Residual Learning for Image Point Descriptors

Rashik Shrestha[1] **Ajad Chhatkuli**[1,2] **Menelaos Kanakis**[2] **Luc Van Gool**[2]
[1]NepAl Applied Mathematics and Informatics Institute for research (NAAMII)
[2]ETH Zürich
rashik.shrestha@naamii.org.np
{ajad.chhatkuli,menelaos.kanakis,vangool}@vision.ee.ethz.ch

Abstract

Local image feature descriptors have had a tremendous impact on the development and application of computer vision methods. It is therefore unsurprising that significant efforts are being made for learning-based image point descriptors. However, the advantage of learned methods over handcrafted methods in real applications is subtle and more nuanced than expected. Moreover, handcrafted descriptors such as SIFT and SURF still perform better point localization in Structure-from-Motion (SfM) compared to many learned counterparts. In this paper, we propose a very simple and effective approach to learning local image descriptors by using a handcrafted detector and descriptor. Specifically, we choose to learn only the descriptors, supported by handcrafted descriptors while discarding the point localization head. We optimize the final descriptor by leveraging the knowledge already present in the handcrafted descriptor. Such an approach of optimization allows us to discard learning knowledge already present in non-differentiable functions such as the hand-crafted descriptors and only learn the residual knowledge in the main network branch. This offers 50X convergence speed compared to the standard baseline architecture of SuperPoint while at inference the combined descriptor provides superior performance over the learned and hand-crafted descriptors. This is done with minor increase in the computations over the baseline learned descriptor. Our approach has potential applications in ensemble learning and learning with non-differentiable functions. We perform experiments in matching, camera localization and Structure-from-Motion in order to showcase the advantages of our approach.

1 Introduction

The impact of feature point localization and description methods [1, 2, 3, 4, 5, 6, 7] in computer vision applications cannot be understated. Key computer vision applications such as Structure-from-Motion (SfM) [8, 9, 10] and sparse Simultaneous Localization and Mapping (SLAM) [11, 12], hinge on the remarkable accuracy of local feature descriptor methods. SfM and SLAM in turn have facilitated successful industrial and scientific applications [13, 14]. As an example, SfM has played no small part in the optimization of Neural Radiance Field (NERF) models by offering accurate camera poses [15] and sparse 3D initialization [16].

State-of-the-art local feature descriptors are either handcrafted or learned. Handcrafted feature descriptors often use spatial gradients [17, 1, 2, 3, 18, 19] to localize interest points on images. These methods provide repeatable interest points despite view point changes and changes in scale with remarkable accuracy. The interest point detection is then followed by the descriptor computation, which often uses histogram features [1, 6, 3] for robustness. The descriptor computation is often performed through non-differentiable functions [1, 3] that are not trainable in classical deep learning. The importance of such will be apparent later on.

Preprint. The work presented in this paper is currently in progress. Our team is actively working on integrating the latest state-of-the-art models to further enhance the results.

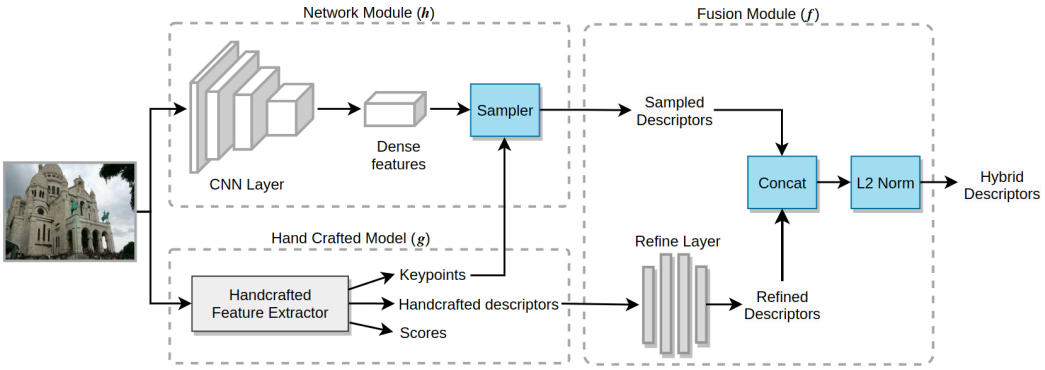

Figure 1: Block Diagram of our approach, consisting three main blocks: Network Module h, Hand-crafted Model g and Fusion Module f. Despite the hand-crafted part being non-differentiable, our method can make use of the function for learning residual knowledge on the network modules.

