Domain Adaptation for Convolutional Neural Networks-Based Remote Sensing Scene Classification

Shaoyue Song, Hongkai Yu, Zhenjiang Miao, *Member, IEEE*, Qiang Zhang, Yuewei Lin, and Song Wang, *Senior Member, IEEE*

***Abstract*— Remote sensing (RS) scene classification plays an important role in the field of earth observation. With the rapid development of the RS techniques, a large number of RS scene images are available. As manually labeling large-scale RS scene images is both labor and time consuming, when a new unlabeled data set is obtained, how to use the existing labeled data sets to classify the new unlabeled images is an important research direction. Different RS scene image data sets may be taken from different type of sensors, and the images may vary from imaging modalities, spatial resolutions, and image scales, so the distribution discrepancy exists among different image data sets. As a result, simply applying convolutional neural networks (CNN) trained on source domain cannot accurately classify the images on target domain. Domain adaptation (DA) can be helpful to solve this problem. In this letter, we design a subspace alignment (SA) and CNN-based framework to solve the DA problem in RS scene image classification. A new SA layer is proposed and added into CNN models for DA, which could align the source and target domains in some feature subspace. Fine-tuning the modified CNN model with the added SA layer makes the CNN model adapt to the aligned feature subspace and helps to relieve the domain distribution discrepancy. The experiments conducted on two public data sets show that adding the SA layer into CNN improves the scene classification on the target domain.**

***Index Terms*— Convolutional neural networks (CNN), domain adaptation (DA), remote sensing (RS), scene classification, subspace alignment (SA).**

I. Introduction

As A fundamental problem in the tasks of understanding high-resolution remote sensing (RS) imagery, image scene classification plays an important role in the field of earth observation. Some works such as [1] and [2] concentrate on the pixel-level analysis of the RS scene images, whereas our task is the general scene image classification [3].

With the rapid development of the RS techniques in recent years, a large number of RS scene images have been taken, and many public RS-related data sets [3]–[6] are available. However, manually labeling a large number of RS scene image samples is both labor and time consuming. It is necessary to study how to take advantage of the existing relevant labeled data sets to classify the newly available unlabeled data sets.

To solve the problem of cross data set RS scene image classification, the discrepancy between image data sets needs to be considered. As different RS scene image data sets may be taken from different type of sensors, the images in different data sets may show different imaging modalities, resolutions, and scales. There are always distribution differences among different data sets. As a result, the domain distribution discrepancy may be large from one data set to another.

Deep neural networks have become important tools in the field of image classification recently. Some existing works [7]–[9] have demonstrated that the pretrained deep convolutional neural networks (CNN) learned from a large-scale data set such as ImageNet [10] can be transferable for image classification on other image data sets. However, directly applying the pretrained CNN on the labeled data set to classify the RS scene images on the unlabeled data set might show low accuracy due to the domain distribution discrepancy. To relieve the cross data set discrepancy, domain adaptation (DA) is considered by adapting models trained on a source domain to a target domain [11] in this letter, which is a special technique for transfer learning. The labeled image data set is considered as the source domain, and the new unlabeled image data set is considered as the target domain. Typically, DA aims to use the information from both the source and target domains to reduce the domain discrepancy [12], [13]. The DA-based RS scene classification problem is shown in Fig. 1.

In the RS scene classification, to adapt the RS image data from the source domain to the target domain, some CNN-based methods have been designed for DA recently [9], [14], [15]. Castelluccio *et al.* [14] explores various training modalities for transferring a pretrained deep CNN to the RS data sets. With the results showed on two publicly available RS data sets, the authors prove that CNN can provide an excellent classification. Wang *et al.* [9] proposes a deep CNN-based feature extraction framework and tries to form

Manuscript received November 1, 2018; revised December 29, 2018; accepted January 13, 2019. Date of publication February 25, 2019; date of current version July 18, 2019. This work was supported in part by NSFC under Grant 61672089, Grant 61273274, Grant 61572064, and Grant 61672376, in part by NSFC-U under Grant 1803264, in part by National Key Technology Research and Development Program of China under Grant 2012BAH01F03, in part by NSF under Grant 1658987, and in part by the Brookhaven National Laboratory, Laboratory Directed Research and Development under Grant 18-009. *(Corresponding authors: Shaoyue Song; Hongkai Yu; Zhenjiang Miao.)*

S. Song, Z. Miao, and Q. Zhang are with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China (e-mail: 14112060@bjtu.edu.cn; zjmiao@bjtu.edu.cn; 11112066@bjtu.edu.cn).

