Adversarial Feature Disentanglement for Place Recognition Across Changing Appearance

Li Tang, Yue Wang, Qianhui Luo, Xiaqing Ding, Rong Xiong

***Abstract*— When robots move autonomously for long-term, varied appearance such as the transition from day to night and seasonal variation brings challenges to visual place recognition. Defining an appearance condition (e.g. a season, a kind of weather) as a *domain*, we consider that the desired representation for place recognition (i) should be domain-unrelated so that images from different time can be matched regardless of varied appearance, (ii) should be learned in a self-supervised manner without the need of massive manually labeled data, and (iii) should be able to train among multiple domains in one model to keep limited model complexity. This paper sets to find domain-unrelated features across extremely changing appearance, which can be used as image descriptors to match between images collected at different conditions. We propose to use the adversarial network to disentangle domain-unrelated and domain-related features, which are named *place* and *appearance* features respectively. During training, only domain information is needed without requiring manually aligned image sequences. Experiments demonstrated that our method can disentangle place and appearance features in both toy case and images from the real world, and the place feature is qualified in place recognition tasks under different appearance conditions. The proposed network is also adaptable to multiple domains without increasing model capacity and shows favorable generalization.**

I. Introduction

Visual place recognition is a vital task for mobile robots. Given sequences of images captured at different conditions, its goal is to find out pair of images corresponding to the same places. Feature extraction is one of the core problems in place recognition while changing appearance is a challenge. Conventional methods use features which are designed to be robust against small illumination variation like HOG [1] and SIFT [2], or use statistics of handcraft local features as global descriptors like DBoW2 [3] and VLAD [4]. However, they are not able to overcome extreme appearance changes, such as illumination changes from day to night and the visual difference in different seasons. Besides, although LiDAR-based methods [5], [6], [7] are shown to be more robust in place recognition under such changes, the high-cost prevents them from widely used.

Recent success in deep learning makes researchers start to study how to apply deep learning features in place recognition. Lots of efforts are devoted to supervised feature learning [8], [9], but the dependence on massive labeled data is hard to ensure. Thus, self-supervised methods like [10] are preferred in such a task. In their work, the whole feature is required to be invariant across different appearances. Meanwhile, some researches like [11] transfer query images to match the style of database images using neural networks. These methods show realistic transferred images, but they are only able to match the two domains used in the training phase. When a new domain comes, more models are needed. In addition, [11] extracts features using existing featurization tool [12]. We think such a process is indirect because style-transfer is unnecessary in place recognition.

Authors are with State Key Laboratory of Industrial Control Technology, Zhejiang University, 310012 Hangzhou, P.R.China. Yue Wang is the corresponding author. Rong Xiong is the co-corresponding author. {ltang,ywang24,qianhuiluo,xqding,rxiong}@zju.edu.cn.

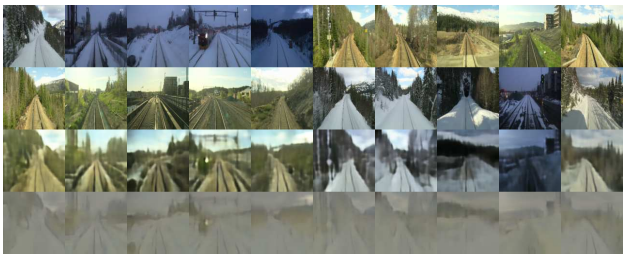


Fig. 1: Translated and zero-appearance images of Nordland. Row 1: input images from $\mathcal{D}_1$; row 2: input images from $\mathcal{D}_2$; row 3: translated images from $\mathcal{D}_1$ to $\mathcal{D}_2$; row 4: zero appearance images of $\mathcal{D}_1$. Column 1 to 5: $\mathcal{D}_1$ is winter and $\mathcal{D}_2$ is spring; column 6 to 10: $\mathcal{D}_1$ is spring and $\mathcal{D}_2$ is winter.

In this paper, we set to find a model that can disentangle *domain*-unrelated and *domain*-related features of an image, where the domain-unrelated feature is used for place recognition (Fig. 1). Specifically, a *domain* denotes a specific appearance condition, such as daytime in spring or nighttime in summer. This idea is motivated by the hypothesis that an image is composed of *place* and *appearance*, where the place is domain-unrelated content of a scene (e.g. corners or edges of buildings), while appearance is domain-related properties (e.g. brightness of sunlight and type of season). Under this hypothesis, we disentangle place and appearance features using two encoders, following the widely used autoencoder structure. To ensure that place features contain only domain-unrelated content, adversarial training is applied explicitly among place features. Besides, another adversarial loss is used to eliminate dependency between place features and appearance features. The model is trained in a self-supervised manner without manually aligned data. Finally, place feature is used as the descriptor in place recognition. Also, the network is shared across different domains, thus our model is adaptable to different domains without an increasing number of parameters. Main contributions of our method are listed as follows:

- A self-supervised feature learning method is proposed to disentangle the *place* and *appearance* features from the given image.
- The proposed architecture is designed to be trained with multiple domains without increasing model complexity.
- A toy case study on MNIST is carried out to validate our hypothesis. Besides, experiments are taken on two public datasets and show good performance. We also open the source code for reproduction [1].

