Geometric Image Correspondence Verification by Dense Pixel Matching

Zakaria Laskar* Iaroslav Melekhov* Hamed R. Tavakoli[2] Juha Ylioinas
Juho Kannala
Aalto University, Espoo, Finland [2]Nokia Technologies, Espoo, Finland

Abstract

This paper addresses the problem of determining dense pixel correspondences between two images and its application to geometric correspondence verification in image retrieval. The main contribution is a geometric correspondence verification approach for re-ranking a shortlist of retrieved database images based on their dense pair-wise matching with the query image at a pixel level. We determine a set of cyclically consistent dense pixel matches between the pair of images and evaluate local similarity of matched pixels using neural network based image descriptors. Final re-ranking is based on a novel similarity function, which fuses the local similarity metric with a global similarity metric and a geometric consistency measure computed for the matched pixels. For dense matching our approach utilizes a modified version of a recently proposed dense geometric correspondence network (DGC-Net), which we also improve by optimizing the architecture. The proposed model and similarity metric compare favourably to the state-of-the-art image retrieval methods. In addition, we apply our method to the problem of long-term visual localization demonstrating promising results and generalization across datasets.

1. Introduction

Image retrieval is a well studied problem in the field of computer vision and robotics with applications in place recognition [5, 12, 36], localization [20, 36, 41], autonomous driving [24], and virtual reality [27] among many others. Given a query image, the image retrieval pipeline returns a ranked list of database images according to its measure of relevance to the query image. As raw pixels are not a good representation, extensive research has gone into finding discriminative and efficient image representations. The seminal work of Sivic and Zisserman [39] proposed Bag-of-Words based image representation using SIFT [21]. Later, more advanced and efficient representations were proposed in the form of VLAD [17] descriptors and Fisher vectors [31]. More recently, off-the-shelf [3, 18, 19] and fine-tuned [1, 11, 28] convolutional neural network (CNN) representations have demonstrated great success in image retrieval. The models encode an input image to a global vector representation which leads to efficient retrieval allowing to use just a dot product as a similarity measure to obtain relevant database images. Once fine-tuned on auxiliary datasets with similar distribution as the target one, those methods have achieved state-of-the-art image retrieval performance [1, 11, 28]. However, the main limitation of such fine-tuned CNN representations are their generalization capabilities which is crucial in the context of city-scale localization where the database images can be quite similar in structure and appearance. Moreover, variations in illumination (*e.g.* night time queries) or occlusion can significantly affect the encoded global representations degrading retrieval performance due to lack of spatial information.

Figure 1: Qualitative results of the proposed method for the task of image retrieval. The first row is a query taken at night-time with a mobile phone camera and the last row is a list of top-4 retrieved database images obtained by our method. All 4 are correct matches.

In this paper we leverage the advances of spatial geom-

*Equal contribution: firstname.lastname@aalto.fi

etry to obtain better ranking of the database images. To this end, we revisit the geometric verification problem in the context of image retrieval. That is, given an initial ranked list, L of database images returned by a CNN model (*e.g.* NetVLAD), we seek to re-rank a shortlist $L' \in L$ of images by using dense pixel correspondences [26] which are verified by the proposed similarity functions. Previously, DGC-Net [26] has been successfully applied only to positive image pairs *i.e.* pairs with overlapping field of view. In this work we extend its applicability to verify positive and negative image pairs in the framework of geometric verification. That is, we demonstrate how dense pixel correspondence methods such as DGC-Net can be used to improve image retrieval by geometric verification.

In summary, the contributions of this work are threefold. First, we improve the baseline DGC-Net by constraining the matching layer to be locally and globally consistent. Second, we replace multiple decoders of the original DGC-Net architecture by the proposed universal decoder, which can be shared for feature maps in different layers of the feature pyramid of DGC-Net. Third, we formulate two similarity functions, which first rank the shortlisted database images based on structural similarity and then re-rank them using appearance based similarity.

2. Related work

This work is closely related to image retrieval and image matching tasks. We provide a brief overview of existing approaches below.

Image retrieval methods can be broadly categorized into two categories: local descriptors [7, 15, 16, 22, 39] and global representations [1, 11, 28]. The approaches of the first category are based on either hand-engineered features such as SIFT [21] or learnt CNNs descriptors on the task of local image patch matching [25, 43]. Similarly, global representations methods can be further categorized into traditional hand-designed descriptors such as VLAD [17], Fisher Vectors [31], Bag-of-Words [39] and CNN based methods [1, 3, 11, 28]. Babenko *et al*. [3] demonstrate that the performance of off-the-shelf CNN models pre-trained on ImageNet [8] fall behind traditional local descriptors. However, when trained on an auxiliary dataset, the performance improves over such hand-engineered descriptors [1, 11, 28].

