Hyperspectral Image Classification Using Random Occlusion Data Augmentation

Juan Mario Haut, *Student Member, IEEE,* Mercedes E. Paoletti, *Student Member, IEEE,* Javier Plaza, *Senior Member, IEEE,* Antonio Plaza, *Fellow, IEEE,* and Jun Li, *Senior Member, IEEE*

***Abstract*— Convolutional neural networks (CNNs) have become a powerful tool for remotely sensed hyperspectral image (HSI) classification due to their great generalization ability and high accuracy. However, owing to the huge amount of parameters that need to be learned and to the complex nature of HSI data itself, these approaches must deal with the important problem of overfitting, which can lead to inadequate generalization and loss of accuracy. In order to mitigate this problem, in this letter, we adopt random occlusion, a recently developed data augmentation (DA) method for training CNNs, in which the pixels of different rectangular spatial regions in the HSI are randomly occluded, generating training images with various levels of occlusion and reducing the risk of overfitting. Our results with two well-known HSIs reveal that the proposed method helps to achieve better classification accuracy with low computational cost.**

***Index Terms*— Convolutional neural networks (CNNs), hyperspectral images (HSIs), random occlusion data augmentation (DA).**

I. Introduction

REMOTELY sensed hyperspectral images (HSIs) comprise hundreds of continuous and narrow spectral bands, where each pixel (vector) characterizes uniquely the observed objects. HSIs have been widely used in many applications, including classification, segmentation, or target detection. Many different machine learning techniques have been used for extracting information from HSIs, including support vector machines (SVMs) [1], extreme learning machines (ELMs) [2], and single-hidden layer feedforward networks (SHLFN) [3]. Recent advances in earth observation missions have allowed capturing more complex HSI images, comprising a larger number of spectral bands and higher spectral and spatial resolution. This has imposed requirements in terms of run times and storage [4]. In this context, convolutional neural networks (CNNs) have emerged as a powerful tool for HSI data interpretation [5], [6], consolidating deep learning CNN-based approaches as the current state of the art in HSI data classification [3].

Most available CNN techniques for HSI data classification suffer from the problem of overfitting. Coupled with the great spectral variability present in HSIs, this problem complicates the learning process. To the best of authors' knowledge, traditional efforts to mitigate the overfitting problem [7] and improve the generalization ability of CNNs are based on increasing the amount of training data by including large spatial patches in the training process, possibly by means of geometric transformations [6]. Other techniques focus on applying regularization methods; for instance, Paoletti *et al.* [5] use dropout and maxpooling techniques. Other recent efforts are aimed at improving the model architecture, e.g., by adding more connections between layers [8] or by developing residual architectures [9] to feed each layer with additional information. These methods have been further extended by other existing strategies, such as pixel-pairs features (PPFs) [10], active learning [11], or fully connected architectures [12]. However, these methods reduce overfitting at the expense of making an extensive use of the output (softmax) layer, which increases computational complexity. For instance, Li *et al.* [12] try to insert new information in this layer using principal component analysis (PCA), while Haut *et al.* [11] improve the model's generalization by incorporating the samples that have more uncertainty. Similarly, the work by Li *et al.* [10] intends to solve the problem of data occlusions by using PPFs in the pixel neighborhood information.

In fact, data occlusion is an important problem in remote sensing, which is related to those areas of the surface of the earth which are not visible from the remote sensor due to external factors, such as the interruption between the sensor and the target 2-D surface, or the presence of nearby 3-D objects. This problem is motivated by the presence of clouds, shadows, or other objects, which result in a loss of information over the scene. Inspired by human reasoning, which is intrinsically based on the interpretation of 3-D spaces [13], several techniques have been developed to address data occlusions. This concept can also be used to enhance the training of machine learning methods. In this letter, we adopt a recently

Manuscript received August 30, 2018; revised October 15, 2018; accepted March 20, 2019. Date of publication April 22, 2019; date of current version October 30, 2019. This work was supported in part by Spanish Ministry under Grant FPU14/02012-FPU15/02090 and Grant ESP2016-79503-C2-2-P, in part by Junta de Extremadura under Grant GR18060, and in part by European Union under Grant 734541-EXPOSURE. *(Corresponding author: Jun Li.)*

J. M. Haut, M. E. Paoletti, J. Plaza, and A. Plaza are with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, 10003 Cáceres, Spain (e-mail: juanmariohaut@unex.es; mpaoletti@unex.es; jplaza@unex.es; aplaza@unex.es).