Learned methods for local image features have gained traction through self-supervised learning [20] and specifically contrastive and metric learning [21, 22] on image augmentations. A method that stands out with self-supervised training on a large image set [23] is SuperPoint [7]. A key advantage of the method is regarding the ease of training in terms of hardware requirement and the simple approach. Latter works have proposed improvements via score prediction [24, 25], outlier rejection [25] and larger transformer networks [26]. Furthermore, self-supervised learning on large image sets alleviates domain shift problems encountered by earlier learned methods [27]. Metric learning [21] on the other hand, can address some of the invariance vs. description tradeoff. Despite these advances, a severe limitation of learned methods is in fact on the sub-pixel point localization. A challenging benchmark [28] highlights the low 'resolution' of point clouds in the SfM reconstructions. [29] uses a more complex training approach using reinforcement learning in order to solve the problem of point localization resolution as well as providing better descriptors. However, the method requires significant computation in training and inference, requiring specialized GPUs for training. Owing to the accuracy provided by hand-crafted keypoints, earlier works [30, 31] learn only the descriptors. Specifically [31] directly uses SIFT [1] in order to compute the interest points. The current state-of-the-art paradigm of local image point description is however, to learn descriptors at interest points identified by a point detection branch trained jointly with the descriptor network.

Two questions naturally arise with these observations. Can traditional methods [1, 2] provide further supervision to the task of self-supervised learning of description networks such as the Superpoint [7], while exploiting the high accuracy of hand-crafted point localization? In other words, can we *improve* the hand-crafted descriptors in conjunction with a standard learning approach for image point descriptors? More importantly, given the low-compute advantage of the handcrafted methods, can these two different approaches be used collaboratively to construct a more powerful description of interest points? In this work, we address the above questions by constructing a simple albeit non-conventional learning architecture. We aim to maximize the use of knowledge encoded by a handcrafted method and learn the residual knowledge in the Deep Neural Nets (DNNs). Here we use 'residual knowledge' in a broader sense for describing knowledge learned on top of existing ones, and not just for a summation operation. Similar to [32], we project the handcrafted feature using a learnable multi-linear perceptron (MLP) followed by concatenation with the DNN feature. The simple architecture forces the DNN to only learn 'residual information' on the network. Thus the whole method makes efficient use of arbitrary but useful non-differential functions for the task. We highlight our approach in Figure 1. We perform extensive evaluations with ablations that showcase superior performance of the proposed method in various metrics including localization and reconstruction.

2 Related Work

We discuss briefly the relevant works on learning feature extractors. Specifically, we are interested in learned local feature extractors. We divide them into those which do not fully learn keypoints and descriptors and those that learn both the keypoints and descriptors.

Learned methods with keypoints or descriptors. Hand-crafted features provide the advantage of speed and accuracy of point localization. Thus earlier methods [30, 31] instead learn only the descriptors. Instead of learning the descriptors, it is also possible to directly learn matching between two images as done in [33, 34].

Learned methods with local features and keypoints. Fully learned local features provide an attractive incentive to the research community in the hope of maximizing the data priors in methods. TILDE [35] and LIFT [36] are seminal works in that regard which use ground-truth matches in order to learn descriptors. Next, several methods proposed self-supervised or unsupervised formulation for training descriptors [30, 37, 38, 7]. A key challenge with any learned descriptor is the domain gap at inference that naturally arises when testing on images of the user's choice. This has been partly solved by using augmentations and learning in a very large image set, together with metric learning [7]. Nonetheless, the question of perspective changes from different viewpoints still remain very recently tackled by [39]. Several recent works have provided architectural improvement [40] and novel training loss [29]. However, in practice handcrafted approaches such as SIFT [1] and standard learned approach of [7] remain highly attractive methods in the community.