In addition to the standard retrieval approaches, there are several methods that attempt to explain the similarity between the query and top ranked database images using a geometric model [6, 23, 41]. The geometric model is estimated by fitting a simple transformation model (*e.g.* planar homography) to the correspondence set obtained using local descriptors such as SIFT, or off-the-shelf CNN descriptors [41]. In this work, we also use pre-trained CNN descriptors. However, in contrast to [41] which uses exhaustive nearest-neighbor search in descriptor space, we model the similarity using a learnt convolutional decoder. Moreover, [41] only uses coarse correspondence estimate, while our similarity decoder allows fine high resolution pixel level correspondence estimation. This is particularly important in city scale localization due to subtle differences in an overall similar architectural style observed in this scenario (*c.f*. Fig. 7).

Image matching. This task relates to the optical flow estimation problem. Recently proposed optical flow methods [14, 40] utilize a local correlation layer that performs spatially constrained matching in a coarse-to-fine manner. DGC-Net [26] extends this process of learning iterative refinement of pixel correspondences using a global correlation layer to handle wide viewpoint changes in the task of instance matching. Such a global correlation layer for instance matching has been used to estimate geometric transformations [29]. Melekhov *et al*. [26] demonstrate that such a method falls behind dense correspondence approaches due to the constrained range of transformations estimated by [29]. Recently, Rocco *et al*. [30] propose locally and globally constrained matching network on top of the global correlation layer which leads to improvement in instance and semantic matching. However, such a global correlation layer can only provide coarse correspondence estimates.

3. Method overview

Our contributions are related to the two last stages of the following three-stage image retrieval pipeline: *1)* Given a query image, we retrieve a shortlist of relevant database images using a fast and scalable retrieval method based on representing images with a descriptor vector; *2)* We perform dense pixel matching between the query and each shortlisted database image in a pairwise manner using a correspondence estimation network; *3)* We determine a set of cyclically consistent dense pixel matches for each image pair and use them to compute a similarity metric, which provides the final re-ranking of the shortlist.

The particular architecture of the aforementioned retrieval pipeline used in this work is illustrated in Fig. 2. That is, we use NetVLAD [1] for the first stage, our own modified version of DGC-Net [26] for the second stage, and the proposed approach with a novel similarity metric for the third stage. Here NetVLAD is used for retrieval, but also other global image level descriptors could be used instead.

Our contributions related to stages *2)* and *3)* above are described in the following sections. The geometric verification method is presented in Section 4 and our modifications to the DGC-Net architecture are described in Section 5.

4. Geometric verification

Dense pixel correspondences produced by [26] do not take into account the underlying model explaining the 3D structure of the scene by the image pair. RANSAC [10]

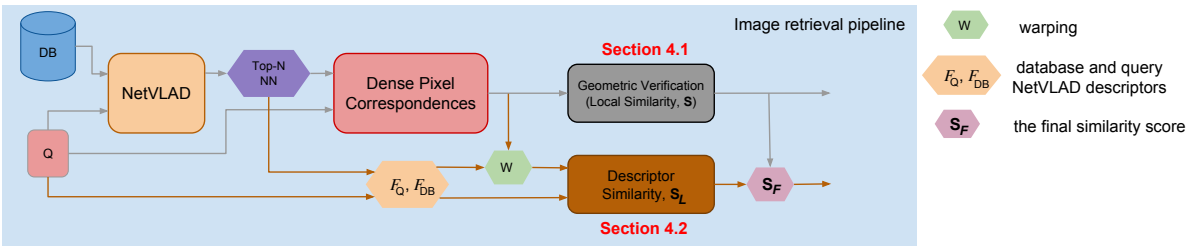


Figure 2: Overview of the proposed pipeline. Given a query image, we first rank the database images based on global similarity (*e.g.* using NetVLAD). In the next stage dense pixel correspondences are computed between the query and top N ranked database images. These correspondences are then verified by the proposed similarity functions utilizing geometry and CNN based image descriptors to re-rank database images according to the input query. See Sec. 4 and 5 for more details.

has been a popular method of choice to find the set of inliers from the whole correspondence set. However, dense pixel correspondences predicted by CNNs [26] are locally smooth due to the shared convolutional filters at different layers. As a result, RANSAC usually finds a large set of inliers even for non-matching image pairs. We propose two methods to eliminate these limitations in the following sections. That is, given an initial ranked shortlist L of database images based on global representation similarity with the query image, we re-rank a new shortlist $L' \subseteq L$ through a series of geometric verification steps (Sec. 4.1 and 4.2).

