Hyperspectral Open Set Classification towards Deep Networks Based on Boxplot

Yu Liu¹, Jing Hou¹, Yuanxi Peng², Yunpeng Xu² and Tian Jiang³*

¹College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410073, China;
²State Key Laboratory of High-Performance Computing, College of Computer, National University of Defense Technology, Changsha 410073, China;
³Beijing Institute for Advanced Study, National University of Defense Technology, Beijing 100010, China
*Corresponding Author Email: tjiang@nudt.edu.cn

Abstract: Recently, hyperspectral imaging (HSI) supervised classification has achieved an astonishing performance by using deep learning. However, most of them take the ideal assumption of ‘closed set’, where all testing classes have been known during training. In fact, in the real world, new classes unseen in training may appear during testing. Obviously, traditional supervised classification methods cannot operate correctly in the real world, which requires classifiers not only to classify known classes, but reject the unknown in order to avoid false positives. This challenge is called ‘open set classification’(OSC). Considering the increased applications of deep learning in the real world, rejecting unknown classes during classification is of vital importance. To tackle it, we present a simple but effective HSI OSC method toward deep networks. In this method, we tighten the decision boundaries of SoftMax function of the last layer of the deep networks by using boxplots to analysis the statistical characteristics of the probability distribution of known classes and generate a proper rejection threshold for each known class. To test the performance of the proposed HSI OSC method, experiments are conducted on three HSI datasets. The results show that the proposed method outperforms existing state-of-the-art HSI OSC methods.

1. Introduction

Hyperspectral imaging (HSI) collects digital images in hundreds of continuous narrow spectral bands, providing spatial information and fine electromagnetic spectrum from visible to infrared wavelength ranges. For HSI, each pixel can be regarded as a high-dimensional vector corresponding to the spectral reflectance. With the advantage of distinguishing spectral difference, HSI has been widely used in various fields [1, 2]. For its applications, HSI classification (assigning each pixel with a certain class label) lays a crucial foundation and has drawn broad attention [3].

So far, numerous HSI classification methods have been put forward, such as support vector machine (SVM), logistic regression, and so on. Recently, with the development of deep learning technology, various representative deep networks have been proposed for HSI classification [3-5], which can automatically learn the representative spectral and spatial features in a hierarchical manner from training sets, and yield better performance compared with traditional classifiers [6, 7]. In [8], a spectral-spatial residual network (SSRN) is proposed. By learning the spectral and spatial features in sequence through three dimensional (3D) and two dimensional (2D) residual structure, SSRN can extract discriminative spectral and spatial features and achieve satisfactory HSI classification performance. Similarly, Wenju Wang et al. [9] proposed a FDSSC deep network, which uses the
densely-connected structures to learn the features. In [10], a deep network called HybridSN is proposed, which applies the principal component analysis (PCA) to remove the spectral redundancy firstly and combines the complementary information of spatial-spectral and spectral in the form of 3D and 2D convolutions. Owing to the deep networks, HSI classification performance has been improved significantly.

However, a key assumption made by these powerful supervised classification models is that classes appearing in testing must have been seen (known) in training, which is called ‘closed set assumption’. To be honest, this assumption can be invalid in the real dynamic open world, where unseen (unknown) situations can emerge unexpectedly, which drastically weakens the robustness of these existing methods [11]. Therefore, it is necessary for HSI classifiers to have the ability of classifying the incoming HSI data to the right known classes and reject those unknown classes in order to avoid false positives. The problem is called ‘open set classification’ (OSC) [12]. To define the OSC problem, we assume a set of known classes \( \{(X_i, y_i)\}_{i=1}^{m} \), where \( X_i \) is the set of the data from the \( i \)-th known class, and \( y_i \in \{1,2,\ldots,m\} = Y \) is \( X_i \)'s label. OSC tends to build a model \( f(x) \) that can classify each input data cube \( x \) to one of the \( m \) known classes in \( Y \) or reject it as unknown class \( (m+1) \) from open space that does not belong to any of the \( m \) known classes. In other words, \( f(x) \) is a \( (m+1) \)-class classifier with the labels \( y_i \in \{1,2,\ldots,m,m+1\} = Y \), where \( m+1 \) indicates the unknown classes.

