Another interesting work is the Self-Improving Visual Odometry algorithm (DeTone et al., 2018a), where the authors first estimate 2D keypoints and descriptors for each image in a monocular sequence using a convolutional network, and then use a bundle adjustment method to classify the stability of those keypoints based on re-projection error, which serves as supervisory signal to retrain the model. Their method, however, is not fully differentiable, so it cannot be trained in an end-to-end manner.

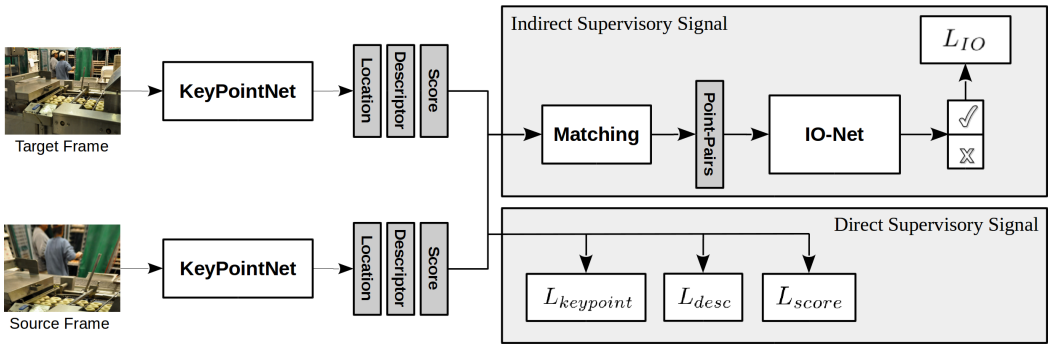


Figure 1: Our proposed framework for self-supervised keypoint detector and descriptor learning with neural outlier rejection. The outlier rejection model produces an indirect supervisory signal to the KeyPointNet, while the other losses provide direct supervisory signals. The additional gradient propagated from our proxy network, InlierOutlierNet (IONet), regularizes the keypoints to have distinguishable inlier and outlier sets.

3 Self-supervised KeyPoint and Descriptor Learning

In this work, we aim to regress a function which takes as input an image and outputs keypoints, descriptors, and scores. Specifically, we define $K : I \rightarrow \{\mathbf{p}, \mathbf{f}, \mathbf{s}\}$, with input image $I \in \mathbb{R}^{3\times H\times W}$, and output keypoints $\mathbf{p} = \{[u, v]\} \in \mathbb{R}^{2\times N}$, descriptors $\mathbf{f} \in \mathbb{R}^{256\times N}$ and keypoint scores $\mathbf{s} \in \mathbb{R}^{N}$; N represents the total number of keypoints extracted and it varies according to an input image resolution, as defined in the following sections. We note that throughout this paper we use $\mathbf{p}$ to refer to the set of keypoints extracted from an image, while p is used to refer to a single keypoint.

Following the work of DeTone et al. (2018b), we train the proposed learning framework in a self-supervised fashion by receiving as input a source image I_s such that $K(I_s) = \{\mathbf{p_s}, \mathbf{f_s}, \mathbf{s_s}\}$ and a target image I_t such that $K(I_t) = \{\mathbf{p_t}, \mathbf{f_t}, \mathbf{s_t}\}$. Images I_s and I_t are related through a known homography transformation $\mathbf{H}$ which warps a pixel from the source image and maps it into the target image. We define $\mathbf{p}_t^* = \{[u_i^*, v_i^*]\} = \mathbf{H}(\mathbf{p}_s)$, with $i \in I$ - e.g. the corresponding locations of source keypoints $\mathbf{p_s}$ after being warped into the target frame.

Inspired by recent advances in Neural Guided Sample Consensus methods (Brachmann & Rother, 2019), we define a second function C which takes as input point-pairs along with associated weights according to a distance metric and outputs the likelihood that each point-pair belongs to an inlier set of matches. Formally, we define $C : \{\mathbf{p_s}, \mathbf{p_t^*}, d(\mathbf{f_s}, \mathbf{f_t^*})\} \in \mathbb{R}^{5\times N} \rightarrow \mathbb{R}^N$ as a mapping which computes the probability that a point-pair belongs to an inlier set. We note that C is only used at training time to choose an optimal set of consistent inliers from possible corresponding point pairs and to encourage the gradient flow through consistent point-pairs.