4.1. Cyclically consistent geometry

We propose a similarity cost function, S that combines RANSAC based geometric model estimation with cyclic consistency. Given a dense pixel correspondence map, $O \in \mathbb{R}^{H \times W \times 2}$, RANSAC outputs a set of inliers, $I \subseteq O$ w.r.t. to a transformation model (*e.g.* planar homography). We then estimate the subset of inliers that are cyclically consistent, $C \subseteq I$ using forward and backward correspondence maps predicted by our network (*i.e.* O^A and O^B). The cyclically consistent matches are those matches for which the combined mapping $O^A \circ O^B$ is close to an identity mapping. For geometrically dissimilar images, cyclic consistency constraint further constrains the number of inliers as the assumption here is that transformation model obtained by RANSAC may be inconsistent in forward and backward directions. We define this similarity function as follows

$$S = \frac{|C|}{|I|} \cdot exp\left(-\frac{\beta}{|C|}\right), \tag{1}$$

where β is a constant. As $|C|/|I|$ is a ratio, the exponential term is added to down-weight the similarity cost for image pairs which have less cyclically consistent correspondences in the inlier set. As β must be greater than $|C|$, we set it to 240x240 which is the maximum value of $|C|$ as our dense correspondence network (Sec. 5) operates on fixed size images of resolution 240x240. The similarity is computed in both directions, S^A, S^B and the final similarity is the maximum of the two values, $S = max(S^A, S^B)$. The shortlist L is re-ranked using S resulting in the new shortlist $\hat{L}$.

4.2. Global and local similarity

Using the geometry based similarity function S to re-rank the shortlist typically improves retrieval accuracy, but the retrieved list may still contain outliers as the global and local appearance similarity is not directly taken into account while computing S. Hence, the top-ranked database images in the geometrically verified shortlist $\hat{L}$ are passed through a second similarity function based on global and local descriptor similarity. The second similarity function is detailed below and more costly to evaluate, as it requires dense image feature extraction on high resolution images (*e.g.* 640x480 or higher) to obtain high resolution feature maps. On the other hand the dense correspondence estimation in Eq. 1 is performed on lower resolution images (240x240) and hence is significantly faster to compute. Therefore we have a two-stage re-ranking, where the second re-ranking is done only for a subset of top-ranked images from the first stage.

To obtain global dissimilarity G we use normalized global descriptors from a pre-trained network NetVLAD [1]. The network was originally trained to learn powerful representations for image retrieval. The Euclidean distance between the global representations is defined as the global dissimilarity value G. To compute local similarity, we extract hypercolumn [13] features from different NetVLAD layers (see Supplementary), L2 normalize and concatenate them along channel dimension. The final features are again L2 normalized resulting in feature maps, $\tilde{F}_A, \tilde{F}_B$, where $\tilde{F} \in \mathbb{R}^{H \times W \times Z}$, and $(H, W), Z$ are the image resolution and the final descriptor length. The local descriptor similarity S_L is then obtained as:

$$S_L = \sum_a \left(f_A^a \cdot f_B^a\right) m^a \tag{2}$$

where $\cdot$ denotes inner product, $f_A^a \in {}^w\tilde{F}_A$ and $f_B^a \in \tilde{F}_B$ are the hypercolumn NetVLAD features at location a in the

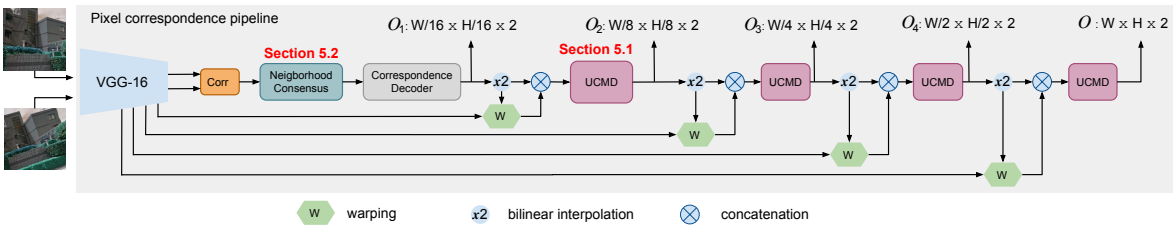


Figure 3: Overview of the dense pixel correspondence network. Pre-trained VGG-16 network is used to create a multiscale feature pyramid P of the input image pair. Correlation and neighborhood consensus layers use features from the top level of P to establish coarse pixel correspondences which are then refined by the proposed unified correspondence map decoder (UCMD). In contrast to DGC-Net [26] with multiple decoders, UCMD can be applied to each level of the multi-scale feature pyramid seamlessly leading to smaller memory footprint.

warped source $\tilde{F}_A$ and target feature map $\tilde{F}_B$, and $m^a \in M$, where M is the mask containing 1s at cyclically consistent pixels. Thus, Eq. 2 computes the cosine similarity between normalized warped source and target hypercolumn descriptors at cyclically consistent pixel locations.

The final similarity function between an image pair is a function of global dissimilarity and local similarities, G and S_L :

$$S_F = \log_{10}(S_L \cdot S) \cdot 10^{-G} \tag{3}$$

Here, local similarity score S_L is weighted by the similarity score S. We use S_F to re-rank the top-ranked images in $\tilde{L}$ to get the final shortlist L' for a given query. The *log* term is added as a normalization to balance the local and global scores. Although there are many possible ways to combine the local and global scores, we perform an extensive evaluation (see Supplementary) and show that the current form of these equations (1 and 3) achieves the best performance.