Deep network-based methods can achieve remarkable performance for HSI classification and have received wide attention from the community. But their increased applications in the real world open them to lots of unknown scenes, hence it is essential to address OSC issues towards deep networks. Although HSI OSC problem is very crucial for its applications in the real world, few related works have been published till now. Recently, a HSI OSC method towards deep networks is proposed in [13]. To reject unknown classes, the authors leverage an algorithm called (OpenMax) to calculate the probability distribution over known classes and unknown classes by using Weibull distribution to fit the logits trained via SoftMax function. However, its performance depends heavily on two hyperparameters: tailsize and \( \alpha \), which need be tuned with validation examples from unknown classes. Besides HSI, the challenges of OSC also exists in the field of pattern recognition. In [14], a OSC method called DOC is proposed to reduce the misclassification of unknown classes in text documents. This method replaces the SoftMax layer with a 1-vs-rest final layer of Sigmoid, and uses Gaussian fitting to tighten the classification boundaries of sigmoid functions. However, the output of sigmoid function for each known class does not follow the Gaussian distribution strictly, and Sigmoid functions is generally used for multi-label classification rather than multi-class classification. Therefore, more appropriate method to tighten the decision boundaries of known classes should be explored.

To solve the HSI OSC problems towards deep networks, we proposed a method Deep Open Classification with Boxplot (DOCBP). Different from the traditional deep learning-based classifiers, DOCBP has the ability of rejecting unknown classes when classifying known classes rightly. And unlike DOC using sigmoid function and depending on Gaussian distribution, DOCBP tightens the decision boundaries of SoftMax function by using boxplot, which depicts numerical data through their quartiles without any assumption of the statistical distribution. Experimental results on three publicly available HSI datasets show that DOCBP outperforms the OpenMax method and DOC method for HSI OSC problems. The main contribution of this letter is summarized as follows: 1) We develop an 00HSI OSC method towards deep learning, which can achieve the goal of classifying known classes rightly and rejecting unknown classes. Compared to the existing HSI OSC methods, our method is simple yet effective. 2) We propose a method for selecting the proper threshold to tighten the classification boundaries of known classes. Besides Gaussian fitting in DOC, we introduce the boxplot method and verify its robustness and effectiveness with different deep networks.

2. The Proposed DOCBP Structure
The flow diagram of the whole DOCBP structure is shown in Figure 1. It comprises deep networks and unknown classes detection mainly. As it shows, DOCBP can be applied to various deep networks,
such as SSRN, FDSSC and HybridSN. In this section we use SSRN as DOCBP’s base and introduce
the boxplot which is used to choose appropriate thresholds for unknown classes rejection.

![Overall structure of the DOCBP.](image)

**Figure 1.** Overall structure of the DOCBP.

### 2.1. SSRN and Feed Forward Layers of DOCBP

SSRN is a supervised spectral-spatial residual network proposed in [8]. It can be regarded as an extension of convolutional layers in convolutional neural networks, applying shortcut connections between every other convolutional layer. Its designed spectral and spatial residual blocks can extract discriminative spectral-spatial features from HSI data, and obtain good performance on different HSI datasets.

Let the HSI data be denoted by $\mathbf{I} \in \mathbb{R}^{H \times W \times B}$, where $H$ is the height, $W$ is the width, and $B$ is the number of spectral bands. In order to use the deep-learning based image classification model, HSI data should be divided into small overlapping 3D patches, the truth labels of which are decided by the labels of their central pixels. The 3D patches, $\mathbf{P} \in \mathbb{R}^{S \times S \times B}$ from $\mathbf{I}$, covering the $S \times S$ spatial window. In this way, we create the training set and testing set for SSRN-based DOCBP.