The remainder of this paper is organized as follows: related work will be discussed and summarized in Sec. II and our method will be presented thoroughly in Sec. III. We will introduce the experiments in detail and show results in Sec. IV. Conclusion will be made in Sec. V.

II. Related Work

Place recognition has been studied for years. Typical pipeline for place recognition includes global feature extraction and matching, optionally followed by temporal fusion [13], among which this paper will focus on the first part.

Handcraft features for place recognition can be classified as global and local features. Traditional global-feature-based methods try to find appearance-invariant features, such as aggregating gradient information using histograms [1] and computing responses to artificial filters [14]. On the other side, traditional local-feature-based methods extract local features [15], [16] as representation and use different statistics strategies [3], [4] to obtain descriptors. However, none of them is found robust across all environments with extreme appearance changes, thus not preferable in place recognition.

As deep neural network achieves success in the computer vision area, researchers in robotics community start to explore how to integrate deep learning into existing research. [17] investigate discriminative ability for place recognition of different layers from AlexNet [18] pre-trained on ImageNet ILSVRC dataset [19], and they find that features from middle layers are robust against changing appearance. However, its performance is not good enough as reported in [10]. [20] use a classification network for location-specific place recognition and achieve comparable results. This work shows the potential for the neural network to distinguish places, but places are fixed once training is finished. Therefore, it cannot be extended to other scenes. To address this problem, [21] introduces a differential VLAD component into the neural network, namely NetVLAD, to compact a feature map into a descriptor vector, which is optimized by triplet loss. Labeled data from Google Street View Time Machine are needed to construct training tuple. [9] uses two datasets that are manually aligned to illustrate the power of supervision. However, these supervised methods need massive labeled data, which might be undesirable in fast deployment.

Another branch of machine learning methods, namely self-supervised learning, do not rely on aligned data. [22] apply PCA to raw images or CNN features to obtain invariant descriptors. It is reported that components with large

[1]https://github.com/dawnos/fdn-pr

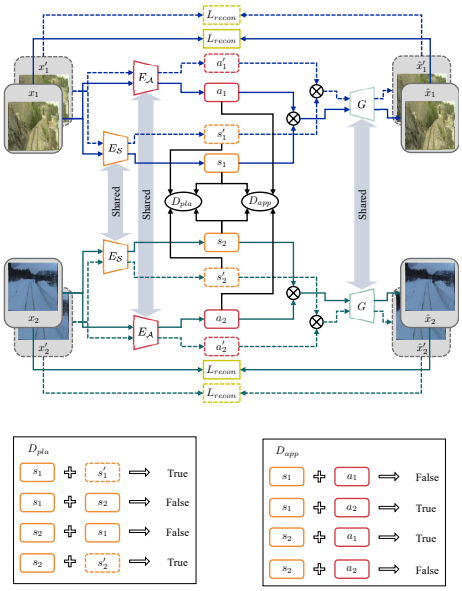


Fig. 2: Proposed network architecture. $E_{\mathcal{S}}$: place encoder; $E_{\mathcal{A}}$: appearance encoder; G: decoder; D_{pla}: place domain discriminator; D_{app}: appearance compatibility discriminator; s_i/a_i: place/appearance feature from domain i; $x_i/\hat{x}_i$: input/reconstructed image from domain i; $\otimes$: concatenate operation. Any symbol with superscript "′" indicates that it is sampled or generated from another image of the same domain.

eigenvalues represent variations in the image. By removing these components, the remaining information is suitable for place recognition. [23] use denoising autoencoder to learn features that can reconstruct original images, while [10] try to reconstruct HOG similarly. The output of encoders in these two methods exhibit robustness in place recognition. One advantage of work by [10] is that their training set is different from the testing set, demonstrating favorable generalization. Inspired by the work of style transfer [24], [25], some researchers try to transfer query images to match the style of database images (such as [11], [26], [27], [28]), following by local feature matching, global descriptor matching or dense matching. For example, [11] transfers nighttime images into daytime ones, and extract features using DenseVLAD [12]. Each model in these methods is targeted at the two domains used in the training phase. When adding new domains, new models are needed, and the number of models increases exponentially. [29] enhances the pretrained NetVLAD [21] with semantic information, which is shown to be viewpoint-invariant. It demonstrates that appearance-based descriptors, such as the proposed method, can be improved to overcome changing viewpoints with techniques in [29].