5. Pixel correspondence estimation

To obtain dense matching between two images we use a CNN network based on the architecture of DGC-net proposed by [26]. In this section, we provide two modifications to DGC-Net leading to more compact but effective model.

5.1. Unified correspondence map decoder

In general, DGC-Net consists of a series of convolutional layers and activation functions as an encoder E with M layers. An input image pair $I_A, I_B \in \mathbb{R}^{H \times W \times 3}$ is fed into the encoder independently to obtain a multi-resolution feature pyramid, $P = \{(F_A^l, F_B^l) | l = 1, 2, ...M\}$. Here $F^l \in \mathbb{R}^{H_l \times W_l \times N_l}$ is the feature map at the output of layer l of the encoder. The encoded feature maps at the top level of P, (F_A^M, F_B^M) are passed through a global correlation layer, that computes exhaustive pairwise features cosine similarity. The output of correlation layer is then passed through a decoder D_1 that estimates the initial correspondence map

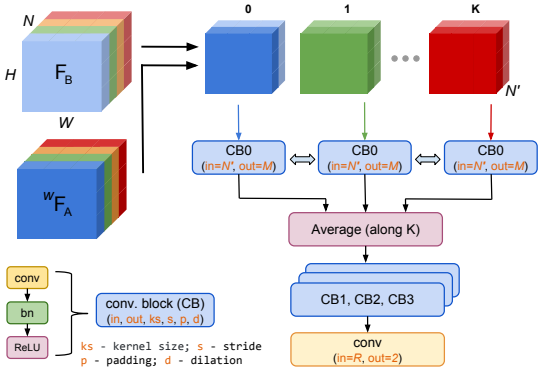


Figure 4: Overview of the unified correspondence map decoder (UCMD) D_c. The feature maps of the target F_B and the warped source ${}^wF_A$ images have been split into k tensors and then concatenated along the channel dimension. Further, each tensor is complemented by the correspondence map estimates $H \times W \times 2$ (expelled from the figure for clarity) and then fed into a convolutional block $CB0$ with N' inputs and shared weights. The output feature maps of $CB0$ are then averaged and processed by the remaining layers of the decoder to produce refined pixel correspondence estimates.

O_1 at the same resolution as F^M. O_1 is then iteratively refined by a series of decoders $D = \{D_2, D_3, ...D_M\}$ to obtain the final correspondence grid O_M at the same resolution as input images. Each decoder, $D_j \in D$ takes in as input $X_j = \{O_{j-1}, {}^wF_A^{M-j+1}, F_B^{M-j+1}\}$, where O_{j-1} is the upsampled correspondence map estimated by the previous decoder D_{j-1}, ${}^wF_A^{M-j+1}$ and F_B^{M-j+1} are the warped source and target feature maps at $l = M - j + 1$. However, since feature maps at level l of P have various number of channels, each decoder D_l has different structure which leads to increased memory costs.

In this work, we propose a unified correspondence map decoder D_c (UCMD) illustrated in Fig. 4. The unified decoder behaves like a recursive refinement function that op-

erates on feature maps across different layers l of P. More specifically, we divide the concatenated input feature maps in X_j into k_j non-overlapping components as shown in Fig. 4. We then propagate each of the k_j concatenated components $X_j^t, t = 1, ..k_j$ through the first convolutional layer (CB0) of our decoder, D_c. The resulting k_j feature maps at the output of CB0 are subsequently averaged and passed through the remaining layers to obtain refined correspondence estimates O_j.

The number of inputs of CB0 is $N' = 2Q + 2$, where Q specifies the number of channels in feature maps ${}^wF_A, F_B$ which are concatenated along the channel dimension. The additional 2 channels comprise of the upsampled coarser pixel correspondence map estimate from the previous layer of P. Therefore, k_l is given by N_l/Q where N_l is the dimensionality of the feature maps at the current layer l.

Inference. During the testing phase, apart from evaluating the trained network directly we additionally follow a second strategy. We infer the pixel correspondences by feed-forwarding each X_j^t through the complete decoder D_c resulting in k correspondence map estimates O^k. The process is applied to each level of the feature pyramid P. The mean $\mathbb{E}(O^k)$ is used as the final pixel correspondence map estimate. This formulation was not used during training as it did not lead to convergence.

5.2. Match consistency

The global correlation layer only measures the similarities in one direction *i.e.* from target to source image. However, many related works in the optical flow have shown that cyclic consistency allows the network to achieve better performance. In [30], a similar kind of global correlation layer was applied with cyclic consistency and neighborhood consensus to learn optimal feature correspondence. The idea is that matches should be consistent both locally and cyclically. That is nearby matches should be locally consistent and also the matches should be consistent in both forward and backward direction. Thereby, we integrated the Neighborhood Consensus Network (NCNet) [30] in our network. In contrast to original DGC-Net, the output of the correlation layer is now passed through NCNet with learnable parameters before being feed-forwarded through the decoders D_M and D_c to obtain dense pixel correspondences O. We refer to this network as `DGC-NC-UCMD-Net`.