The output of the FC layer before activation function is $\mathbf{p} = \left[ p_1, \ldots, p_m \right]$ is the output of SoftMax, containing the probabilities $p_i$ for $i$th known class. To train the model, SoftMax loss function is minimized.

$$p_i = \text{soft max}(\xi_i) = \frac{\exp(\xi_i)}{\sum_{j=1}^{m} \exp(\xi_j)}$$  \hspace{1cm} (1)

For SSRN’s last fully connected (FC) layer, its number of neurons is changed to be consistent with the number of the known classes in HSI dataset, and its activation function is SoftMax function. The formula of SoftMax is show in Equation (1), where $\mathbf{g} = [\xi_1, \ldots, \xi_m]$ is the output of the FC layer before activation function. $\mathbf{p} = [p_1, \ldots, p_m]$ is the output of SoftMax, containing the probabilities $p_i$ for $i$th known class.

### 2.2. Unknown Class Rejection

During testing, we reinterpret the predication of SoftMax function to allow rejection. For the output of SoftMax, we check if its maximum probability $\text{max}(\mathbf{p})$ is larger than the threshold $t_{\text{arg max}(\mathbf{p})}$ that belongs to the Class $\text{arg max}(\mathbf{p})$. If larger for an example, it belongs to the Class $\text{arg max}(\mathbf{p})$; otherwise, it will be rejected as unknown Class $(m+1)$ from open space. Formally, the process of OSC is achieved by:

$$\hat{y} = \begin{cases} \text{arg max}(\mathbf{p}), & \text{if } \max(\mathbf{p}) > t_{\text{arg max}(\mathbf{p})} \\ (m+1) & \text{otherwise} \end{cases}$$ \hspace{1cm} (2)

Obviously, the unknown classes rejection is achieved by setting thresholds on SoftMax’s output. Some papers have stressed the approach and thought that it was not effective in their experiments [15, 16], for they set a single threshold for all known classes. In fact, different known classes have different
activation ranges; a single threshold is not enough to handle these differences. Therefore, in order to achieve a better performance for HSI OSC, it is necessary to select the proper threshold for each known class.

2.3. Tightening the Decision boundaries with Boxplot

To choose a proper threshold for each known class, DOC algorithm used the idea of outlier detection in statics: if a value/point is a certain number ($\gamma$) of standard deviations ($\sigma$) away from the mean ($\mu$), it is considered an outlier. This method is based on the assumption that the predicted probabilities calculated by sigmoid functions of all training data of each known class follow the Gaussian distribution with $\mu=1$. However, for the data from real world, the predicted probabilities of each known class do not follow one half of the Gaussian distribution strictly in fact. And thus, this assumption is weak to do unknown classes rejection. Moreover, the sigmoid function is generally used for multi-label classification rather than multi-class classification. Given these considerations, we propose to use the method of boxplot to generate the rejection threshold for each known class according to the output of SoftMax during training, and achieve the unknown rejection during testing. Compared with the weak Gaussian distribution, our method using boxplot which ignores the statistical distribution performs better.

![Boxplot Structure](image.png)

**Figure 2.** Structure of a boxplot. IQR is short for interquartile range, i.e., $IQR = Q_3 - Q_1$.

Boxplot is an outlier detection method in descriptive statistics which graphically describes groups of data through their quartiles. It is based on robust statistics which are more resistant (robust) to the presence of outliers than the classical statistics based on the normal distribution. And it is less influenced the dirty samples in known classes. [17]. As Figure 2 shows, boxplot displays data based on five indexes: the minimum, the first quartiles ($Q_1$), the median, the third quartiles ($Q_3$) and the maximum. Maximum is the upper extreme value limit $Q_3 + \eta \times IQR$. Minimum is the lower extreme value limit $Q_1 - \eta \times IQR$. Commonly, $\eta$ is set to be 1.5. Theoretically, boxplot is non-parametric, which does not make any assumption of the statistical distribution. Its most evident application is to identify samples with extreme characteristics. Thus, boxplot has the potential to find a proper threshold for each known class in HSI. Besides, boxplot is a simple method without complex hyperparameters tuning process, and does not take up too much computing space.