H. Yu is with the Department of Computer Science, The University of Texas Rio Grande Valley, Edinburg, TX 78539 USA (e-mail: hongkai.yu@utrgv.edu).

Y. Lin is with the Brookhaven National Laboratory, Upton, NY 11973 USA (e-mail: ywlin@bnl.gov).

S. Wang is with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA, and also with the School of Computer Science and Technology, Tianjin University, Tianjin 300350, China (e-mail: songwang@cec.sc.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2019.2896411

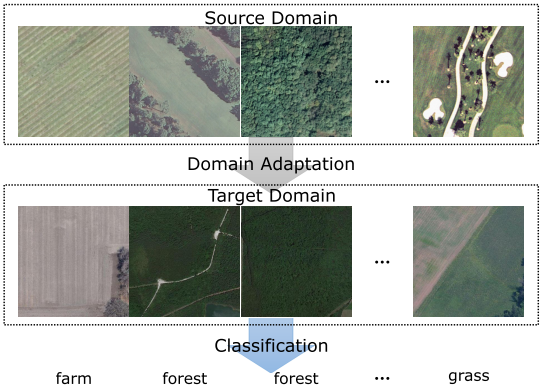

Fig. 1. RS scene classification with DA from source domain (labeled) to target domain (unlabeled). The domain distribution discrepancy is considered in DA.

a baseline for transferring pretrained deep CNN to other RS tasks. Othman *et al.* [15] design a DA network composed of pretrained CNN model and extra one hidden layer network for tackling the cross-scene classification. Different with previous methods, we design and add a new layer in CNN for DA in this letter. Specifically, a new subspace alignment (SA) [12] layer is proposed and added into CNN to relieve the domain distribution discrepancy. The new added SA layer can be used for DA to improve the classification. As a representative algorithm for DA, SA learns transformation matrices to align the features between source and target domains on some subspace. Previous SA methods, like [12], are based on dimension reduction and feature mapping, which is separated with CNN models. In this letter, we propose a new and effective way to embed the SA into CNN for DA.

In our framework, we first use a pretrained CNN on source domain to extract features on source and target domains. On the basis of the extracted features, we apply SA, leading to a new SA layer. This new generated SA layer can be easily embedded into the pretrained CNN model for fine-tuning. The experimental results show that our method is able to reduce the domain distribution discrepancy so as to improve the RS scene classification.

II. Method

To alleviate the domain discrepancy between source and target domains, we incorporate SA-based DA into a CNN model for better RS scene image classification.

A. Subspace Alignment for Domain Adaptation

SA [12] is a classic and representative algorithm for DA, so we first review the work of SA for DA.

We suppose that the data in an existing data set with manually labeled ground truth are the source domain S, and the newly obtained data set without ground truth is the target domain T. Both of the source domain S and the target domain T lie in a given D-dimensional space and drawn independent identically distributed according to a fixed but unknown source distribution $\mathcal{D}_S$ and target distribution $\mathcal{D}_T$ [12]. By using principal component analysis (PCA), d eigenvectors X_S and X_T (X_S, $X_T \in \mathbb{R}^{D\times d}$) are selected as the bases of the source subspace and target subspace, respectively. The shift between the source and the target domains are then learned by aligning X_S and X_T.

In order to align X_S to X_T, a transformation matrix M is defined, and M can be obtained by minimizing the Bregman matrix divergence

$$M^* = \underset{M}{\operatorname{argmin}}(||X_S M - X_T||_F^2) \tag{1}$$

where $||\cdot||_F^2$ is the Frobenius norm. As introduced in [12], there is a simple closed-form solution of (1)

$$M^* = X_S' X_T \tag{2}$$

where $'$ means the transpose operation. We can use M^* to transform the source subspace X_S to the target subspace X_T as in the following:

$$X_a = X_S M^* = X_S X_S' X_T \tag{3}$$

where X_a is the learned matrix used to align the source subspace domain to the target subspace domain. Using the transformations, we can get aligned subspace S_A and T_A for source and target domains by projecting the source domain data to the target aligned source subspace and projecting the target domain to the target subspace