6. Experiments

We discuss the experimental settings and evaluate the proposed method on two closely related tasks, *i.e.* establishing dense pixel correspondences between images (image matching) and retrieval-based localization.

Method	Viewpoint ID				
	I	II	III	IV	V
FlowNet2 [14]	5.99	15.55	17.09	22.13	30.68
PWC-Net [40]	4.43	11.44	15.47	20.17	28.30
Rocco [29]	9.59	18.55	21.15	27.83	35.19
DGC-Net [26]	1.55	5.53	8.98	11.66	16.70
DGC-NC-UCMD-Net	1.90	5.02	9.08	10.18	13.24
DGC-NC-UCMD-Net (avg. est.)	1.51	4.46	8.66	**9.59**	**12.62**
DGC-NC-Net	**1.24**	**4.25**	**8.21**	9.71	13.35

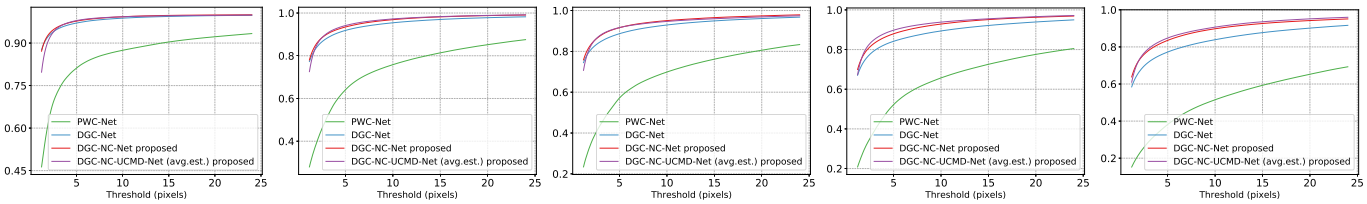
Table 1: AEPE metric for different viewpoint IDs of the HPatches dataset (lower is better).

6.1. Image matching

For this task we compare our approach with DGC-Net [26], which can handle strong geometric transformations between two views. We use training and validation splits proposed by [26] to compare both approaches fairly. More specifically, diverse synthetic transformations (affine, TPS, and homography) have been applied to Tokyo Time Machine dataset [1] to generate 20k training samples. Similarly to [26], the proposed network has been trained by minimizing $L1$ distance between the ground-truth and estimated correspondence map O_i at each level of the feature pyramid P (*c.f.* Fig. 3). Details of the training procedure are given in supplementary.

We evaluate our method on HPatches dataset [4] and report the average endpoint error (AEPE) of the predicted pixel correspondence map. HPatches dataset consists of several sequences of real images with varying photometric changes. Each image sequence represents a reference image and 5 corresponding source images taken under a different viewpoint with the estimated ground-truth homography $\mathbf{H}$. As predicting a dense pixel correspondence map is closely related to optical flow estimation, we provide AEPE for strong optical flow (OF) baseline methods, *i.e.* FlowNet2 [14] and PWC-Net [40] respectively.

We calculate AEPE over all image sequences belonging to the same Viewpoint ID of the HPatches dataset and report the results in Tab. 1. Here, DGC-NC-Net refers to the original DGC-Net architecture complemented by NC layer (Sec. 5.2) with a set of independent decoders at each level of the spatial feature pyramid P. Compared to DGC-Net, this model can achieve better performance reducing the overall EPE by 20% for the most extreme viewpoint difference between the reference and source images (Viewpoint V). According to Tab. 1, DGC-NC-UCMD-Net with one universal correspondence map decoder (Sec. 5.1) falls slightly behind of DGC-NC-Net (by 12% in average across all Viewpoint IDs) but it demonstrates advantages in terms of computation and memory costs (c.f. Sec. 6.3 and Supplementary). However, DGC-NC-UCMD-Net performance can be improved further if, at inference time, rather than averaging k feature maps produced by the first convolutional block of UCMD



(a) Viewpoint I (b) Viewpoint II (c) Viewpoint III (d) Viewpoint IV (e) Viewpoint V

Figure 5: PCK metric calculated for different Viewpoint IDs of the HPatches dataset. The proposed architectures (DGC-NC-*) substantially outperform all strong baseline methods with a large margin.

(*c.f*. Fig. 4) we average *predicted pixel correspondence estimates* for each input k feature map. We refer this model as DGC-NC-UCMD-Net (avg. est.).

In addition, we report a number of correctly matched pixels between two images by calculating PCK (Percentage of Correct Keypoints) metric with different thresholds. As shown in Fig. 5, the proposed DGC-NC-* models outperform DGC-Net by about 4% and correctly match around 62% pixels for the case where geometric transformations are the most challenging (Viewpoint V).

