






Feature Transformation for Cross-domain Few-Shot Remote Sensing Scene Classification

Qiaoling Chen¹ , Zhihao Chen¹ , and Wei Luo^{1,2}  

¹ South China Agricultural University, Guangzhou 510000, GD, China

² Pazhou Lab, Guangzhou 510330, GD, China

cswluo@gmail.com

Abstract. Effectively classifying remote sensing scenes is still a challenge due to the increasing spatial resolution of remote imaging and large variances between remote sensing images. Existing research has greatly improved the performance of remote sensing scene classification (RSSC) in recent years. However, these methods are not applicable to cross-domain few-shot problems where target domain is with very limited training samples available and has a different data distribution from source domain. To improve the model's applicability, we propose the feature-wise transformation module (FTM) in this paper. FTM transfers the feature distribution learned on source domain to that of target domain by a very simple affine operation with negligible additional parameters. Moreover, FTM can be effectively learned on target domain in the case of few training data available and is agnostic to specific network structures. Experiments on RSSC and land-cover mapping tasks verified its capability to handle cross-domain few-shot problems. By comparison with finetuning methods, FTM achieves better performance and possesses better transferability and fine-grained discriminability.

Keywords: Remote sensing scene classification · Few-shot learning · Cross-domain

1 Introduction

Remote sensing scene classification (RSSC) has attracted much attention in the field of optical remote sensing image processing and analysis in recent years, both due to the availability of high spatial-resolution images and its key role in wide applications, e.g., disaster detection [12], environmental monitoring [2], urban planning [37]. However, effectively classifying scenes from a newly obtained remote sensing image (RSI) is still nontrivial owing to the rich content brought by high-resolution, imaging conditions, seasonal changes and so on. Together with the difficulty of collecting sufficient labeled training samples, these factors make the robust-performance of RSSC a very challenging task.

To improve the performance of RSSC, deep learning methods [10, 15, 35] have been widely used in RSSC. The deep learning based RSSC methods made use of

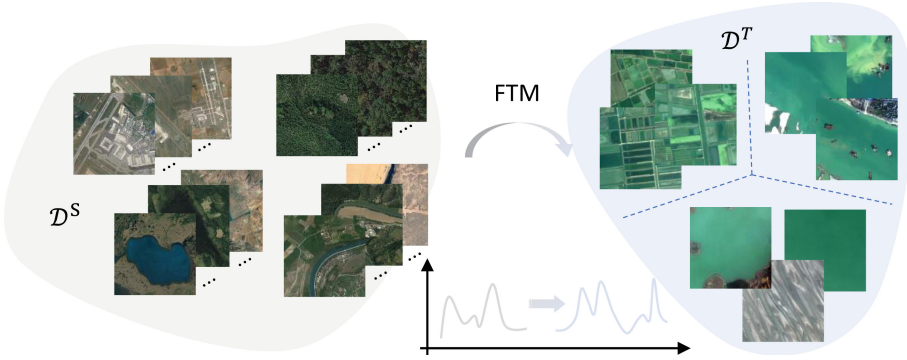


Fig. 1. Illustration of our motivation. Source domain \mathcal{D}^S has sufficient training samples for each class as shown here airplane, forest, lake, and river. Target domain \mathcal{D}^T may have different classes from \mathcal{D}^S and provides only few training samples for each class (here 3 samples for paddy field, river, and lake, respectively). As shown here, \mathcal{D}^S and \mathcal{D}^T have a significant domain gap. The proposed FTM tries to transfer the feature distribution learned on \mathcal{D}^S to matching that of \mathcal{D}^T by an affine transformation with a negligible number of additional parameters, thus improving the applicability of models learned on \mathcal{D}^S to cross-domain few-shot tasks.

the hierarchical network structure and feature abstraction ability of deep models to extract robust features for classification [3, 17, 26] and have achieved a great success, although they usually set aside the distribution differences between the training and testing data. While in a more realistic setting, the distribution difference was explicitly taken into consideration (under the framework of domain adaption) to build more applicable RSSC models like [23, 28, 46]. These methods usually require the same class distribution in the source and target domains. In addition, existing methods are almost all built on the prerequisite that sufficient training samples are available on target domain. This is, however, a very strict constraint on many real RSSC applications, especially in those target samples from a different distribution.

