2020 13th International Conference on Intelligent Computation Technology and Automation (ICICTA) | 978-1-7281-8666-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICICTA51737.2020.00065

3D Convolutional Neural Network for Hyperspectral Image Classification Using Generative Adversarial Network

QiRui Yang, Yu Liu, Tong Zhou, YuanXi Peng, YuHua Tang*

National University of Defense Technology
E-mail: yangqirui95@163.com

***Abstract*—Recently, hyperspectral image (HSI) classification has become a hot spot in the field of machine learning. Unlike traditional black and white dual-channel images or R, G, B three-channel images, HSI has multiple channels in the spectral dimension, one channel captures light of a specified wavelength. Hyperspectral data sets are usually three-dimensional. In order to extract the spectral spatial features in such images and differentiate images from unknown classes, We have introduced domain adaptation technology, which is a well established technique for appling an algorithm trained in one or more "source domains" to a different (but related) "target domain", and proposed a framework based on Three-dimensional Convolutional Neural Network (3D-CNN), combined with Generative Adversarial Networks (GAN) to deal with unknown classes hyperspectral image classification. Our framework can achieve high-precision classification on hyperspectral images. Compared with other deep learning based methods, Our model has advantages in many aspects, such as universality for different hyperspectral data sets, and requiring fewer parameters and epochs during training. We use datasets of Pavia University scene to evaluate the adaptability of our method on the open set domain. The classification accuracy of the proposed method for unknown categories is 87.16% on PaviaU and 99.43% on Salinas.**

Keywords- Hyperspectral image classification; Machine learning; Three-dimensional Convolutional Neural Network; Generative Adversarial Networks

I. INTRODUCTION

In recent years, hyperspectral imaging technology has developed rapidly, and this technology has been widely used in a series of scenarios such as agriculture [1], Environmental sciences [2] and wild-land fire tracking. The advantages of hyperspectral imaging technology are as follows:

1. High spectral resolution with adequate spectral wave bands not only can obtain almost continuous spectral characteristic curves of ground features, but also select or extract specific wave bands to highlight target features according to needs;

2. The quantitative continuous spectral curve data provides convenience for the image classification of the ground object.

However, every coin has two sides, There are still many difficulties in the classification of hyperspectral images. The image contains dozens to hundreds of bands, which makes the amount of data very large. If not handled properly, the redundancy of data will become noise, affecting the final classification accuracy. At the same time, The correlation between the bands is high, so the number of training samples required for classification is gradually increasing. If the training samples are insufficient, the results of the model will be unreliable (Curse of Dimensionality). Therefore, the classification models and methods designed for conventional images cannot meet the needs of hyperspectral image classification.

For the past few years, machine learning depicts a remarkable performance in many fields such like Natural Language Processing (NLP) [3], computer vision [4] and machine translation [5]. Hyperspectral images are very complex, and each pixel in the image plays a crucial role in the final classification. In order to solve the problem of hyperspectral image classification, researchers have proposed a large number of methods using machine learning and deep learning technologies [6] [7] [8] [9].

Traditional hyperspectral image classification methods are usually based only on spectral information and do not include adjacent spatial information. In the article written by S. Nirmal et al. [10], they presented a novell approach for dealing with HSI classification using Open Set domain adaptation and GAN. Domain adaptation is a part of Transfer Learning (TL) [11], where a features of a target dataset are a subset of features of source dataset [12]. GAN [13][14] is a classification technique, including generator and classifier.

The generators in the network take random data as input and constantly try to imitate the original data. The classifier in the network is trained to distinguish the original data and the data generated by the generator from the input data. Due to this continuous training and confrontation between the generator and the classifier in the GAN network, fewer data samples can be used to obtain higher accuracy. They make use of OS domain adaptation by backpropogation, for training unknown class samples, as well as increasing the individual class accuracy.