6.2. Localization

We study the performance of our pipeline in the context of image retrieval for image based city-scale localization. For evaluating the performance of our pipeline, we consider three localization datasets: Tokyo24/7 [42], Aachen Day-Night [34], and extended CMU-Seasons [34]. For all the datasets, we follow the same procedure outlined below. For a given query we first obtain a ranked list of database images, L based on Euclidean distance between their global NetVLAD representations, G. The top 100 ranked database images, $\hat{L} \subseteq L$ are re-ranked according to their geometric similarity score based on S. From these geometrically verified re-ranked database images, we pass the top 20, $L' \subseteq \hat{L}$ through the more expensive and stricter representation similarity function, S_F. Based on this final similarity, the final re-ranking is done on L'.

Localization metrics. The performance on the Tokyo24/7 dataset is evaluated using Recall@N, which is the number of queries that are correctly localized given N nearest-neighbor database images returned by the model. The query is considered correctly localized if at least one of the relevant database images is presented in the top N ranked database images. In contrast, the localization performance on Aachen Day-Night and extended CMU-Seasons is measured in terms of accuracy of the estimated query pose. The accuracy is defined as the percentage of queries with their estimated 6DOF pose lying within a pre-defined threshold to the ground-truth pose.

Tokyo 24/7. We compare the proposed approach with several strong baseline methods for place recognition. The

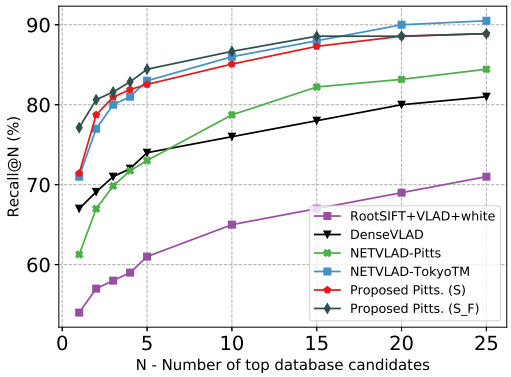


Figure 6: Comparison of the proposed methods versus state-of-the approaches for place recognition.

Methods	Recall		
	r@1	r@5	r@10
DenseVLAD [42]	67.1	74.2	76.1
NetVLAD-Pitts	61.27	73.02	78.73
NetVLAD-TokyoTM [1]	71.1	83.1	86.2
SIFT	73.33	80.0	84.4
Inloc [41]	62.54	67.62	70.48
Proposed (S) Pitts	71.43	82.54	85.08
Proposed (S_F) Pitts	**77.14**	**84.44**	**86.67**

Table 2: Localization performance on the Tokyo24/7 dataset (higher is better). Our proposed method outperforms Inloc and SIFT based geometric verification.

hand-crafted methods are represented by DenseVLAD [42] which aggregates densely extracted SIFT descriptors [21]. As our re-ranking method aims to improve the initial ranking by NetVLAD representations, we consider NetVLAD as a baseline. In particular, we use a publicly available PyTorch implementation of NetVLAD trained on Pittsburgh dataset (NetVLAD-Pitts). As a reference, NetVLAD-Pitts obtains $85.2/94.8/97.0\%$ compared to $84.1/94.6/95.5\%$ by NetVLAD [1] on Pitts-30k [1] validation set. In addition, we also consider Inloc [41] which uses dense NetVLAD descriptors in a geometric verification setting to obtain the final shortlist of ranked database images.

The Recall@N for the baseline methods are presented in

Methods	Condition, $5m, 10°$				
	Aachen Day-Night		CMU-Seasons		
	day	night	urban	suburban	park
HF-Net [33]	94.2	76.5	97.9	92.7	80.4
D2-Net [9]	93.4	74.5	-	-	-
Active Search [35]	96.6	43.9	-	-	-
NetVLAD-Pitts	81.7	64.3	78.9	77.0	63.2
Proposed	*84.7*	*68.4*	*89.1*	*77.1*	*63.3*

Table 3: Localization performance on the Aachen and CMU-Seasons datasets (higher is better). The best performance among *image retrieval* based approaches is highlighted as *italic*.

Fig. 6. Our geometric verification based pipeline achieves the state-of-the-art performance at Recall@1-10. The proposed approach significantly outperforms NetVLAD-Pitts and other baseline methods for all Recall@N thresholds (*c.f*. Tab. 2). Moreover, it is noteworthy that our method pushes the generalization performance of NetVLAD-Pitts above the NetVLAD-TokyoTM which was trained on images with similar distribution as Tokyo24/7. We also compared against traditional SIFT [2] based geometric verification which achieved 73.33% for Recall@1. We used COLMAP [37, 38] to extract SIFT features, followed by fundamental matrix based geometric verification to compute the inlier count.

