

# Domain Adaptation of Learned Features for Visual Localization

Sungyong Baik<sup>†2</sup>  
dsybaik@snu.ac.kr

Hyo Jin Kim<sup>1</sup>  
kimhyojin@fb.com

Tianwei Shen<sup>1</sup>  
tianweishen@fb.com

Eddy Ilg<sup>1</sup>  
eddyilg@fb.com

Kyoung Mu Lee<sup>2</sup>  
kyoungmu@snu.ac.kr

Chris Sweeney<sup>1</sup>  
sweeneychris@fb.com

<sup>1</sup> Facebook Reality Labs  
Redmond, WA, USA

<sup>2</sup> ASRI, Department of ECE  
Seoul National University  
Seoul, South Korea

---

## Abstract

We tackle the problem of visual localization under changing conditions, such as time of day, weather, and seasons. Recent learned local features based on deep neural networks have shown superior performance over classical hand-crafted local features. However, in a real-world scenario, there often exists a large domain gap between training and target images, which can significantly degrade the localization accuracy. While existing methods utilize a large amount of data to tackle the problem, we present a novel and practical approach, where only a few examples are needed to reduce the domain gap. In particular, we propose a few-shot domain adaptation framework for learned local features that deals with varying conditions in visual localization. The experimental results demonstrate the superior performance over baselines, while using a scarce number of training examples from the target domain.

## 1 Introduction

Visual localization is the problem of estimating the 6-DoF camera pose of a query image, given a set of reference images [67]. It is one of the key technologies for robot navigation [1, 37, 39], autonomous driving [50], and augmented and virtual reality [46, 51]. Visual localization is challenging because often the query and its corresponding reference images are taken in different viewpoints, illumination, time of day, weather, and seasons [40, 52, 58]. In addition, occlusions, ubiquitous objects (geographically prevalent objects, such as traffic lights), and transient objects (objects that move or change its appearance, such as cars or trees) can cause confusion in the localization process, making the problem even more difficult [30, 31, 33].

---

<sup>†</sup> Work done while at Facebook Reality Labs.

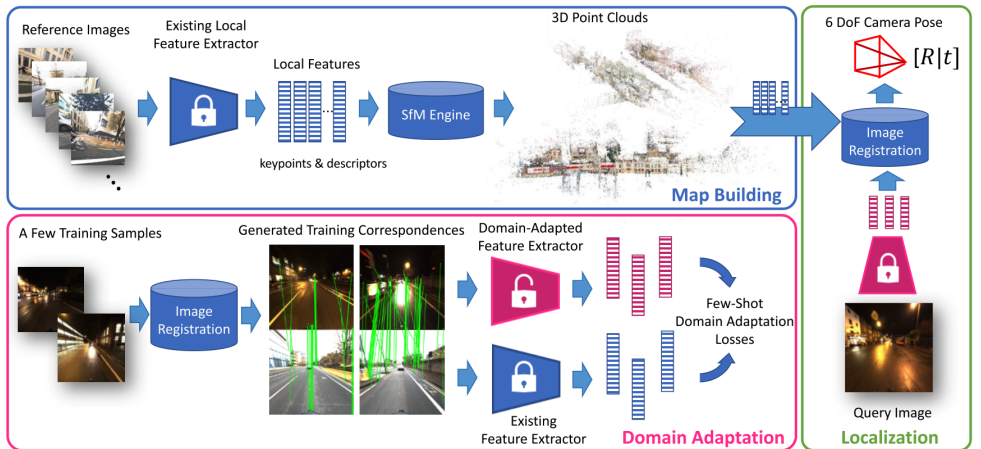


Figure 1: Overview of the proposed framework. See Sec 3.1 for a detailed description. The lock symbol indicates whether the network parameters are fixed or trained.

Recent learned local features [12, 13, 14, 15] based on deep neural networks have demonstrated increased robustness against photometric and geometric changes over traditional local features [6, 12, 19], owing to the higher representative power and the end-to-end training paradigm. Still, there exists a severe performance gap when the target image domain is different from the training image domain [16, 17]. For example, feature extraction models trained on sunny or daytime images do not perform well on snowy or nighttime images.

To tackle this problem, there has been a recent push on using image translation techniques to convert a source domain image to a target domain image [2, 19]. However, such approaches require a significant amount of training images. Collecting data for all possible conditions is difficult and cumbersome. Furthermore, deploying these methods requires an additional routing stage to apply the relevant translation model, which may introduce unrecoverable errors due to mis-classification [1, 20, 17, 14, 15].

In this work, we instead focus on a more common, practical scenario where only a few examples of target domain images are available, which is often the case in real-world scenarios. To this end, we propose a few-shot domain adaptation framework for learned local features. To the best of our knowledge, ours is the first work to explore few-shot domain adaptation of CNN-based local features in the context of visual localization. Our framework assumes that the reference representation stays fixed, and does not require re-extraction of features for the reference 3D models as a result of domain adaptation, which is computationally heavy. Together with the fact that the training match pairs are automatically generated based on the current representation, it opens up the possibility for continuous online training. We introduce three objective functions for this task, that can be used in addition to the correspondence loss between the target and the source domain descriptors. The first loss aligns the overall distribution of local descriptors from the source and target domains that belong to the same visual element. Our second loss provides more granular level of constraint by enforcing the pairwise distances between the examples in the source manifold subspace to be similar to that of the corresponding examples in the target subspace. Finally, we propose an auxiliary loss that penalizes the feature matches that conflict with the given pose.