However, this approach used Principal Components Analysis (PCA) [15] to one-dimensionalize the hyperspectral image data. A clear disadvantage of this approache is that it do not preserve the spatial information and do not use it well, using only spectral information. There are two ways to incorporate spatial information for HSI classification: spatialized input and postprocessing [16]. The spatialized input methods impose feature engineering step on 3-D cubes obtained from HIS [17]. The postprocessing approaches have taken the prior knowledge of smoothness into consideration that neighboring pixels with similar spectral information are likely to belong to the same land-cover categories.

To take the advantage of both spectral and spatial information features in HIS [18], we herein propose a pixel-level classification method that uses 3D-CNN and GAN to process hyperspectral image classification. In 3D

978-1-7281-8666-5/20/$31.00 ©2020 IEEE
DOI 10.1109/ICICTA51737.2020.00065

convolution operation, 3D kernels are used to extract spatial features and spectral features from HSI simultaneously. Unlike traditional CNNs, the 3D-CNN used for hyperspectral image classification here consists only of convolutional layers and fully connected layers. The reason why the pooling operation is not applied is that the pooling operation will reduce the spatial resolution, affecting the final classification accuracy.

The remainder of this paper is organized as follows. Our classification method will be described in detail in Section 2. In Section 3, we conducted experiments on hyperspectral data set of Pavia University. Sections 4 discuss the experimental. Finally, we conclude this paper in Section 5.

II. METHODOLOGY

In this paper, we design a model similar to the generating adversarial network for hyperspectral image classification [19]. First, we divided the original three-dimensional hyperspectral data into source set and target set. The data x_s and label y_s from the source (X_s; Y_s) and the data x_t from the target X_t are considered. The general architecture diagram is shown in the Figure 1 and 2.

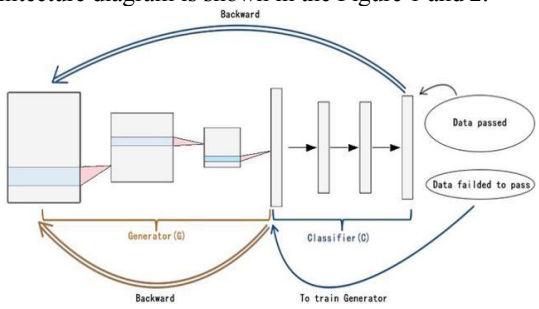

Figure 1. The architecture we proposed is mainly composed of Generators(G) and Classifier(C). First, the input data(x) is passed into the network, the generator will try to increase the error rate by generating images. This generated images G(x) is then passed as input to the classifier. The Classifier will try to get a boundary between known and unknown target class.

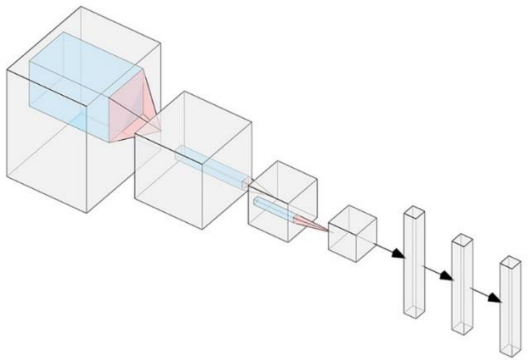

Figure 2. The features of the hyperspectral image are extracted through the three-dimensional convolutional neural network and mapped to the feature domain.

At the same time, in order to carry out pixel-level classification for hyperspectral images, we first need to create a model to extract relevant information of pixels in hyperspectral images. For illustrative purposes, we have refined the proposed method into three steps:

Step 1: Extracting training samples (3D cubes). When hyperspectral image data is placed into the model, the filtering window slides over the small spatial neighborhood (not the entire image) along the entire spectral band, extracting pixels to form an image cube as initial data, we use the class label of the center pixel as the label of the cube. The processing of feature cube extraction is shown in the Figure 3.

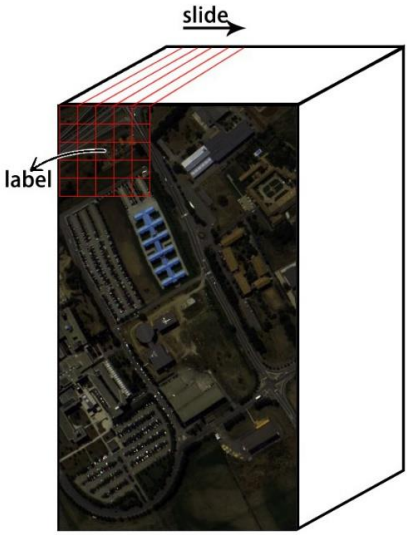

Figure 3. Extracting feature cube from HSI.