3 Method

In order to formalize our approach, we consider two functions $g(\mathbf{I}) \rightarrow \{(x,\ y_1)\}, y_1 \in \mathbb{R}^{d_1}$ – the non-differentiable model which is the handcrafted feature descriptor function and the learned descriptor function $h(\mathbf{I}, \{x\}) \rightarrow \{y_2\} \in \mathbb{R}^{m \times d_2}$. The hand-crafted model g takes in an image and outputs m keypoint locations $\{x\}$, along with a descriptor of dimension d_1 for each point. The learned feature descriptor function h uses the image and the point locations $\{x\}$ to output descriptors $\{y\}$, each of dimension d_2. Here, (x, y_2) is a tuplet of point location and corresponding descriptor respectively. Additionally, a feature fusion module $f(\{(y_1, y_2)\}) \rightarrow \{y\}, y \in \mathbb{R}^d$ takes in the two corresponding feature set outputs of f and g in order to output m descriptors of dimension d. In our case, only h and f are trainable functions, implemented as neural networks. Our goal is to learn the functions h and f in order to produce descriptors y which has better performance compared to y_1 or the descriptor obtained by training f independently of g.

The above formalization helps us to understand the problem as that learning $h,\ f$ conditioned on g, rather than only feature fusion. In semi-supervised learning [41] the use of two different functions g and h, where one is an expert is used to train the network h efficiently through the so-called self-learning. Two important differences exist in our case, the expert g is computationally more efficient than h but is not sufficiently good. Furthermore, we have an effective approach of metric learning in order to supervise h and thus f similar to previous works [7, 24, 25, 42].

3.1 Framework Design

Our framework is divided into three parts: the non-differentiable function g, the network module h and the fusion/projection module f. We illustrate their constructions and interactions through Figure 1. Below we detail each of the block and their design choices.

Handcrafted method g. We mainly consider the two most popular full precision handcrafted descriptor methods: SIFT [1, 43] and SURF [2]. Both use gradient information for detection of interest keypoints. The main differences between the two, from the design perspective, are the speed and feature size they return. SIFT is computationally more complex and uses a descriptor length of 128 integers. SURF on the other hand, is faster and uses float vectors which can be of lengths 32 or 64. Although these two methods may be combined with yet another fusion module, we choose to use only one of them in order to focus on analysis over the performance.

Neural network h. We use a standard architecture [7] for learning the dense descriptors in an image. The network h takes in an image $\mathbf{I}$ to compute dense descriptors. Before they are used in the loss function, these dense descriptors are sampled at locations $\{x\}$, provided by the handcrafted method g. [7] uses VGG style encoder layer to reduce the dimensionality of input image and descriptor decoder head to generate final descriptors.

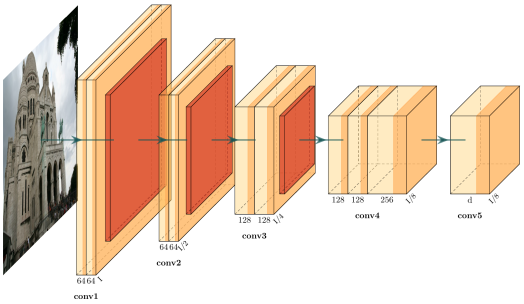

Figure 2: Trimmed SuperPoint Architecture: We use the SuperPoint [7] as our baseline architecture for h without the point decoder head.

Fusion module f. Feature fusion is often used in prior works in order to combine complementary features [44] or for knowledge distillation [45], to name a few. We use a small 3-layer MLP which we call the Refine layer in order to project the handcrafted descriptors obtained from g. We then concatenate the output of the refine layer and the network module h to obtain the final descriptors. Note that alternate design choices can be used instead of concatenation, for example, addition or element-wise product. However, we opt for concatenation for simplicity and favorable results from our initial experiments. We use ℓ_2 normalization in order to bound the descriptors after the concatenation.

3.2 Network Architecture

Our setup uses two learning networks, one CNN layer to extract deep features from the image h and one small Refine layer f to project the handcrafted features before concatenation.

For the **CNN layer**, we use a trimmed version of the standard SuperPoint architecture by removing its point decoder head. It has a VGG style encoder layer to reduce the dimensionality of input image and descriptor decoder head to generate the dense descriptors. We use RGB image for input, so for an image of size $3 \times H \times W$, the network generates features of dimension $d_s \times H_c \times W_c$ dimension, where $H_c = H/8$ and $W_c = W/8$. Figure 2 shows the architecture in details.

