MLIFeat: Multi-level information fusion based deep local features

Yuyang Zhang[1,2] *, Jinge Wang[3], Shibiao Xu[1,2] **,
Xiao Liu[3], and Xiaopeng Zhang[1,2]

[1] National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] Megvii Technology
{yuyang.zhang,shibiao.xu,xiaopeng.zhang}@nlpr.ia.ac.cn
liuxiao@foxmail.com,wjg172184@163.com

Abstract. Accurate image keypoints detection and description are of central importance in a wide range of applications. Although there are various studies proposed to address these challenging tasks, they are far from optimal. In this paper, we devise a model named MLIFeat with two novel light-weight modules for multi-level information fusion based deep local features learning, to cope with both the image keypoints detection and description. On the one hand, the image keypoints are robustly detected by our Feature Shuffle Module (FSM), which can efficiently utilize the multi-level convolutional feature maps with marginal computing cost. On the other hand, the corresponding feature descriptors are generated by our well-designed Feature Blend Module (FBM), which can collect and extract the most useful information from the multi-level convolutional feature vectors. To study in-depth about our MLIFeat and other state-of-the-art methods, we have conducted thorough experiments, including image matching on HPatches and FM-Bench, and visual localization on Aachen-Day-Night, which verifies the robustness and effectiveness of our proposed model. Code at: `https://github.com/yyangzh/MLIFeat`

1 Introduction

For a long time, image keypoints detection and their local feature description have been active and open research problems in computer vision. It is an essential processing step for various visual-based applications such as SfM[1], SLAM[2–6], Visual Localization[7, 8], and Image Retrieval[9]. With the industry's rapid development, these applications is required to deal with more complex and challenging scenarios (various conditions such as day, night, and seasons). As the image keypoints detection and description are the critical components of these

[*] Part of the contribution was made by Y. Zhang when he was an intern at Megvii Research Beijing, Megvii Technology, China.
[**] S. Xu is the corresponding author(shibiao.xu@nlpr.ia.ac.cn)

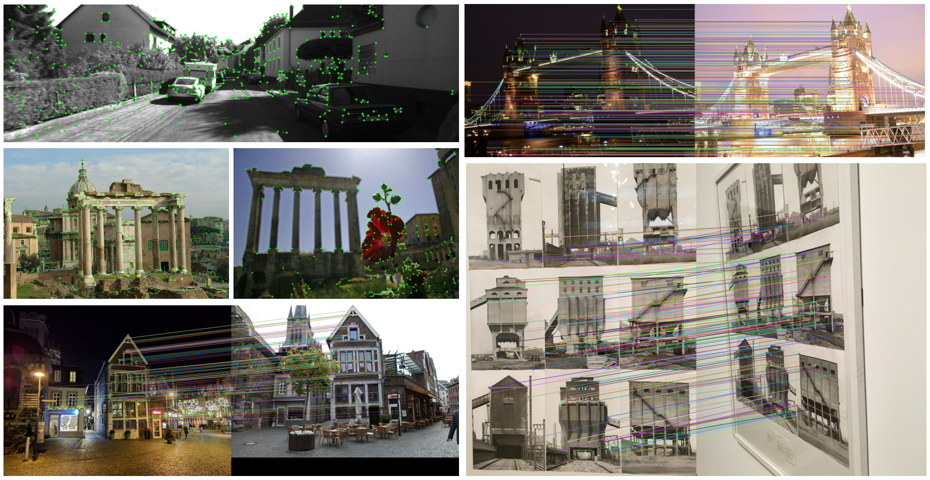

Fig. 1. Visualization samples of detecting and matching on FM-Bench[20], Aachen-Day-Night[21] and HPatches[22]. The proposed method can successfully find image correspondences even under large illumination or viewpoint changes.

high-level algorithms, there is an urgent need to improve their precision, which is of great significance.

Over the past two decades, there are many excellent algorithms proposed to solve the above problem. Both the traditional hand-crafted methods[10–14] and the deep-learning-based methods[15–17] have made a breakthrough. Especially the deep-learning-based algorithms, such as SuperPoint[15], D2-net[16], and R2D2[18], have greatly improved the accuracy of both the keypoints detection and the local feature description. However, most previous methods[15, 16, 18, 19] deploy the top-layer feature map to detect keypoints and extract descriptors, which is problematic. Firstly, detecting keypoints on the top-layer feature map with reduced spatial size will inevitably enlarge the detection error. More importantly, it is hard for the descriptors extracted from the top-layer feature to distinguish keypoints with the same high-level semantics but the different local structures, as they lack the low-level structural information. Motivated by such observation, we propose two novel lightweight modules to mitigate each limitation separately. Specifically, to reduce the systematic detection error, we design a Feature Shuffle Module (FSM), which can efficiently reorganize the feature maps from low-resolution to high-resolution with marginal computing cost and detect the keypoints with high precision from these shuffled feature maps. To encode necessary structural information to each descriptor, we further devise a Feature Blend Module (FBM), capable of collecting rich information from the multi-level convolutional features and constructing the most discriminative descriptor.

In brief, there are three main contributions in this paper: 1) we design a novel Feature Shuffle Module (FSM) to detect the keypoints accurately; 2) we devise a novel Feature Blend Module (FBM) to generate robust descriptors; 3) with the power of the two lightweight modules, we present a novel model named MLIFeat to detect keypoints and extract descriptors jointly. To analyze the proposed method's strengths, we have conducted comprehensive experiments on HPatches[22], FM-Bench[20], and Aachen-Day-Night[21], which show that our proposed MLIFeat reaches state-of-the-art performance.