An overview of our method is presented in Figure 1. We define the model K parametrized by θ_K as an encoder-decoder style network. The encoder consists of 4 VGG-style blocks stacked to reduce the resolution of the image $H \times W$ to $Hc \times Wc = H/8 \times W/8$. This allows an efficient prediction for keypoint location and descriptors. In this low resolution embedding space, each pixel corresponds to an 8×8 cell in the original image. The decoder consists of 3 separate heads for the keypoints, descriptors and scores respectively. Thus for an image of input size $H \times W$, the total number of keypoints regressed is $(H \times W)/64$, each with a corresponding score and descriptor. For every convolutioanl layer except the final one, batch normalzation is applied with activation funcion leakyReLU. A detailed description of our network architecture can be seen in Fig. 2. The Inlier-Outlier model IO is a 1D CNN parametrized by θ_{IO}, for which we follow closely the structure from Brachmann & Rother (2019) with 4 default setting residual blocks and the original activation function for final layer is removed.

3.1 KeyPoint Learning

Following Christiansen et al. (2019), the keypoint head outputs a location relative to the 8×8 grid in which it operates for each pixel in the encoder embedding: $[u'_i, v'_i]$. The corresponding input

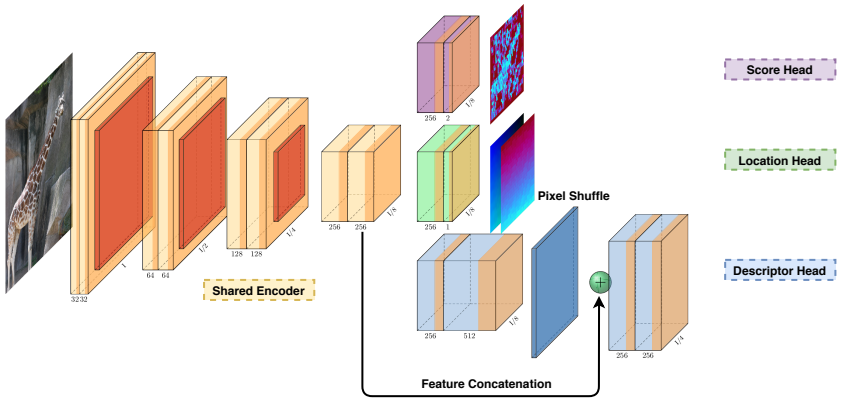


Figure 2: The KeyPointNet architecture. The encoder part consists of 4 VGG-style blocks, and the decoder part consists of 3 separate heads for keypoint locations, scores and descriptors respectively.

image resolution coordinates $[u_i, v_i]$ are computed taking into account the grid's position in the encoder embedding. We compute the corresponding keypoint location $[u_i^*, v_i^*]$ in the target frame after warping via the known homography $\mathbf{H}$. For each warped keypoint, the closest corresponding keypoint in the target frame is associated based on Euclidean distance. We discard keypoint pairs for which the distance is larger than a threshold ϵ_{uv}. The associated keypoints in the target frame are denoted by $\hat{\mathbf{p}}_t = \{[\hat{u}_t, \hat{v}_t]\}$. We optimize keypoint locations using the following self-supervised loss formulation, which enforces keypoint location consistency across different views of the same scene:

$$L_{loc} = \sum_i ||\mathbf{p}_t^* - \hat{\mathbf{p}}_t||_2 \,. \tag{1}$$

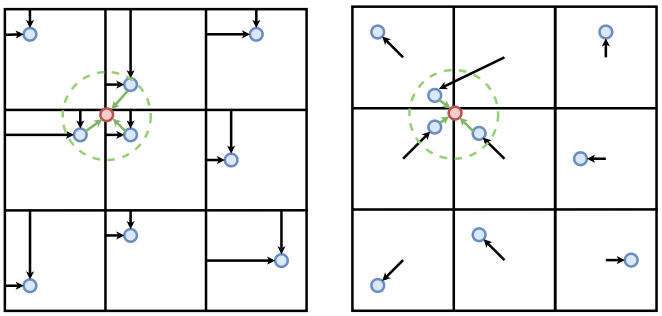
While Christiansen et al. (2019) predict keypoints which are evenly distributed within the cells, they enforce that the predicted keypoint locations do not cross cell boundaries (i.e. each cell predicts a keypoint inside it). However, this leads to suboptimal performance when stable keypoints appear near cell borders. As illustrated in Figure 3, while keypoint association is done via 2D Euclidean distance, Christiansen et al. (2019) force keypoint locations to "stay" inside cells, while they may be naturally pushed outside towards salient points which lie on the boundary of adjacent cells. We propose a novel formulation which allows us to effectively aggregate keypoints across cell boundaries. Specifically, we map the relative cell coordinates $[u'_s, v'_s]$ to input image coordinates via the following function:

$$[v_i, u_i] = ([row_i^{center}, col_i^{center}] + ([v'_i, u'_i]))\frac{\sigma_1(\sigma_2 - 1)}{2} \,, \quad v'_i, u'_i \in (-1, 1) \tag{2}$$

with $\sigma_2 = 8$, i.e. the cell size, and σ_1 is a ratio relative to the cell size. By setting σ_1 larger than 1, we allow the network to predict keypoint locations across cell borders. Our formulation predicts keypoint locations with respect to the cell center, and allows the predicted keypoints to drift across cell boundaries. In the ablation study (Section 4.3), we quantify the effect of this contribution and show that it significantly improves the performance of our keypoint detector.

3.2 Descriptor Learning

As recently shown by Pillai et al. (2018) and Guizilini et al. (2019), subpixel convolutions via pixel-shuffle operations (Shi et al., 2016) can greatly improve the quality of dense predictions, especially in the self-supervised regime. In this work, we include a fast upsampling step before regressing the descriptor, which promotes the capture of finer details in a higher resolution grid. The architectural



(a) (b)

Figure 3: Illustrative example of cross border detection. We show as example how a warped point (in red) can be associated to multiple predicted points (in blue) based on a distance threshold. In this case, forcing a keypoint prediction in the same cell can cause converging issues, since these points can only be pulled to the border, as shown in (a). In our method, we design the network to predict the localization from the cell center, which allows keypoints to be outside the border for better aggregation, as shown in (b).

diagram of the descriptor head is show in Figure 2. In the ablative analysis (Section 4.3), we show that the addition of this step greatly improves the quality of our descriptors.

We employ metric learning for training the descriptors. While the contrastive loss (Hadsell et al., 2006) is commonly used in the literature for this task, we propose to use a per-pixel triplet loss (Schroff et al., 2015) with nested hardest sample mining as described in Tang et al. (2018) to train the descriptor. Recall that each keypoint $p_i \in \mathbf{p_s}$ in the source image has associated descriptor f_s, which we obtain by sampling the appropriate location in the dense descriptor map $\mathbf{f_s}$ as described in DeTone et al. (2018b). The associated descriptor $f^*_{i,+} = f^*_i$ in the target frame is obtained by sampling the appropriate location in the target descriptor map $\mathbf{f_t}$ based on the warped keypoint position p^*_i. The nested triplet loss is defined as:

$$L_{desc} = \sum_i \max(0, \|\mathbf{f}_i, \mathbf{f}^*_{i,+}\|_2 - \|\mathbf{f}_i, \mathbf{f}^*_{i,-}\|_2 + m) \,, \tag{3}$$

where $+$ and $-$ indicate positive and negative samples relative to the anchor descriptor, while m denotes the distance margin.

3.3 Score Learning

The third head of the decoder is responsible for outputting the score associated with each descriptor. At test time this value will indicate the most reliable keypoints from which a subset will be selected. Thus the score is trained to self-calibrate the quality of predicted keypoints while maintaining consistency across frames. Following the work of Christiansen et al. (2019), we define the score loss as the following:

$$L_{score} = \sum_i \left[\frac{(s_i + \hat{s_i})}{2} \cdot (d(p_i, \hat{p_i}) - \bar{d}) + (s_i - \hat{s_i})^2 \right] , \tag{4}$$

where $\bar{d}$ is the average reprojection error of associated points in the current frame, $\bar{d} = \sum_i^L \frac{d(p_i, \hat{p_i})}{L}$, with d being the feature distance in 2D Euclidean space and L is that total number of feature pairs. We note that the aim of L_{score} is two-fold: (i) we want to ensure that feature-pairs have consistent scores and (ii) the network should learn that good keypoints are the ones with low feature point distance d.