To address the difficulty of cross-domain RSSC tasks with few training samples, we propose a feature-wise transformation module (FTM) in deep CNNs with a two-stage training strategy. FTM borrows the idea from feature-wise linear modulation (FiLM) [29] but works in the unconditional setting and can be inserted in every convolutional layer. It attacks the cross-domain problem by transforming the distribution of features learned on source domain into matching that of target domain (see Fig 1). To achieve this, a pair of scale and shift vectors is applied to convolutional layers element-wisely. This pair of vectors, however, is not learned on source domain with the backbone network parameters, but instead trained on target domain without touching those already learned backbone parameters on source domain, which is different from [27, 29, 38] where the FiLM parameters are learned with the backbone network in an end-to-end manner. This two-stage training strategy can also alleviate the phenomenon of

overfitting on target tasks with few labeled training samples due to the parsimonious parameters involved in the second training stage. Generally, the two-stage training strategy and the parsimonious usage of parameters in FTM make it well adapted to scenarios with limited labeled training samples and class distribution mismatching between domains. We compare FTM with finetuning methods in this study and show its better prediction performance, transferability, and fine-grained discriminability. We notice that there is no existing work to deal with this problem in RSSC and we approach this problem in this study with the following contributions:

- We propose FTM for cross-domain few-shot RSSC. FTM transforms the feature distribution of source data into that of matching the target data via an affine transformation.
- We propose a two-stage training strategy in which only FTM parameters are involved in the second training stage on target tasks, thus alleviating the overfitting problem.
- We validate the effectiveness of FTM on two cross-domain few-shot RSSC tasks and demonstrate its applicability to land cover mapping tasks.

2 Related Work

Remote sensing scene classification (RSSC) has gained great progress in recent years since the publication of several benchmark datasets such as AID [43] and NWPU [5], which promote the application of deep models in RSSC. In early studies, researches focus on directly transferring deep features [26] or exploring deep network structures to utilize multi-layer [11, 17, 24] or multi-scale features [21, 22, 42] for classification, thus fully exploiting granularity information in RSIs [41]. Another line of research highlights the importance of local structures and geometries and proposes to combine them with global features for more discriminative representation [18, 19, 45]. Recently, the attention mechanism is further incorporated in selectively attending informative areas [40] or assigning objects with different weights for feature fusion [3]. In addition, nonlocal attentions are also studied to integrate long-range spatial relationships for RSSC [9]. Although the mainstream deep learning methods are absorbed quickly by the RSSC field and much progress has been achieved, these methods, however, are not applicable to the setting in this paper where the training and testing data have different distributions.

Few-shot learning (FSL) has attracted much attention in recent years where the target tasks have very few training samples available. To tackle this problem, a large-scale labeled dataset is usually supposed to be available for prior knowledge learning and the learned prior knowledge can be adapted to guide the model learning on target tasks, thus alleviating the overfitting problem in few-shot scenarios. The methodologies can be roughly grouped into three categories. The metric-learning based methods [33, 34, 39] target at learning an embedding space where an off-the-shelf or learned metric can be performed well. In contrast, the meta-learning based methods [8, 16, 31] aim to make the learned

model fast adapt to unseen novel tasks at the test stage. Recently, the finetuning based methods [4] report exciting results by exploiting multiple subspaces [20] or assembling multiple CNN features [6]. Meanwhile, FSL is also developed in settings like incremental learning [32, 36], cross-domain learning [30, 38], etc. However, very few works investigate FSL in RSSC while it is widely admitted as a practical problem in RSSC.

Domain adaption (DA) has gone through thorough studies and has been introduced to RSSC for a long time. The research on DA in RSSC mainly borrows ideas of existing DA approaches such as finetuning models on target domain [37], minimizing the maximum mean discrepancy between the source and target data distributions [28]. Specifically, [46] argues that the conditional distribution alignment is also important to cross-scene classification, thus they propose to combine the marginal and conditional distributions for more comprehensive alignment. To achieve fine-grained alignment, [47] tries to capture complex structures behind the data distributions for improved discriminability and reduce the local discrepancy of different domains to align the relevant category distributions. In addition, the class distribution misaligned problem is investigated in [23] by multisource compensation learning. Nevertheless, these methods assume sufficient training samples available on target domain. [44] studies the cross-domain task with limited target samples in RSSC, their training samples on the target domain is, however, orders of magnitude larger than ours.