2 Related Work

For a long time, the hand-crafted methods are the preference of most high-level algorithms. Among them, SIFT[10] plays a vital role in computer vision[1], which utilizes Difference-of-Gaussian to detect the keypoints and then constructs the corresponding descriptors through gradients of their surrounding pixels. Besides, ORB[11] is a commonly used algorithm due to its fast and robust features. More comprehensive evaluation results can be found in [23, 24, 14].

With the development of deep learning technology, many learned local features [17, 25, 26] emerge, which detect keypoints based on the hand-crafted methods and extract the descriptors via neural network. Among them, L2-Net[17] proposed a network architecture stacking by several convolution layers to extract the descriptor of an image patch and deployed an n-pair loss to train the model end-to-end. Hardnet[25] proposed a hard-mining strategy to train the network more efficiently, which improved the model performance significantly. SOS-Net [26] used the second-order similarity to regularize the descriptors' potential distribution. Since these methods take an image patch as input, the performance of their descriptors is still limited in some challenge scenarios[14, 15].

In contrast to the above hybrid methods, many unified architectures have proposed to detect the keypoints and describe[27, 28, 15, 16] the local feature jointly in recent years. Among them, LIFT[27] and LF-Net[28] both proposed a two-stage algorithm to first detect the keypoints via a score map predicted by one sub-network and then input the corresponding image patches to another sub-network to generate the descriptors. Different from the above two-stage methods, SuperPoint[15] raised a more unified architecture constructed by a common encoder followed by two separate branches to detect the keypoints and extract the descriptors. DELF[29] and D2-Net[16] proposed a describe-and-detect approach that utilizes the dense feature descriptors to detect the keypoints. R2D2[18] raised an algorithm that trains the model to detect and describe the keypoints only in the discriminate image region. UnsuperPoint[30] deployed an unsupervised pipeline to learn both the keypoints detector and the local feature descriptor. Recently, ASLFeat[19] utilized the powerful DCN to extract the descriptors, which can correctly match under challenging scenarios. However, most of these methods ignore the importance of low-level structural information(e.g., shape, scales) to the keypoints detection and descriptors extraction, resulting in suboptimal performance. To mitigate this limitation, in this paper, we carefully devise two novel and intuitive light-weight modules to take the advantages of multi-level feature maps to largely promote the precision of keypoints and the robustness of descriptors.

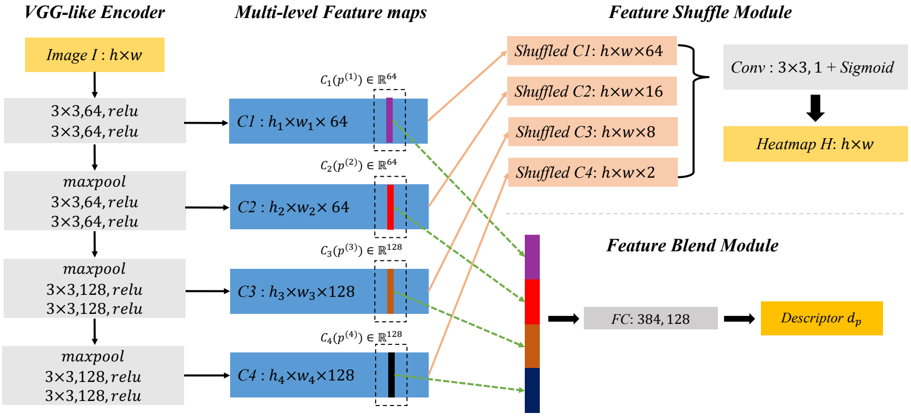

Fig. 2. The network architecture of our **MLIFeat**, which is designed by integrating the common used VGG-like encoder with the Feature Shuffle Module(FSM) and the Feature Blend Module(FBM). Specifically, the backbone encoder takes a single-scale image I as input and output the feature maps at scales. The FSM further utilizes these feature maps to predict the heatmap H. Besides, given a point $p \in I$ and its down-sampled location $p^{(m)}$in each feature map C_m, the corresponding feature vector $C_m(p^{(m)})$ is looked up from C_m with bi-linear interpolation. Then the FBM blends all feature vectors to generate the descriptor d_p.

3 Proposed Method

3.1 Network Architecture

Our model consists of three core components: the backbone feature encoder, the Feature Shuffle Module (FSM), and the Feature Blend Module (FBM). The backbone feature encoder takes a single-scale image as input and generates a series of convolutional feature maps with semantic information from low to high. The well-designed Feature Shuffle Module and Feature Blend Module further take these feature maps as input and output the detected keypoints and their corresponding descriptors. Since the detection and description are relatively independent of the feature extracting, we take the commonly used VGG-like[31, 15, 16] encoder as our backbone network due to its efficiency and accuracy. The whole network architecture can be seen in Fig. 2.