3.4 OUTLIER REJECTION

Keypoint and descriptor learning is a task which is tightly coupled with outlier rejection. In this work, we propose to use the latter as a proxy task to supervise the former. Specifically, we associate keypoints from the source and target images based on *descriptor distance*: $\{\mathbf{p_s}, \mathbf{p_t^*}, x(\mathbf{f_s}, \mathbf{f_t^*})\}$. In addition, only keypoints with the lowest K predicted scores are used for training. Similar to the hardest sample mining, this approach accelerates the converging rate and encourages the generation of a richer supervisory signal from the outlier rejection loss. To disambiguate the earlier association of keypoint pairs based on reprojected distance defined in Section 3.1, we denote the distance metric by x and specify that we refer to Euclidean distance in descriptor space. The resulting keypoint pairs along with the computed distance are passed through our proposed Inlier-Outlier(IO)-Net which outputs the probability that each pair is an inlier or outlier. Formally, we define the loss at this step as:

$$L_{IO} = \sum_i \frac{1}{2}(r_i - \mathbf{sign}(||p_i^* - \hat{p}_i||_2 - \epsilon_{uv}))^2 \ , \tag{5}$$

where r is the output of the IO-Net, while ϵ_{uv} is the same Euclidean distance threshold used in Section 3. Different from a normal classifier, we also back propagate the gradients back to the input sample, i.e., $\{\mathbf{p_s}, \mathbf{p_t^*}, d(\mathbf{f_s}, \mathbf{f_t^*})\}$, thus allowing us to optimize both the location and descriptor for these associated point-pairs in an end-to-end differentiable manner.

The outlier rejection task is related to the neural network based RANSAC (Brachmann & Rother, 2019) in terms of the final goal. In our case, since the ground truth homography transform H is known, the random sampling and consensus steps are not required. Intuitively, this can be seen as a special case where only one hypothesis is sampled, i.e. the ground truth. Therefore, the task is simplified to directly classifying the outliers from the input point-pairs. Moreover, a second different with respect to existing neural RANSAC methods arises from the way the outlier network is used. Specifically, we use the outlier network to explicitly generate an additional proxy supervisory signal for the input point-pairs, as opposed to rejecting outliers.

The final training objective we optimize is defined as:

$$\mathcal{L} = \alpha L_{loc} + \beta L_{desc} + \lambda L_{score} + L_{IO} \ , \tag{6}$$

where $[\alpha, \beta, \lambda]$ are weights balancing different losses.

4 EXPERIMENTAL RESULTS

4.1 DATASETS

We train our method using the COCO dataset (Lin et al., 2014), specifically the 2017 version which contains $118k$ training images. Note that we solely use the images, without any training labels, as our method is completely self-supervised. Training on COCO allows us to compare against SuperPoint (DeTone et al., 2018b) and UnsuperPoint (Christiansen et al., 2019), which use the same data for training.

We evaluate our method on the HPatches dataset (Balntas et al., 2017), which contains of 57 illumination scenes and 59 viewpoint scenes. Each scene consists of a reference image and 5 target images with varying photometric and geometric changes for a total of 580 image pairs. In Table 2 and Table 3 we report results averaged over the whole dataset. And for fair comparison, we evaluate results generated without applying Non-Maxima Suppression (NMS).

To evaluate our method and compare with the state-of-the-art, we follow the same procedure as described in (DeTone et al., 2018b; Christiansen et al., 2019) and report the following metrics: Repeatability, Localization Error, Matching Score (M.Score) and Homography Accuracy. For the Homography accuracy we use thresholds of 1, 3 and 5 pixels respectively (denoted as Cor-1, Cor-3 and Cor-5 in Table 3). The details of the definition of these metrics can be found in the appendix.