We performed evaluation with varying number of training samples on the standard benchmark datasets for visual localization. The results demonstrate superior performance of our method over the baselines when only a few examples from the target domain are available.

We also provide the comparison of different combinations of the loss terms. Beyond the detailed innovations, our framework can be applied to any CNN-based local feature extractor.

## 2 Related Work

**Visual Localization.** Traditionally, visual localization is performed by registering the given query image to the 3D point cloud model reconstructed from a set of reference images through the Structure-from-Motion technique [82]. Specifically, 2D-3D matching is performed based on the local descriptors from the query image and the point cloud. Then, the camera pose is estimated from 2D-3D matches using a PnP solver [23]. Efficient matching can be done by prioritized matching [58, 52, 64] or by performing image retrieval to reduce the search space [65, 66, 57, 74]. This line of approach is often called the 3D-structure-based approach [66] and our method also belongs to this category. There are other streams of work such as the 2D-based approach, where the query pose is approximated using the visually similar reference images [3, 9, 110, 51, 53], and the regression-based methods that estimate the camera pose directly from the input image using deep neural networks [1, 8, 9, 76, 77, 83].

**Learned Local Features.** Traditional local features like SIFT [42] and SURF [6] have dominated the area for almost two decades, due to their generalization ability. Yet, these hand-crafted local features do not utilize the full representation power inherited in the data. There is more and more evidence showing that CNN-based learned local descriptors outperform traditional ones in a number of tasks [12, 43, 44] by a large margin. CNN-based local descriptor learning is typically formulated as a metric learning problem [22, 72, 88]. Different aspects of the problem, such as hard negative mining [53], adding regularization terms [75, 89], and incorporating context information [49] were also explored. There is also a recent trend to jointly optimize the detector and the descriptor [13, 14, 87]. SuperPoint [13] computes both keypoints and descriptors based on the notion of homographic adaptation. D2-Net [14] extends this idea by sharing and optimizing the parameters jointly for the detector and the descriptor. The above methods, though addressing learning local features in different perspectives, have fundamental drawbacks in domain generalization ability. In this paper, we address the challenge of reducing this domain gap.

**Domain Adaptation for Deep Neural Networks.** Domain adaptation methods typically aim to find a transformation that aligns the source and the target feature spaces [15, 73]. There is a large body of literature for shallow models [12, 19, 20, 25, 28, 54, 60, 77, 86]. For deep neural networks, it is traditionally achieved through fine-tuning [18]. However, fine-tuning often results in over-fitting due to the imbalance between the source and the target domain examples [79] and is not applicable when no target label is provided. To overcome this problem, several methods were proposed [47, 52, 61, 79, 81]. These include the unsupervised confusion losses based on the first- and the second-order statistics [21, 41, 73, 79], and adversarial losses by training domain classifiers [16, 17, 52, 71, 80, 81].

**Domain Adaptation for Visual Localization.** Our work is mostly related to recent work that focuses on reducing the domain gap between the training and the test images [2, 56, 59]. Porav *et al.* [59], use CycleGAN [91] to transform a source domain image to a target domain image (e.g., night-to-day, winter-to-summer), to enhance the local feature matching. In a similar spirit, Anoosheh *et al.* introduced ToDayGAN [2] to convert the images from nighttime to daytime to improve the image retrieval stage with DenseVLAD [78]. Mueller *et al.* [66] use view synthesis to enhance data augmentation. However, all these methods require a significant number of training images and can moreover introduce undesired artifacts. Some methods utilize the modalities that are invariant to changes, such as semantic or depth

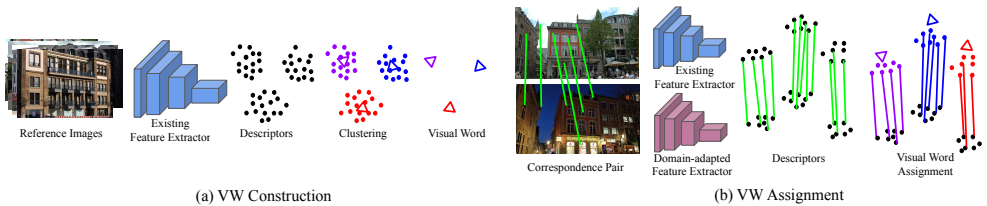


Figure 2: (a) Visual word construction: The visual words are obtained through K-means clustering on the source descriptors (Sec. 3.4). (b) Visual word assignment: The source and the target descriptors are extracted from a pair of reference and query images. The source descriptors are assigned to the closest visual words. The visual word assignment for a target descriptor is determined by that of its corresponding source descriptor, based on the training feature correspondences (Sec. 3.2.2-3.2.3).

information [65, 68, 69, 70, 76]. However, getting such modalities consistently from RGB inputs under changing conditions is challenging by itself unless such variances are reflected in the training data [66].

In this work, we propose a few-shot domain adaptation framework that presents a practical yet effective setting, where only few examples from the target domain are available. In this new setting, we adopt one of the classical domain adaptation losses, but use pseudo labels (coarse visual words) to further capture the structure of the data, based on the feature correspondences that we generate (Sec. 3.2.2). We also introduce two additional loss terms (Sec. 3.2.3-3.2.4) that are effective for few-shot domain adaptation of learned local features.