J. Li is with the Guangdong Provincial Key Laboratory of Urbanization and Geosimulation, Center of Integrated Geographic Information Analysis, School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China (e-mail: lijun48@mail.sysu.edu.cn).

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.2019.2909495

developed technique for data augmentation (DA) [14] that reduces the overfitting of CNN models by modifying randomly (in each learning batch and in each iteration) some of the input data by replacing a part of them with an empty patch of random dimensions. This technique, known as random occlusion, is computationally efficient, easy to implement, and can be inserted into CNN-based HSI classification frameworks in a straightforward manner, without penalty at runtime.

The main innovative contributions of this letter can be summarized as follows.

1) We introduce a new deep CNN model for HSI classification that eliminates the need to use large spatial patches in the training phase in order to reduce overfitting. The use of large spatial patches has been adopted in several works [12], [15], but this leads to high computational complexity. Specifically, the aforementioned approaches use patch sizes of 27×27 [15] and 48×48 [12] pixels, respectively. Our proposed method is able to achieve similar results to those reported by the aforementioned methods with smaller patch size (i.e., 23×23).
2) The proposed method offers a new way to increase the amount of available training data in the sense that by generating different samples with different occluded areas, the number of available samples increases, and the associated spatial content varies considerably. As a result, in addition to introducing variations in spatial features (making the learning process more robust), the proposed method can be considered as an effective DA technique.

Furthermore, the adopted strategy is quite simple and easy to implement/integrate in HSI classification frameworks, and it does not significantly increase the computational load as opposed to some widely used DA methods, which are quite complex and require extra parameter learning and/or high memory consumption. Also, it can be complemented by other traditional DA techniques based on crops or rotations. Section II describes the proposed method in detail. The experimentation over two HSIs (Section III) demonstrates that the proposed method exhibits good results with very few training samples, improving the performance of a spatial CNN (2-DCNN)—an architecture that is traditionally hampered by overfitting—due to the variability of the information contributed by means of the adopted random occlusion DA method. Finally, Section IV concludes this letter with some remarks and hints at plausible future research lines.

II. Methodology

An HSI data set $\mathbf{X}$ is composed by $n_1 \times n_2$ pixel vectors, where each pixel $\mathbf{x}_i \in \mathbb{R}^{n_{\text{bands}}}$ collects the spectral information in n_{bands} spectral bands, creating a huge data cube, $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_{\text{bands}}}$.

In order to process the HSI image $\mathbf{X}$ using a traditional 2-DCNN model, a preprocessing stage is often carried out, consisting of two main steps [16]: 1) dimensionality reduction, normally using PCA [17], which reduces the n_{bands} to one PC, reducing the spectral dimensionality and the requirements in terms of runtime and 2) grouping of the HSI pixels in regions in order to create, for each pixel vector $\mathbf{x}_i$, a $d \times d$ spatial patch $\mathbf{p}_i \in \mathbb{R}^{d \times d}$ that contains the spatial neighborhood around the pixel. These patches are used to extract feature maps that will represent and characterize the original input data. Although 2-DCNN models outperform pixel-based (1-DCNN) models by the inclusion of spatial information, they are quite sensitive to the quality of the spatial information contained in the patches, in the sense that small variations produced by noise or occlusions, together with overfitting, can lead the model to misclassifications and oversmoothing of small objects [18]. In order to address this issue, a large number of training samples $\mathcal{D}_{\text{train}} = \{\mathbf{p}_i, y_i\}_{i=1}^{L}$ (being y_i the corresponding label of $\mathbf{p}_i$) or additional data variations $L + L_+$ (via DA) are often required.