The size of initial data is w×h×c, w×h is the spatial size (also the filtering window size), and c represents the number of spectral bands. In the 3D convolution operation, input data is convolved with 3D kernels to extract spectral spatial features (see Figure 4) [20].

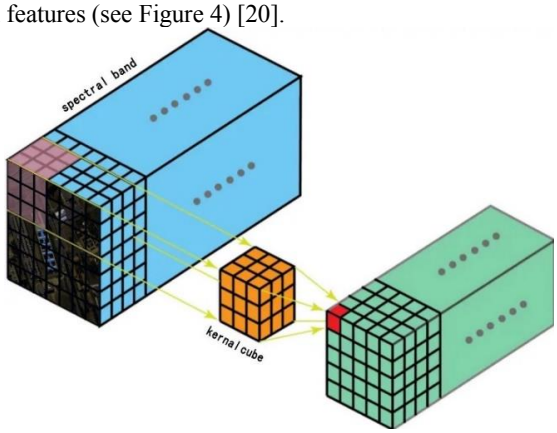

Figure 4. The 3D convolution operation.

before going through the activation function to form the output data (i.e, feature cubes). This operation can be formulated as

$$v_{lij}^{xyz} = f\left(\sum_{h=0}^{H_l-1}\sum_{w=0}^{W_l-1}\sum_{c=0}^{C_l-1} k_{lj}^{hwc} v_{(l-1)i}^{(x+h)(y+w)(z+c)} + b_{lj}\right)$$

where l is the layer that is considered, j indicates the number of feature cubes in this layer (filtering window s lides j time over the entire image). (x, y, z) is the coordi nates of the center point in jth feature cube. v_{lij}^{xyz} is the output at the position (x, y, z) that is calculated by convolving the ith feature cube of the preceding layer with the jth kernal of the l th layer, b is the bias, $f()$ is the activation function , H_l and W_l and C_l being the height, width, spectral dep th of the 3D kernal respectively. And the k_{lj}^{hwc} is the (h, w, c)th value of the kernel connected to the ith featur e cube in the preceding layer. As such, the output data of the lth convolution lay er contains $i \times j$ 3D feature cubes.

Step 2: Training generator. The class dimension of source data is K. We first connect the generator and the classifier in series, send the training samples x_s (extracte dfrom the source data X_s) into the model for training, an d use CrossEntropy as the loss function. After that, the tr aining samples x_t (extracted from the target data X_t) with K+1 classes (K known classes and unknown classes) are also sent into the model for training. Typically, the number of input and output classes in a model must be the same. Unlike other open set domain adaptation models, we do not provide any data to train unknown classes separately. To compensate for the missing class data, the data gener ated by the generator that fails to pass the classifier will be used to train unknown classes in the source domain. Therefor e, there is no need to provide data for unknown classes separately during training.

Step 3: Training Classifier. When the input data x is transferred to the network, the generator will generate an image G(x) based on x. The generated image G(x) is then passed as input to the classifier. The data generated by the generator that fails to pass through the classifier will be used to train the unknown class in the source. At the same time, the classifier will try to draw a boundary between the known classes and the unknown classes. The loss function can be formulated as

$$l(x, y) = L = \{l_1, \dots, l_N\}^T$$
$$l_n = -\frac{1}{2}[t \cdot \log x_n + (1-t) \cdot \log(1-x_n)]$$

where N is the batch size, l is the loss, t will be expla ined below. The softmax function is used in the model t o convert the logit of unknown class into probability P_{k+1} [21]. In GAN, this limit is generally 0.5, so in this experiment w e set $t = 0.5$ as the boundary between the known classe s and the unknown classes.