To find the proper threshold for each known class, we collect the predicted probabilities of those correctly classified training samples of each class during training firstly. And then with the help of boxplot, we choose the minimum probability of each known class’s boxplot as the rejection threshold to tighten the decision boundary of SoftMax function. Note that due to the application of the boxplot, different Class $y_i$ can have a different rejection threshold $t_i$.

3. Experiments and Discussion

3.1. Datasets Description and Experimental Setup

In our experiments, three HSI datasets are chosen for performance evaluation, including Salinas (SA), Pavia University (UP) and Kennedy Space Centre (KSC). SA has $512 \times 217$ pixels and 204 spectral bands after removing water absorption bands, and it contains 16 different classes. UP is $610 \times 340$ pixels in size and has 103 spectral bands after corrected. And it is differentiated into 9 classes. For
KSC, it contains 512×614 pixel, 176 spectral bands and 13 classes. All three HSI datasets are publicly available: (http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes).

For a fair comparison with OpenMax, we make the same OSC datasets as [13]. Following [13], we choose the same 9 classes as known classes and the remaining 7 classes as unknown classes in SA. In UP, the same 6 classes are chosen as known classes and the remaining 3 classes as unknown classes. In KSC, same 7 classes are chosen as known classes and the remaining 6 classes are treated as unknown classes. Figure 3 shows the ground-truth maps of the three HSI OSC datasets. For simplification and emphasis, only unknown classes of each dataset are indicated in Figure 3. Besides, the truth labels of the three HSI OSC datasets are relabeled from 1.

![Figure 3](image)

**Figure 3.** Ground-truth maps of different HSI OSC datasets. (a) SA. (b) UP. (c) KSC. The color bar below each map indicates the distributions of the unknown classes.

To quantitatively evaluate the performance of the HSI OSC methods, class accuracy, overall accuracy (OA), average accuracy (AA), and macro $F_1$ score ($F_1$) on the testing sets (over $m+1$ classes) are utilized as the performance indexes. We run the experiments for five times with randomly selected training data and calculate the mean of the indexes. All experiments are conducted on a computer with an NVIDIA GTX 1060 GPU with 16-GB RAM, and implemented with Keras using TensorFlow as the backend. The learning rates of the Adam optimizer for three datasets are 0.00003. The batch sizes are 32 and networks are trained for 100 epochs. 10% of the data are randomly divided into training sets for each class for SA and UP. For KSC, 20% are divided into training sets. To balance the computational cost and spatial information, we choose 5×5 as the spatial window size of the input data for all datasets (i.e., $S = 5$).

### 3.2. Experimental Results

To evaluate the efficiency of SoftMax and boxplot relative to the sigmoid function and Gaussian fitting, we conduct the following experiments using SSRN as deep network of OSC. We compare our SoftMax-boxplot (Soft_B) with SoftMax-Gaussian-fitting (Soft_G), Sigmoid-boxplot (Sig_B), Sigmoid-Gaussian-fitting (i.e., DOC or Sig_G) and OpenMax (OM) on the three HSI OSC datasets.
Table 1. OSC accuracy (%) on SA