$$S_A = S X_a \tag{4}$$

$$T_A = T X_T. \tag{5}$$

B. Proposed Method

In our method, we first train a CNN model on the source domain S, then a pretrained CNN model on S can be obtained. Using this pretrained CNN model, the features of both source domain data S and target domain data T before the final fully connected (fc) layer are extracted. Let F_S and F_T denote the features of source domain S and target domain T, respectively, whose feature dimensions are D. Then, we select d-dimension eigenvectors as bases of the source and target domain subspace for PCA-based dimension reduction as described in Section II-A. Based on (4) and (5), we can compute X_a and X_T to align the subspace of source domain S and target domain T. Using the transformation X_a and X_T, we define a new SA layer L_{SA}. Because X_a and X_T are just for matrix's dot product, each of them can be easily represented as the parameters of one fc layer in CNN. The new added SA layer L_{SA} is actually a kind of fc layer for SA. The algorithm flow for computing the SA layer L_{SA} is shown in Algorithm 1.

The whole pipeline of the proposed method is shown in Fig. 2. First, we train a CNN model on the source domain S and then apply the pretrained CNN model to extract features F_S and F_T from the source domain (S) and target domain (T). Then, Algorithm 1 is used to align the subspace and construct the new SA layer L_{SA}. Finally, we add L_{SA} before the final fc layer of the pretrained CNN for DA.

For continuous training on source domain S, usually called fine-tuning, we use X_a to construct L_{SA}. For testing on the target domain T, we use X_T to construct L_{SA}. In this way,

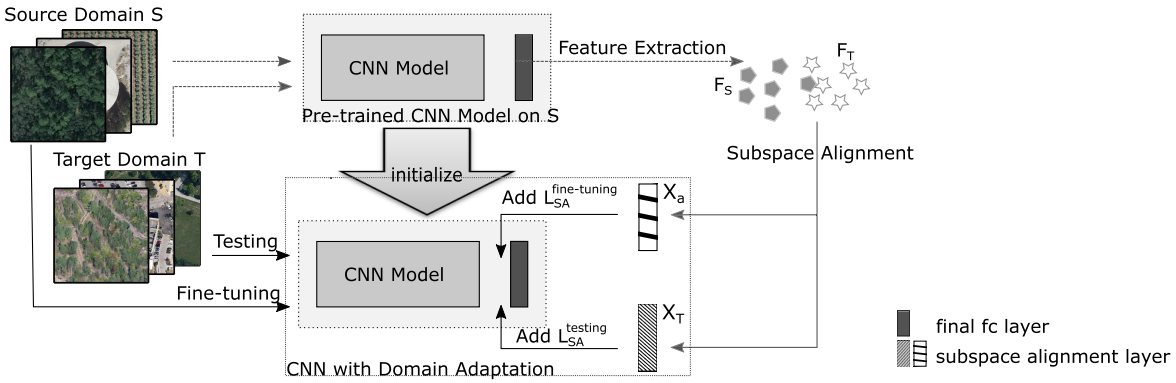

Fig. 2. Pipeline of the proposed method. First, a pretrained CNN on source domain S is applied to extract features of source domain (S) and target domain (T). Then, features F_S and F_T are used for SA, resulting in new SA layers $L_{\text{SA}}^{\text{fine-tuning}}$ and $L_{\text{SA}}^{\text{testing}}$. Finally, we add L_{SA} into the pretrained CNN for DA. Note that the parameters for L_{SA} are fixed in the CNN fine-tuning and testing after SA.

Algorithm 1 Computing the SA Layer

Input: source domain feature F_S, target domain feature F_T, subspace dimension d.

Output: SA layer L_{SA}.

1: $X_S \leftarrow PCA(F_S)$
2: $X_T \leftarrow PCA(F_T)$
3: $X_a \leftarrow X_S X_S' X_T$
4: $L_{\text{SA}}^{\text{fine-tuning}} \leftarrow X_a$
5: $L_{\text{SA}}^{\text{testing}} \leftarrow X_T$

the subspace of F_S and F_T is aligned to reduce the domain distribution discrepancy. Using the CNN structure AlexNet [16] as an example, let us explain the implementation details. The dimension of extracted features before the final fc layer (FC8) is $D = 4096$. Suppose we use PCA's $d = 1024$ dimension for the subspace, X_a and X_T will be both 4096×1024 matrices. For fine-tuning the CNN on S, X_a is used to define a new fc layer, i.e., $L_{\text{SA}}^{\text{fine-tuning}}$, and inserted into AlexNet before the final fc layer. For testing the CNN on T, X_T is used to define a new fc layer, i.e., $L_{\text{SA}}^{\text{testing}}$, and replace the $L_{\text{SA}}^{\text{fine-tuning}}$ layer in the already fine-tuned AlexNet model. It is worth mentioning that the parameters for L_{SA} are fixed in the CNN fine-tuning and testing after SA. To be consistent in dimension after adding L_{SA}, the final fc layer in AlexNet is modified to reduce the dimension from $d = 1024$ to the number of classes for classification. The proposed method is very flexible that L_{SA} can be easily inserted before the final fc layer of different CNN structures, such as AlexNet [16], VGGNet [17], and ResNet [18].