III. Dataset and Model Description

In this section, we introduced the HSI dataset of Pavia University and Salinas, explained the structure of this hyperspectral dataset, and evaluated the proposed methods using classification metrics, such as overall accuracy (OA) and average accuracy (AA). Then, we explained the structure of the convolutional neural network model.

Pavia University dataset was acquired by the ROSIS sensor during a flight campaign over Pavia, Nothern Italy in 2001. The number of spectral bands is 103. And the picture size of Pavia University is 610×610 pixels, but some of the samples contain no information and have to be discarded before the analysis. After discard, the picture size of Pavia University is 610×340 pixels. The geometric resolution is 1.3 meters. The image groundtruths of Pavia University have 9 classes, which are listed in Table 1.

Table 1. Pavia University dataset.

#	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947

With Pavia University dataset, we designed a scenario that the data set contains unknown classes. We selected 5 classes (Asphalt, Bare Soil, Gravel, Trees and Painted metal sheets, The remaining classes are put into unknown classes) from the Pavia University dataset, and selected 1000 samples from each class for source domain training. Then we used the complete Pavia University dataset (about 42,775 samples with 9 classes) as the target domain, and also selected 1000 samples each classes from target domain for training, The rest of data as the test set. The details of the data set are mentioned in Table 2.

Table 2. Pavia University Source dataset and target dataset.

Label	Features	PaviaU (Source)	PaviaU (Target)
0	Asphalt	1000	6631
1	Bare Soil	1000	5029
2	Gravel	1000	2099
3	Trees	1000	3064

4	Painted metal sheets	1000	1345
5	Unknown(Meadows, Self-Blocking Bricks, and Shadows)	0	24607
Total samples	#	5000	42775

Salinas dataset was collected by the 224-band AVIRIS sensor over Salinas Valley, California, and is characterized by high spatial resolution (3.7-meter pixels). The area covered comprises 512 lines by 217 samples. The dimension of the data is $512 \times 217 \times 224$ with total of 16 classes, which are listed in Table 3.

Table 3. Salinas dataset.

#	Class	Samples
1	Brocoli_green_weeds_	12009
2	Brocoli_green_weeds_	23726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807

An small subscene of Salinas image, denoted Salinas-A, is usually used too. It comprises $86 \times 83 \times 224$ pixels located within the same scene and includes 6 classes, which are listed in Table 4.

Table 4. Salinas-A dataset.

#	Class	Samples
1	Brocoli_green_weeds_	1391
2	Corn_senesced_green_weeds	1343
3	Lettuce_romaine_4wk	616
4	Lettuce_romaine_5wk	1525
5	Lettuce_romaine_6wk	674
6	Lettuce_romaine_7wk	799

For one of our experiment, we consider, Salinas-A dataset with 6 classes (Source) and Salinas dataset with 8 classes (Target) is used. The class and sample for each class is mentioned in Table 5.

Table 5. Dataset Description for SalinasA(source) and Salinas(target).

Label	Features	Source(SalinasA)	Target(Salinas)
0	Brocoli_green_weeds_1	391	2008
1	Corn_senesced_green_weeds	1343	3278
2	Lettuce_romaine_4wk	616	1068
3	Lettuce_romaine_5wk	1524	1927
4	Lettuce_romaine_6wk	675	916
5	Lettuce_romaine_7wk	799	1070
6	Unknown(Brocoli_green_weeds_2 and Fallows)	0	5702
Total	#	5348	15969

In the model, we used 3D-CNN to build a generator including three layers. Each layer uses the ReLu function as an activation function to train three-dimensional hyperspectral image data. After the convolution and fully connected layers, we used 3D batch standardization.

IV. EXPERIMENTAL RESULTS

The parameters which gives the best results for our model during the experiment are mentioned below.

In the experiment, the network is trained for 150 epochs and the batch size is 32, patch size (size of the spatial neighbourhood) is 27. SGD optimization method with learning rate of (0.003) is used for training the network. And we use Cross-Entropy as the loss function.