Instead of obtaining L_+ new training samples, our adopted method increases randomly the variability of training samples by performing random occlusions over the data, which is a simple and inexpensive form of DA. In this regard, the pool of training samples $\mathcal{D}_{\text{train}}$ is first shuffled, where each $\mathbf{p}_i$ grouped into different batches $\mathbf{B}_j$ in each epoch of the network's training process. These batches are initially preprocessed before being fed to the 2-DCNN model, assigning an initial random probability from a Gaussian distribution, p, to every patch $\mathbf{p}_i \in \mathbf{B}_j$. Those $\mathbf{p}_i$ with higher values of p are modified by random occlusions, until a percentage of randomly occluded samples per batch (denoted by n) is reached.

The occlusion strategy works as follows. For each $\mathbf{p}_i$, we calculate a rectangular region $\mathbf{p}_i^* \in \mathbb{R}^{d_1^* \times d_2^*}$ allocated inside the patch. In order to obtain both the spatial dimension $d_1^* \times d_2^*$ and the exact location of $\mathbf{p}_i^*$ in $\mathbf{p}_i$, the occlusion strategy first obtains the area of the original input patch $\mathbf{p}_i$ as $a_i = d \cdot d$. From this value, the method calculates a smaller region (whose size is between a minimum and maximum threshold) over the original patch area, $a_i^* = \text{rand}(t_{\max} \cdot a_i, t_{\min} \cdot a_i)$. This a_i^* becomes the area of $\mathbf{p}_i^*$, which complies with the expression $a_i^* = d_1^* \cdot d_2^*$. The next step is to obtain d_1^* and d_2^* as follows:

$$d_1^* = \sqrt{a_i^* \cdot r_i} \quad \text{and} \quad d_2^* = \sqrt{\frac{a_i^*}{r_i}} \tag{1}$$

where r_i is a randomly selected value between a minimum and maximum threshold value: $r_i = \text{rand}(r_{\min}, r_{\max})$, whose goal is to avoid nondesirable shapes (such as horizontal and vertical lines). Finally, the process randomly allocates the $\mathbf{p}_i^*$ by making sure that it does not exceed the margins of the original $\mathbf{p}_i$, and fills the area with a predetermined value, that is, the occluded pixels have been set to 0.5 in our experiments.

The aforementioned process is repeated at every epoch of the training process, over all the batches that conform the training data set, allowing the model to be fed with a rich set of variations over the same training data. The procedure is illustrated in Fig. 1, which shows a graphical result of this processing method. The details of the 2-DCNN model architecture adopted in our work to implement the aforementioned strategy are given in Table I.

III. Experiments

A. Experimental Configuration

With the aim of testing the performance of the proposed DA method for HSI classification using the baseline 2-DCNN architecture given in Table I, a set of experiments have been

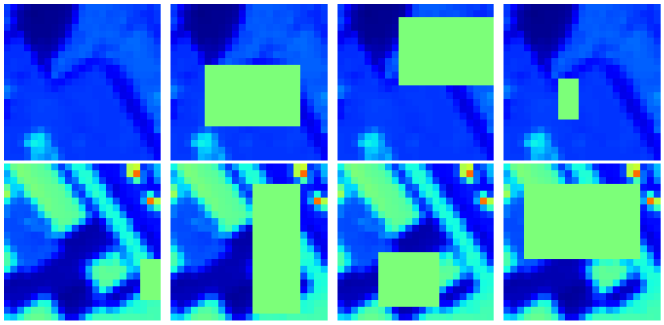

Fig. 1. Example of random occlusion DA in two different input patches. (Leftmost column) Original (not occluded) patches, while the other patches exhibit occluded zones shown in green.