3 Approaches

In this section, we propose FTM in deep CNNs that adapts the feature distribution learned on source domain to that of target domain. Assuming a well-labeled large-scale dataset and a newly acquired RS image with a small number of labeled samples annotated from it, we define two domains, the source domain \mathcal{D}^S and the target domain \mathcal{D}^T , respectively. The data of the two domains may from different classes, $\mathcal{C}^S \neq \mathcal{C}^T$ and $\mathcal{C}^S \cap \mathcal{C}^T \neq \emptyset$. Our approach first learns a backbone network on \mathcal{D}^S , and then adapts the backbone feature maps by FTM on \mathcal{D}^T without touching the backbone network parameters. In the following, we start by introducing FTM, followed by describing its training strategy, and then present the FTM network (Fig. 2).

3.1 Feature-Wise Transformation Module

Modern deep CNNs usually include BN [13] layers that reduce internal covariate shift and preserve feature distributions via a learned affine transformation for training efficiency. This operation inspires us to model different feature distributions by adjusting the feature map activations of a learned CNN, expecting it can perform well on a different domain with few training examples.

Supposing a backbone network has been trained on \mathcal{D}^S . Feature-wise transformation module (FTM) transforms the feature map by a pair of scale and shift vectors (γ, β) . Concretely, assuming the feature map of an input $\mathbf{X} \in \mathbb{R}^{3 \times H \times W}$

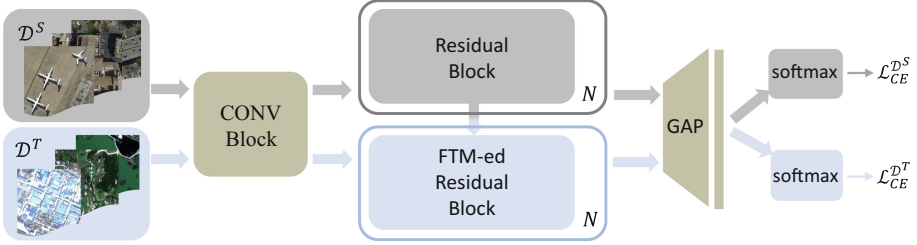


Fig. 2. Overview of the proposed FTM network. The detail of the FTM-ed residual block is depicted in Fig. 3(b). Our approach first trains a backbone network (shaded by gray blocks) on the source domain \mathcal{D}^S and then uses it to initialize a corresponding FTM network (shaded by blue blocks). The aligned parts between the two networks are then fixed and only the remained parts of the FTM network are learned on the target domain \mathcal{D}^T by $\mathcal{L}_{CE}^{\mathcal{D}^T}$. The light green blocks are shared. Best viewed in color. (Color figure online)

from the l -th layer is $\mathbf{f}^l \in \mathbb{R}^{C \times H' \times W'}$, FTM transforms the distribution of \mathbf{f}^l by modulating its activations:

$$\tilde{\mathbf{f}}_c^l = \gamma_c^l \odot \mathbf{f}_c^l + \beta_c^l, \quad (1)$$

where the subscript c represents feature channel indices and \odot means element-wise multiplication, $\gamma^l, \beta^l \in \mathbb{R}^C$ are learnable parameters. FTM approaches the change of distribution of \mathbf{f}^l by independently changing the activations of each feature channel. Compared to FiLM [29], where (γ, β) are generated by a conditioning network, FTM works in a unconditional setting and simply initializes γ and β to $\mathbf{1}$ and $\mathbf{0}$, respectively. Moreover, FTM is learned on target domain instead of source domain. By noting that the BN transform recovers feature activations through an affine operation, FTM further adapts it to a larger range and recovers the BN transform at $\gamma = \mathbf{1}$ and $\beta = \mathbf{0}$.

3.2 Optimization

To alleviate the overfitting phenomenon of deep CNNs with FTM on target domain with few labeled training samples, we study a two-stage learning strategy for optimization. Recalling that our target is transforming the feature distribution learned on source domain into that of target domain, we prefer to keep the backbone parameters unchanged and only train FTM on target data. To this end, we first optimize the backbone network by regular training on \mathcal{D}^S , then we fix the backbone network parameters and optimize FTM parameters $\{\gamma, \beta\}$ on \mathcal{D}^T through SGD.