## 3 Proposed Method

### 3.1 Overview

The overview of our approach is shown in Fig. 1. We target the scenario where we are given a 3D point cloud generated from the set of reference images from a known source domain (e.g. daytime) and a few training images from unknown target domain (e.g. nighttime). The point cloud is built from the existing local feature representation and is fixed. The reason why it is kept fixed is because re-building the point cloud from new features takes a significant amount of time and is not feasible in real-world scenarios. When a few training images under an unknown condition are given, we register them to the 3D point cloud to generate training feature correspondences, which can be further refined when the ground-truth pose is given. For reliability of the training data, we only use images that are registered with inliers greater or equal to 15. We then use these correspondences to adapt the existing representation to the unknown target domain. Finally, the query images can be registered with better accuracy using the domain adapted features with only a small number of training samples.

### 3.2 Training Objective

This section describes how we adapt an existing local feature extraction network to a target domain with only few training examples. While our method is applicable to any CNN-based feature extractor, we use the state-of-the-art D2-Net [14] to assess our method in the paper.

#### 3.2.1 Correspondence Loss

As discussed in the overview (Sec. 3.1), we generate the training correspondence pairs from a small number of training images by registering them to the 3D point cloud. We use these training correspondence pairs to minimize the distance between the source and the corresponding target descriptors in the feature space (positive pairs) and do the opposite for irrel-

evant pairs (negative pairs). More details on generating the positive and the negative pairs can be found in Sec. 3.3. Specifically, we use the the detection-score-weighted triplet margin ranking loss proposed for D2-Net [4] as our correspondence loss:

$$\mathcal{L}_{\text{Corres}} = \sum_{i=1}^N \frac{s_i^S s_{i+}^T}{\sum_{j=1}^N s_j^S s_{j+}^T} \max(0, \|\mathbf{x}_i^S - \mathbf{x}_{i-}^T\|_2^2 - \|\mathbf{x}_i^S - \mathbf{x}_{i+}^T\|_2^2 + m), \quad (1)$$

where  $\mathbf{x}_{i+}^T$  and  $\mathbf{x}_{i-}^T$  are the positive and the negative target descriptors that correspond to the source descriptor  $\mathbf{x}_i^S$ , respectively.  $s_i^S$  and  $s_{i+}^T$  are the soft keypoint detection scores for the positive pair,  $\mathbf{x}_i^S$  and  $\mathbf{x}_{i+}^T$ ,  $N$  is the number of triplets, and  $m$  is the margin. One baseline that we compare against – the vanilla fine-tuning – uses the correspondence loss only.

### 3.2.2 Per Visual Word Correlation Alignment Loss (VW-CORAL)

However, the fine-tuning alone only leads to over-fitting when trained with very few examples, as we demonstrate in the experimental results (Table 1-3). To overcome this problem, we propose two additional loss terms to regularize the training of target descriptors. First, we adopt the CORAL loss [43] to minimize the difference in second-order statistics between the source and the target domain. Unlike the original CORAL loss, rather than blindly aligning the distributions over all descriptors, we align distributions per coarse visual word (Fig. 3(a)):

$$\mathcal{L}_{\text{VW-CORAL}} = \frac{1}{K} \sum_{k=1}^K \frac{1}{4d^2} \|\mathbf{C}_k^S - \mathbf{C}_k^T\|_F^2, \quad (2)$$

where  $\mathbf{C}_k^S$  and  $\mathbf{C}_k^T$  are the covariance matrix of the source and the target descriptors that belong to  $k$ -th visual word,  $\|\cdot\|_F$  is the Frobenius norm and  $d$  is the descriptor dimension. For each training correspondence pair, the visual word assignment of the source descriptor determines that of its corresponding target descriptor (Fig. 2(b)). Thus, our loss aligns the distribution of the source and target descriptors that *ought to* belong to the same visual word.

### 3.2.3 Cross-Domain Second Order Similarity Loss (CD-SOS)

We can further preserve the structure of the source domain in the target domain by considering the pairwise relationship between the descriptors in each domain. To this end, we introduce the cross-domain second-order similarity (CD-SOS) loss, inspired by the SOS regularizer [44]. The original SOS regularizer was proposed to aid metric learning by making distance between a descriptor pair to be similar to the distance between their positives. Similarly, our CD-SOS loss enforces the pairwise distances between training examples  $\mathbf{x}_i^S$  in the source domain, to be similar to the distances between the corresponding examples  $\mathbf{x}_{i+}^T$ 's in the target domain (Fig. 3(b)). Similar to VW-CORAL, we apply the CD-SOS loss per coarse visual word basis and observe better performance. Our CD-SOS loss then becomes:

$$\mathcal{L}_{\text{CD-SOS}} = \frac{1}{K} \sum_{k=1}^K \frac{1}{N_k} \sqrt{\sum_{j \neq i}^{N_k} (\|\mathbf{x}_{i(k)}^S - \mathbf{x}_{j(k)}^S\| - \|\mathbf{x}_{i(k)+}^T - \mathbf{x}_{j(k)+}^T\|)^2}, \quad (3)$$

where  $\mathbf{x}_{i(k)}^S$  and  $\mathbf{x}_{j(k)}^S$  indicates the source descriptors that are assigned to  $k$ -th visual word and  $\mathbf{x}_{i(k)+}^T$  and  $\mathbf{x}_{j(k)+}^T$  are their corresponding target descriptors.  $N_k$  is the number of such pairs.