For the **Refine layer**, we use a simple MLP with two hidden layers of size 256, making the architecture $d_h \rightarrow 256 \rightarrow 256 \rightarrow d_r$.

In our experiments, we use $d_s = 128$ and $d_r = 128$. For SIFT and extended SURF descriptors, $d_h = 128$. We use simple ReLU activation after CNN and linear layers. Note that several other design choices exist for the Refine layer and the feature fusion thereafter. We do not use additional layers and keep the Refine layer small as per the results of our initial experiments.

3.3 Self-Supervised Training

We train all models including [7] in the large MS COCO [23] dataset, using the self-supervised learning proposed in [7]. Specifically, [7] uses metric learning on augmentations that combine geometric and non-geometric transformations for robust learning. The loss function is the following:

$$\mathcal{L} = \sum_i max(0, y(a,p) - y(a,n) + m) \tag{1}$$

Here, y is the final descriptor of a method with (a, p) denoting a positive for an anchor point a and (a, n) denoting a negative for the same anchor point. We use the margin of $m = 2$, which is the maximum possible distance for the normalized vectors.

4 Experiments

In this section, we provide the necessary implementation details, experimental setups and evaluations.

Figure 3: **Matches Visualization.** We visualize the matches produced by Superpoint (left), Upright RootSIFT (middle), and our method (right) for four stereo pairs in the test set of Phototourism dataset. Matches are coloured red to green, according to their reprojection error (high to low).

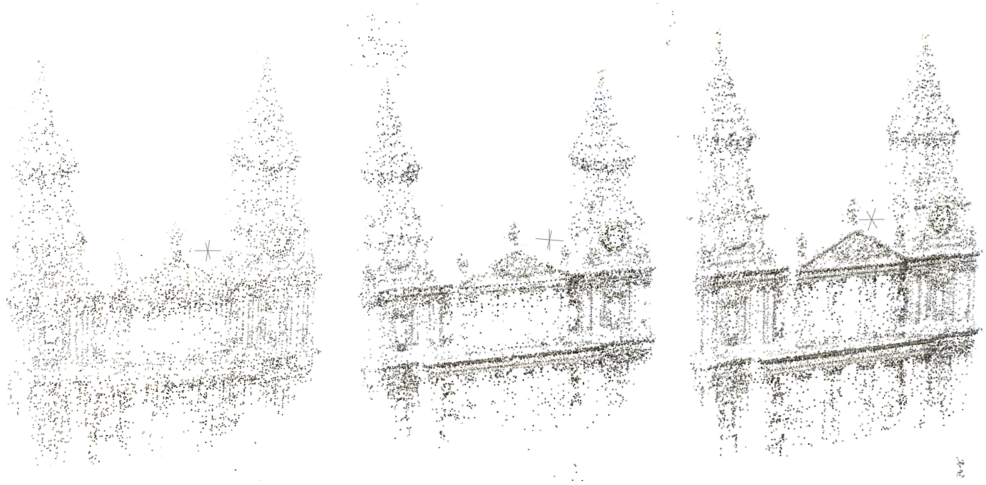

Figure 4: **Point Cloud Visualization.** We build SFM model of "St Pauls Cathedral" scene of Phototourism dataset using 25 images. We visualize point clouds produced by Superpoint, RootSIFT Upright, and our method (from left to right), having 7k, 8k and 11k landmarks respectively.

4.1 Implementation Details

Train. Following the common training paradigm of [7], we use MS-COCO 2017 training dataset split having $118k$ images to train our model. Each image is warped using a random homography transformation. Known homography allows us to generate a set of corresponding keypoints in the original image and their respective warped counterparts, which serves as baseline matches to train our model. Moreover, we use standard non-geometric data augmentation techniques such as addition of random Gaussian noise, random changes to brightness, contrast, saturation and hue to improve the network's robustness.

We implement the model using PyTorch framework. We use a batch size of 4 and ADAM optimizer. The learning rate was set to 10^{-3} throughout the training. We optimize the model for just 2 epochs, which is significantly less than alternative learned methods, such as 100 for SuperPoint, since we found that longer training did not significant improvement the model's performance.