III. Adversarial Feature Disentanglement for Place Recognition

Under the hypothesis that an image is composed of place and appearance, our method uses the convolutional network as a feature extractor to disentangle place and appearance

features. Specifically, autoencoder is used as the backbone. An image is embedded into feature space, and the feature is transformed back to image space. In this paper, part of feature encodes *place* and other encodes *appearance*, which are disentangled into place space $\mathcal{S}$ and appearance space $\mathcal{A}$ respectively.

We begin from 2 domains, denoted by $\mathcal{D}_1$ and $\mathcal{D}_2$. As shown in Fig. 2, for any image $x_i \in \mathcal{D}_i (i = 1, 2)$, place encoder $E_\mathcal{S}$ and appearance encoder $E_\mathcal{A}$ are used to extract place feature $s_i = E_\mathcal{S}(x_i)$ and appearance feature $a_i = E_\mathcal{A}(x_i)$ respectively. Decoder G jointly leverages these two features to produce reconstructed image $\hat{x}_i = G(s_i, a_i)$ (Sec. III-A). To ensure domain-unrelated property of place latent space, *place domain discriminator* D_{pla} is introduced for adversarial training (Sec. III-B). Besides, *appearance compatibility discriminator* D_{app} is designed to eliminate the dependency between these two features (Sec. III-C).

A. Reconstruction Loss

As in many self-supervised feature learning literature, encoders ($E_\mathcal{S}$ and $E_\mathcal{A}$) and decoder G are used to reconstruct original input image cooperatively. To measure reconstruction quality, L2 distance is used, thus the overall reconstruction loss is expressed as:

$$\begin{aligned} L_{recon} = &\frac{1}{2}\mathbb{E}_{x_1 \sim p(x_1)} \left\| G(E_\mathcal{S}(x_1), E_\mathcal{A}(x_1)) - x_1 \right\|_2^2 + \\ &\frac{1}{2}\mathbb{E}_{x_2 \sim p(x_2)} \left\| G(E_\mathcal{S}(x_2), E_\mathcal{A}(x_2)) - x_2 \right\|_2^2 \end{aligned} \tag{1}$$

where x_i is image sampled from some prior distribution $p(x_i)$ of domain $\mathcal{D}_i$.

B. Place Domain Discriminator

Pure autoencoder cannot guarantee that the place feature only captures domain-unrelated information. It may contain appearance content, which is not penalized by reconstruction loss. To overcome this problem, we use adversarial learning, where place domain discriminator D_{pla} is introduced to constrain place features to lie in the same latent space. In each training iteration, two place features are extracted from two images, which may come from the same or different domains. D_{pla} tries to tell whether they are from the same domain, as shown in Fig. 2 (top right). For example, given place feature s_1 from domain $\mathcal{D}_1$, we sample the other two place features s_1' and s_2 from $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively. The loss function for this case can be written as

$$\begin{aligned} L_D^{pla,1} = &\frac{1}{2}\mathbb{E}_{s_1 \sim p(s_1), s_1' \sim p(s_1)}[(D_{pla}(s_1, s_1') - 1)^2] + \\ &\frac{1}{2}\mathbb{E}_{s_1 \sim p(s_1), s_2 \sim p(s_2)}[(D_{pla}(s_1, s_2) - 0)^2] \end{aligned} \tag{2}$$

where $p(s_i)$ is derived by $E_\mathcal{S}(x_i), x_i \sim p(x_i)$. The discriminator D_{pla} outputs 1 if the given two place features are from the same domain and 0 for those from different domains.