III. Experiments

A. Data Sets

1) RSSCN7 Data Set: The RSSCN7 [3] data set contains 2800 RS images, which consists of the following seven typical RS scene categories: grassland, farmland, industrial region, river and lake, forest, residential region, and parking lot. The images are collected from the Google Earth, the 400 images in each class are sampled on four different scales and 100 images are selected per scale. All the images in this data set has a size of 400×400 pixels. The whole image data set is divided into two equal-size subsets: one is training set (total 1400 images: 7 classes, 200 images per class) and the other is testing set (total 1400 images: 7 classes, 200 images per class).

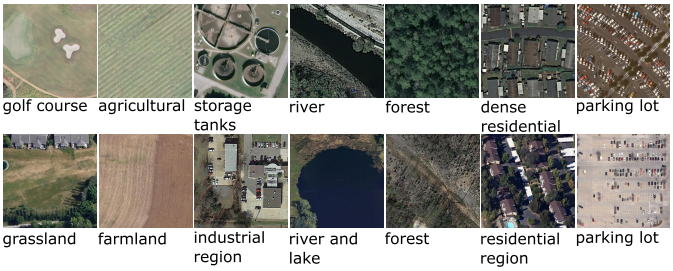

Fig. 3. Sample images from two public data sets used in our experiments (seven classes in total). The top row is from UC Merced data set [4] and the bottom row is from RSSCN7 data set [3]. Each column presents the corresponding class of these two data sets.

2) UC Merced Data Set: The UC Merced Land Use data set [4] is commonly used for RS scene image classification. It consists of 21 land-use classes and each class contains 100 images of 256×256 pixels. The 2100 images were downloaded from the United States Geological Survey National Map. To match the seven classes in RSSCN7 data set, we select the corresponding seven similar classes from UC Merced Land Use data set for experiments: golf course, agricultural, storage tanks, river, forest, dense residential, and parking lot. We denote the selected subset (total 700 images: 7 classes, 100 images per class) of the UC Merced Land Use data set as UC Merced data set in this letter.

Some sample images of corresponding classes from the RSSCN7 and UC Merced data sets are shown in Fig. 3.

B. Baseline CNN Models

For the baseline CNN models, we use three kinds of popular CNN structures, i.e., AlexNet [16], VGGNet [17], and ResNet [18] in our experiments of RS scene image classification. AlexNet consists of five convolution layers and three fc layers. VGGNet shows that increasing depth of the network architecture with very small filters can improve the accuracy in the large-scale image recognition. The Vgg16 model is

used for our experiment. ResNet introduces a residual network to ease the training of deeper networks. The ResNet50 and ResNet152 models are used for our experiment.

C. Experiment Settings

In experiments, we consider two scenarios separately.

1) Scenario I: In this scenario, we conduct the DA from the source domain of UC Merced data set to the target domain of RSSCN7 testing data set. The source domain and target domain are from two different data sets, so the differences of image types, scales, and contents are significant as shown in Fig. 3. Thus, the domain distribution discrepancy in Scenario I is large.

2) Scenario II: In this scenario, we conduct the DA from the source domain of RSSCN7 training data set to the target domain of RSSCN7 testing data set. The source domain and target domain are from the same data set, so the image difference is very small. Thus, the domain distribution discrepancy in Scenario II is very small.