Test. We test our model on HPatches dataset [46] for Homography estimation task, AachenV1.1 Day night dataset [47, 48] for Pose estimation task, and on the Image matching challenge for stereo and multi-view tasks. A more detail explanations can be found in their corresponding sections.

Models	240x320 - 300 points				480x640 - 1000 points			
	Cor-1↑	Cor-3↑	Cor-5↑	MS↑	Cor-1↑	Cor-3↑	Cor-5↑	MS↑
ORB	0.112	0.369	0.474	0.206	0.217	0.528	0.621	0.200
SIFT	0.583	0.841	0.884	0.268	**0.590**	**0.867**	0.914	0.289
SURF	0.397	0.702	0.762	0.255	0.421	0.745	0.812	0.230
SuperPoint	0.491	0.833	0.893	0.318	0.509	0.834	0.900	0.281
Ours (SIFT)	**0.498**	**0.868**	**0.905**	0.400	0.547	0.859	**0.933**	**0.407**
Ours (SURF)	0.492	0.821	0.901	**0.401**	0.531	0.821	0.917	0.401

Table 1: Descriptor evaluation on HPatches dataset. The table shows comparison of Matching Scores and Homography estimation accuracy with 3 different pixel distance thresholds (1,3 and 5). We highlight the best method in **bold**.

4.2 HPatches

Hpatches is comprised of 116 scenes, each with 6 images. All images in a scene are related by a known homographic transformation. 57 scenes have illimunation variations and 59 has viewpoint variations. We report descriptor evaluation metrics, namely Matching Score (MS) and Homography Accuracy (Cor) with thresholds of 1, 3 and 5 pixels. The first image of each scene is used as a reference image to match with the other 5. Hence we have 116*5=580 match pairs.

We perform two sets of experiments with low and high resolution images. For low-resolution images of size 240x320, number of keypoints is limited to 300. For high-resolution images of size 480x640, number of keypoints is limited to 100. Our approach gave better results than SuperPoint and other handcrafted method. We find that extending both SIFT or SURF with our method, consistently improves performance to the handcrafted alternative as well as when using just the SuperPoint baseline. This is thanks to the residual learning of additional knowledge in the deep network branch not captured by the hand-crafted descriptors.

4.3 Aachen Day Night

To further investgiate the generalization capabilities of our method, we evaluate the efficacy of our model on AachenV1.1 [48]. AachenV1.1 is comprised of challenging real-life images captured during different times of the day, namely, day and night. Specifically, the dataset has 824 daytime and 191 nighttime query images. We report the localization accuracy with the threshold values of ($0.25m$,2°), ($0.5m$,5°) and ($5m$,10°). We utilize the HLoc [49] framework for localization, and for each query image, we retrieve the 30 closest images based on global descriptors extracted using NetVLAD [50]. These images act as localization candidates for the query image to localize within the 3D map.

We compare our model with standard handcrafted methods, namely SIFT and SURF, while also evaluating the baseline learned method SuperPoint. For SuperPoint, we used provided pre-trained weights. We test our model with SIFT and SURF features.

We lower the detection thresholds in all our experiments to get enough points. For SIFT, contrast threshold is set to 0.005. For SURF, hessian threshold is set to 30. For SuperPoint, keypoint detection threshold is set 10^{-4}. For all, we take only top 4096 points, filtered on the basis of keypoint scores.

Our method with SIFT worked better than others for Day time images. We can see that handcrafted methods SIFT and SURF does not show good performance for night time images. But a small boost from our model has significantly improved their performance, better or comparable to that of Superpoint. Our model has about 20% of boost in improvement as compared to their hand crafted counterparts for night time images. The difference is more significant when the threshold value is low.

Table 2 shows the results of our experiment. Our method with SIFT features worked consistently better in all the day time metrics.