TABLE I

ARCHITECTURAL DETAILS OF THE PROPOSED 2-DCNN FOR HSI CLASSIFICATION

Layer ID	Kernel/Neurons	Max Pooling	Act. function
Conv1	$50 \times 3 \times 3 \times 1$	No	ReLU
Conv2	$100 \times 5 \times 5 \times 50$	2×2	ReLU
Conv3	$200 \times 5 \times 5 \times 100$	2×2	ReLU
Conv4	$400 \times 2 \times 2 \times 200$	No	ReLU
FC1	300	-	ReLU
FC2	$n_{classes}$	-	Softmax

performed over a hardware environment composed by a sixth generation Intel Core i7-6700K processor with 8 M of Cache and up to 4.20 GHz (4 cores/8 way multitask processing), 40 GB of DDR4 RAM with a serial speed of 2400 MHz, a GPU NVIDIA GeForce GTX 1080 with 8-GB GDDR5X of video memory and 10 Gb/s of memory frequency, a Toshiba DT01ACA HDD with 7200 RPM and 2 TB of storage capacity, and an ASUS Z170 pro-gaming motherboard. In order to efficiently implement the proposed approach, it has been parallelized over the GPU using Cuda language over Pytorch framework [19]. All the codes and examples presented in this letter are available online.[1]

B. Hyperspectral Data Sets

Two well-known HSIs have been used to perform experiments. The first one is the $145 \times 145 \times 200$ Indian Pines (IP) data set, captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor [20] in 1992 over several agricultural fields in Northwestern Indiana. The 16 different classes are available in this scene. The second data set is the University of Pavia (UP) scene, acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor [21] over a $610 \times 340 \times 113$ urban area, comprising nine different class labels.

C. Results and Discussion

1) Experiment 1: In this experiment, we compare the performance of the 2-DCNN model (implemented using the proposed strategy) with seven standard HSI classifiers which traditionally suffer from the overfitting problem. Specifically, six pixel-wise classifiers have been selected: SVM implemented with radial basis function (SVM-RBF), random forest (RF), SHLFN, ELM, kernel ELM (KELM), and a spectral-based CNN (1-DCNN). In addition, the baseline spatial-based 2-DCNN without the proposed strategy has also been included. The resulting values from RF, SVM-RBF, SHLFN, ELM, KELM, and 1-DCNN have been extracted from Ghamisi *et al.* [3], while the two 2-DCNN models (with and without DA) have been implemented in accordance with the architectural details given in Table I. The reported results correspond to the mean of five Monte Carlo runs, and we use five different training percentages: 1%, 3%, 5%, 10%, and 15% of randomly selected labeled data from the two considered data sets. The rest of the data has been used for testing.

[1]https://github.com/mhaut/ROhsi

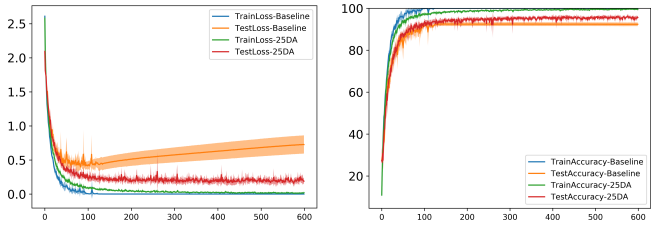

Fig. 2. Evolution of the (Left) loss and (Right) OA as a function of the number of training epochs when using the 2-DCNN with and without the proposed random occlusion DA method (occluding 25% of the IP data in each training batch).

Table II reports the overall accuracies (OAs) obtained after the conducted experiments. On the leftmost part of the table, we report the results obtained by traditional classifiers. On the rightmost side of the table, we report the results obtained by a 2-DCNN implemented with the proposed DA approach (2-DCNN-n), with n set to 5%, 15%, 25%, and 50% occlusion percentages.