Intuitively, we put FTM between the BN layers and nonlinear activations. It seems weird at first that applies two affine transformations – BN and FTM successively, but the separated mechanism can bring advantages to optimization and introduce different functions (see experiments). This operation, however, will

cause the shift of mid-level feature activations if we keep the backbone network parameters untouched, thus complicating optimization. To this end, we free the statistics of BN layers by making them adapt to input changes and leave the shift in activations to be compensated by $\{\gamma, \beta\}$.

3.3 The FTM Network

We instantiate our FTM network on the backbone of ResNet-34 [10]. It is worth noting that FTM is agnostic to specific CNN structures and we choose ResNet-34 just for simplicity. ResNet-34 includes one convolutional stem and 4 stages with each several residual blocks. Each residual block has two convolutional layers to form a shortcut connection. We construct the corresponding FTM network by inserting FTM after the BN layer of the second convolutional layer of the last residual block in one or several stages. For simplicity, we insert FTM after the BN layer of the last stage in ResNet-34 to illustrate its strength in this work. The transformed feature maps are then rectified by ReLU [25] and globally averaged pooled to be fed into a softmax function for classification. Figure 3 shows the FTM-ed residual block in conv5.

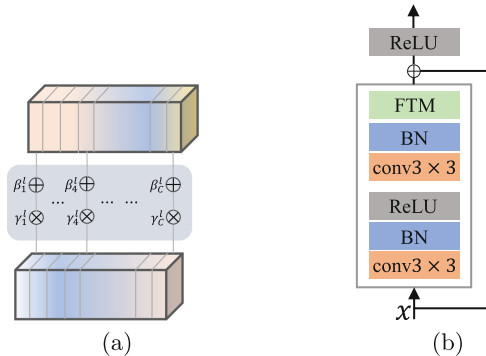


Fig. 3. (a) The shaded part is FTM, which operates on feature maps channel-wise. \otimes and \oplus represent element-wise multiplication and addition. (b) A FTM-ed residual block.

4 Experiments

In this section, we evaluate the transferability of the FTM network on two cross-domain few-shot applications: an RSSC task and a land-cover mapping task.

Table 1. Learning hyper-parameters of FTM and FT on \mathcal{D}^T .

	batch	epochs	lr	step	decay	opt
FT	64	50	0.001	15	0.1	Adam
FT-full	64	50	0.0001	15	0.1	Adam
FTM/FT-bn	64	50	0.003	15	0.1	Adam

4.1 Datasets

Two datasets – NWPU-RESISC45 [5] (hereinafter called NWPU) and AID [43] are separately employed as source domain \mathcal{D}^S in our experiments. Both of them are collected from Google Earth RGB images but with different pixel resolution ranges. NWPU images are with pixel resolutions ranging from 30m to 0.5m. It has 700 sizes of 256×256 images for each class with a total of 45 scene classes such as residential areas, river, and commercial areas. AID has 220 to 420 images with each of size 600×600 and pixel resolutions ranging from 8m to 0.5m in each class with a total of 30 classes, e.g., farmland and port. The two datasets have 19 classes in common that share the same semantic label. In addition, NWPU captures more fine-grained labels than AID. For example, the farmland class in AID is further divided into circular farmland, rectangular farmland, and terrace in NWPU.

The target domain data are from the R, G, and B channels of GID [37] multi-spectral images, which are collected from Gaofen-2 satellite with a spatial resolution of 4m. GID provides two subsets – a large-scale classification set (Set-C) and a fine land-cover classification set (Set-F). Set-C includes 150 and 30 training and validation images of size 6800×7200 with each pixel annotated into 5 coarse categories. Set-F has a subset of 6,000 image patches with train/val/testing 1500/3750/750 respectively. The image patches are of size 224×224 and belong to 15 fine-grained categories, which are subcategories of the 5 coarse categories of Set-C. Set-F is used as \mathcal{D}^T and Set-C is only used for land-cover mapping evaluation. We report the average performance over 5 trials on the RSSC tasks.