Models	Localization Accuracy ↑					
	Day			Night		
	.25 2	.5 5	5 10	.25 2	.5 5	5 10
SIFT	82.3	91.6	97.0	45.0	58.6	72.8
SURF	83.4	91.7	96.6	41.4	56.0	70.7
SuperPoint	86.8	*93.8*	**97.9**	*62.3*	**81.7**	*94.8*
Ours (SIFT)	**87.0**	**94.8**	**97.9**	*62.3*	80.6	**95.3**
Ours (SURF)	*86.9*	93.3	*97.7*	**63.4**	*81.2*	*94.8*

Table 2: Visual Localization accuracy on Aachen v1.1 dataset with different threshold values for day and night time images. We highlight the best method in **bold** and *italicize* the second-best.

4.4 Image Matching Challenge (IMC) [28]

	Method	Stereo Task				Multiview Task				
		NM	NI	mAA(5)	mAA(10)	NM	NL	TL	mAA(5)	mAA(10)
2k keypoints	RootSIFT	163.1	86.0	0.2161	0.3112	164.1	1179.5	3.78	0.3712	0.4699
	RootSIFT Upright	197.4	114.6	*0.2619*	*0.3658*	201.0	1400.6	4.06	*0.4333*	*0.5413*
	SURF	145.9	56.3	0.1277	0.2020	147.8	900.2	3.38	0.2476	0.3269
	SuperPoint	*211.6*	103.2	0.2045	0.3015	*215.7*	1392.7	4.24	0.3982	0.5117
	Ours (RootSIFT)	185.5	*116.3*	0.2426	0.3424	187.4	*1454.2*	*4.10*	0.4142	0.5235
	Ours (RootSIFT Up)	**216.7**	**134.7**	**0.2766**	**0.3852**	**219.6**	**1582.8**	**4.14**	**0.4575**	**0.5713**
	Ours (SURF)	160.2	103.9	0.1907	0.3125	159.0	1248.2	4.01	0.3974	0.4517
8k keypoints	RootSIFT	613.3	327.8	0.3667	0.4837	624.9	4526.9	4.12	0.5699	0.6731
	RootSIFT Upright	528.0	360.5	0.3940	0.5120	544.4	4417.6	4.36	0.5755	0.6760
	SURF	370.2	118.4	0.1594	0.2402	375.6	2439.7	3.38	0.3001	0.4099
	SuperPoint	599.5	273.6	0.2667	0.3546	406.8	3556.8	3.87	0.3900	0.5022
	Ours (RootSIFT)	*607.8*	*397.7*	*0.4020*	*0.5265*	*626.1*	*5130.1*	*4.42*	**0.6084**	**0.7128**
	Ours (RootSIFT Up)	**884.3**	**561.3**	**0.4279**	**0.5524**	**910.0**	**6287.6**	**4.50**	*0.6062*	*0.6994*
	Ours (SURF)	449.1	210.7	0.2914	0.4215	514.7	4025.8	4.19	0.5567	0.5941

Table 3: **Image Matching Challenge results.** We report mean Average Accuracy at the threshold of 5°and 10°. Also, we report Number of Matches (NM), which is fed to RANSAC for stereo task and COLMAP for Multiview task. For stereo task, we report number of inlier matches (NI) after passing through RANSAC. For multiview task, we report number of landmarks (NL) and track length (TL). We highlight the best method in **bold** and *italicize* the second-best, for each keypoint category (2k and 8k).

We evaluate our method on a benchmark provided by Image Matching Challenge [28]. It assesses the effectiveness of local features in two tasks: stereo matching and multi-view reconstruction.

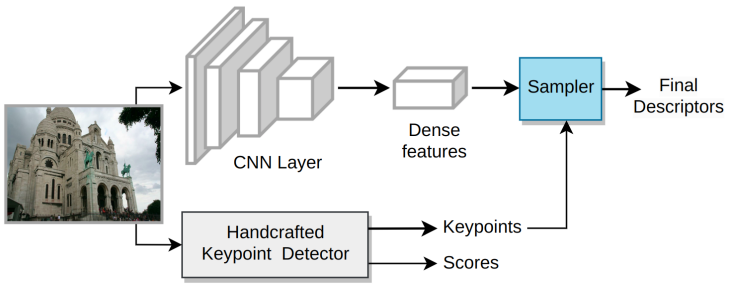

Figure 5: **Ablation Pipeline.** Fusion Module f is removed

Models	Localization Accuracy ↑ Day .25 2°	.5 5°	5 10°	Night .25 2°	.5 5°	5 10°
Ours (SIFT)*	83.7	91.7	97.1	56.0	75.9	90.1
Ours (SURF)*	83.6	92.2	97.2	58.1	76.4	90.6
Small (SIFT)	86.8	93.9	98.4	60.7	78.5	91.6
Small (SIFT)*	84.0	92.0	96.6	55.5	71.2	86.4

Table 4: **Ablation Study -** Here, * represents the model without fusion module. "Small" is the model half the size of our original model, made by removing random CNN layers.