Focusing on the results obtained for the IP data set and looking at the leftmost part of the table, we can see how the OA is increased as we add more training samples to the classifiers, being the 2-DCNN the one with best OA values. This is mainly due to the efficient use of the spatial information contained in the input data, which improves the generalization ability. However, these OAs are lower than those achieved by the 2-DCNN implemented with the proposed DA approach. In the rightmost part of Table II, we can see how the proposed method increases the accuracy by more than 3% points with only occluding 5% of each batch, until reaching final solutions with OAs that are 4% or 5% points superior to those achieved by the standard 2-DCNN model when occluding the 50% of the batch. This reveals that even with very little effort (i.e., by simply occluding 10%–15% of the data), the proposed method is able to significantly increase the OA. If we now focus on the UP data set, we can also observe that the OA values of standard classification methods (at the leftmost part of Table II) are increased when more training data are used, being the SVM-RBF the best classifier when few training samples are available. These results are improved by those obtained by the traditional 2-DCNN when 3% and higher training percentages are used. If we compare the baseline 2-DCNN with the proposed implementation in the rightmost part Table II, we can see that by adding only 5% of occlusions into batches, the proposed method is able to outperform the OA in more than 1% point, a difference that is increased as more occlusions is

TABLE II

OAs OBTAINED BY DIFFERENT CLASSIFICATION METHODS FOR THE IP AND UP SCENES. THE LEFTMOST PART SHOWS THE RESULTS OBTAINED BY TRADITIONAL CLASSIFIERS WITHOUT THE PROPOSAL. THE RIGHTMOST PART SHOWS THE RESULTS OBTAINED BY A 2-DCNN WITH THE PROPOSED DA APPROACH (2-DCNN-n), WITH n SET TO 5%, 15%, 25%, AND 50% OCCLUSION PERCENTAGES

	T. Size	RF	SVM-RBF	SHLFN	ELM	KELM	1DCNN	2DCNN	2DCNN-5%	2DCNN-15%	2DCNN-25%	2DCNN-50%
INDIAN P.	1%	-	-	-	-	-	-	50.78 ±2.24	54.5 ±1.13	53.48 ±1.51	53.62 ±2.27	**55.02** ±0.74
	3%	-	-	-	-	-	-	70.44 ±1.44	73.40 ±1.16	73.04 ±1.40	74.17 ±1.53	**76.66** ±1.64
	5%	69.00	74.52	77.13	72.23	80.38	75.37	80.01 ±1.08	83.7 ±1.03	85.33 ±1.14	86.07 ±0.53	**86.70** ±1.64
	10%	75.58	81.00	83.10	78.88	84.85	82.66	91.47 ±1.26	94.92 ±0.54	95.15 ±0.71	95.38 ±0.54	**96.46** ±0.54
	15%	78.19	83.91	85.28	81.59	86.81	86.13	94.75 ±0.31	97.95 ±0.34	98.14 ±0.28	98.35 ±0.33	**98.63** ±0.40
PAVIA U.	1%	81.35	88.91	88.11	80.28	82.05	85.70	84.28 ±1.23	86.85 ±1.98	87.86 ±1.28	87.93 ±1.07	**87.94** ±1.87
	3%	-	-	-	-	-	-	93.84 ±0.59	94.88 ±0.93	95.72 ±0.87	**96.00** ±0.74	95.91 ±0.66
	5%	87.36	93.43	93.19	85.35	87.07	91.01	96.73 ±0.51	97.84 ±0.40	98.23 ±0.20	98.26 ±0.32	**98.43** ±0.21
	10%	89.51	94.45	94.36	86.72	88.69	94.13	98.96 ±0.26	99.50 ±0.18	99.61 ±0.03	**99.72** ±0.01	99.67 ±0.16
	15%	90.48	94.89	94.91	87.24	89.41	95.29	99.50 ±0.09	99.84 ±0.03	99.86 ±0.06	99.91 ±0.02	**99.92** ±0.01