4.2 Implementation

We experiment with a FTM network based on the ResNet-34 backbone. The ResNet-34 pretrained on ImageNet [7] is first employed to learn on \mathcal{D}^S , where random crops of size 224×224 are used for training and 60 and 100 images from each class are kept for validation on AID and NWPU, respectively. We train ResNet-34 by Adam [14] on \mathcal{D}^S for 30 epochs with batch size 128, lr 10^{-4} , and decay lr by 0.1 every 10 epochs. After this stage, we select the best-performed one to initialize the FTM network, keep the aligned parameters fixed, and learn the remained parameters on \mathcal{D}^T for the few-shot RSSC tasks. The learning hyper-parameters are presented in Table 1.

For the land-cover mapping task, we classify every pixel into one of the 5 coarse classes by combining the output probabilities of subcategories that belong to the same coarse category.

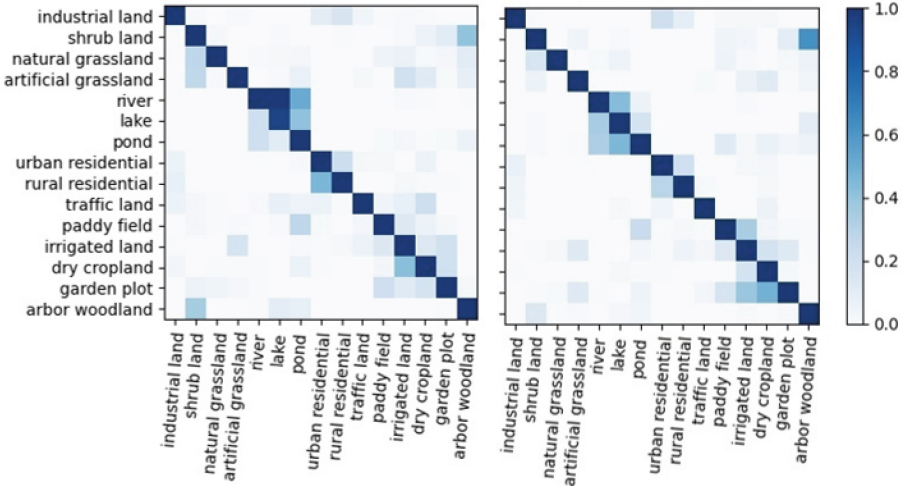


Fig. 4. Confusion matrices of FT-bn (left) and FTM (right) networks on the testing set of Set-F. Both networks are trained with 10 shots on Set-F.

Baseline: we compare FTM network with the finetuning (FT) method, which only finetunes the classification head of ResNet-34 trained on \mathcal{D}^S on \mathcal{D}^T . In addition, FT-bn additionally finetunes the last BN layer on FT. See Table 1 for learning hyper-parameters.

4.3 Experimental Results

RSSC Results. Table 2 and 3 compare the performance of FT, FT-bn and FTM under various available shots on \mathcal{D}^T . The results are obtained from the testing set of Set-F, and show that FTM improves the performance over FT (and FT-bn) by $> 3.1\%$ and $> 4.0\%$ on average respectively, demonstrating the advantages of FTM. Interestingly, the performance of FT-bn is only comparable to FT, lagging behind FTM apparently. This illustrates the different functions of BN and FTM in a residual block and signifies that additional affine transformation after BN can achieve additional effects that are beyond the effects brought in by BN. In addition, Table 2 and 3 illustrate that the performance of FT, FT-bn and FTM can be steadily improved with more training shots and the improvement of FTM over FT and FT-bn is relatively stable independent of the number of available training shots. These observations validate that FTM possesses the ability to transform the feature distribution learned on \mathcal{D}^S into that of target domain even with very limited training shots available on the target domain, thus alleviating the tendency to overfitting on target domain.

To understand which aspects of advantages brought by FTM, we make an analysis of the confusion matrices of FTM and FT-bn networks trained with 10 shots on Set-F and with \mathcal{D}^S NWPU in Fig. 4. It can be seen that FTM has a more concentrated diagonal distribution than FT-bn, indicating its better

Table 2. Accuracy on Set-F under different shots with \mathcal{D}^S NWPU ($\sqrt{\sigma} < 0.03$).

	3	5	10	15	20	30	50
FT	0.50	0.57	0.65	0.71	0.73	0.73	0.81
FT-bn	0.50	0.55	0.65	0.71	0.72	0.73	0.81
FTM	0.53	0.59	0.69	0.73	0.77	0.77	0.84

Table 3. Accuracy on Set-F under different shots with \mathcal{D}^S AID ($\sqrt{\sigma} < 0.03$).