4.2 IMPLEMENTATION DETAILS

We implement our networks in PyTorch (Paszke et al., 2017) and we use the ADAM (Kingma & Ba, 2014) optimizer. We set the learning rate to 10^{-3} and train for 50 epochs with a batch size of 8; we

	Repeat. ↑	Loc. Error ↓	Cor-1 ↑	Cor-3 ↑	Cor-5 ↑	M.Score ↑
V0-Baseline	0.633	1.044	0.503	0.796	0.868	0.491
V1-Cross	**0.689**	0.935	0.491	0.805	0.874	0.537
V2-CrossUpsampling	0.686	0.918	0.579	0.866	**0.916**	**0.544**
V3-IONet	0.685	**0.885**	**0.602**	0.836	0.886	0.520
V4 - Proposed	0.686	0.890	0.591	**0.867**	0.912	**0.544**

Table 1: Ablative comparison for 5 different configurations where V0: baseline, V1: V0 + cross border detection, V2: V1 + descriptor upsampling, V3: V2 + L_{IO}, and finally the Proposed method : V3 + L_{desc}.

halve the learning rate once after 40 epochs of training. The weights of both networks are randomly initialized.

We set the weights for the total training loss as defined Equation (6) to $\alpha = 1, \beta = 2, \lambda = 1$. These weights are selected to balance the scales of different terms. We set $\sigma_1 = 2$ in order to avoid border effects while maintaining distributed keypoints over image, as described in Sec. 3.1. The triplet loss margin m is set to 0.2. The relaxation criteria c for negative sample mining is set to 8. When training the outlier rejection network described in Sec. 3.4 we set $K = 300$, i.e. we choose the lowest 300 scoring pairs to train on.

We perform the same types of homography adaptation as in DeTone et al. (2018b) which consists of crop, translation, scale, rotation, and symmetric perspective transform. After cropping the image with 0.7 (relative to the original image resolution), the amplitudes for other transforms are sampled uniformly from a pre-defined range: scale $[0.8, 1.2]$, rotation $[0, \frac{\pi}{4}]$ and perspective $[0, 0.2]$. Following Christiansen et al. (2019), we then apply non-spatial augmentation separately on the source and target frames to allow the network to learn illumination invariance. We add random per-pixel Gaussian noise with magnitude 0.02 (for image intensity normalized to $[0, 1]$) and Gaussian blur with kernel sizes$[1, 3, 5]$ together with color augmentation in brightness $[0.5, 1.5]$, contrast $[0.5, 1.5]$, saturation $[0.8, 1.2]$ and hue $[-0.2, 0.2]$. In addition, we randomly shuffle the color channels and convert color image to gray with probability 0.5.

4.3 Ablative study

In this section, we evaluate five different variants of our method. All experiments described in this section are performed on images of resolution 240x320. We first define V0-V2 as (i) V0: baseline version with cross border detection and descriptor upsampling disabled; (ii) V1: V0 with cross border detection enabled; (iii) V2: V1 with descriptor upsampling enabled. These three variants are trained *without* neural outlier rejection, while the other two variants are (iv) V3: V2 with descriptor trained using *only* L_{IO} and *without* L_{desc} and finally (v) V4 - proposed: V3 together with L_{desc} loss. The evaluation of these methods is shown in Table 1.

We notice that by avoiding the border effect described in Sec. 3.1, *V1* achieves an obvious improvement in Repeatablity as well as the Matching Score. Adding the descriptor upsampling step improves the matching performance greatly without degrading the Repeatability, as can be seen by the numbers reported under *V2*. Importantly, even though *V3* is trained without the descriptor loss L_{desc} defined in Sec. 3.2, we note further improvements in matching performance. This validates our hypothesis that the proxy task of inlier-outlier prediction can generate supervision for the original task of keypoint and descriptor learning. Finally, by adding the tripled loss, our model reported under *V4 - Proposed* achieves good performance which is well balanced across all the metrics we compute.

To quantify our runtime performance, we evaluated our model on a desktop with an Nvidia Titan Xp GPU on images of 240x320 resolution. We recorded 174.5 FPS and 194.9 FPS when running our model with and without the descriptor upsampling step.