The main purpose of the experiment is to identify the model effectiveness in classifying the unknown class. So precision, recall (classwise accuracy) and f1-score are calculated with contingency values obtained using CM and are used to evaluate the model. The receiver operating characteristic curve (ROC curve) is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings, in which the area under the curve (AUC) represents the accuracy of various classifier (see Figure 5).

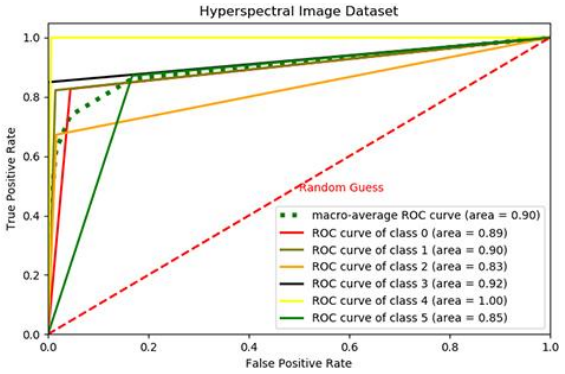

Figure 5. The receiver operating characteristic curve of Pavia University dataset.

Table shows the results for 1st experimental setup using selected PaviaU (Source, 5 class) and complete PaviaU (Target, 9 class). The average accuracy rate is 84.04\%. To get this average accuracy only only 150 epochs were conducted,
compared with 2000 epochs using 1D-CNN, efficiency has been greatly improved.

From the confusion matrix of Pavia University in Figure 6, we can know that about 2784 samples out of 20341 unknown samples are misclassified into other classes and the classification accuracy rate of 87.96% is obtained over unknown class. We can also learn from the confusion matrix of Salinas(Figure 7) that only 10 samples out of 5702 unknown samples are misclassified into other classes, which reaches classification accuracy of 99.82% and AUC of 0.99%. When the proportion between known and unknown class sample is taken into consideration, this further proves that the model works well in classifying unknown class samples. On Pavia universality dataset, We got an average rate of over 86% in precision and above 87% in recall, the f1-score is over 85% (see Table 6). On Salinas dataset, We got an average rate of over 99% in precision and above 98% in recall, the f1-score is over 98% (see Table 7).These predicted result further proves that our method can be implemented will in HSI classification.

According to the experimental results in Table6 and Table 7 (the performance results of 1D-CNN is get from paper [10]) presented above, we can draw the following conclusions: Compared to 1D-CNN, 3D-CNN works even better for spectral–spatial feature extracting. On Pavia University dataset, The Unknown Class Accuracy of 3D-CNN is 87.96%, while the Unknown Class Accuracy of 1D-CNN is only 81.65%(see Figure 8). On Salinas dataset, The Overall Accuracy of 3D-CNN is 99.39%, while the Overall Class Accuracy of 1D-CNN is only 96.38%(see Figure 9). It is observed that the 3D-CNN model could distinguish the unknown class from known classes in Hyperspect dataset better than 1D-CNN, because 3D-CNN can extrat the spatial information well and model finer spectral information attributed to the 3D convolution operations.

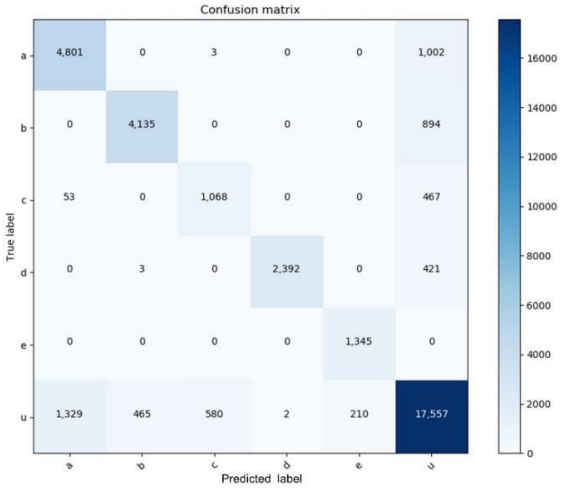

Figure 6. Confusion matrix of Pavia University dataset.