	3	5	10	15	20	30	50
FT	0.51	0.53	0.63	0.67	0.71	0.71	0.79
FT-bn	0.50	0.57	0.64	0.67	0.71	0.71	0.79
FTM	0.55	0.59	0.67	0.73	0.75	0.75	0.83

classification performance, especially in those subcategories belonging to the same coarse category. The same phenomenon is also observed on Set-F with \mathcal{D}^S AID. Specifically, we find that FTM can well separate urban residential from rural residential and distinguish between river, lake, and pond effectively, which are respectively from the same coarse categories – built-up and water, while these are confused by the FT-bn. This signifies that FTM has the ability to transform the original feature space into a more delicate and discriminative space where the subtle differences between fine-grained categories can be better ascertained, even in the case of very limited training shots available.

Land Cover Mapping Results. To verify that FTM can improve models’ applicability to across-domain tasks, we perform the land-cover mapping task on two randomly selected GID images from the Set-C validation set. The two GID images are taken from different locations and seasons showing a big domain gap to the images in \mathcal{D}^S . For simplicity, we do not annotate additional training samples from the two GID images as the target domain data but directly use the Set-F training samples as target domain data since they are obtained from the same satellite. In addition, we only compare FTM to FT in this task because of the better performance of FT than FT-bn in RSSC.

To achieve pixel-level mapping, we on the one hand segment the full GID image into 224×224 patches and classify them by using the FTM (or FT) networks, on the other hand, we segment the full GID image into 100 superpixels by using SLIC [1] and align them with the 224×224 patches. Finally, we assign labels to superpixels by assembling the labels of 224×224 patches within the corresponding superpixels and labeling them by winner-take-all.

Table 4 shows the average F1 scores of the FT and FTM networks evaluated on the 224×224 patches of the two GID images. By comparison, FTM shows a clear advantage over FT, achieving higher performance in all categories. Noting that there is no meadow class because the image has no pixels belonging to it. Further, it is worth special attention that the improvement on farmland is very significant raising from 55.3% to **86.3%**. These improvements further validate

Table 4. F1 scores (%) of FT and FTM networks on land-cover mapping tasks with 3 training samples each class.

	FT3	FTM3
Farmland	55.3	86.3
Built-up	80.6	90.0
Forest	35.2	53.0
Water	84.8	90.5
Average	64.0	80.0

the wide applicability of FTM to cross-domain few-shot tasks considering that we even do not annotate training samples from the target image.

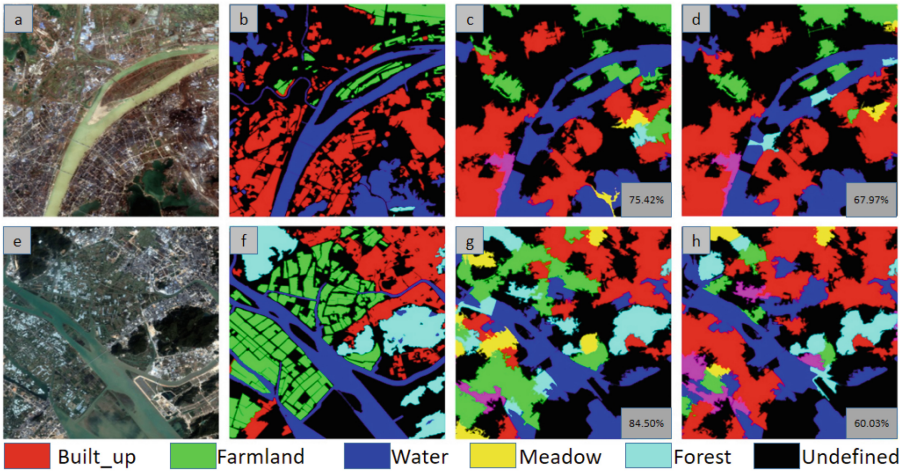


Fig. 5. The land-cover mapping results. (a, e) are RGB images from GID validation set. (b, f) are ground-truth annotations. (c, g) and (d, h) are mapping results from FTM and FT, respectively. The numbers at the bottom of (c, d, g, h) are class average F1 scores evaluated on 224×224 image patches.