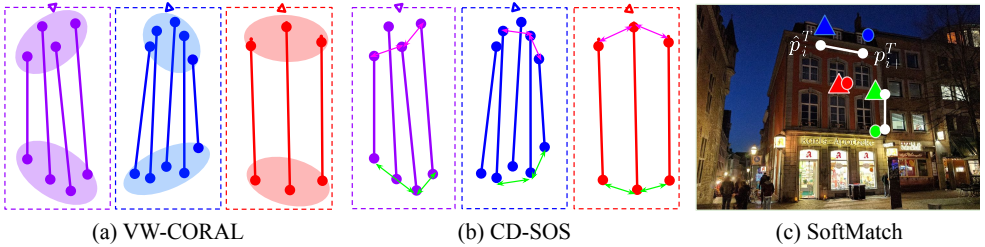


Figure 3: (a) VW-CORAL loss aligns the distribution of the source and target descriptors for each VW-based group (Sec 3.2.2). (b) CD-SOS loss enforces the pairwise distances of the target descriptors (green) to be similar to those of the source descriptors (magenta) for each VW-based group (Sec 3.2.3). (c) SoftMatch loss penalizes its matched keypoints  $\hat{p}_i^T$  in the target image  $I^T$  that are far from the actual corresponding keypoints  $p_{i+}^T$  (Sec 3.2.4). More details can be found in the supplementary material.

### 3.2.4 Soft Matching Loss (SoftMatch)

We observed that minimizing the correspondence loss does not always translate to achieving higher pose estimation accuracy. Therefore, we propose a loss that explicitly penalizes the feature matches that conflict with the given pose during training. To make the matching differentiable, we use the soft-argmax function by Luvizon *et al.* [45] to perform matching between the source and the target descriptors. For a source descriptor  $\mathbf{x}_i^S$  at keypoint  $p_i^S$ , we find the location  $\hat{p}_i^T$  of its match in the target image as follows:

$$\hat{p}_i^T = \text{SoftArgMax2D}(\mathbf{M}(\mathbf{x}_i^S)), \quad (4)$$

where  $\mathbf{M}(\mathbf{x}_i^S)$  is the 2D heatmap of the matching scores between a source descriptor  $\mathbf{x}_i^S$  and all detected local features  $(\mathbf{x}_j^T, p_j^T)$  in the target image  $I^T$ . We then compute the distance between the matched feature keypoint  $\hat{p}_i^T$  and its actual corresponding point  $p_{i+}^T$ , obtained as a result of reprojecting the 3D reference point to  $I^T$  using the generated training pose (or ground-truth pose, if available) (Fig. 3(c)). To summarize, our soft matching loss is defined as follows, where  $l$  is the diagonal length of the image and  $n$  is the number of matches:

$$\mathcal{L}_{\text{SoftMatch}} = \frac{1}{n} \sum_{i=1}^n \frac{1}{l} \|\hat{p}_i^T - p_{i+}^T\|_2. \quad (5)$$

Our overall loss function then becomes as follows, with  $\lambda_i$  denoting the weight for each term:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{Corres}} + \lambda_2 \mathcal{L}_{\text{VW-CORAL}} + \lambda_3 \mathcal{L}_{\text{CD-SOS}} + \lambda_4 \mathcal{L}_{\text{SoftMatch}}. \quad (6)$$

## 3.3 Generating Training Correspondence Pairs

To generate the positive training correspondence pairs for a given target image, we perform image registration against the 3D point cloud that was built from the reference images using COLMAP [68]. We used SIFT [42] features for this step in the experiments as it was slightly less noisy compared to the D2-Net [14] correspondences. While registration using the existing representation may only provide sparse and noisy correspondence pairs, we empirically show that our framework manages to use these sparse correspondences to improve the localization performance through domain adaptation. The negative pairs are generated via hard negative mining, following [14], which looks for the descriptor with the highest similarity score to the source descriptor, outside a local neighborhood of its positive pair.

### 3.4 Visual Vocabulary Construction

For each dataset, we extract local descriptors from randomly sampled reference images using the existing feature extractor (pre-trained D2-Net [14]), and perform K-means clustering to obtain visual words (Fig. 2(a)). In this paper, we used 1k images and sampled 3k descriptors, resulting in 3M descriptors for clustering. We evaluated the localization accuracy on the validation set using the different number of visual words ( $k = 32, 64, 128$ ), and empirically determined the number of clusters as 64. This part is done offline before training.

### 3.5 Implementation Details

For the baseline feature extractor, we use the pre-trained D2-Net [14] provided by the authors. Similar to how D2-Net was originally trained, we optimize the last layer (conv4\_3) and freeze other layers. For all our trained models, including fine-tuning, we used Adam [52] optimizer with learning rate of  $10^{-5}$ . To compensate for the sparsity of the generated ground-truth correspondences, we use a larger batch size of 10 and use full image resolution, whereas cropped images were used with batch size of 1 in [14]. We set the margin  $m$  for the correspondence loss to be 1. The weight for each loss term  $\lambda_i$  is determined such that the loss values are in a similar range and they sum up to 1. Specifically, we compute the means  $\mu_i$  and standard deviations  $\sigma_i$  of each loss term over the training set and set  $\lambda_i$  to be  $\frac{1}{4(\mu_i + 3\sigma_i)}$ .