Simultaneously, place encoder $E_\mathcal{S}$ are encouraged to confuse D_{pla} by outputting place features following the same distribution across domains, so that they are invariant against varied appearance. The autoencoder is trained in an adversarial paradigm against D_{pla}, and its loss function can be expressed as:

$$\begin{aligned} L_{Adv}^{pla,1} = &\frac{1}{2}\mathbb{E}_{x_1 \sim p(x_1), x_1' \sim p(x_1)}[(D_{pla}(E_\mathcal{S}(x_1), E_\mathcal{S}(x_1')) - 0)^2] + \\ &\frac{1}{2}\mathbb{E}_{x_1 \sim p(x_1), x_2 \sim p(x_2)}[(D_{pla}(E_\mathcal{S}(x_1), E_\mathcal{S}(x_2)) - 1)^2] \end{aligned} \tag{3}$$

Eq. (2) and (3) use least square adversarial loss [30], where L2 loss is used without sigmoid function. It's worth noticing that $E_\mathcal{S}(x_1)$, $E_\mathcal{S}(x_1')$ and $E_\mathcal{S}(x_2)$ in Eq. (3) are in fact s_1, s_1' and s_2 in Eq. (2). We replace them to remind the readers that $E_\mathcal{S}$ is fixed when updating D_{pla} (Eq. (2)), while D_{pla} is fixed when updating $E_\mathcal{S}$ (Eq. (3)).

Eq. (2) and (3) are formulated for the case that the first input image is sampled from $\mathcal{D}_1$. When the first image is from $\mathcal{D}_2$, $L_D^{pla,2}$ and $L_{Adv}^{pla,2}$ and can be derived by exchanging two domains in Eq. (2) and (3).

C. Appearance Compatibility Discriminator

Only applying place domain discriminator is not enough. To see it, one can assume that output features of $E_\mathcal{A}$ might carry some information about the place. In this case, the combination of s_i and a_i are still able to reconstruct the original image, and D_{pla} is not affected. However, place and appearance features are not independent. To eliminate dependency of those features, we propose *appearance compatibility discriminator* D_{app} to tell if given place and appearance features are *independent*.

As images x_1 and x_2 sampled from different domains are independent, place feature $s_1 = E_\mathcal{S}(x_1)$ from $\mathcal{D}_1$ and appearance feature $a_2 = E_\mathcal{A}(x_2)$ from $\mathcal{D}_2$ are also independent. On the other hand, if place and appearance features are not disentangled, $s_1 = E_\mathcal{S}(x_1)$ and $a_1 = E_\mathcal{A}(x_1)$ from the same image x_1 are not independent. Now that we have positive and negative pairs for D_{app}, which can be leveraged to construct loss function for D_{app}:

$$\begin{aligned} L_D^{app,1} = &\frac{1}{2}\mathbb{E}_{s_1, a_1 \sim p(s_1, a_1)}[(D_{app}(s_1, a_1) - 1)^2 + \\ &\frac{1}{2}\mathbb{E}_{s_1 \sim p(s_1), a_2 \sim p(a_2)}[(D_{app}(s_1, a_2) - 0)^2] \end{aligned} \tag{4}$$

where $p(s_1, a_1)$ is given by $(E_\mathcal{S}(x_1), E_\mathcal{A}(x_1)), x_1 \sim p(x_1)$, and $p(a_2)$ is given by $E_\mathcal{A}(x_2), x_2 \sim p(x_2)$.

Similar to Sec.III-B, encoders $E_\mathcal{S}$ and $E_\mathcal{A}$ are encouraged to confuse D_{app} by outputting independent place and appearance features. When D_{app} fails, place and appearance features will be independent. Again, the loss function for autoencoder can be expressed as

$$\begin{aligned} L_{Adv}^{app,1} = &\frac{1}{2}\mathbb{E}_{x_1 \sim p(x_1)}[(D_{app}(E_\mathcal{S}(x_1), E_\mathcal{A}(x_1)) - 0)^2] + \\ &\frac{1}{2}\mathbb{E}_{x_1 \sim p(x_1), x_2 \sim p(x_2)}[(D_{app}(E_\mathcal{S}(x_1), E_\mathcal{A}(x_2)) - 1)^2] \end{aligned} \tag{5}$$

Similarly, we can have $L_D^{app,2}$ and $L_{Adv}^{app,2}$.

Like [10], we generate labels for Eq. (2)-(5) with unaligned data, thus we claim that the proposed method is self-supervised.

D. Extension: Multiple Domains Case

The autoencoder in our method is shared across domains. Besides, discriminators use information *between* two domains, and they are domain-unrelated. Thus, it is easy to extend to multiple domains. Assume that there are N domains, denoted by $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N$. In each training iteration, two domains $\mathcal{D}_i$ and $\mathcal{D}_j$ are randomly drawn, where $i, j = 1, 2, \cdots, N$ and $i \neq j$. Then images from $\mathcal{D}_i$ and $\mathcal{D}_j$ are sampled respectively from training set, which are input data of encoders (Eq. (3)). In testing phase, images from different domains are fed into $E_{\mathcal{S}}$ to obtain place features.