Backbone Feature Encoder. The process of the encoder is a feed-forward computation of the backbone network, which produces the feature maps at several scales with a scaling step of 2. Considering the original image as $I \in \mathbb{R}^{h \times w}$, the corresponding feature maps at scales can be denoted as $C_m \in \mathbb{R}^{h_m \times w_m \times d_m}$, where $m \in \{1, 2, 3, 4\}$ and $d \in \{64, 64, 128, 128\}$. The size of C_m and the size of I satisfies $h = h_m \times 2^{m-1}, w = w_m \times 2^{m-1}$. This feature extraction process can

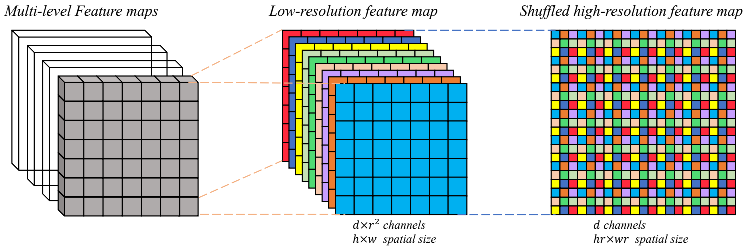

Fig. 3. The visualization of the pixel shuffle operation. Each depth channel's features are scattered into the corresponding spatial region according to a scale ratio r, resulting in a high-resolution feature map with reduced depth channel. The whole process is fast and casts no extra memory resources, which is very suitable for real-time keypoints detection.

be formulated as:

$$C_1, C_2, C_3, C_4 = Encoder(I). \tag{1}$$

Keypoint Detection with Feature Shuffle Module. Inspired by the pixel shuffle operation raised in [32], we propose a novel Feature Shuffle Module (FSM) that takes the multi-level feature maps as input and predicts the keypoint heatmap with the same resolution as the input image.

Specifically, our Feature Shuffle Module first reorganizes each low-resolution feature map $C_m \in \mathbb{R}^{h_m \times w_m \times d_m}$ to a high-resolution one $C_m^s \in \mathbb{R}^{h \times w \times d_m / 4^{m-1}}$ via the pixel shuffle operation, which is shown in Fig.3. Since the shuffled feature maps have the same spatial size, they can be processed by a unified $Conv$ layer to generate the final heatmap, which implicitly fuses multi-level semantics and naturally leads to a prediction with high precision. And the whole process can be abstracted as:

$$H = FSM(C_1, C_2, C_3, C_4). \tag{2}$$

During the model inference, the Non-Maximum-Suppression (NMS) is first applied to the predicted heatmap. A point is then marked as a keypoint while its response value in H exceeds a fixed detection threshold α.

Local Feature Description with Feature Blend Module To further make full use of the multi-level semantics, we design a novel Feature Blend Module (FBM) that can extract the most discriminative information from the multi-level feature vectors to construct the descriptor.

For a point $p = [x, y]^T$ in the original image, its location in each feature map $C_m \in \mathbb{R}^{h_m \times w_m \times d_m}$ can be computed by $p^{(m)} = p/2^i = [x/2^m, y/2^m]^T$ and the corresponding feature vector $C_m(p^{(m)}) \in \mathbb{R}^{d_m}$ is bi-linear interpolated from the feature map C_m. After generating all the feature vectors corresponding to the same point, a long feature vector C_{cat} is constructed by concatenation.

Though C_{cat} already contains multi-level semantics from low to high, directly using this feature vector as a descriptor will certainly introduce noise and useless information. Therefore, one fully-connected layer is further deployed to filter noise and compress the valid semantics to produce a compact descriptor $d_p \in \mathbb{R}^{dim}$, where $dim = 128$. The FBM is illustrated in Fig. 2 and the whole process can be generalized as:

$$D = FBM(C_1, C_2, C_3, C_4, P), \tag{3}$$

where $P = \{p_1, p_2, ..., p_n\}$ denotes a bunch of keypoints and their corresponding descriptors are denoted as $D = \{d_1, d_2, ..., d_n\}$.

3.2 Data Preparation for Joint Training.

To train our MLIFeat with FSM and FBM jointly, we use the COCO[33] and MegaDepth[34] as our training dataset. The former are collected from plenty of diverse scenes, which ensures the robustness of the whole model. And the latter contains image pairs with known poses and depth, which can further enhance the local features' distinguishability.

Image Keypoints Supervising. As the original COCO and MegaDepth do not have ground truth labels for the keypoint detection, we deploy the Iterative Homographic Adaptation[15] to generate the keypoints pseudo-ground truth label $Y \in \mathbb{R}^{h \times w}$ for each image in both datasets: 1) Construct a synthetic dataset as source dataset; 2) Use the source dataset to train a detector; 3) Label the target dataset (COCO and MegaDepth); 4) Change the source to the newly labeled target datasets and back to the step two until converged. More details can be found in our supplementary material.

Correspondences Generation. For the descriptor training, the correspondences between the image pair are required. Different from the MegaDepth, the images in COCO are relatively independent. Thus, for an image I in COCO, a random homography is sampled and an image I' is synthesized based on the homography, resulting in the pairwise image. Then, for both dataset, n randomly sampled correspondences are constructed based either on the homography in COCO or on the pose in MegaDepth, which can be formulated as:

$$P = RandomSample(\cdot) \quad P', V = Transform(P), \tag{4}$$

where P, P' are the corresponding points between the image pair and $V \in \mathbb{B}^n$ is a valid mask denoting the validity of each projected point, as not all the transformed points are located in the image boundaries.