## 4 Experimental Results

### 4.1 Evaluation Protocol

**Datasets.** For evaluation, we used the publicly available datasets including RobotCar Seasons [48] and Aachen Day-Night [67]. The Aachen Day-Night [67] dataset consists of  $\sim 4k$  reference images and 922 query images, where 98 images are captured in night, and the rest are daytime images. Because the Aachen Day-Night [67] dataset does not provide data splits for training and testing, we randomly divide the nighttime query images into disjoint sets and obtain 20 training, 16 validation, and 62 test images. No pose is given for the query images. Recently, Zhang *et al.* [90] have updated the dataset with more accurate ground-truth poses. In this paper, the updated ground-truth poses are used for evaluation.

RobotCar Seasons [48] provides  $\sim 32k$  images in total, where  $\sim 20k$  of them are reference images, and the rest are the query images are taken under nine different conditions. The reference images are provided with known poses, which are used for reconstructing the 3D model. We use a variant of the dataset called the CSC (Cross-Seasons Correspondence) RobotCar Seasons dataset [59] that provides  $\sim 5k$  training images with ground-truth poses and the total of 2,072 test images across the nine conditions. Because we consider few-shot domain adaptation, we only use a small subset ( $\sim 35$  images per condition) of available training images, similar to the number of training examples used for Aachen Day-Night [67].

**Evaluation Metric.** Following the standard protocol for above datasets [8, 48, 63], we evaluate the performance by measuring the pose error: the difference between the ground-truth pose ( $\mathbf{c}, \mathbf{R}$ ) and the estimated pose ( $\hat{\mathbf{c}}, \hat{\mathbf{R}}$ ), where pose consists of camera position  $\mathbf{c}$  and camera rotation matrix  $\mathbf{R}$ . The pose error consists of the position error  $\varepsilon_t = \|\mathbf{c} - \hat{\mathbf{c}}\|_2$  and orientation error  $\varepsilon_r = \arccos(\frac{1}{2}(\text{trace}(\mathbf{R}^{-1}\hat{\mathbf{R}}) - 1))$ .  $\varepsilon_t$  measures the Euclidean distance between the ground-truth and estimated positions, while  $\varepsilon_r$  measures the the minimum angle needed to align the axes of both rotations [24]. Finally, the pose accuracy is measured by the

Table 1: Recall on the subset under night condition from Aachen Day-Night [63]

Method	night
	.25/.50/5.0 (m) 2/5/10 (deg)
D2-Net [44] + ToDayGAN [0]	38.7 / 62.9 / 79.0
D2-Net [44]	<b>67.8 / 80.6 / 98.3</b>
fine-tune	<b>67.8 / 80.6 / 97.7</b>
<b>Ours</b>	<b>71.0 / 83.9 / 100</b>

Table 2: Overall performance on day and night conditions on CSC RobotCar Seasons [65]. The first and second best results are highlighted.

Method	all-day	all-night
	.25/.50/5.0 (m) 2/5/10 (deg)	.25/.50/5.0 (m) 2/5/10 (deg)
FGSN [66]	<b>61.6 / 93.5 / 99.7</b>	11.0 / 28.4 / 45.2
D2-Net [44]	59.3 / 89.5 / <b>98.5</b>	<b>20.0 / 39.8</b> / 49.4
fine-tune	59.5 / 89.5 / 98.3	19.2 / 39.6 / <b>50.9</b>
<b>Ours</b>	<b>61.0 / 89.6 / 98.5</b>	<b>20.9 / 42.6 / 52.1</b>

proportion of query images that are correctly localized within pose error thresholds of (1) 0.25 m and 2 deg, (2) 0.5 m and 5 deg, and (3) 5 m and 10 deg.

## 4.2 Quantitative Results

In Table 1, we compare the localization recall of our method with the baselines on Aachen Day-Night [67]. The baselines include fine-tuning which uses the correspondence loss (Eq. 1) only, the pre-trained D2-Net [44], and D2-Net on the night-to-day converted images using ToDayGAN [0]. Our method significantly outperforms the baselines while only using an average of 20 training images, whereas fine-tuning failed to show any improvement.

A similar trend is observed for the CSC RobotCar Seasons [65] dataset in Table 2, where our method consistently outperforms the pretrained D2-Net and fine-tuning. For this dataset, we also compare our results with the fine-grained segmentation network (FGSN) [66] with semantic match consistency (SSMC) [76]. Although we use much smaller portion (~6.7%) of the available training images compared to FGSN [66], our method yields much better recall on nighttime conditions, while achieving competitive performance on daytime conditions.

In Table 3, we compare performance of the proposed method and fine-tuning as we change the number of training images, for each weather condition in the CSC RobotCar Seasons dataset [65]. It shows that fine-tuning often performs worse than the pre-trained model when less training images are available. On the other hand, the proposed method consistently achieves better or comparable accuracy. It can be also seen that our method performs well for nighttime and winter conditions, where the domain gap is most significant.

We continue to compare the proposed method with ToDayGAN [0] on CSC RobotCar Seasons [65] in Table 4. ToDayGAN [0] also aims to reduce the domain gap by translating nighttime into daytime images, thereby greatly improving the image retrieval performance. However, the translated images contain artifacts that deteriorate the local features, which leads to a decrease in the registration performance. The proposed method, on the other hand, focuses on adapting the local features to the target domain, without the direct modification of images. While ToDayGAN [0] is trained on ~7k images, including 868 nighttime images from RobotCar Seasons, it is outperformed by our method that uses very few training images.