One advantage of our method is that only one model is needed in a specific area for long-term deployment. When collecting new data with different appearances in the same area, one can improve the model by retraining with new data without additional parameters. On the contrary, style-transfer-based methods need new models to transfer new data into known style. When new data come periodically, this will lead to exponentially increasing parameters. Thus, our method can be plugged into any long-term localization framework [31], [32] as the feature extractor.

E. Implementation and Training

Training Strategy During training, discriminators (D_{pla} and D_{app}) and autoencoder ($E_{\mathcal{S}}$, $E_{\mathcal{A}}$ and G) are updated alternatively by back-propagation:

$$\min_{D_{pla}} L_D^{pla,1} + L_D^{pla,2} \tag{6}$$

$$\min_{D_{app}} L_D^{app,1} + L_D^{app,2} \tag{7}$$

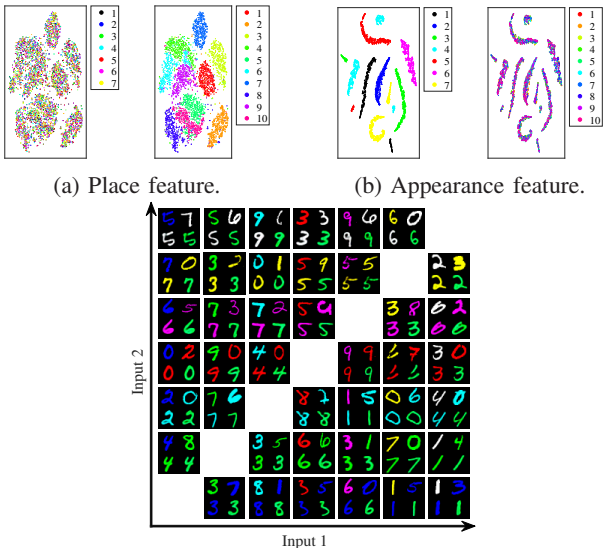
$$\min_{E_{\mathcal{S}}, E_{\mathcal{A}}, G} L_{recon} + \lambda_1 (L_{Adv}^{app,1} + L_{Adv}^{app,2}) + \lambda_2 (L_{Adv}^{pla,1} + L_{Adv}^{pla,2}) \tag{8}$$

where λ_1 and λ_2 are hyper-parameters to balance reconstruction and disentanglement. In case of two domains, images x_1, x_1', x_2, x_2' are drawn from two domains in each training iteration. For multiple domains, images are drawn as described in Sec. III-D.

Network Architecture Encoders ($E_{\mathcal{S}}$ and $E_{\mathcal{A}}$), decoder G and discriminators (D_{app} and D_{app}) are all convolutional networks. They are shared among different domains. The autoencoder is bottleneck architecture, where encoders downsample an image as two feature maps and the decoder jointly upsamples them back to the original resolution. The appearance feature is a vector of dimension $n_{\mathcal{A}}$. Input features of discriminators, (s_i, s_j) or (s_i, a_j), are concatenated together before going into the discriminators, and they are downsampled to one dimension as output.

Place Recognition To demonstrate the discriminative ability of the learned features, we use place features as descriptors in place recognition. Specifically, given a sequence of already observed images, we extract their place features by feeding those images into place encoder $E_{\mathcal{S}}$, constituting the database feature set $\{s_{DB,i}\}$, where $i = 1, \cdots, N_{DB}$ and N_{DB} is number of database images. When a new query image comes, the query feature s_Q is extracted from $E_{\mathcal{S}}$. Then, the best-matched image x_m in the database is determined by

$$m = \underset{i=1,\cdots,N_{DB}}{\arg\max} \left(\frac{s_Q}{\|s_Q\|} \cdot \frac{s_{DB,i}}{\|s_{DB,i}\|} \right) \tag{9}$$



(a) Place feature. (b) Appearance feature.

(c) Example translation between different domains. Each block contains the first input image (upper left), the second input image (upper right), the translated image (lower left) and the zero-appearance image (lower right). The first input images of the same column share the same domain, while the second input images of the same row share the same domain.

Fig. 3: Visualization of features and image translation between different domains of toy case experiment. Fig. 3a and Fig. 3b are visualization of *place* and *appearance* features using t-SNE respectively, where labels are colors (left, $1 \sim 7$) and digits (right, $1 \sim 10$).

IV. Experimental Results

We firstly carry out a toy case experiment to validate our hypothesis (Sec. IV-A). Later, experiments are conducted on two real datasets to illustrate its basic performance (Sec. IV-B) and generalization ability (Sec. IV-C) under place recognition scenario with two domains. The last experiment demonstrates an extension to multiple domains (Sec. IV-D).