We further visualize the mapping results in Fig. 5. From it we find that large variances exist between GID images. This poses great challenges to models applicability where a large number of annotated training samples are usually needed to be recollected to retrain the model. However, FTM can alleviate the annotation requirements. The third and fourth columns of Fig. 5 show prediction results. By comparison, we conclude that FTM can effectively predict the main areas in the image and keep the smoothness between neighboring superpixels. In contrast, FT fails to achieve these effects and results in fragmented superpixels. For example, large areas of farmland are mismapped into built-up by FT

while correctly mapped by FTM. This is because seasonal changes cause large differences between the source and target domains in the farmland class, thus when the labeling information of the target data is limited, it is incapable of FT to effectively represent contextual properties of this scene class. Although the visualization effects are far behind satisfaction, we, however, should note that our purpose is to validate the adaptability of FTM across domains while not the mapping accuracy, which can be achieved via much smaller image patches and more superpixels.

5 Conclusion

In this paper, we studied a feature-wise transformation module (FTM) that adapts feature distributions learned on source domain to that of target domain and verified that it has better transferability and fine-grained discriminability relative to fine-tuning methods, especially in cases of limited training shots available. Although it is simple, FTM shows great applicability to the RS field where large domain gaps exist and available training samples are extremely limited.

Problems remain. We notice that FTM still cannot well separate samples from visually similar classes, thus limiting its performance to a certain degree. This can be observed from the confusion matrices. The reason may be due in part to the affine transformation of FTM, which cannot nonlinearly scale features thus limiting its ability to explore more discriminative space. In addition, the performance of FTM still lacks robustness, although it performs better than FT and FT-bn on average. This is reflected in the land cover mapping tasks on Set-C, where the performance of FTM on different trials with different training samples varies. This phenomenon also indicates the weakness of FTM to reshape the feature space. We will explore these in our future work.

Acknowledgements. This work was supported in part by NSFGD (No.2020A1515010813), STPGZ (No.202102020673), Young Scholar Project of Pazhou Lab (No.PZL2021KF0021), and NSFC (No.61702197).

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P.V., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**, 2274–2282 (2012)
2. Alcántara, C., Kuemmerle, T., Prishchepov, A.V., Radeloff, V.C.: Mapping abandoned agriculture with multi-temporal modis satellite data. *Remote Sens. Environ.* **124**, 334–347 (2012)
3. Cao, R., Fang, L., Lu, T., He, N.: Self-attention-based deep feature fusion for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **18**, 43–47 (2021)
4. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y., Huang, J.B.: A closer look at few-shot classification. In: *ICLR* (2019)

5. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: benchmark and state of the art. *Proc. IEEE* **105**, 1865–1883 (2017)
6. Chowdhury, A., Jiang, M., Jermaine, C.: Few-shot image classification: Just use a library of pre-trained feature extractors and a simple classifier. In: *ICCV* (2021)
7. Deng, J., et al.: Imagenet: a large-scale hierarchical image database. In: *CVPR* (2009)
8. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: *ICML* (2017)
9. Fu, L., Zhang, D., Ye, Q.: Recurrent thrifty attention network for remote sensing scene recognition. *IEEE Trans. Geosci. Remote Sens.* **59**, 8257–8268 (2021)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR* (2016)
11. He, N., Fang, L., Li, S., Plaza, A.J., Plaza, J.: Remote sensing scene classification using multilayer stacked covariance pooling. *IEEE Trans. Geosci. Remote Sens.* **56**, 6899–6910 (2018)
12. Huang, X.Z., Han, X., Ma, S., Lin, T., Gong, J.: Monitoring ecosystem service change in the city of shenzhen by the use of high-resolution remotely sensed imagery and deep learning. *Land Degrad. Dev.* **30**(12), 1490–1501 (2019)
13. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *ICML* (2015)
14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *CoRR* abs/1412.6980 (2015)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *NIPS* (2012)
16. Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10649–10657 (2019)
17. Li, E., Xia, J., Du, P., Lin, C., Samat, A.: Integrating multilayer features of convolutional neural networks for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **55**, 5653–5665 (2017)
18. Li, F., Feng, R., Han, W., Wang, L.: High-resolution remote sensing image scene classification via key filter bank based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **58**, 8077–8092 (2020)
19. Li, Z., Xu, K., Xie, J., Bi, Q., Qin, K.: Deep multiple instance convolutional neural networks for learning robust scene representations. *IEEE Trans. Geosci. Remote Sens.* **58**, 3685–3702 (2020)
20. Lichtenstein, M., Sattigeri, P., Feris, R.S., Giryes, R., Karlinsky, L.: Tafssl: task-adaptive feature sub-space learning for few-shot classification. In: *ECCV* (2020)
21. Liu, Q., Hang, R., Song, H., Li, Z.: Learning multiscale deep features for high-resolution satellite image scene classification. *IEEE Trans. Geosci. Remote Sens.* **56**, 117–126 (2018)
22. Liu, Y., Zhong, Y., Qin, Q.: Scene classification based on multiscale convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **56**, 7109–7121 (2018)
23. Lu, X., Gong, T., Zheng, X.: Multisource compensation network for remote sensing cross-domain scene classification. *IEEE Trans. Geosci. Remote Sens.* **58**, 2504–2515 (2020)
24. Lu, X., Sun, H., Zheng, X.: A feature aggregation convolutional neural network for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **57**, 7894–7906 (2019)
25. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: *ICML* (2010)