4.4 Performance Evaluation

In this section, we compare the performance of our method with the state-of-the-art, as well as with traditional methods on images of resolutions 240×320 and 480×640 respectively. For the results

	Repeat. ↑		Loc. Error ↓	
Method	240x320	480x640	240x320	480x640
ORB	0.532	0.525	1.429	1.430
SURF	0.491	0.468	1.150	1.244
BRISK	0.566	0.505	1.077	1.207
SIFT	0.451	0.421	0.855	1.011
LF-Net(indoor) (Ono et al., 2018b)	0.486	0.467	1.341	1.385
LF-Net(outdoor) (Ono et al., 2018b)	0.538	0.523	1.084	1.183
SuperPoint (DeTone et al., 2018b)	0.631	0.593	1.109	1.212
UnsuperPoint (Christiansen et al., 2019)	0.645	0.612	**0.832**	0.991
Proposed	**0.686**	**0.684**	0.890	**0.970**

Table 2: Keypoint detector performance.

	240x320, 300 points				480 x 680, 1000 points			
Method	Cor-1	Cor-3	Cor-5	M.Score	Cor-1	Cor-3	Cor-5	M.Score
ORB	0.131	0.422	0.540	0.218	0.286	0.607	0.71	0.204
SURF	0.397	0.702	0.762	0.255	0.421	0.745	0.812	0.230
BRISK	0.414	0.767	0.826	0.258	0.300	0.653	0.746	0.211
SIFT	**0.622**	0.845	0.878	0.304	**0.602**	0.833	0.876	0.265
LF-Net(indoor)	0.183	0.628	0.779	0.326	0.231	0.679	0.803	0.287
LF-Net(outdoor)	0.347	0.728	0.831	0.296	0.400	0.745	0.834	0.241
SuperPoint	0.491	0.833	0.893	0.318	0.509	0.834	0.900	0.281
UnsuperPoint	0.579	0.855	0.903	0.424	0.493	0.843	0.905	0.383
Proposed	0.591	**0.867**	**0.912**	**0.544**	0.564	**0.851**	**0.907**	**0.510**

Table 3: Homography estimation performance.

obtained using traditional features as well as for LF-Net (Ono et al., 2018b) and SuperPoint (DeTone et al., 2018b) we report the same numbers as computed by (Christiansen et al., 2019). During testing, keypoints are extracted in each view keeping the top 300 points for the lower resolution and 1000 points for the higher resolution from the score map. The evaluation of keypoint detection is shown in Table 2. For Repeatibility, our method notably outperforms other methods and is not significantly affected when evaluated with different image resolutions. For the localization error, UnsuperPoint performs better in lower resolution image while our method performs better for higher resolution.

The homography estimation and matching performance results are shown in Table 3. In general, self-supervised learning methods provide keypoints with higher matching score and better homography estimation for the *Cor-3* and *Cor-5* metrics, as compared to traditional handcrafted features (e.g. SIFT). For the more stringent threshold *Cor-1*, SIFT performs the best, however, our method outperforms all other learning based methods. As shown in Table 1, our best performing model for this metric is trained using only supervision from the outlier rejection network, without the triplet loss. This indicates that, even though fully self-supervised, this auxiliary task can generate high quality supervisory signals for descriptor training. In addition, we show qualitative results of our method in the appendix.

5 Conclusion

In this paper, we proposed a new learning scheme for training a keypoint detector and associated descriptor in a self-supervised fashion. Different with existing methods, we used a proxy network to the generate extra supervisory signal from a task tightly connected to keypoint extraction: the outlier rejection. We show that even without an explicit loss for the descriptor, the supervisory signal from the auxiliary task can be effectively propagated back to the keypoint network to generate distinguishable descriptors. Using the combination of proposed method as well as improved structure, we achieved competitive results in the homography estimation benchmark.

REFERENCES

Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *European conference on computer vision*, pp. 29–42. Springer, 2010.

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*, pp. 5173–5182, 2017.

Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. *arXiv preprint arXiv:1905.04132*, 2019.

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. *arXiv preprint arXiv:1907.04011*, 2019.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Self-improving visual odometry daniel. *arXiv preprint arXiv:1812.03245*, 2018a.

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*, pp. 224–236, 2018b.

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, and Adrien Gaidon. Packnet-sfm: 3d packing for self-supervised monocular depth estimation. *arXiv preprint arXiv:1905.02693*, 2019.