In stereo matching, local features of an image pair is matched and fed to RANSAC [51], for calculating their relative pose. In multi-view, COLMAP [8] is used to construct SFM model using subset of 5, 10, and 25 images. Both tasks measure the performance in terms of the quality of the estimated poses, by using mean average accuracy (mAA) at a 5°and 10°error threshold.

Hypermarameter Tuning. The accuracy is very sensitive to multiple tunable hyperparameters of the pipeline. We tune the ratio threshold of ratio test and inlier threshold of DEGENSAC[52] to get the best performance for each method. We tune the hyperparameters using the validation set of Phototourism dataset. It has three scenes with a total of 274 images.

Fine Tuning Model. We fine-tune our model on the training set of the Phototourism dataset. It has images from 10 different scenes. We randomly sample 120k image pairs and fine-tune our model in a single epoch.

Test. We use the phototourism test set for testing. It has images from 9 scenes. For stereo matching, we calculate feature point matches between each possible image pair for every scene. For multi-view, we construct 10 SFM models with 5 images, 5 SFM models with 10 images and 3 SFM models with 25 images, for each scene. We used ratio threshold of 0.94 and inlier threshold of 0.5 in our models.

Our Model using RootSIFT Upright descriptor performed better than SuperPoint and other hand-crafted methods. Figure 3 shows the comparison of matches produced by three different methods. We see, our method produces better matches in comparison to the other two. In the case of multi-view reconstruction, point cloud given by Superpoint is more dispersed than the one produced by SIFT detector, and hence doesn't produce a good 3D map.

Moreover, our method can be thought of a way to add more performance to handcrafted descriptors. When applied to an already better RootSIFT Upright, the results are better as well. Whereas, when applied to slightly less performing SURF, our model produced slight worse results.

	Method	Stereo Task NM	NI	mAA(5)	mAA(10)	Multiview Task NM	NL	TL	mAA(5)	mAA(10)
2k	Ours (SIFT)	148.6	46.3	0.0770	0.1295	156.2	1578.5	4.21	0.3214	0.4126
8k	Ours (SIFT)	697.2	218.7	0.1584	0.2492	711.5	5405.7	3.98	0.3336	0.4640

Table 5: **Ablation study in Image Matching Challenge.**

4.5 Ablation Study

In our ablation study, we show that descriptors given by hand crafted model g is indeed required for the model to learn quickly and generate better descriptors. We completely omit the handcrafted descriptors from the pipeline. We sample the dense features produced by network module h using handcrafted keypoints, and obtain final descriptors, as illustrated in 5. We omit the fusion module f as well since there is only a single descriptor.

Note that the descriptor size is kept constant as before i.e complete 256 dimension of the descriptor is extracted from CNN Layer. Previously, first half of descriptors (128 dim) was obtained from CNN network, whereas last half (128 dim) was obtained from handcrafted method.

We show that, concatenated descriptors perform better than CNN-only descriptors by a good margin.

In other ablation, we study the effect of using lighter model for CNN layer. We use a smaller model with half the size than our previous one. The lighter model perform comparable to of the pretained SuperPoint model for day time images with only slight less accuracy for night time images.

We do the same ablation for image matching challenge as well, both for 2k and 8k category. We found a huge drop in performance when excluding the handcrafted descriptors. Also, the convergence rate is slower as the model needs to learn the entire descriptor from scratch. This ablation performs as the SuperPoint model but instead of learning points on its own, it uses SIFT points. This gives the advantage for the network to learn faster, but it is still slow than training by incorporating the handcrafted descriptors.