3.3 Definition of Loss Function

Detector Loss. Given a heatmap $H \in \mathbb{R}^{h \times w}$ predicted from Eq.(2) and its corresponding keypoints pseudo-ground truth label Y, the weighted binary cross entropy loss can be formulated as:

$$L_{bce}(H,Y) = \frac{1}{hw}\sum_{u,v}^{h,w}(-\lambda Y_{u,v}log(H_{u,v}) - (1-Y_{u,v})log(1-H_{u,v})), \tag{5}$$

where λ is used for balancing the ratio between positive and negative samples because the number of positive samples is much smaller than the number of negative samples. And in our paper, we empirically set $\lambda = 200$.

Descriptor Loss. Given the points set P in I and their corresponding points set P' in I' generated from Eq.(4), the descriptors D, D' of these points can be extracted from FBM respectively. Then, for a descriptor $d_{p_i} \in D$, its *positive pair distance* is defined as:

$$p(d_{p_i}) = ||d_{p_i} - d_{p'_i}||_2, \tag{6}$$

where $d_{p'_i} \in D'$ is the corresponding descriptor of d_{p_i}. And its *hardest negative pair distance* is formulated as:

$$n(d_{p_i}) = ||d_{p_i} - d_{p'_{k^*}}||_2, \tag{7}$$

where

$$k^* = \arg\min_{k \neq i} ||d_{p_i} - d_{p'_k}||_2 \ \& \ ||p'_k - p'_i||_2 > \theta \ \& \ p'_k \text{ within the boundaries.} \tag{8}$$

The empirical threshold $\theta = 16$ is used to ensure that the spatial distance between p'_{k^*} and p'_i is beyond a certain value, as the two descriptors are too similar to distinguish from each other when they are very close in the image, which is harmful to the training. Besides, the selected negative sample $d_{p'_{k^*}}$ is also required to locate within the image boundaries, or it is invalid. Given $p(d_{p_i})$ and $n(d_{p_i})$, we define our hardest triplet descriptor loss as:

$$l_{triplet}(d_{p_i}) = \max(0, p(d_{p_i}) - n(d_{p_i}) + 1). \tag{9}$$

And the whole loss constructed for the descriptors D, D' is summed as:

$$L_{triplet}(D, D', V) = \sum_{i=1}^{n} \frac{l_{triplet}(d_{p_i})v_i}{\sum_{j=1}^{n} v_j}, \tag{10}$$

where $v_i \in V$ indicating the validity of the correspondence between d_{p_i} and $d_{p'_i}$.

Total Loss. Based on above definition, the total loss is formulated as:

$$L_{total}(H, H', D, D'; Y, Y', V) = L_{bce}(H, Y) + L_{bce}(H', Y') + L_{triplet}(D, D', V). \tag{11}$$

The sampling of both the transformation and correspondences is processed with the training procedure in parallel, which prevents the network from overfitting.

3.4 Parameters Setting

For model training, we use Adam optimizer [35] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $lr = 0.001$ and $weight\,decay = 10^{-4}$. The training image size is set to 240×320 with the training batch size setting to 16. The whole training process typically converges in about 30 epochs. Besides, during the model evaluation, the NMS radius is set to 4 pixels. And the detection threshold α is set to 0.9 to balance the number and reliability of the keypoints.

4 Experiments

4.1 Image Matching on HPatches

Dataset. We use the popular HPatches[22], which includes 116 scenes with 580 image pairs exhibiting a large change in either illumination or viewpoints. The ground truth homography between each image pair is provided for the evaluation. Following D2net[16], we exclude eight high-resolution sequences, leaving 108 scenes for a fair comparison.

Evaluation protocols. For a comprehensive evaluation, three standard metrics are used: 1) Homography accuracy ($\%HA$), a.k.a the ratio of correct estimated homography. 2) Matching score ($\%M.S.$), a.k.a the ratio of correct matches and the minimum number of keypoints in the shared view. 3) Mean matching accuracy ($\%MMA$), a.k.a the ratio of correct matches and possible matches. Here, the matches are found by the mutual nearest search for all methods, and a match is defined to be correct if the point distance is below some error threshold after projecting from one image to another. Besides, the homography is estimated based on the matches, and it is defined to be correct when its warping error is below some error thresholds[15].

Comparative methods. We compare our methods with 1) hand-craft method ROOT-SIFT[36] and DSP-SIFT[13]. 2) learned shape estimator HesAffNet[37] plus learned patch descriptors HardNet++[25]. 3) Joint local feature learning state-of-the-art approaches including SuperPoint[15], D2net[16], R2D2[18], and recent ASLFeat [19]. To ensure the fairness and reproducibility of results, we report all the results based on the public implementations with default parameters. Except for speed evaluation, all evaluations are conducted based on the original resolution images in HPatches.

Baseline. In this paper, we use the same backbone as SuperPoint and present our reimplementation of SuperPoint(*our impl*) as our baseline. Specifically, *our impl* is differs from the original SuperPoint(*orig*) in mainly two aspects: 1) Different training dataset (COCO and MegaDepth vs. only COCO). 2) Different loss formulation (hardest-triplet[25] vs. pairwise-contrastive[15]). Under the same training protocol, it is fair to compare our MLIFeat with the new baseline.