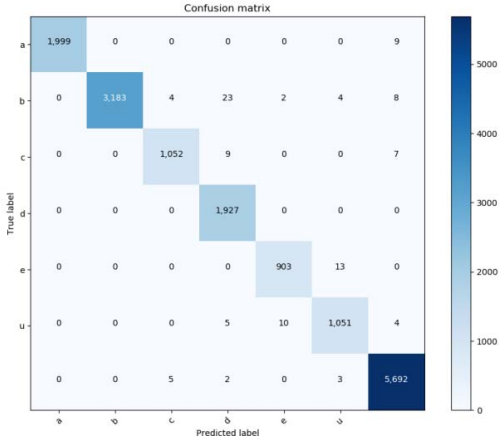

Figure 7. Confusion matrix of Salinas dataset.

Table 6. Classwise Accuracy for PaviaU (Source) and PaviaU (Target) obtained using the 1D-CNN and the 3D-CNN method.

1D-CNN with 2000 epochs (cited from [10])			
Label	precision	recall	f1-score
0	0.78	0.98	0.93
1	0.91	0.59	0.72
2	0.61	0.87	0.72
3	0.69	0.85	0.71
4	1.00	0.99	0.99
5	0.88	0.83	0.85
average	0.81	0.86	0.83
3D-CNN with 150 epochs			
Label	precision	recall	f1-score
0	0.78	0.83	0.80
1	0.90	0.82	0.86
2	0.65	0.69	0.66
3	1.00	0.85	0.92
4	0.86	1.00	0.93
5	0.86	0.87	0.87
average	0.86	0.87	0.85

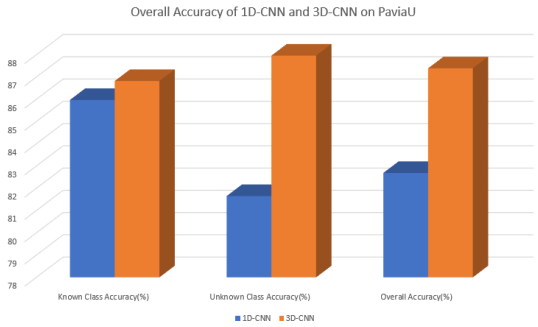

Figure 8. Accuracy rate of 1D-CNN and 3D-CNN on PaviaU.

Table 7. Classwise Accuracy for SalinasA (Source) and Salinas (Target) obtained using the 1D-CNN and the 3D-CNN method.

1D-CNN with 2000 epochs (cited from [10])			
Label	precision	recall	f1-score
0	1.00	0.93	0.96
1	1.00	0.93	0.96
2	0.96	0.93	0.95
3	0.94	0.85	0.97
4	0.97	0.97	0.97
5	0.96	0.94	0.95
6	0.95	0.99	0.97
average	0.97	0.96	0.96
3D-CNN with 150 epochs			
Label	precision	recall	f1-score
0	1.00	0.97	0.99
1	1.00	0.98	0.98
2	0.99	0.96	0.98
3	0.97	0.99	0.99
4	0.98	1.00	0.93
5	0.99	0.97	0.97
6	1.00	0.99	0.98
average	0.99	0.98	0.97

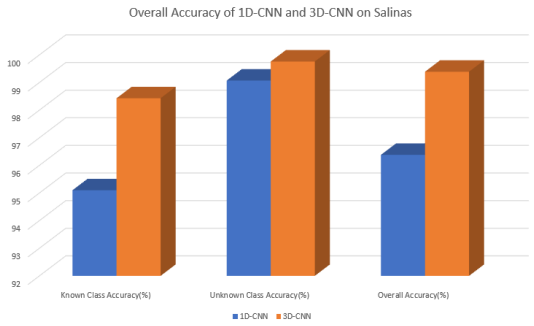

Figure 9. Accuracy rate of 1D-CNN and 3D-CNN on Salinas.