| Class  | Sig_G | Sig_B | Soft_G | Soft_B | OM  |
|--------|-------|-------|--------|--------|-----|
| 1 Known | 98.81 | 93.81 | 99.15  | 89.45  | 97.57 |
| 2 Known | 99.30 | 93.07 | 99.42  | 91.78  | 92.97 |
| 3 Known | 99.34 | 92.77 | 99.31  | 90.79  | 95.09 |
| 4 Known | 98.91 | 90.90 | 97.80  | 89.25  | 98.18 |
| 5 Known | 99.99 | 94.95 | 99.94  | 95.88  | 98.75 |
| 6 Known | 96.63 | 93.57 | 99.86  | 94.72  | 95.68 |
| 7 Known | 98.58 | 90.47 | 98.55  | 88.70  | 97.50 |
| 8 Known | 100.00| 95.74 | 100.00 | 95.23  | 97.87 |
| 9 Known | 100.00| 97.30 | 100.00 | 94.21  | 97.93 |
| 10 Unknown | 50.87 | 86.89 | 56.98  | 96.20  | 89.03 |
| \(F_1\) | 80.63 | 91.26 | 81.64  | 93.96  | 94.82 |

Result of SA are shown in Table 1. Although the employment of Gaussian fitting is helpful to achieve better classification accuracies for known classes, it is too weak to reject unknown classes: 50.87% for sigmoid and 56.98 % for SoftMax. Comparatively, using boxplot increases the rejection rate of unknown classes dramatically, 86.89% for sigmoid and 96.20% for SoftMax, at the small cost of accuracy reduction for known classes which is acceptable. As for OpenMax, it achieves the highest \(F_1\) and its rejection rate for unknown classes is 89.03%, which is just a bit less than the proposed method. However, note that OpenMax’s best performance is achieved by tuning the hyperparameters \(\alpha\) and tailsizer (\(\alpha = 2\), tailsizer = 10), which need to use validation samples from unknown classes. Owing to this process, OpenMax method is tedious and impractical.

Table 2. OSC accuracy (%) on UP

| Class  | Sig_G | Sig_B | Soft_G | Soft_B | OM  |
|--------|-------|-------|--------|--------|-----|
| 1 Known | 98.30 | 90.68 | 98.06  | 88.77  | 94.49 |
| 2 Known | 96.92 | 88.77 | 96.00  | 85.27  | 92.36 |
| 3 Known | 99.80 | 95.95 | 99.85  | 95.93  | 85.03 |
| 4 Known | 98.02 | 88.49 | 99.09  | 89.25  | 89.44 |
| 5 Known | 91.51 | 85.99 | 92.28  | 83.94  | 84.09 |
| 6 Known | 99.65 | 93.76 | 99.96  | 92.92  | 87.58 |
| 7 Unknown | 21.70 | 45.84 | 26.25  | 52.56  | 47.72 |
| \(F_1\) | 78.79 | 82.00 | 80.18  | 83.56  | 81.02 |

Table 3. OSC accuracy (%) on KSC

| Class  | Sig_G | Sig_B | Soft_G | Soft_B | OM  |
|--------|-------|-------|--------|--------|-----|
| 1 Known | 99.52 | 94.52 | 95.56  | 96.83  | 93.69 |
| 2 Known | 100.00| 92.03 | 96.51  | 93.16  | 95.96 |
| 3 Known | 99.65 | 96.02 | 99.60  | 94.37  | 95.55 |
| 4 Known | 100.00| 96.11 | 99.94  | 96.73  | 98.94 |
| 5 Known | 99.54 | 92.98 | 93.89  | 92.27  | 97.22 |
| 6 Known | 100.00| 98.84 | 100.00 | 96.22  | 97.37 |
| 7 Known | 97.79 | 93.02 | 97.38  | 91.81  | 95.88 |
| 8 Unknown | 16.33 | 44.98 | 37.14  | 50.57  | 47.24 |
| \(F_1\) | 73.62 | 79.71 | 78.50  | 81.41  | 82.43 |
Results of UP and KSC are shown in Table 2 and 3, respectively. It can be summarized from Table 1, 2 and 3, our proposed method SoftMax-boxplot achieves the best performance for unknown classes rejection and does not decrease the classification accuracies for known classes markedly, compared to DOC and OpenMax. With these experimental comparisons, we verify the superiority of our method: simple yet effective.