## 4.3 Qualitative Results

The qualitative results on the Aachen Day-Night [67] and the CSC RobotCar Seasons [65] datasets are shown in Fig. 4 and Fig. 5, respectively. In each figure, we visualize the inlier matches between a reference-query pair. We compare our method with the pre-trained D2-Net [44] and fine-tuned models. It can be seen that the proposed method improves the performance across diverse conditions, while the fine-tuning rarely improves the pre-trained model.



Table 3: Recall for each condition on CSC RobotCar Seasons dataset [35]. The second column lists the average number of training images  $I^T$  of target domain and that of image pairs ( $I^S, I^T$ ) across the different conditions. **first** and **second** best results are highlighted.

Method	train avg #img / #pair	night	night-rain	dawn	dusk	OC-summer
		.25/.50/5.0 (m) 2/5/10 (deg)	.25/.50/5.0 (m) 2/5/10 (deg)	.25/.50/5.0 (m) 2/5/10 (deg)	.25/.50/5.0 (m) 2/5/10 (deg)	.25/.50/5.0 (m) 2/5/10 (deg)
D2-Net [14]	-	21.9 / 42.6 / 51.7	17.9 / 36.6 / 46.8	55.1 / 90.7 / 99.3	76.4 / 94.3 / 100	38.3 / 85.5 / 94.4
fine-tune	13 /	21.1 / 42.3 / 53.2	17.1 / 35.4 / 45.5	55.1 / 90.7 / 99.3	74.5 / 94.4 / 100	37.0 / 84.2 / 95.7
<b>Ours</b>	305	<b>24.4 / 44.8</b> / 53.2	<b>19.1 / 40.6 / 47.2</b>	55.1 / 89.4 / <b>100</b>	<b>77.9</b> / 95.9 / 100	<b>39.5</b> / 85.5 / 95.7
fine-tune	20 /	22.6 / 42.3 / 52.8	17.1 / 37.4 / <b>47.2</b>	55.1 / 90.7 / 99.3	<b>77.9</b> / 94.4 / 100	37.0 / 85.5 / 95.7
<b>Ours</b>	610	<b>23.7</b> / 44.1 / <b>53.9</b>	<b>20.3 / 39.8</b> / 46.7	<b>56.4</b> / 90.7 / <b>100</b>	<b>79.4</b> / 95.9 / 100	<b>39.5</b> / 85.5 / 95.7
fine-tune	35 /	21.5 / 41.9 / 52.8	16.7 / 37.0 / <b>48.8</b>	55.1 / 90.7 / 99.3	76.4 / 95.9 / 100	38.3 / 85.5 / 94.4
<b>Ours</b>	1220	22.5 / <b>45.2 / 55.0</b>	<b>19.1 / 39.8 / 48.8</b>	<b>56.4</b> / 90.7 / 99.3	<b>79.4</b> / 95.9 / 100	<b>40.8 / 86.7</b> / 95.7

Method	train avg #img / #pair	OC-winter	snow	rain	sun
		.25/.50/5.0 (m) 2/5/10 (deg)	.25/.50/5.0 (m) 2/5/10 (deg)	.25/.50/5.0 (m) 2/5/10 (deg)	.25/.50/5.0 (m) 2/5/10 (deg)
D2-Net [14]	-	54.4 / 92.3 / 100	62.0 / 91.8 / 99.2	<b>79.2</b> / 94.5 / 100	53.1 / 77.8 / 95.3
fine-tune	13 /	54.4 / 92.3 / 100	63.2 / 91.8 / 99.2	76.5 / 94.5 / 100	51.8 / 76.6 / 96.3
<b>Ours</b>	305	<b>56.1</b> / 92.3 / 100	63.2 / <b>93.0</b> / 99.2	<b>79.2</b> / 94.5 / 100	<b>54.3 / 79.0</b> / 96.3
fine-tune	20 /	54.4 / 92.3 / 100	63.2 / 91.8 / 99.2	76.5 / 94.5 / 100	51.8 / 76.6 / 96.3
<b>Ours</b>	610	<b>57.7</b> / 90.7 / 100	<b>64.5</b> / 91.8 / 99.2	<b>77.9</b> / 94.5 / 100	<b>55.6 / 79.0</b> / 96.3
fine-tune	35 /	54.4 / 92.3 / 100	<b>64.5</b> / 91.8 / 99.2	<b>77.9</b> / 94.5 / 100	53.1 / 77.8 / 96.3
<b>Ours</b>	1220	<b>57.7</b> / 92.3 / 100	<b>64.5</b> / 91.8 / 99.2	<b>77.9</b> / 94.5 / 100	<b>54.3</b> / 77.8 / 96.3

Table 4: Comparison with ToDayGAN [2] on CSC RobotCar Seasons dataset [35]. We compare the localization recall with two image retrieval methods: DenseVLAD on original query images, and DenseVLAD on ToDayGAN generated images (ToDayGAN + DenseVLAD).