Table 1. Ablation experiments of proposed modules. *orig* means the SuperPoint publicly released model, and *impl* is the reimplemented baseline under our training protocol. SuperPoint + FSM replaces the SuperPoint detection head with our Feature Shuffle Module. SuperPoint + FBM replaces the SuperPoint description head with our Feature Blend Module. And MLIFeat is the backbone of SuperPoint plus two proposed modules that significantly improves the baseline model's performance.

HPatches dataset(error threshold @3px)									
Configs	***Total***			***Illumination***			***Viewpoint***		
	M.S.	*MMA*	*HA*	*M.S.*	*MMA*	*HA*	*M.S.*	*MMA*	*HA*
SuperPoint *orig*	0.424	0.645	0.726	0.456	0.694	0.892	0.394	0.599	0.571
SuperPoint *impl*	0.456	0.683	0.713	**0.502**	0.734	0.889	0.413	0.637	0.557
SuperPoint + FSM	0.464	0.710	0.730	0.489	0.742	0.896	0.439	0.679	0.575
SuperPoint + FBM	0.460	0.698	0.734	0.496	0.748	**0.915**	0.427	0.651	0.575
MLIFeat	**0.475**	**0.728**	**0.756**	0.500	**0.763**	0.892	**0.453**	**0.696**	**0.629**

Ablation on Training protocol. Due to the newly added dataset and more powerful loss function, as shown in Tab.1, *our impl* outperforms *orig* in *%MMA* and *%M.S.*. However, it's interesting to find that the *%HA* of *our impl* is slightly worse than the *orig*. It lies in that the *%HA* is not a direct metric to assess and is affected by both the homography estimation algorithm's accuracy and the quality of the matched points. Generally speaking, only when the matching is sufficiently good can the corresponding estimated homography be improved.

Ablation on FSM. When replacing the original detection head in SuperPoint with our proposed Feature Shuffle Module, it is evident in Tab.1 that this variant outperforms the baseline in almost metrics. Such improvement is reasonable that FSM detects keypoints from the high-resolution multi-level information fused feature map. Especially when viewpoint changes, points detected from low-resolution prone to large errors. However, applying FSM, the accuracy of the keypoints improves obviously, e.g., %MMA from 0.637 to 0.679, which indicates that FSM will reduce the systematic errors caused by low-resolution feature map.

Ablation on FBM. Similarly, utilizing the Feature Blend Module yields better results, for it promotes the discriminability of the descriptors by the multi-level feature vectors. The lower-level feature vector contains more structural information about the neighbor of the keypoints. Meanwhile, the high-level feature vector encodes more semantics information from a wider spatial region. Such a combination is simple but effective. And it is convenient to take our FBM in most current methods to further improve their descriptors' performances.

Comparisons with other methods. The comprehensive comparisons results with other methods are illustrated in Fig.4. Within a small threshold (3px),

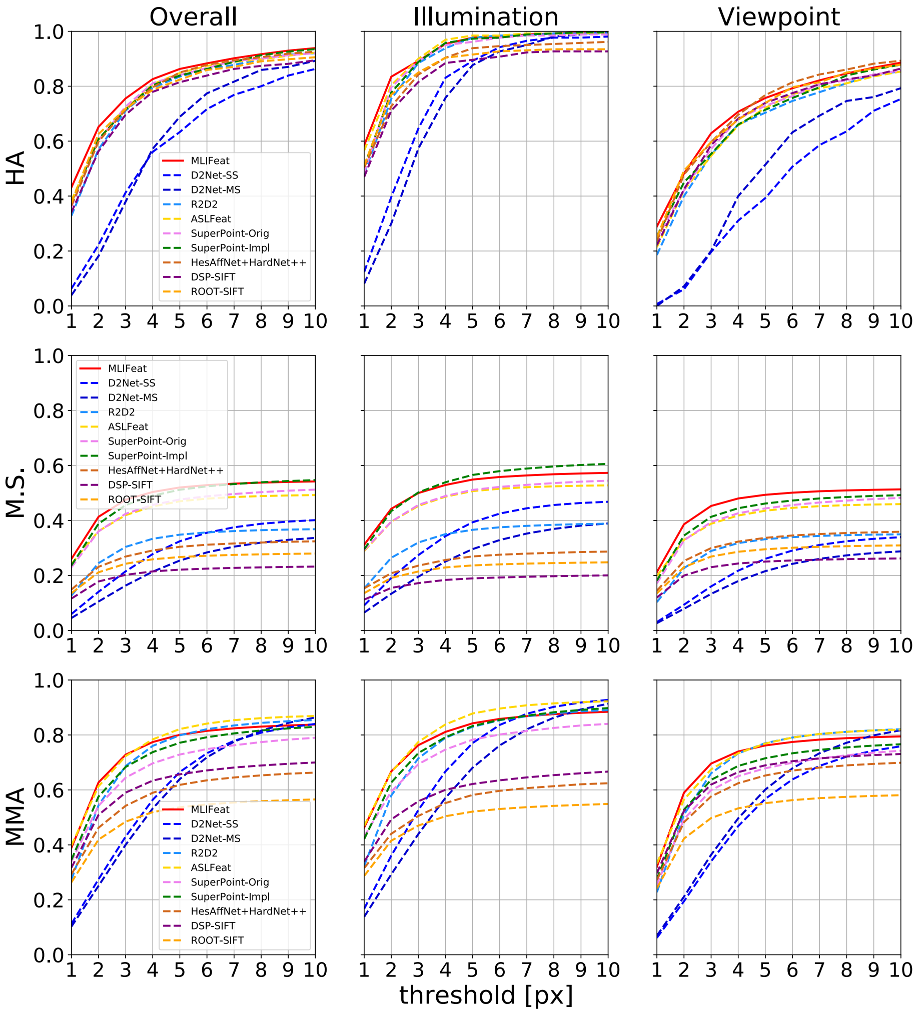

Fig. 4. Comparisons on HPatches Dataset[22] with Homography Accuracy (*%HA*), Matching Scores (*%M.S.*), and Mean Matching Accuracy (*%MMA*). Our method achieves either the best or the comparable performances within a threshold of 3px.