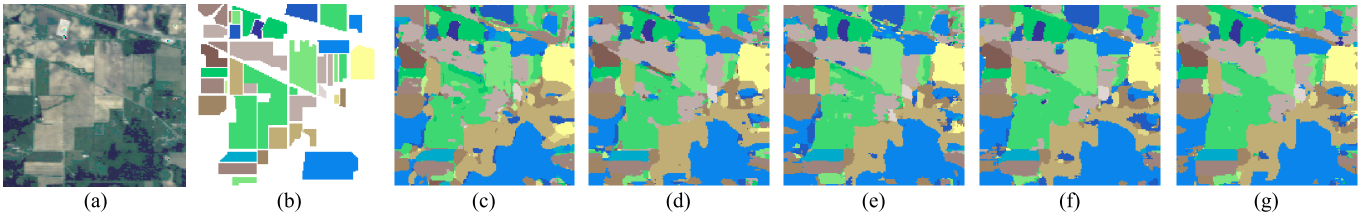

Fig. 3. Classification maps for the IP data set. (a) Simulated RGB composition of the scene. (b) Ground-truth classification map. (c)–(h) Classification maps corresponding to Table III. Note that, the OA values are shown in brackets and the best result is highlighted in bold font. (a) RGB. (b) GT. (c) 2-DCNN (92.73%). (d) 2-DCNN-5% (96.00%). (e) 2-DCNN-15% (96.94%). (f) 2-DCNN-25% (96.94%). (g) 2-DCNN-50% (97.50%).

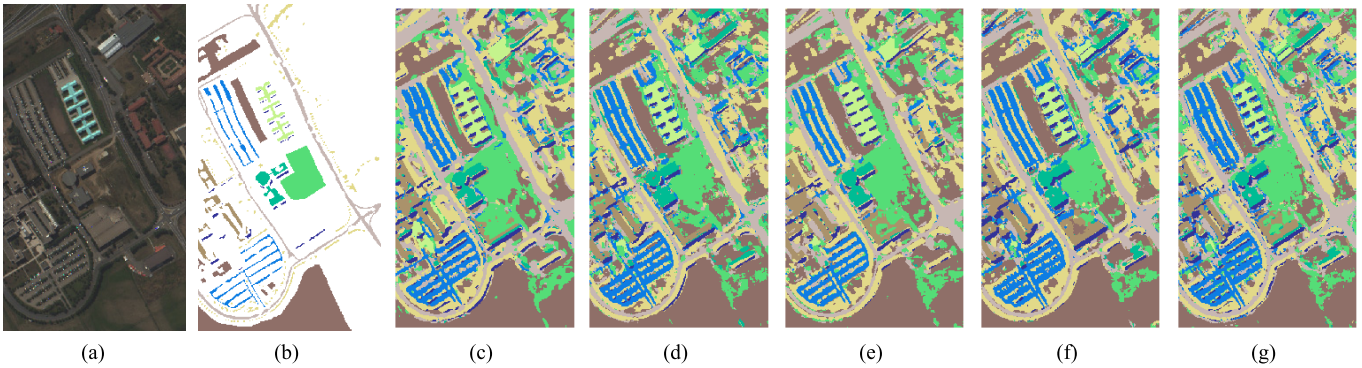

Fig. 4. Classification maps for the UP data set. (a) Simulated RGB composition of the scene. (b) Ground-truth classification map. (c)–(h) Classification maps corresponding to Table IV. Note that, the OA values are shown in brackets and the best result is highlighted in bold font. (a) RGB. (b) GT. (c) 2-DCNN (98.33%). (d) 2-DCNN-5% (98.88%). (e) 2-DCNN-15% (99.12%). (f) 2-DCNN-25% (99.17%). (g) 2-DCNN-50% (99.09%).

added to the training data, until reaching the best OA (99.92%) with 15% of training data and 50% of occlusions.

2) Experiment 2: Our second experiment compares the proposed approach with traditional spatial-based CNN models. Specifically, we report the results obtained by the spatial model proposed by Chen *et al.* [15] (which is shallower than the proposed 2-DCNN and fed with bigger spatial patches), the proposed baseline 2-DCNN architecture without the proposal improvement, and the 2-DCNN framework with different percentages of the proposed random occlusion DA strategy. To facilitate the comparison, we have considered (for each of the two considered scenes) the number of pixels per class indicated by Chen *et al.* [15].