Registration	Retrieval	night		night-rain	
		ToDayGAN + DenseVLAD	DenseVLAD	ToDayGAN + DenseVLAD	DenseVLAD
ToDayGAN [2]		0.7 / 9.1 / 56.8	-	3.7 / 16.7 / 52.0	-
ToDayGAN + D2-Net		25.9 / 53.2 / <b>76.9</b>	19.7 / 35.3 / 51.7	22.0 / 49.6 / 61.0	14.6 / 31.3 / 42.7
<b>Ours</b>		<b>26.6 / 56.1</b> / 76.5	<b>22.6 / 45.2 / 55.0</b>	<b>28.0 / 54.5 / 66.7</b>	<b>19.1 / 39.8 / 48.8</b>

Table 5: Ablation study of different loss term combinations on CSC RobotCar Seasons [43]

	all day	all night	mean
$\mathcal{L}_{\text{Corres}}$ (fine-tune)	59.4 / 89.5 / 98.3	19.2 / 39.6 / 50.9	50.9 / 78.6 / 87.9
$\mathcal{L}_{\text{Corres}} + \mathcal{L}_{\text{VW-CORAL}}$	<b>61.0</b> / 89.5 / 98.5	20.6 / 41.1 / 50.0	52.4 / 78.9 / 87.8
$\mathcal{L}_{\text{Corres}} + \mathcal{L}_{\text{CD-SOS}}$	60.8 / <b>89.8 / 98.7</b>	<b>21.2</b> / 41.1 / 50.6	52.4 / 79.2 / 88.0
$\mathcal{L}_{\text{Corres}} + \mathcal{L}_{\text{SoftMatch}}$	60.5 / <b>89.8 / 98.7</b>	19.8 / 43.2 / 50.3	51.8 / <b>79.7</b> / 87.9
$\mathcal{L}_{\text{Corres}} + \mathcal{L}_{\text{VW-CORAL}} + \mathcal{L}_{\text{CD-SOS}}$	60.7 / 89.6 / <b>98.7</b>	20.7 / <b>43.4 / 52.1</b>	52.2 / 79.6 / <b>88.3</b>
$\mathcal{L}_{\text{Corres}} + \mathcal{L}_{\text{VW-CORAL}} + \mathcal{L}_{\text{CD-SOS}} + \mathcal{L}_{\text{SoftMatch}}$ ( <b>Ours</b> )	<b>61.0</b> / 89.6 / 98.5	20.9 / 42.6 / <b>52.1</b>	<b>52.5</b> / 79.4 / <b>88.3</b>

## 4.4 Ablation Study

We evaluate the effectiveness of each proposed loss term in Table 5, by applying them individually in addition to the correspondence loss in our proposed few-shot domain adaptation framework. Each proposed loss term provides significant performance improvement over the fine-tuning only. This indicates the effectiveness of regularization achieved by VW-CORAL and CD-SOS in adapting descriptors to the target domain, given only few examples. The performance improvement by SoftMatch also illustrates that it provides stronger supervision for pose estimation, compared to the correspondence loss. The combination of all loss terms, and the combination of VW-CORAL and CD-SOS provide the best overall performance.



Figure 4: Qualitative results on the Aachen Day Night dataset [63]: The inlier matches are visualized for each pair of the retrieved database image (left) and the the query image (right).

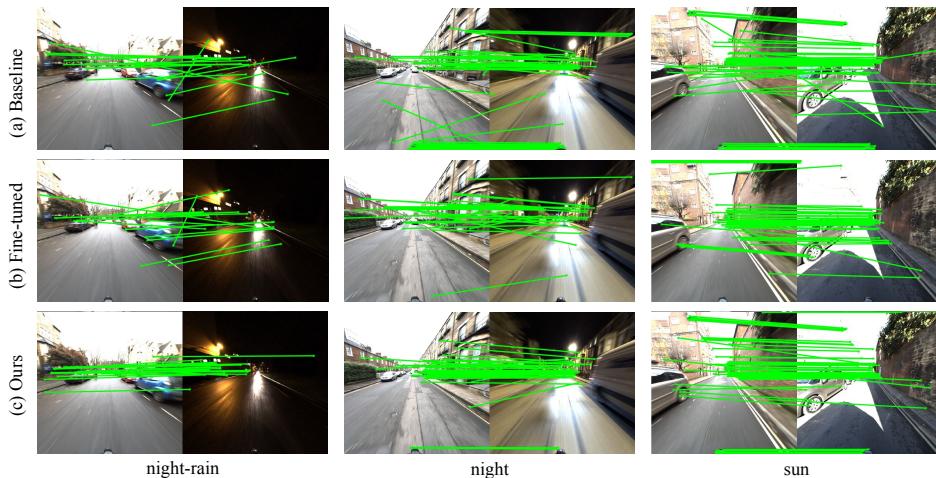


Figure 5: Qualitative results on the RobotCar Seasons Dataset [48]: The inlier matches are visualized for each pair of the retrieved database image (left) and the the query image (right).

## 5 Conclusion

In this paper, we investigate the domain gap between the training and the test images for visual localization. We propose a novel few-shot domain adaptation framework for learned local features to address this issue. Experimental results demonstrate that the proposed method works well on challenging scenarios where the training and test data have divergent modality. As the data in real world would always possess a distribution different from the training set, this work presents a extensible and practical solution to the visual localization problem. Future directions include online domain adaptation with streaming data.

**Acknowledgements** We would like to thank Torsten Sattler for providing CSC RobotCar Seasons dataset [65] and Asha Anooseh for providing the results for ToDayGAN [2].