A. Toy Case

We use MNIST, a handwritten digits dataset, to validate our hypothesis that image is composed of place and appearance. To simulate different domains, data images are colored randomly in 7 colors. In this case, place refers to the digit, while appearance refers to the color. Sample digits can be seen in Fig. 3c. Dimension of appearance feature $n_{\mathcal{A}}$ is 8. In training, λ_1 and λ_2 are set as 0.1.

We firstly investigate disentanglement by visualizing place and appearance features with t-SNE [33] (Fig. 3). As shown in Fig. 3a, place features lie in the manifold where digits can be distinguished easily, while colors are randomly distributed. On the other side, appearance features are clustered by colors, and they are unrelated to digits. These results

Method	Aligned?	Nordland		Alderley	
		AUC	Accuracy	AUC	Accuracy
DBoW2	✗	0.09	1.33%	0.00	0.22%
HOG	✗	0.17	17.0%	0.02	1.85%
[10]	✗	0.28	15.8%	0.10	1.26%
NetVLAD	✗	0.22	19.9%	0.02	2.65%
Ours	✗	**0.70**	**49.4%**	**0.34**	**21.0%**
[9]	✓	N/A[1]	92%	N/A[1]	7.82%
NetVLAD	✓	0.74	83.0%	0.13	15.8%

[1] Not Available (N/A) because they do not use AUC as criteria.

TABLE I: AUC and accuracy of different methods on Nordland and Alderley.

Methods	ToA[1]?	Night vs Day		Day vs Night	
		AUC	Accuracy	AUC	Accuracy
DBoW2	✗	0.00	0.22%	0.00	0.13%
HOG	✗	0.02	1.85%	0.02	0.43%
[10]	✗	0.10	1.26%	0.02	1.02%
[9]	✗	N/A[2]	1.73%	N/A[2]	N/A[2]
[11]	✗	0.01	0.6%	0.00	0.3%
NetVLAD	✗	0.04	5.56%	0.02	2.65%
Ours	✗	**0.13**	**8.28%**	**0.10**	**3.17%**
[9]	✓	N/A[2]	7.82%	N/A[2]	N/A[2]
NetVLAD	✓	0.16	19.2%	0.13	15.8%
Ours	✓	**0.39**	**23.7%**	**0.34**	**21.0%**

[1] Short for **T**rain **o**n **A**lderley.
[2] **N**ot **A**vailable because they do provide these results and the code.

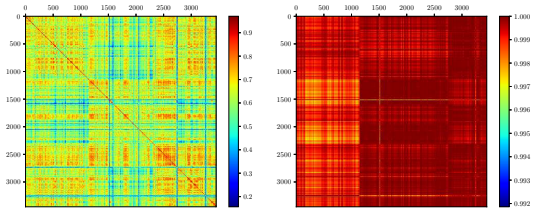
TABLE II: Generalization results. Models except for the last 3 rows do not see Alderley before testing. Results in the last 3 rows are trained on Alderley, and they are placed here for comparison. We also try different orders of input domains. For example, "Night vs Day" means $\mathcal{D}_1$ is nighttime and $\mathcal{D}_2$ is daytime.

illustrate that our method can effectively disentangle place and appearance features in this toy case.

To understand better what has place feature learned, the vector of appearance feature is set to zero, and the output of the decoder is displayed in Fig. 3c (called *zero-appearance image*). One can see that when the appearance feature is fixed, all decoded images have the same color. It illustrates that the decoupled place feature contains only place information (digit) of any input digit image. Fig. 3c also presents reconstructed images decoded from place and appearance features of different domains (called *translated image*). Specifically, the decoder combines the place feature from the first input image and appearance feature from the second input image to reconstruct digit. Results show that the digits of translated images are determined by place features, while colors are controlled by appearance features.

B. Basic Performance

We use *Partitioned Nordland Dataset* [34] and *Alderley Day/Night Dataset* [13] to validate our method in place recognition. Nordland is collected on a train in four seasons (spring, summer, fall, and winter), which are treated as four domains in this paper. Alderley is captured by a camera



(a) Place. (b) Appearance.

Fig. 4: Distance matrixes of place feature (left) and appearance (right) feature. Cosine distance is used.

mounted in a car in two sessions, one rainy nighttime, and one sunny daytime. Ground truths for both datasets are determined by GPS, and the partition of training and testing sets follows [9]. A match is considered correct if the distance between predicted and ground truth images is less than 3 frames, as used by [9]. Two criteria are used in quantity analysis: (i) AUC (Area Under Curve) and (ii) accuracy (overall true positive rate). The dimension of appearance feature $n_{\mathcal{A}}$ is 8. In training, λ_1 and λ_2 are set as 0.003 and 0.01 respectively.