Aachen Day-Night and Extended CMU-Seasons. Most localization systems involve an image retrieval stage where our proposed method can be directly applied. We did experiments on Aachen (day/night) and CMU Seasons datasets to show that our method retrieves more relevant database images compared to NetVLAD that leads to accurate query camera pose estimation. For each query, 20 images from the final shortlist produced by our method and NetVLAD were fed into a baseline localization pipeline, which uses a RANSAC PnP solver to register the query using 2D-3D matches (produced by performing 2D matching between the query and database images using our network and then utilizing known semi-dense point cloud for database images) and does hypothesis selection based on inlier count. We report the proportion of correctly localized queries for the threshold $(5m, 10°)$ in the following. Aachen day: 81.7% (NetVLAD-Pitts), **84.7**% (ours). Aachen night: 64.3% (NetVLAD-Pitts), **68.4**% (ours). CMU: 78.9% (NetVLAD-Pitts), **89.1**% (ours). Better accuracy in query camera pose estimation *given the same localization pipeline and image matching method* shows that our approach retrieves higher quality database images compared to NetVLAD-Pitts (*c.f*. Tab. 3). Our verification framework is generic and can be plugged in to other localization systems, such as HF-Net [33] and D2-Net [9].

Qualitative image retrieval results on Tokyo 24/7 and Aachen Day-Night datasets are illustrated in Fig. 7.

Methods	Recall		
	r@1	r@5	r@10
Proposed (S) Pitts	71.43	82.54	85.08
Proposed (S) (MNetv2 enc.) Pitts	73.02	81.9	85.4
Proposed (S_F) Pitts	**77.14**	**84.44**	**86.67**
Proposed (S_F) (MNetv2 enc.) Pitts	76.51	83.17	84.13

Table 4: **Ablation study**. Localization performance on the Tokyo24/7 dataset (MobileNetv2 decoder).

6.3. Ablation study

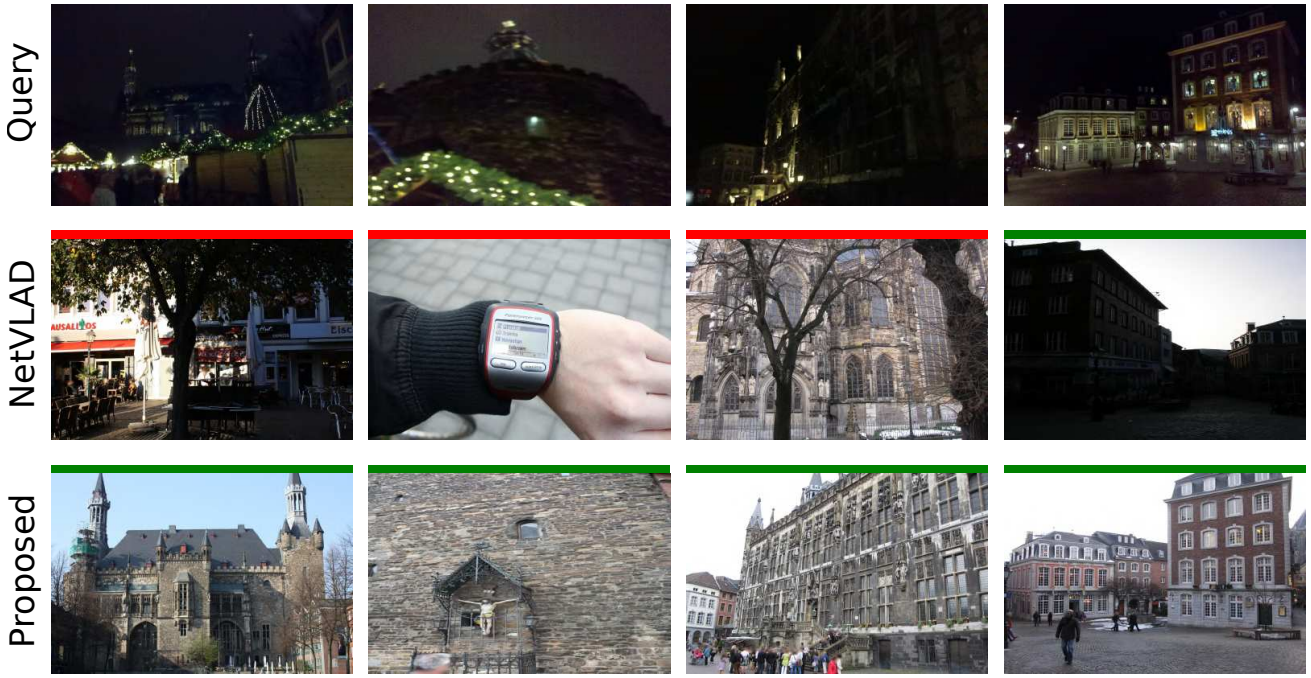
As the proposed UCMD decoder, D_c is defined as a refinement function operating on the space of representation similarity, it should be invariant to the representations themselves. This allows us to replace the VGG-16 encoder with a much light weight encoder MobileNetv2 (MNetv2) [32] at test time without further re-training while keeping the same decoder trained on the features produced by VGG-16. This leads to a highly compact model. In practice, localization problem is most relevant in the context of mobile devices, thus the model compactness is crucial also at test time. This led to comparable performance on the challenging Tokyo24/7 dataset (*c.f*. Tab. 4) and reduced the total number of network parameters from 8**M** (VGG16:$\sim$7**M**, UCMD:$\sim$0.9**M**) to 1**M** (MNetv2:$\sim$0.07**M**, UCMD:$\sim$0.9**M**). In the latter case UCMD provides notable memory savings compared to the original DGC-net (10**M**). One feed-forward pass through the DGC-NC-UCMD-Net with MNetv2 encoder requires 60ms compared to 80ms with VGG16 encoder providing savings in computation time.