MLIFeat outperforms other methods on almost all error metrics. Even within a relaxed error bound, our method is still at the top three ranks in all models. Furthermore, when comparing with the most recent ASLFeat who utilizes the complex Deformable Convolutional network to generate descriptors with high precision, our MLIFeat still generates comparable results, which strongly verifies the effectiveness of the proposed two modules.

In addition, experiments are conducted to compare the size and speed of the proposed model and other joint learning methods, which is shown in Tab.2.

Table 2. Size and speed comparisons of the joint learning methods. The speed is averaged on HPatches(480×640) with TitanV. We can see that our MLIFeat reaches the fastest speed under the same experimental protocol.

	MLIFeat	SuperPoint	ASLFeat	R2D2	D2Net
Size	2.6Mb	5.0Mb	5.2Mb	**1.9Mb**	29Mb
Speed	**32fps**	28fps	21fps	8fps	6fps

Specifically, the speed is average on HPatches with the same image size(480×640) under TitanV, and the size is the sum of all the parameters contained in each model. With the light-weight FSM and FBM, our MLIFeat reaches the fastest speed. Though R2D2 has the smallest model size, it cast too much time to detect keypoints and extract descriptors, making the whole algorithm very slow.

4.2 Image Matching on FM-Bench

The widely used HPatches dataset may not comprehensively reflect the algorithm's performance in real applications[19], since it exhibits only homography. Therefore, the same as ASLFeat, we resort to the newly proposed FM-Bench [20] to further evaluate each method's matching performance.

Dataset. FM-Bench comprises four datasets captured in practical scenarios: the TUM dataset[38] in indoor SLAM settings, the KITTI dataset[39] in driving scenes, the Tanks and Temples dataset(T&T)[40] for wide-baseline reconstruction and the Community Photo Collection(CPC)[41] for wild reconstruction from web images. For each dataset, 1000 overlapping image pairs are chosen for evaluation, with ground-truth fundamental matrix pre-computed.

Evaluation protocols. A full matching pipeline including outlier rejection (ratio test) and geometric verification(RANSAC) is performed, and the final estimated pose accuracy is evaluated. FM-Bench utilizes ground-truth pose to generate a set of virtual correspondences, then use the estimated pose to measure the average of normalized symmetric epipolar distance, and finally computes the ratio of correct estimates as *%Recall*. A pose is defined as correct for its distance error is below a certain threshold (0.05 as default). Besides, FM-Bench also reports intermediate results such as the inlier ratio (*%Inlier*/*%Inlier-m*) and correspondence number (*%Corr*/*%Corr-m*) after/before RANSAC.

Comparisons with other methods. As we can observe in Tab.3, for the *Recall* metric, MLIFeat is superior to other methods in T&T and CPC dataset, which are scenes with wide baseline, and it is slightly inferior to ASLFeat in TUM and KITTI dataset, which are scenes with short baseline. Since the baseline of image pair in TUM and KITTI dataset is short[20], image from one to another does not

Table 3. Evaluation results on FM-Bench[20] for pair-wise image matching, where *Recall* denotes the percentage of accurate pose estimation(within the error threshold 0.05), *Inlier* and *Inlier-m*, *Corrs* and *Corrs-m* denote the inlier ratio and correspondence number after/before RANSAC. The results of other methods come from the paper[19] except ASLFeat, ROOT-SIFT, and DSP-SIFT, which the are evaluated using their publicly released models with the default setting. The best and the second best are marked red and blue, respectively.