4.6 Limitations

Though our model is small in size, it still adds some computational overhead to the handcrafted descriptors. The increase in computation over the deep neural network SuperPoint is about 10%. Application fields such as robotics may require realtime feature extraction and matching using low-powered devices. Our solution might not be ideal for such a scenario without network quantization. Although the approach we propose is general enough to learn a DNN in conjunction with multiple hand-crafted descriptors, or for that matter, a mixture of hand-crafted and pre-trained model, a stringent requirement is that they must share the same keypoint extractor. Another drawback of our approach is that the final concatenation increases the dimension of the descriptors which results in slightly longer matching time.

5 Conclusions

In this paper, we studied the problem of self-supervised learning of descriptors conditioned on hand-crafted descriptors. We opted a simple training strategy and architecture owing the the baseline method's success and showcased how the 'knowledge' encoded in hand-crafted descriptors can be augmented by learning through a deep network. We call the knowledge added by the deep neural network as residual knowledge as it naturally avoids learning the same function as the hand-crafted descriptor. We obtained a significant improvement in performance on several challenging datasets and evaluation tasks compared to our baselines. Such an approach may be used in future in order to incorporate useful non-differentiable functions in the 'residual learning' paradigm for different tasks.

References

[1] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[3] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011.

[4] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *ECCV*, 2012.

[5] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *ECCV*, 2010.

[6] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *T-PAMI*, 32(5):815–830, 2009.

[7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018.

[8] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[9] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80:189–210, 2008.

[10] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013.

[11] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE T-RO*, 31(5):1147–1163, 2015.

[12] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *T-PAMI*, 29(6):1052–1067, 2007.

[13] Riccardo Giubilato, Sebastiano Chiodini, Marco Pertile, and Stefano Debei. An experimental comparison of ros-compatible stereo visual slam methods for planetary rovers. In *2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 386–391. IEEE, 2018.

[14] Kemal E Ozden, Konrad Schindler, and Luc Van Gool. Multibody structure-from-motion in practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1134–1141, 2010.

[15] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[16] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5511–5520, 2022.

[17] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 525–531. IEEE, 2001.

[18] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244, 1988.

[19] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 430–443. Springer, 2006.

[20] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2051–2060, 2017.

[21] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. *Advances in neural information processing systems*, 27, 2014.

[22] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[23] 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 *ECCV*, 2014.

[24] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: repeatable and reliable detector and descriptor. In *NeurIPS*, 2019.

[25] Jiexiong Tang, Hanme Kim, Vitor Guizilini, Sudeep Pillai, and Rares Ambrus. Neural outlier rejection for self-supervised keypoint learning. In *ICLR*, 2020.

[26] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 2746–2762, December 2022.

[27] Johannes L Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1482–1491, 2017.

[28] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

[29] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020.

[30] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. *Advances in neural information processing systems*, 29, 2016.

[31] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *Advances in neural information processing systems*, 30, 2017.

[32] Mihai Dusmanu, Ondrej Miksik, Johannes L Schönberger, and Marc Pollefeys. Cross-descriptor visual localization and mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6058–6067, 2021.

[33] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[34] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Probabilistic warp consistency for weakly-supervised semantic correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8708–8718, 2022.

[35] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5279–5288, 2015.

[36] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016.

[37] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 100–117. Springer, 2016.

[38] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6325–6333, 2018.

[39] Dominik Muhle, Lukas Koestler, Krishna Murthy Jatavallabhula, and Daniel Cremers. Learning correspondence uncertainty via differentiable nonlinear least squares. 2023.

[40] Jongmin Lee, Byungjin Kim, and Minsu Cho. Self-supervised equivariant learning for oriented keypoint detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4847–4857, 2022.

[41] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[42] Simon Maurer, Menelaos Kanakis, Matteo Spallanzani, Ajad Chhatkuli, and Luc Van Gool. Zippypoint: Fast interest point detection, description, and matching through mixed precision discretization. *arXiv preprint arXiv:2203.03610*, 2022.

[43] Relja Arandjelović 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.

[44] Saihui Hou, Xu Liu, and Zilei Wang. Dualnet: Learn complementary features for image recognition. In *Proceedings of the IEEE International conference on computer vision*, pages 502–510, 2017.

[45] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[46] 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.

[47] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018.

[48] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012.

[49] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.

[50] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016.

[51] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[52] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane, 2005.