V. CONCLUSION

In this work, for the purpose of improving HSI classification, we applied GAN to achieve open set domain adaptation in hyperspectral dataset. Further more, we introduced a novel 3D-CNN classification framework that takes full advantage of both spectral and spatial information contained within HSI data. The obtained results shows that, open set domain adaptaion and 3D-CNN classification framework works well with HSI data and it will increase the efficiency and the accuracy of HSI classification models obviously. In particular, we have compared our 3D-CNN approach against the 1D-CNN HSI classification methods on PaviaU dataset, the experimental results demonstrated that the proposed method of 3D-CNN-based HSI classification achieved the best Unknown Class Accuracy and Overall Accuracy on PaviaU dataset.

As the future scope of the present work, we try to use the hyperparamter tuning to increase the efficiency of the proposed model. In the meantime, we plan to performe investigation on more effective 3D-CNN-based HSI classification techniques. In HSI, the supervised classification methods need lots of labeled samples, but obtaining labeled data is very laborious. So we committed to finding an effective use of unlabeled samples.

REFERENCES

[1] Lacar, F. M., M. M. Lewis, and I. T. Grierson. "Use of hyperspectral imagery for mapping grape varieties in the Barossa Valley, South Australia." *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217)*. Vol. 6. IEEE, 2001.

[2] Bioucas-Dias, José M., et al. "Hyperspectral remote sensing data analysis and future challenges." *IEEE Geoscience and remote sensing magazine* 1.2 (2013): 6-36.

[3] Young, Tom, et al. "Recent trends in deep learning based natural language processing." *ieee Computational intelligenCe magazine* 13.3 (2018): 55-75.

[4] Voulodimos, Athanasios, et al. "Deep learning for computer vision: A brief review." *Computational intelligence and neuroscience* 2018 (2018).

[5] Cho, Kyunghyun, et al. "On the properties of neural machine translation: Encoder-decoder approaches." *arXiv preprint arXiv:1409.1259* (2014).

[6] Mou, Lichao, Pedram Ghamisi, and Xiao Xiang Zhu. "Deep recurrent neural networks for hyperspectral image classification." *IEEE Transactions on Geoscience and Remote Sensing* 55.7 (2017): 3639-3655.

[7] Zhao, Wenzhi, and Shihong Du. "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach." *IEEE Transactions on Geoscience and Remote Sensing* 54.8 (2016): 4544-4554.

[8] Li, Wei, et al. "Hyperspectral image classification using deep pixel-pair features." IEEE Transactions on Geoscience and Remote Sensing 55.2 (2016): 844-853.

[9] Hu, Wei, et al. "Deep convolutional neural networks for hyperspectral image classification." *Journal of Sensors* 2015 (2015).

[10] Nirmal, S., V. Sowmya, and K. P. Soman. "Open Set Domain Adaptation for Hyperspectral Image Classification Using Generative Adversarial Network." *Inventive Communication and Computational Technologies*. Springer, Singapore, 2020. 819-827.

[11] Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2009): 1345-1359.

[12] Raina, Rajat, et al. "Self-taught learning: transfer learning from unlabeled data." *Proceedings of the 24th international conference on Machine learning*. 2007.

[13] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

[14] Bousmalis, Konstantinos, et al. "Unsupervised pixel-level domain adaptation with generative adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[15] Astel, Aleksander, et al. "Comparison of self-organizing maps classification approach with cluster and principal components analysis for large environmental data sets." *Water research* 41.19 (2007): 4566-4578.

[16] Zhong, Zilong, et al. "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework." *IEEE Transactions on Geoscience and Remote Sensing* 56.2 (2017): 847-858.

[17] Li, Ying, Haokui Zhang, and Qiang Shen. "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network." *Remote Sensing* 9.1 (2017): 67.

[18] Chen, Yi, Nasser M. Nasrabadi, and Trac D. Tran. "Hyperspectral image classification using dictionary-based sparse representation." *IEEE transactions on geoscience and remote sensing* 49.10 (2011): 3973-3985.

[19] Camps-Valls, Gustavo, and Lorenzo Bruzzone. "Kernel-based methods for hyperspectral image classification." *IEEE Transactions on Geoscience and Remote Sensing* 43.6 (2005): 1351-1362.

[20] Tran, Du, et al. "Learning spatiotemporal features with 3d convolutional networks." *Proceedings of the IEEE international conference on computer vision*. 2015.

[21] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).