7. Conclusion

We have presented novel methods for CNN based dense pixel to pixel correspondence learning and its application to geometric verification for image retrieval. In particular, we have proposed a compact but effective CNN model for dense pixel correspondence estimation using the universal correspondence map decoder block. Due to the universal nature of the decoder, we are able to obtain memory and computational savings at evaluation time.

In addition, we have integrated the matching layer in our model with neighborhood consensus [30] which further enhances the matching performance. This modified dense correspondence model along with the proposed geometric similarity functions are then applied to improve the initial ranking of database images given by NetVLAD descriptor. We have evaluated our approach on three challenging city-scale localization datasets achieving state-of-the-art retrieval results.

(a) Tokyo24/7



(b) Aachen Day-Night

Figure 7: **Qualitative results** produced by NetVLAD [1] (rows 2 and 5) and the proposed method (rows 3 and 6) on two localization datasets: Tokyo24/7 and Aachen Day-Night. Each column corresponds to one test case: for each query (row 1 and 4) top-1 (Recall@1) nearest database image has been retrieved. The green and red strokes correspond to correct and incorrect retrieved images, respectively. The proposed approach can handle different illumination conditions (day/night) and significant viewpoint changes (the second column in Fig. 7b). More examples presented in the supplementary.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. CVPR*, 2016. 1, 2, 3, 5, 6, 8

[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012. 7

[3] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky. Neural Codes for Image Retrieval. In *Proc. ECCV*, 2014. 1, 2

[4] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. CVPR*, 2017. 5

[5] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *Proc. CVPR*, 2011. 1

[6] O. Chum and J. Matas. Matching with PROSAC - progressive sampling consensus. In *CVPR*, 2005. 2

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. *Proc. ICCV*, 2007. 2

[8] J. Deng, W.Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 2

[9] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 7

[10] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6), 1981. 2

[11] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep Image Retrieval: Learning global representations for image search. In *Proc. ECCV*, 2016. 1, 2

[12] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *Proc. CVPR*, 2013. 1

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015. 3

[14] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proc. CVPR*, 2017. 2, 5

[15] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010. 2

[16] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *Proc. CVPR*, 2007. 2

[17] H. Jgou, M. Douze, C. Schmid, and P. Prez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010. 1, 2

[18] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional Weighting for Aggregated Deep Convolutional Features. In *Proc. ECCVW*, 2016. 1

[19] Z. Laskar and J. Kannala. Context aware query image representation for particular object retrieval. In *Proc. SCIA*, 2017. 1

[20] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proc. ICCVW*, pages 929–938, 2017. 1

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, Nov 2004. 1, 2, 6

[22] A. Makadia. Feature tracking for wide-baseline image retrieval. In *Proc. ECCV, year =*. 2

[23] J. Matas and O. Chum. Optimal randomized RANSAC with SPRT. In *ICCV*, 2005. 2

[24] C. McManus, W. Churchill, W. Maddern, A. D. Stewart, and P. Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *Proc. ICRA*, 2014. 1

[25] I. Melekhov, J. Kannala, and E. Rahtu. Image Patch Matching using Convolutional Descriptors with Euclidean Distance. In *Proc. ACCVW*, 2016. 2

[26] I. Melekhov, A. Tiulpin, T. Sattler, M. Pollefeys, E. Rahtu, and J. Kannala. DGC-Net: Dense Geometric Correspondence Network. In *Proc. WACV*, 2019. 2, 3, 4, 5

[27] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-dof localization on mobile devices. In *Proc. ECCV*, 2014. 1

[28] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *TPAMI*, 2018. 1, 2

[29] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, 2017. 2, 5

[30] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic. Neighbourhood consensus networks. In *Proc. NeurIPS*, 2018. 2, 5, 7

[31] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *IJCV*, 105(3):222–245, 2013. 1, 2

[32] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 7

[33] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. 2019. 7

[34] T. Sattler, W. Maddern, A. Torii, J. Sivic, T. Pajdla, M. Pollefeys, and M. Okutomi. Benchmarking 6DOF Urban Visual Localization in Changing Conditions. In *Proc. CVPR*, 2018. 6

[35] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. In *Proc. CVPR*. 7

[36] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *Proc. BMVC*. 1

[37] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[38] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 7

[39] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003. 1, 2

[40] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. CVPR*, 2018. 2, 5

[41] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *Proc. CVPR*, 2018. 1, 2, 6

[42] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 Place Recognition by View Synthesis. In *Proc. CVPR*, 2015. 6

[43] S. Zagoruyko and N. Komodakis. Learning to Compare Image Patches via Convolutional Neural Networks. In *Proc. CVPR*, 2015. 2