In experiments, we compare performances of three settings: "Original," "Proposed," and "Proposed+." "Original" is a pretrained CNN model on the source domain S. In experiments, we borrow the pretrained CNN model on ImageNet as the initialization to fine-tune on the source domain S so as to obtain the "Original" model. "Proposed" denotes the proposed method: after adding L_{SA} into the "Original" model, we fix the parameters of all the network layers except the final fc layer and then fine-tune the CNN model with S and test it on T. "Proposed+" denotes the proposed method: after adding L_{SA} into the "Original" model, we only fix the parameters of L_{SA} and then fine-tune the CNN model with S and test it on T. We implement the L_{SA} and conduct experiments using PyTorch [19]. All of the images are normalized into the size of 224×224. During training, we set the initial learning rate at 0.001 and decayed with a factor of 0.9 of every seven epochs. The momentum is 0.9 in our experiments and the maximum iteration number is 100. We set the batch size of four in all the experiments. To calculate the SA layer, we use the code provided in [12] directly. We set the PCA subspace dimension $d = 1024$ in Algorithm 1 uniformly for AlexNet, Vgg16, ResNet50, and ResNet152. The feature dimensions extracted from the final fc layer in AlexNet and Vgg16 before PCA are $D = 4096$ while they are $D = 2048$ for ResNet50 and ResNet152, which are determined by their default network structures.

D. Experimental Results

The experimental results for Scenario I and Scenario II are given in Tables I and II, respectively. We can see that the proposed method with L_{SA} added can help to reduce the domain difference and improve the classification accuracy. Because the Scenario I's domain difference is large, so the performance improvement is significant in this case. While the Scenario II's domain difference is very small, the performance improvement is very little. This result is consistent with the definition of the added L_{SA}. L_{SA} is designed to reduce the domain difference between source and target domains in CNN.

TABLE I
AVERAGE CLASSIFICATION ACCURACY OF SCENARIO I WHERE DOMAIN DIFFERENCE IS LARGE: DA FROM UC MECERD DATA SET TO RSSCN7 TESTING DATA SET (UNDERLINED ITALIC: IMPROVED, BOLD: BEST)

Model	Original(%)	Proposed(%)	Proposed+(%)
AlexNet	44.00	*45.64*	**49.53**
Vgg16	43.79	*45.36*	**48.93**
ResNet50	46.71	*46.86*	**49.79**
ResNet152	47.43	46.71	**49.36**
Average	45.48	*46.14*	**49.40**

TABLE II
AVERAGE CLASSIFICATION ACCURACY OF SCENARIO II WHERE DOMAIN DIFFERENCE IS VERY SMALL: DA FROM RSSCN7 TRAINING DATA SET TO RSSCN7 TESTING DATA SET. AS A COMPARISON, THE ACCURACY OF ZOU *et al.* [3] IS 77.0% BY A DBN-BASED FEATURE SELECTION METHOD

Model	Original(%)	Proposed(%)	Proposed+(%)
AlexNet	88.86	88.50	88.36
Vgg16	93.00	**93.43**	**93.43**
ResNet50	92.29	**93.14**	*92.36*
ResNet152	91.36	*92.14*	**92.50**
Average	91.38	**91.80**	*91.66*

If the domain difference is large, L_{SA} will help to improve the performance significantly. However, if the domain difference is very small, L_{SA} can only help a little. In Scenario I, we see that "Proposed+" obtains obviously improved accuracy versus "Original" and "Proposed." It demonstrates that fine-tuning the whole CNN including the convolution layers is better than only fine-tuning the final fc layer for the task of RS scene classification. We think the reason is that "Proposed" only performs SA while "Proposed+" applies SA and feature selection. Specifically, fine-tuning the whole CNN with fixed L_{SA} ("Proposed+") affects the feature selection in the previous convolution layers for a better classification.

Fig. 4 shows the classification accuracy of "Proposed+" for each image class in Scenario I, Scenario II by different CNN models and the classification accuracy in [3] is also displayed for comparison. The method in [3] is a supervised method, which uses RSSCN7 training data set for deep belief network (DBN) training. We can see that the ranking of classification accuracy is: our method in Scenario II > the method in [3] > our method in Scenario I. Without using any data in RSSCN7 data set for CNN training, our method in Scenario I still achieves good and acceptable classification on RSSCN7 testing data set. It is amazing to see that in some classes, such as grass and river, our method in Scenario I obtains comparable performance with the supervised method in [3]. Our method in Scenario II using RSSCN7 data set for CNN training obtains much better accuracy than that in [3].