## References

- [1] Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani. Network of experts for large-scale image categorization. In *ECCV*, 2016.
- [2] Asha Anooosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc van Gool. Night-to-day image translation for retrieval-based localization. In *ICRA*, 2019.
- [3] Relja Arandjelović and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *ACCV*, 2014.
- [4] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016.
- [5] Hernán Badino, D Huber, and Takeo kanade. Visual topometric localization. In *IV*, 2011.
- [6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.
- [7] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *CVPR*, 2018.
- [8] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017.
- [9] Samarth Brahmabhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018.
- [10] David Chen, Georges Baatz, K Koser, Sam Tsai, Ramakrishna Vedantham, Timo Pylvanainen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *CVPR*, 2011.
- [11] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6):647–665, 2008.
- [12] Hal Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2009.
- [13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018.
- [14] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *CVPR*, 2019.
- [15] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.
- [16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

- [17] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 2016.
- [18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
- [19] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.
- [20] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.
- [21] Arthur Gretton, Alexander J. Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. *Covariate shift and local learning by distribution matching*, pages 131–160. MIT Press, Cambridge, MA, USA, 2009.
- [22] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015.
- [23] Robert Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13:331–356, 1994.
- [24] Richard Hartley, Jochen Trunpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, 2013.
- [25] Judy Hoffman, Brian Kulis, Trevor Darrell, and Kate Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, 2012.
- [26] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017.
- [27] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015.
- [28] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.
- [29] Hyo Jin Kim and Jan-Michael Frahm. Hierarchy of alternating specialists for scene recognition. In *ECCV*, 2018.
- [30] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Predicting good features for image geo-localization using per-bundle vlad. In *ICCV*, 2015.
- [31] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *CVPR*, 2017.
- [32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [33] Jan Knopp, Josef Sivic, and Tomas Pajdla. Avoiding confusing features in place recognition. In *ECCV*, 2010.

- [34] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.
- [35] Mans Larsson, Erik Stenborg, Lars Hammarstrand, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. A cross-season correspondence dataset for robust semantic segmentation. In *CVPR*, 2019.
- [36] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *ICCV*, 2019.
- [37] Ang Li, Huiyi Hu, Piotr Mirowski, and Mehrdad Farajtabar. Cross-view policy learning for street navigation. In *ICCV*, 2019.
- [38] Yunpeng Li, Noah Snavely, and Daniel Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*. 2010.
- [39] Hyon Lim, Sudipta N Sinha, Michael F Cohen, and Matthew Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *CVPR*, 2012.
- [40] Chris Linegar, Winston Churchill, and Paul Newman. Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *ICRA*, 2016.
- [41] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.
- [42] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.
- [43] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *ECCV*, 2018.
- [44] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *CVPR*, 2019.
- [45] Diogo C Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85:15–22, 2019.
- [46] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *RSS*, 2015.
- [47] Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *CVPR*, 2019.
- [48] Will Maddern, Geoffrey Pascoe, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *International Journal of Robotics Research*, 36(1):3–15, 2017.
- [49] Jiri Matas, Ondrej Chum, Martin Urban, and Tomas Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

- [50] Colin McManus, Winston Churchill, Will Maddern, Alexander D Stewart, and Paul Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *ICRA*, 2014.
- [51] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *ECCV*, 2014.
- [52] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *ICRA*, 2012.
- [53] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor’s margins: Local descriptor learning loss. In *NIPS*, 2017.
- [54] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *NIPS*, 2017.
- [55] Arsalan Mousavian and Jana Košec̆ka. Semantically aware bag-of-words for localization. In *ICRA*, 2015.
- [56] Markus S. Mueller, Torsten Sattler, Marc Pollefeys, and Boris Jutzi. Image-to-image translation for enhanced feature matching, image retrieval and visual localization. In *ISPRS Annals of Photogrammetry*, 2019.
- [57] Venkatesh N Murthy, Vivek Singh, Terrence Chen, R Manmatha, and Dorin Comaniciu. Deep decision network for multi-class image classification. In *CVPR*, 2016.
- [58] Tayyab Naseer, Gabriel L Oliveira, Thomas Brox, and Wolfram Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *ICRA*, 2017.
- [59] Horia Porav, Will Maddern, and Paul Newman. Adversarial training for adverse conditions: Robust metric localisation. In *ICRA*, 2018.
- [60] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.
- [61] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019.
- [62] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, 2012.
- [63] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012.
- [64] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *ICCV*, 2015.
- [65] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *CVPR*, 2016.
- [66] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3d models really necessary for accurate visual localization? In *CVPR*, 2017.

- [67] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018.
- [68] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.
- [69] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *CVPR*, 2018.
- [70] Tianxin Shi, Shuhan Shen, Xiang Gao, and Lingjie Zhu. Visual localization using sparse semantic 3d map. In *ICIP*, 2019.
- [71] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *ICLR*, 2018.
- [72] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015.
- [73] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.
- [74] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018.
- [75] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sonet: Second order similarity regularization for local descriptor learning. In *CVPR*, 2019.
- [76] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *ECCV*, 2018.
- [77] Tatiana Tommasi and Barbara Caputo. Frustratingly easy nbnn domain adaptation. In *ICCV*, 2013.
- [78] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015.
- [79] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- [80] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015.
- [81] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

- [82] Shimon Ullman. The interpretation of structure from motion. *The Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.
- [83] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *ICCV*, 2017.
- [84] David Warde-Farley, Andrew Rabinovich, and Dragomir Anguelov. Self-informed neural network structure learning. In *ICLRW*, 2014.
- [85] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition. In *ICCV*, 2015.
- [86] Jun Yang, Rong Yan, and Alexander G Hauptmann. Adapting svm classifiers to data with shifted distributions. In *ICDMW*, 2007.
- [87] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016.
- [88] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.
- [89] Xu Zhang, Felix X Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *ICCV*, 2017.
- [90] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for visual localization via learned features and view synthesis. *arXiv preprint arXiv:2005.05179*, 2020.
- [91] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.