26. Nogueira, K., Penatti, O.A.B., dos Santos, J.A.: Towards better exploiting convolutional neural networks for remote sensing scene classification. *ArXiv abs/1602.01517* (2017)
27. Oreshkin, B.N., Rodriguez, P., Lacoste, A.: Tadam: task dependent adaptive metric for improved few-shot learning. In: *NeurIPS* (2018)
28. Othman, E., Bazi, Y., Melgani, F., Alhichri, H.S., Alajlan, N.A., Zuair, M.A.A.: Domain adaptation network for cross-scene classification. *IEEE Trans. Geosci. Remote Sens.* **55**, 4441–4456 (2017)
29. Perez, E., Strub, F., Vries, H.D., Dumoulin, V., Courville, A.: Film: visual reasoning with a general conditioning layer. In: *AAAI* (2018)
30. Phoo, C.P., Hariharan, B.: Self-training for few-shot transfer across extreme task differences. In: *ICLR* (2021)
31. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: *ICLR* (2017)
32. Ren, M., Liao, R., Fetaya, E., Zemel, R.S.: Incremental few-shot learning with attention attractor networks. In: *NeurIPS* (2019)
33. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: *NIPS* (2017)
34. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208 (2018)
35. Szegedy, C., et al.: Going deeper with convolutions. In: *CVPR* (2015)
36. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
37. Tong, X.Y., et al.: Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sens. Environ.* **237**, 111322 (2018)
38. Tseng, H.Y., Lee, H.Y., Huang, J.B., Yang, M.H.: Cross-domain few-shot classification via learned feature-wise transformation. In: *ICLR* (2020)
39. Vinyals, O., Blundell, C., Lillicrap, T.P., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: *NeurIPS* (2016)
40. Wang, Q., Liu, S., Chanussot, J., Li, X.: Scene classification with recurrent attention of VHR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **57**, 1155–1167 (2019)
41. Wang, S., Guan, Y., Shao, L.: Multi-granularity canonical appearance pooling for remote sensing scene classification. *IEEE Trans. Image Process.* **29**, 5396–5407 (2020)
42. Wang, X., Wang, S., Ning, C., Zhou, H.: Enhanced feature pyramid network with deep semantic embedding for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **59**, 7918–7932 (2021)
43. Xia, G.S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., pei Zhang, L., Lu, X.: Aid: a benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **55**, 3965–3981 (2017)
44. Yan, L., Zhu, R., Mo, N., Liu, Y.: Cross-domain distance metric learning framework with limited target samples for scene classification of aerial images. *IEEE Trans. Geosci. Remote Sens.* **57**, 3840–3857 (2019)
45. Yuan, Y., Fang, J., Lu, X., Feng, Y.: Remote sensing image scene classification using rearranged local features. *IEEE Trans. Geosci. Remote Sens.* **57**, 1779–1792 (2019)

46. Zhu, S., Du, B., pei Zhang, L., Li, X.: Attention-based multiscale residual adaptation network for cross-scene classification. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–15 (2022)
47. Zhu, S., Luo, F., Du, B., pei Zhang, L.: Adversarial fine-grained adaptation network for cross-scene classification. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp. 2369–2372 (2021)