To demonstrate basic performance, the first experiment is targeted at two domains case. For Nordland, winter and spring are chosen as $\mathcal{D}_1$ and $\mathcal{D}_2$. For Alderley, both domains are used, where nighttime and daytime are $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively. In testing, images from $\mathcal{D}_1$ are used to build a database, while those from $\mathcal{D}_2$ are seen as query images.

As a comparison, several methods are used to illustrate the discriminative ability of our learned features. We use the model provided by [10] without retraining on Nordland or Alderley, as done in their paper. Besides, the implementations of DBoW and HOG in [10] are used. For [9], results are obtained from their paper as they do not provide their code. For NetVLAD, we not only validate it by using the pretrained model from [21], but also retrain it on those two datasets for comparison. Results are shown in Tab. I. Our method shows large improvement over classical features including DBoW2 and HOG in both datasets. It is also better than the self-supervised method by [10]. The supervised methods like [9], [21] can obtain better performance in some datasets (such as Nordland), but it highly depends on the quality of ground truth. By watching images frame by frame, we find that two domains in Alderley dataset also have perspective differences, due to the motion characteristic of a car (such as changing lane). That's why our method exhibits higher accuracy than [9] and [21] in Alderley.

To validate that appearance features only contain place-unrelated information, we plot the distance matrix across appearance features of two domains in Nordland (Fig. 4b). As a comparison, we also plot the matrix of place features (Fig. 4a). It can be seen that the appearance feature shows no difference in different places, while the place features are more meaningful in the context of place recognition.

Similar to Sec.IV-A, some translated and zero-appearance images from Nordland dataset are present in Fig. 1. Although

$\mathcal{D}_1$ \ $\mathcal{D}_2$	Spring	Summer	Fall	Winter	Mean
Spring	-	0.86/**0.92**	0.92/**0.94**	**0.78**/0.73	0.85/**0.86**
Summer	0.86/**0.91**	-	0.98/**0.98**	0.68/**0.71**	0.84/**0.87**
Fall	0.88/**0.92**	0.98/**0.98**	-	0.66/**0.76**	0.84/**0.89**
Winter	**0.70**/0.59	**0.64**/0.58	**0.63**/0.63	-	**0.66**/0.60

(a) AUC.

$\mathcal{D}_1$ \ $\mathcal{D}_2$	Spring	Summer	Fall	Winter	Mean
Spring	-	70.3%/**77.7%**	78.3%/**81.3%**	**56.3%**/52.3%	68.3%/**70.4%**
Summer	70.5%/**77.8%**	-	96.1%/**96.5%**	44.8%/**46.7%**	70.5%/**73.7%**
Fall	76.9%/**80.1%**	96.2%/**97.0%**	-	43.7%/**50.2%**	72.3%/**75.8%**
Winter	**49.4%**/37.9%	**39.1%**/38.6%	**41.5%**/40.0%	-	**43.3%**/38.8%

(b) Accuracy.

TABLE III: Result of multiple domains case. Each item contains criterion (AUC or accuracy) of two/multiple domains. Number of parameters: 213.6M/17.8M (two/multiple domains).

the reconstructed images are not as realistic as in [11], it is acceptable because our goal is to obtain disentangled features instead of image reconstruction or style-transfer. From the last row, we can see that the learned place feature is unrelated to different appearance conditions.

C. Generalization Performance

We also experiment to see the generalization performance of our disentangled features. Models trained on datasets other than Alderley are used to perform place recognition on Alderley (Tab. II). Methods can be divided into 3 categories. The first kind of method (DBoW, HOG, NetVLAD, and [10]) is designed for general purposes. The second one ([11]) is targeted at the same appearance conditions (daytime and nighttime) but trained with data from other places from a different dataset (Robotcar [35]). The third one is trained in different appearance conditions and places.

It is found that our method achieves the highest performance. Models by [9], [10] and [21] try to learn unified features, which is, in fact, hard to fulfill across domains. Thus, when they are transferred to a new domain, the features are not as robust as ours. The accuracy of our method without training on Alderley is even slightly higher than the method by [9] with Alderley as the training set. This shows that our model is as robust as supervised learning without disentanglement. In comparison with [11], one can find that our method is much stronger. It is because [11] tries to find invariant representation in image space. Our method directly constrains the features, which is more useful in finding invariant features for place recognition. Also, style-transfer-based methods maybe not good for generalization, as error built on the image requires the quality of image details, which can be less useful for place recognition. These results verified the superior generalization of the proposed method.