In order to choose the dimension d after the PCA dimension reduction, experiments using the "Proposed+" method in "Scenario I" are considered by changing d from 4 to 2048. The detailed classification results are given in Table III. We follow

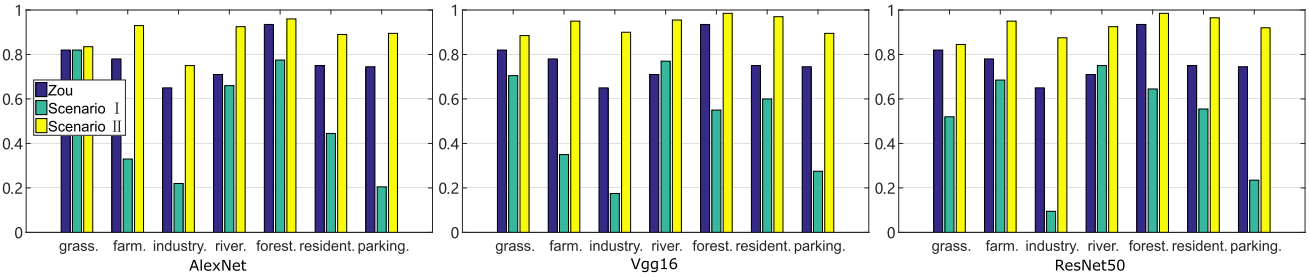

Fig. 4. Classification accuracy on RSSCN7 testing data set of "Proposed+" for each class in Scenario I and Scenario II by different CNNs. In some classes (like grass and river), our method in Scenario I not using RSSCN7 data set for CNN training can obtain comparable accuracy with the supervised method in [3] using RSSCN7 data set for training. Our method in Scenario II using RSSCN7 data set for CNN training obtains much better accuracy than that in [3].

TABLE III
AVERAGE CLASSIFICATION ACCURACY (%) OF THE "PROPOSED+" METHOD IN SCENARIO I. d IS THE PCA SUBSPACE DIMENSION

d	4	8	16	32	64	128	256	512	1024	2048
AlexNet	36.79	50.71	40.36	45.86	46.29	48.29	43.50	47.71	49.53	42.79
Vgg16	39.07	47.71	49.21	47.64	49.21	47.86	47.43	46.64	48.93	45.79
ResNet50	27.21	50.43	49.00	51.79	45.07	48.50	48.29	49.07	49.79	46.29
ResNet152	45.29	45.43	49.00	50.79	48.57	48.86	49.57	47.00	49.36	49.43
Average	37.09	48.57	46.89	49.02	47.29	48.38	47.20	47.61	**49.40**	46.08

two rules to select d: 1) If d is too large, say > 2048, it will generate a large number of parameters in L_{SA} leading to training difficulties and 2) If d is too small, say 4, 8, or 16, some valuable feature information might be lost leading to low accuracy. The experimental results given in Table III also verify the two rules. The best average performance is achieved when $d = 1024$, so we suggest $d = 1024$ for this task.

IV. CONCLUSION

In this letter, we propose a DA method for RS scene image classification. With the proposed strategy embedding a SA layer into CNN models, we transfer the information from the source domain to target domain. The proposed method is very flexible that the SA layer can be easily incorporated into most CNN models. The experimental results on two public RS data sets show that the proposed method can help to improve the scene classification when there is a significant domain difference between source and target domains.

REFERENCES

[1] Q. Wang, Z. Yuan, Q. Du, and X. Li, "GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 3–13, Jan. 2018.

[2] Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 911–923, Feb. 2018.

[3] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.

[4] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.

[5] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Maître, "Structural high-resolution satellite image indexing," in *Proc. ISPRS TC 7th Symp.-100 Years (ISPRS)*, vol. 38, 2010, pp. 298–303.

[6] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019.

[7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[8] P. Peng *et al.*, "Unsupervised cross-dataset transfer learning for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1306–1315.

[9] J. Wang, C. Luo, H. Huang, H. Zhao, and S. Wang, "Transferring pre-trained deep CNNs for remote scene classification with general features learned from linear PCA network," *Remote Sens.*, vol. 9, no. 3, p. 225, 2017.

[10] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[11] Y. Qin, L. Bruzzone, and B. Li. (2018). "Tensor alignment based domain adaptation for hyperspectral image classification." [Online]. Available: https://arxiv.org/abs/1808.09769

[12] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.

[13] Y. Lin *et al.*, "Cross-domain recognition by identifying joint subspaces of source domain and target domain," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1090–1101, Apr. 2017.

[14] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva. (2015). "Land use classification in remote sensing images by convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1508.00092

[15] E. Othman, Y. Bazi, F. Melgani, H. Alhichri, N. Alajlan, and M. Zuair, "Domain adaptation network for cross-scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4441–4456, Aug. 2017.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[17] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[19] *Pytorch*. Accessed: 2018. [Online]. Available: https://pytorch.org/