Tables III and IV, respectively, provide the obtained results for the IP and UP scenes, reporting the OA and also the average accuracy (AA), kappa statistic, and runtime in the considered computing environment (in seconds).

The results reported in Table III indicate that the baseline 2-DCNN provides better results than the method in Chen *et al.* [15] in terms of accuracies and runtime. These results are improved when incorporating the random occlusion DA strategy to the 2-DCNN, increasing the OA in about 6%–8% points over the method in Chen *et al.* [15] and in about 3%–5% points over the standard 2-DCNN. Fig. 2 graphically shows how the standard 2-DCNN tends to quickly overfit, reaching a very low loss value in the training stage but being unable to reduce the loss in the test stage, which ultimately reduces the test accuracy. However, when the random occlusion DA strategy is adopted, several variations are introduced that prevent the training from reaching the lowest loss value, and this allows the model to continue learning, which leads to low test errors and high test accuracies. On the other hand, it is important to emphasize that the runtime of the 2-DCNN, with and without the random occlusion DA strategy,

TABLE III

OA, AA, KAPPA STATISTIC, AND RUNTIME OBTAINED BY DIFFERENT SPATIAL-BASED CLASSIFICATION METHODS FOR THE IP SCENE

Class	Samples	Chen *et al.* [15]	2DCNN -	2DCNN 5%	2DCNN 15%	2DCNN 25%	2DCNN 50%
1	30	**99.65**	99.57	99.13	99.57	99.57	99.57
2	150	90.64	84.47	92.45	92.82	93.75	**94.27**
3	150	**99.11**	93.93	96.22	97.52	97.86	98.22
4	100	100.00	99.75	99.75	100.00	100.00	99.92
5	150	98.84	98.26	98.30	98.84	99.17	**99.09**
6	150	97.95	98.58	**99.67**	99.45	99.51	99.62
7	20	100.00	100.00	100.00	100.00	100.00	100.00
8	150	**100.00**	99.75	99.87	99.79	99.96	99.92
9	15	100.00	100.00	100.00	100.00	100.00	100.00
10	150	95.33	94.22	94.79	98.27	96.67	96.60
11	150	78.21	85.67	92.70	93.94	94.35	**95.89**
12	150	99.39	96.53	97.91	98.89	98.45	**99.76**
13	150	100.00	100.00	100.00	100.00	100.00	100.00
14	150	97.71	99.10	99.30	**99.81**	99.59	99.59
15	50	**99.31**	97.15	98.08	99.02	96.68	97.46
16	50	99.22	99.35	**99.57**	99.14	99.35	98.71
Overall Accuracy		89.99	92.73	96.00	96.94	96.94	**97.50**
Average Accuracy		97.19	96.64	97.98	98.57	98.43	**98.66**
Kappa		87.95	91.76	95.44	96.52	96.51	**97.15**
Runtime (s.)		357.00	**82.90**	83.40	83.32	83.59	83.88

TABLE IV

OA, AA, KAPPA STATISTIC, AND RUNTIME OBTAINED BY DIFFERENT SPATIAL-BASED CLASSIFICATION METHODS FOR THE UP SCENE

Class	Samples	Chen *et al.* [15]	2DCNN -	2DCNN 5%	2DCNN 15%	2DCNN 25%	2DCNN 50%
1	200	97.11	98.11	98.61	98.86	99.00	**99.03**
2	200	87.66	97.57	98.44	98.81	**98.92**	98.59
3	200	99.69	98.95	98.94	99.30	99.49	99.69
4	200	98.49	99.45	99.56	99.63	99.41	**99.74**
5	200	100.00	99.91	99.97	100.00	99.82	99.87
6	200	98.00	98.78	99.11	99.13	99.11	**99.24**
7	200	99.89	99.77	99.47	**99.95**	99.79	99.92
8	200	99.70	99.19	99.76	**99.90**	99.84	99.84
9	200	97.11	99.75	**99.92**	99.60	99.73	99.83
Overall Accuracy		94.04	98.33	98.88	99.12	**99.17**	99.09
Average Accuracy		97.52	99.05	99.31	99.47	99.46	**99.53**
Kappa		92.43	97.79	98.51	98.84	**98.90**	98.80
Runtime (s.)		607.11	174.68	**174.64**	174.81	174.97	175.52