D. Multiple Domains

To show that the proposed network is easy to adapt to multiple domains, four domains from Nordland are used to train a unified model, following the training strategy described in Sec. III-E. AUC and accuracy are computed for the testing set of every two seasons from Nordland using this model. Besides, to see benefits brought by fusing multiple domains, we also trained a two-domains model for every two seasons and compute those criteria.

From Tab. III, we can see that by fusing more domains, the model can get comparable AUC and accuracy in most of the time. We should notice that the model trained from 4 domains has the same number of parameters as one model trained from 2 domains. On the contrary, with only two-domain models (such as [24], [25]), we have to train 12 networks. Thus, by fusing more domains, our method can achieve comparable results without increasing the capacity of the model.

V. Conclusion

In this paper, we propose a self-supervised feature learning method to disentangle the place and appearance features, and the place feature is leveraged in the place recognition task. An autoencoder, including place encoder, appearance encoder, and decoder, is trained as a self-supervised feature extractor. To make place feature domain-unrelated, place domain discriminator is proposed. Besides, appearance compatibility discriminator is used to eliminate dependency between place and appearance features. We start from a toy case to illustrate the disentanglement effect. Experiments on real datasets show that the disentangled place feature is suitable for the place recognition task. It achieves comparable results to several existing methods. We also present the generalization ability of our method. An extension strategy is shown to prove that our method is easy to add new domains, which is also found to be beneficial to reduce model complexity without sacrificing place recognition performance.

VI. Acknowledgment

This work was supported by the National Key R&D Program of China (2018YFB1309300), the National Nature Science Foundation of China (U1609210, 61903332), and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[2] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *iccv*, vol. 99, no. 2, 1999, pp. 1150–1157.

[3] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[4] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*. IEEE Computer Society, 2010, pp. 3304–3311.

[5] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong, "3d lidar-based global localization using siamese neural network," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[6] L. He, X. Wang, and H. Zhang, "M2dp: A novel 3d point cloud descriptor and its application in loop closure detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 231–237.

[7] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.

[8] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli, "Learning context flexible attention model for long-term visual place recognition," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4015–4022, 2018.

[9] J. M. Facil, D. Olid, L. Montesano, and J. Civera, "Condition-invariant multi-view place recognition," *arXiv preprint arXiv:1902.09516*, 2019.

[10] N. Merrill and G. Huang, "Lightweight unsupervised deep loop closure," in *Proc. of Robotics: Science and Systems (RSS)*, Pittsburgh, PA, June 2018, (accepted).

[11] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool, "Night-to-day image translation for retrieval-based localization," *arXiv preprint arXiv:1809.09767*, 2018.

[12] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1808–1817.

[13] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1643–1649.

[14] A. C. Murillo, G. Singh, J. Kosecká, and J. J. Guerrero, "Localization in urban environments using a panoramic gist descriptor," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 146–160, 2013.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.

[17] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4297–4304.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[20] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep learning features at scale for visual place recognition," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3223–3230.

[21] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[22] S. Lowry and M. J. Milford, "Supervised and unsupervised linear learning techniques for visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 600–613, 2016.

[23] X. Gao and T. Zhang, "Unsupervised learning to detect loops using deep neural networks for visual slam system," *Autonomous robots*, vol. 41, no. 1, pp. 1–18, 2017.

[24] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in neural information processing systems*, 2017, pp. 700–708.

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks. arxiv," 2016.

[26] Y. Latif, R. Garg, M. Milford, and I. Reid, "Addressing challenging place recognition tasks using generative adversarial networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2349–2355.

[27] H. Porav, W. Maddern, and P. Newman, "Adversarial training for adverse conditions: Robust metric localisation using appearance transfer," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1011–1018.

[28] L. Clement and J. Kelly, "How to train a cat: learning canonical appearance transformations for direct visual localization under illumination change," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2447–2454, 2018.

[29] S. Garg, N. Suenderhauf, and M. Milford, "Semantic–geometric visual place recognition: a new perspective for reconciling opposing views," *The International Journal of Robotics Research*, p. 0278364919839761, 2019.

[30] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[31] L. Tang, Y. Wang, X. Ding, H. Yin, R. Xiong, and S. Huang, "Topological local-metric framework for mobile robots navigation: a long term perspective," *Autonomous Robots*, vol. 43, no. 1, pp. 197–211, 2019.

[32] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.

[33] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[34] D. Olid, J. M. Fácil, and J. Civera, "Single-view place recognition under seasonal changes," in *PPNIV Workshop at IROS 2018*, 2018.

[35] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.