Raia Hadsell, Sumit Chopra, and LeCun Yann. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, June 2006.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6399–6408, 2019.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL https://doi.org/10.1023/B:VISI.0000029664.99615.94.

David G Lowe et al. Object recognition from local scale-invariant features. In *iccv*, volume 99, pp. 1150–1157, 1999.

Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Y. LF-Net: Learning local features from images. In *NIPS*, 2018a.

Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *Advances in Neural Information Processing Systems*, pp. 6234–6244, 2018b.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Sudeep Pillai, Rares Ambrus, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *arXiv preprint arXiv:1810.01849*, 2018.

Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 430–443, 2006.

Edward Rosten, Reid B. Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):105–119, 2010. URL `http://dblp.uni-trier.de/db/journals/pami/pami32.html#RostenPD10`.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, pp. 2. Citeseer, 2011.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: Unsupervised learning to rank for interest point detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. CVPR*, June 2015.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016.

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *arXiv preprint arXiv:1902.09212*, 2019.

Jiexiong Tang, John Folkesson, and Patric Jensfelt. Geometric correspondence network for camera motion estimation. *IEEE Robotics and Automation Letters*, 3(2):1010–1017, 2018.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv preprint arXiv:1904.01355*, 2019.

Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 391–408, 2018.

Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2566–2576, 2019.

Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned Invariant Feature Transform. In *ECCV*, 2016.

A HOMOGRAPHY ESTIMATION EVALUATION METRIC

We evaluated our results using metric same as DeTone et al. (2018b). The Repeatability, Localization Error and Matching Score are generated with correct distance threshold 3. All the four metrics are evaluated from both view points for each image pair.

Repeatability. The repeatability is the ratio of correctly associated points after warping into target frame. The association is performed by selecting closest in-view point and compare the distance with correct distance threshold.

Localization Error. The localization error is the calculated by averaging the distance of points and their associated points.

Matching Score (M.Score). The matching score is the success rate of retrieve correct associated points by performance nearest neighbour matching using descriptors.

Homography Accuracy. The homography accuracy is the success rate of correctly estimating the homographies. The mean distance between four corners of the image planes and the warped image planes using estimated and groundtruth homography matrices are compared with distance $1, 3, 5$.

B QUALITATIVE RESULTS

In this section, we show qualitative results of our best model. Figure 4 to 6 denoting examples of successful matching under strong illumination, rotation and perspective transformation.

Figure 4: Qualitative results of our proposed methods for illumination cases.