is very similar (less than one second difference), being in both cases lower than that of the method in [15]. This makes the adopted strategy a very attractive one to easily increase the performance of the network at very low computational cost.

The results obtained with the UP scene provide similar observations (see Table IV), with improvements of over 5% points in OA when compared to the method in [15]. In our experiment, the proposed approach is faster than the 2-DCNN baseline when occluding 5% of the data in each batch.

For illustrative purposes, Figs. 3 and 4, respectively, show some of the obtained classification maps for the IP and UP data sets, using the 2-DCNN baseline model and the proposed approach (occluding 5%, 15%, 25%, and 50% of training samples in each batch). Once again, the proposed method is able to improve the performance of the model at no cost.

IV. CONCLUSION

This letter evaluates a simple DA approach for HSI classification which is based on randomly occluding areas of the input data to reduce overfitting problems in CNN models. The obtained results demonstrate that the adopted approach is quite efficient, as it is able to improve the generalization ability of CNNs without increasing the computational cost. Since the adopted approach is not restricted to spatial-based 2-DCNNs, in the future, we plan to incorporate it to spatial–spectral 3-DCNN architectures.

REFERENCES

[1] B. Waske, S. van der Linden, J. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.

[2] J. M. Haut, M. E. Paoletti, J. Plaza, and A. Plaza, "Fast dimensionality reduction and classification of hyperspectral images with extreme learning machines," *J. Real-Time Image Process.*, vol. 15, no. 3, pp. 439–462, Oct. 2018. doi: 10.1007/s11554-018-0793-9.

[3] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017.

[4] J. Haut, M. Paoletti, J. Plaza, and A. Plaza, "Cloud implementation of the K-means algorithm for hyperspectral image analysis," *J. Supercomput.*, vol. 73, no. 1, pp. 514–529, Jan. 2017.

[5] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogram. Remote Sens.*, vol. 145, Part A, pp. 120–147.

[6] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[7] J. A. Richards and X. Jia, "Using suitable neighbors to augment the training set in hyperspectral maximum likelihood classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 774–777, Oct. 2008.

[8] J. M. Haut, R. Fernandez-Beltran, M. E. Paoletti, J. Plaza, A. Plaza, and F. Pla, "A new deep generative network for unsupervised remote sensing single-image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6792–6810, Nov. 2018.

[9] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[10] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[11] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, Nov. 2018.

[12] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi, and J. Hu, "Classification of hyperspectral imagery using a new fully convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 292–296, Feb. 2018.

[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 746–760.

[14] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. (2017). "Random erasing data augmentation." [Online]. Available: https://arxiv.org/abs/1708.04896

[15] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[16] L. Zhang, L. Zhang and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016. doi: 10.1109/MGRS.2016.2540798.

[17] Q. Sun, X. Liu, and M. Fu, "Classification of hyperspectral image based on principal component analysis and deep learning," in *7th IEEE Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2017, pp. 356–359.

[18] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2015, pp. 4959–4962.

[19] A. Paszke *et al.*, "Automatic differentiation in pytorch," Tech. Rep., 2017. [Online]. Available: https://github.com/pytorch/pytorch/issues/4126

[20] R. O. Green *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, Sep. 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425798000649

[21] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, H. van der Piepen, and M. Schroder, "ROSIS (Reflective Optics System Imaging Spectrometer)-A candidate instrument for polar platform missions," in *Optoelectronic Technologies for Remote Sensing from Space*, vol. 868. International Society for Optics and Photonics, 1988, pp. 134–142.