In Figure 4, we visualize the OSC results of the proposed method for SA, UP and KSC. Compared to the ground-truth maps in Figure 3, we find that most pixels from unknown classes have be rejected when classifying the known classes rightly by SSRN-based DOCBP. However, we have to acknowledge that the OSC performance for UP and KSC is still unsatisfactory, which needs further exploration and improvement.

The impact of spatial window size of the input over the performance of the proposed method is reported in Table 4. The impact of training samples sizes is shown in Figure 5. The results show that the spatial window size and training samples size have a limited influence to the performance of the proposed method. In other words, the proposed method is a robust and stable algorithm for HSI OSC tasks.

Furthermore, we verify the DOCBP’s efficiency with another two deep networks: FDSSC and HybridSN. FDSSC is an end-to-end, fast, and dense spectral-spatial convolution framework [9]. It uses two sequential dense blocks to extract abundant spectral features and spatial features in HSIs. In this experiment, the spatial window size of input is set as 9 × 9, and 10 % labelled samples are randomly selected as training groups for all HSI OSC datasets. HybridSN [10] is a hybrid 3D and 2D CNN model for HIS classification. Different from SSRN and FDSSC, HybridSN applies principle component analysis (PCA) to reduce the spectral redundancy of original HSI data firstly. Following [10], the number of spectral bands after PCA is set to be 30 for SA and KSC, 15 for UP. The spatial window size of input is set as 25×25 for all OSC datasets. The training sample sizes for three OSC datasets are set as 30%. The experimental results are shown in Table 5. It can be found that DOCBP can work effectively with these two deep networks. What’s more, a powerful deep network is helpful for improving the OSC performance of DOCBP.

| Window | SA Unknown | SA F1 | UP Unknown | UP F1 | KSC Unknown | KSC F1 |
|--------|------------|------|------------|------|-------------|------|
| 5 × 5  | 96.20      | 93.96| 52.56      | 83.56| 50.57       | 81.41|
| 7 × 7  | 97.17      | 94.16| 56.20      | 84.05| 45.08       | 79.63|
| 9 × 9  | 95.13      | 92.68| 59.42      | 84.59| 45.41       | 79.54|
| 11 × 11| 97.07      | 92.83| 67.32      | 85.84| 50.57       | 80.73|

Figure 4. Visualization of the OSC results of the SSRN-based DOCBP. (a) SA. (b) UP. (c) KSC. The color bar below each map indicates the predicted distribution of unknown classes.
Figure 5. Impact of the training samples sizes on the proposed method. (a) SA. (b) UP. (c) KSC.

Table 5. OSC performance of the proposed method with different deep networks as its bases (%).

| Index     | FDSSC   | HybridSN |
|-----------|---------|----------|
|           | SA  | UP  | KSC | SA  | UP  | KSC |
| Unknown   | 96.58| 61.35| 59.91| 90.76| 83.08| 85.10|
| OA        | 93.28| 81.91| 79.71| 94.31| 92.54| 85.66|
| AA        | 91.35| 83.19| 87.95| 95.90| 91.31| 85.78|
| F1        | 93.07| 83.37| 81.82| 93.87| 92.18| 87.58|

4. Conclusion
Deep networks-based supervised methods have achieved astonishing performance for HSI closed set classification. However, in the real dynamic open world, a more important problem is OSC, where unknown classes may appear during testing. Till now, HSI OSC is still an open problem which has not been paid enough attention to. In this letter, we propose a new method to address the HSI OSC problems towards deep networks. The method employs boxplots to tighten the decision boundaries of SoftMax, called DOCBP. Evaluating on three publicly available HSI datasets, we show that DOCBP performs dramatically better than the OpenMax and DOC with less computation complexity. Besides, the method can work with different deep networks, efficiently and stably.

In our future work, we plan to improve the OSC performance during testing by designing more powerful deep networks. We also plan to enable the OSC method to possess the abilities to discover the hidden unseen classes in the rejected examples and learn the new classes incrementally.