FM-Bench Dataset(error threshold @0.05)								
Methods	**TUM**[38](indoor SLAM settings)				**KITTI**[39](driving SLAM settings)			
	Recall	*Inlier*	*Inlier-m*	*Corrs(-m)*	*Recall*	*Inlier*	*Inlier-m*	*Corrs(-m)*
ROOT-SIFT[36]	58.40	75.33	62.46	68 (308)	**92.20**	98.34	91.11	158 (520)
DSP-SIFT[13]	55.60	74.54	56.44	66 (380)	**92.40**	98.22	87.60	154 (573)
HesAffNet+HardNet++[37]	51.70	75.70	62.06	101 (657)	90.40	98.09	90.64	233 (1182)
D2Net-MS[16]	34.50	67.61	49.01	74 (1279)	71.40	94.26	73.25	103 (1832)
R2D2 [18]	**57.70**	73.70	61.53	260 (1912)	78.80	97.53	86.49	278 (1804)
ASLFeat[19]	**59.10**	76.17	69.13	149 (742)	92.00	98.64	96.27	446 (1459)
SuperPoint *orig*[15]	45.80	72.79	64.06	39 (200)	86.10	98.11	91.52	73 (392)
SuperPoint *impl*	49.80	73.95	68.32	43 (193)	87.70	98.28	93.95	76 (367)
MLIFeat	52.90	74.10	67.29	65 (358)	89.10	98.25	95.07	140 (772)
	T&T[40](wide-baseline reconstruction)				**CPC**[39](wild reconstruction)			
ROOT-SIFT[36]	78.00	81.38	63.38	93 (756)	41.20	78.31	62.27	65 (369)
DSP-SIFT[13]	74.50	79.80	60.07	90(846)	34.00	75.83	56.29	58(367)
HesAffNet+HardNet++[37]	82.50	84.71	70.29	97 (920)	47.40	82.58	72.22	65 (405)
D2Net-MS[16]	68.40	71.79	55.51	78 (2603)	31.30	56.57	49.85	84 (1435)
R2D2 [18]	73.00	80.81	65.31	84 (1462)	43.00	82.40	67.28	91 (954)
ASLFeat[19]	**88.60**	85.56	79.08	297 (2070)	**52.90**	87.88	82.29	177 (1062)
SuperPoint *orig*[15]	81.80	83.87	70.89	52 (535)	40.50	75.28	64.68	31 (225)
SuperPoint *impl*	85.00	85.95	78.00	57 (491)	44.60	86.16	79.98	40 (273)
MLIFeat	**88.80**	86.21	78.63	103 (1006)	**53.50**	86.52	80.78	72 (535)

vary too much. The transformation between the image pair can be approximated to an affine transformation, which is precisely the advantage of ASLFeat whose descriptors are generated by the affine-constraint *DCN*[19].

In contrast, the image pair in T&T and CPC dataset exhibits large viewpoint and illumination changes. To correctly match the keypoints, it is required that the descriptors contain not only the local structural information but high-level semantics as well. Though FBM is not as powerful as *DCN* to extract the affine-invariant descriptors, the multi-level semantics fused descriptors are much more robust in these challenging wide-baseline dataset.

4.3 Visual Localization

In this section, we evaluate our MLIFeat and other methods under the task of visual localization[42, 18], where the goal is to retrieve the pose of an image within a given environment. In this benchmark, methods will face challenges such as day-night transitions and significant viewpoint changes between scene modeling

Table 4. Evaluation results on the Aachen-Day-Night dataset. We report the average feature number of each method, the descriptor's dimension, and the percentages of successfully localized images within three error thresholds. The best and second best are marked in red and blue, respectively. It can be observed that our MLIFeat achieves the best results within the most strict threshold.

Aachen-Day-Night Dataset					
Methods	**#Features**	**Dim**	**Correctly localized queries(%)**		
			$0.25m, 2^\circ$	$0.5m, 5^\circ$	$5m, 10^\circ$
ROOT-SIFT[36]	11K	128	49.0	53.1	61.2
DSP-SIFT[13]	11K	128	41.8	48.0	52.0
HesAffNet+HardNet++[37]	11K	128	52.0	65.3	73.5
D2Net-SS[16]	19K	512	72.4	**88.8**	**100**
D2Net-MS[16]	14K	512	75.5	**88.8**	**100**
R2D2[18]	10K	128	74.5	85.7	**99.0**
ASLFeat[19]	10K	128	**77.6**	**87.8**	98.0
SuperPoint[15] *Orig*	7K	256	73.5	79.6	88.8
SuperPoint *Impl*	7K	256	76.5	86.7	94.9
MLIFeat	7K	128	**78.6**	**88.8**	96.9

and image localization. It is particularly meaningful to evaluate each method's performance under this real-world application because it further reflects the local feature's robustness.

Dataset. The evaluation is conducted on the Aachen-Day-Night dataset[21]: For each of the 98 night-time images in the dataset, up to 20 relevant day-time images with known camera poses are given. After exhaustive feature matching between the day-time images in each set, their known poses are used to triangulate the scenes' 3D structure. Finally, these resulting 3D models are used to localize the night-time query images[16].

Evaluation protocols. We follow the public evaluation pipeline proposed in *The Visual Localization Benchmark*, which takes the custom features as input, then relies on COLMAP [43] for image registration, and finally generates the percentages of successfully localized images within three tolerances $(0.25m, 2^\circ)$ / $(0.5m, 5^\circ)$ / $(5m, 10^\circ)$. It is noting that the evaluation rule and tolerances are changed after recent updating in the website, and all of our results are based on the new rule.

Comparisons with other methods. The comparison results are illustrated in Tab.4. Consistent with the above evaluation, our MLIFeat outperforms other methods in the most strict tolerance. However, it is interesting to find that D2Net recovers all the query images' poses for the most relaxed tolerance$(5m, 10^\circ)$. On the one hand, D2Net is fine-tuned from the VGG pre-trained on ImageNet, making its descriptors implicitly contain much more semantics than others. On the other hand, the dataset MegaDepth used for fine-tuning D2Net is close to the scenes contained in Aachen. Therefore, despite having large keypoints localization error, the matched keypoints still belong to the same place, which ensures the recovery of poses within the most relaxed tolerance.