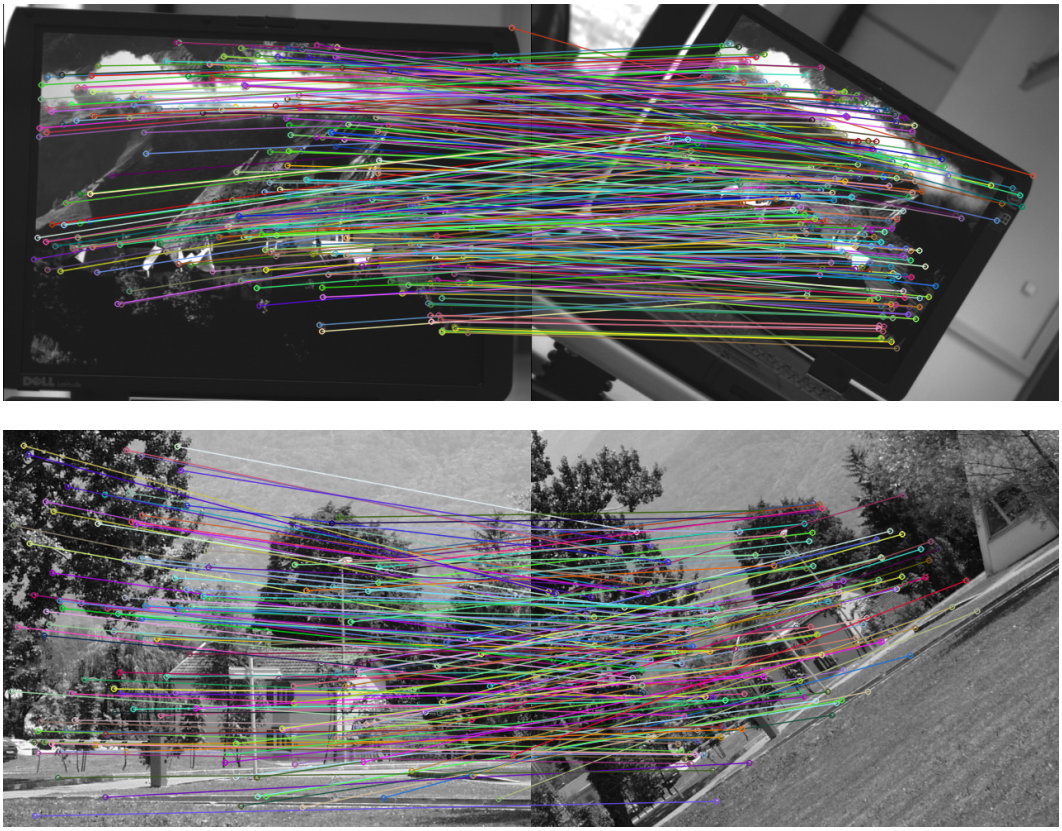


Figure 5: Qualitative results of our proposed methods for rotation cases.

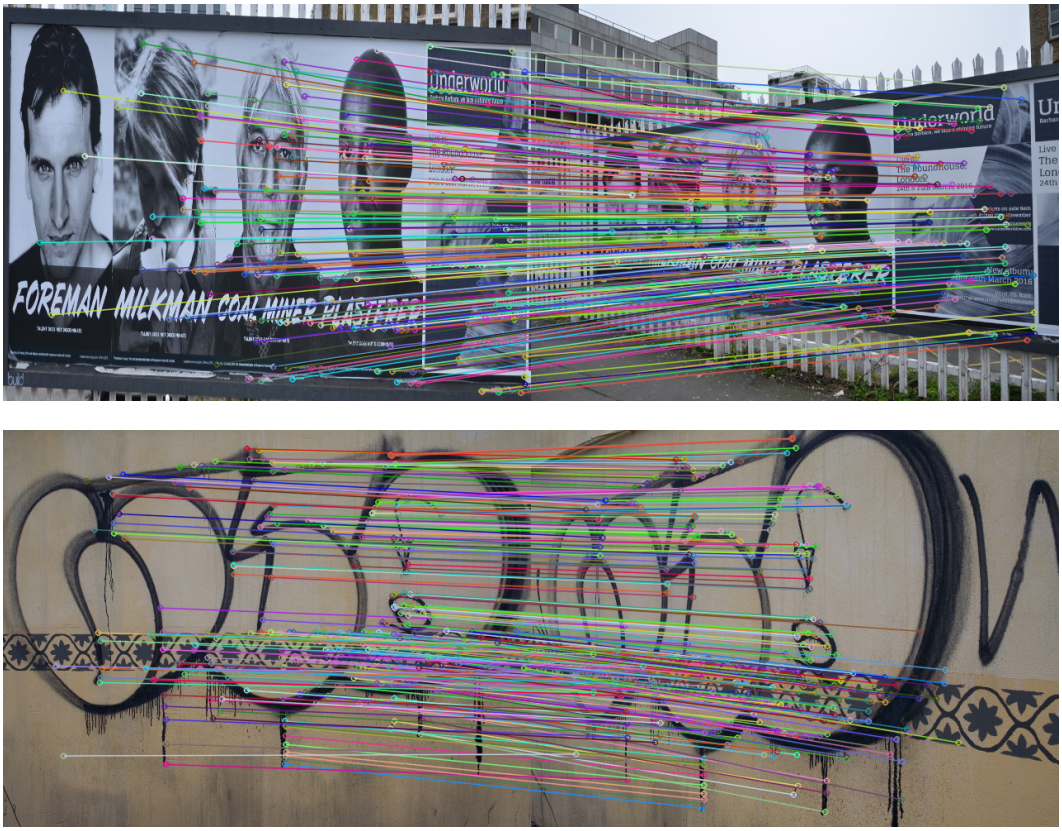


Figure 6: Qualitative results of our proposed methods for perspective cases.