5. References
[1] M.N. Islam, M.J. Freeman, L.M. Peterson, K. Ke, A. Ifarraguerri, C. Bailey, F. Baxley, M. Wager, A. Absi, J. Leonard, H. Baker, M. Rucci, Field tests for round-trip imaging at a 1.4 km distance with change detection and ranging using a short-wave infrared super-continuum laser, Appl. Opt., 55 (2016) 1584-1602.
[2] Ø. Ødegård, A.A. Mogstad, G. Johnsen, A.J. Sørensen, M. Ludvigsen, Underwater hyperspectral imaging: a new tool for marine archaeology, Appl. Opt., 57 (2018) 3214-3223.
[3] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, J.A. Benediktsson, Deep Learning for Hyperspectral Image Classification: An Overview, IEEE Transactions on Geoscience and Remote Sensing, (2019) 1-20.
[4] L. Huang, Y. Chen, Dual-Path Siamese CNN for Hyperspectral Image Classification With Limited Training Samples, IEEE Geoscience and Remote Sensing Letters, (2020) 1-5.
[5] Z. Dong, Y. Cai, Z. Cai, X. Liu, Z. Yang, M. Zhuge, Cooperative Spectral-Spatial Attention Dense Network for Hyperspectral Image Classification, IEEE Geoscience and Remote Sensing Letters, (2020) 1-5.
[6] A. Signoroni, M. Savardi, A. Barionio, S. Benini, Deep Learning Meets Hyperspectral Image Analysis: A Multidisciplinary Review, Journal of Imaging, 5 (2019) 52.
[7] L. Ying, H. Zhang, S.J.R.S. Qiang, Spectral–spatial classification of hyperspectral imagery with
3D convolutional neural network, Remote Sensing, 9 (2017) 67.

[8] Z. Zhong, J. Li, Z. Luo, M. Chapman, Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework, IEEE Transactions on Geoscience and Remote Sensing, 56 (2018) 847-858.

[9] W. Wang, S. Dou, Z. Jiang, L. Sun, A Fast Dense Spectral–Spatial Convolution Network Framework for Hyperspectral Images Classification, Remote Sensing, 10 (2018) 1068.

[10] S.K. Roy, G. Krishna, S.R. Dubey, B.B. Chaudhuri, HybridSN: Exploring 3D-2D CNN Feature Hierarchy for Hyperspectral Image Classification, IEEE Geoscience and Remote Sensing Letters, (2019).

[11] C. Geng, S.-J. Huang, S. Chen, Recent Advances in Open Set Recognition: A Survey, CoRR, abs/1811.08581 (2018).

[12] W.J. Scheirer, D.R.R. Anderson, S. Archana, T.E. Boult, Towards Open Set Deep Networks, IEEE transactions on pattern analysis and machine intelligence, 35 (2013) 1757-1772.

[13] Y. Liu, Y. Tang, L. Zhang, L. Liu, M. Song, K. Gong, Y. Peng, J. Hou, T. Jiang, Hyperspectral open set classification with unknown classes rejection towards deep networks, International Journal of Remote Sensing, 41 (2020) 6355-6383.

[14] S. Lei, X. Hu, L. Bing, DOC: Deep Open Classification of Text Documents, in: Conference on Empirical Methods in Natural Language Processing Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 2911–2916.

[15] A. Bendale, T.E. Boult, Towards Open Set Deep Networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1563-1572.

[16] M.O. Lawrence Neal, Xiaoli Fern, Weng-Keen Wong, Fuxin Li, Open Set Learning with Counterfactual Images, The European Conference on Computer Vision (ECCV), (2018) 613-628.

[17] D. Massart, J. Smeyers-Verbeke, X. Capron, K. Schlesier, Visual presentation of data by means of box plots, LC-GC Europe, 18 (2005) 215-218.