Analogously, when contrast SuperPoint *impl* and SuperPoint *orig* in Tab.4, there is an evident improvement from *orig* to *impl* (79.6 to 86.7 and 88.8 to 94.9). With the above analysis of D2Net, it is easy to find that such an improvement is mainly due to the extra MegaDepth training dataset. Since the scenes in MegaDepth are close to Aachen, the descriptors trained from MegaDepth perform much better in Aachen than that in other test datasets (HPathces and FM-Bench). Furthermore, it is interesting to find the DSP-SIFT, ROOT-SIFT and HardNet++ performs much worse in this task. It might due to the descriptors from these methods are extracted from the image patch, which lacks enough global semantics to handle large illumination changes(day v.s night). Thus, for the challenge localization task, to learn descriptors with rich semantics or add auxiliary semantic learning, e.g., classification, will both increase the accuracy of such a problem.

5 Conclusion

In this paper, we propose a novel deep model for multi-level information fusion based deep local features learning (MLIFeat), to cope with the image keypoints detection and description simultaneously. Two novel feature fusion modules, Feature Shuffle Module (FSM) and the Feature Blend Module (FBM), are cascaded to the commonly used encoder (SuperPoint backbone used in our paper). The Feature Shuffle Module can efficiently utilize the multi-level feature maps to detect the keypoints with high precision via the pixel shuffle operation. And the Feature Blend Module can make the full use of the multi-level feature vectors to generate the discriminative descriptors. To evaluate our model and other state-of-the-art methods, we have conducted extensive experiments, including image matching on HPatches and FM-Bench, and visual localization on Aachen-Day-Night. These evaluation results not only validate the effectiveness of our MLIFeat but also give insight into the performances of current methods under different tasks, which is beneficial to the development of the related algorithm.
Future work. To further improve the deep local feature's precision, better keypoints supervisory signals should be developed, as the current pseudo-ground label still contains noise. Besides, as analyzed above, additional semantic information should be embedded in the descriptors, enabling the model to handle more challenging scenarios.

Acknowledgement.

This work was supported in part by the National Natural Science Foundation of China (Nos. 91646207, 61620106003, 61971418, 61771026, 62071157 and 61671451).

References

1. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
2. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31** (2015) 1147–1163
3. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics **33** (2017) 1255–1262
4. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European conference on computer vision, Springer (2014) 834–849
5. Forster, C., Pizzoli, M., Scaramuzza, D.: Svo: Fast semi-direct monocular visual odometry. In: 2014 IEEE international conference on robotics and automation (ICRA), IEEE (2014) 15–22
6. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence **40** (2017) 611–625
7. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al.: Benchmarking 6dof outdoor visual localization in changing conditions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 8601–8610
8. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: Inloc: Indoor visual localization with dense matching and view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7199–7209
9. Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R., Robertson, N.M.: Ranked list loss for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 5207–5216
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60** (2004) 91–110
11. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision, Ieee (2011) 2564–2571
12. Alcantarilla, P.F., Bartoli, A., Davison, A.J.: Kaze features. In: European Conference on Computer Vision, Springer (2012) 214–227
13. Dong, J., Soatto, S.: Domain-size pooling in local descriptors: Dsp-sift. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 5097–5106
14. Schonberger, J.L., Hardmeier, H., Sattler, T., Pollefeys, M.: Comparative evaluation of hand-crafted and learned local features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1482–1491
15. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2018) 224–236
16. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint detection and description of local features. In: CVPR 2019. (2019)
17. Tian, Y., Fan, B., Wu, F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 661–669

18. Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., Humenberger, M.: R2d2: Repeatable and reliable detector and descriptor. arXiv preprint arXiv:1906.06195 (2019)
19. Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., Li, S., Fang, T., Quan, L.: Aslfeat: Learning local features of accurate shape and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 6589–6598
20. Bian, J.W., Wu, Y.H., Zhao, J., Liu, Y., Zhang, L., Cheng, M.M., Reid, I.: An evaluation of feature matchers for fundamental matrix estimation. arXiv preprint arXiv:1908.09474 (2019)
21. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image retrieval for image-based localization revisited. In: BMVC. (2012)
22. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 5173–5182
23. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. International Journal of computer vision **37** (2000) 151–172
24. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE transactions on pattern analysis and machine intelligence **27** (2005) 1615–1630
25. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. In: Advances in Neural Information Processing Systems. (2017) 4826–4837
26. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 11016–11025
27. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European Conference on Computer Vision, Springer (2016) 467–483
28. Ono, Y., Trulls, E., Fua, P., Yi, K.M.: Lf-net: learning local features from images. In: Advances in neural information processing systems. (2018) 6234–6244
29. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proceedings of the IEEE international conference on computer vision. (2017) 3456–3465
30. Christiansen, P.H., Kragh, M.F., Brodskiy, Y., Karstoft, H.: Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. arXiv preprint arXiv:1907.04011 (2019)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
32. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: 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. (2016) 1874–1883
33. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740–755
34. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2041–2050
35. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

36. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2012) 2911–2918
37. Mishkin, D., Radenovic, F., Matas, J.: Repeatability is not enough: Learning affine regions via discriminability. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 284–300
38. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE (2012) 573–580
39. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2012)
40. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36** (2017) 1–13
41. Wilson, K., Snavely, N.: Robust global translations with 1dsfm. In: European Conference on Computer Vision, Springer (2014) 61–75
42. Svärm, L., Enqvist, O., Kahl, F., Oskarsson, M.: City-scale localization for cameras with known vertical direction. IEEE transactions on pattern analysis and machine intelligence **39** (2016) 1455–1461
43. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4104–4113