Unsupervised Segmentation of Hyperspectral Images Using 3-D Convolutional Autoencoders

Jakub Nalepa, *Member, IEEE*, Michal Myller, Yasuteru Imai, Ken-Ichi Honda, Tomomi Takeda, and Marek Antoniak

***Abstract*— Hyperspectral image analysis has become an important topic widely researched by the remote sensing community. Classification and segmentation of such imagery help understand the underlying materials within a scanned scene since hyperspectral images convey detailed information captured in a number of spectral bands. Although deep learning has established the state-of-the-art in the field, it still remains challenging to train well-generalizing models due to the lack of ground-truth data. In this letter, we tackle this problem and propose an end-to-end approach to segment hyperspectral images in a fully unsupervised way. We introduce a new deep architecture which couples 3-D convolutional autoencoders with clustering. Our multifaceted experimental study—performed over the benchmark and real-life data—revealed that our approach delivers high-quality segmentation without any prior class labels.**

***Index Terms*— Autoencoder, clustering, deep learning, hyperspectral imaging (HSI), unsupervised segmentation.**

I. Introduction

HYPERSPECTRAL imaging (HSI) provides detailed information about the material within a captured scene. It registers a number of spectral bands, commonly up to hundreds of them, and can be exploited to understand the location and characteristics of the objects in the process of HSI *classification* and *segmentation*. In classification, we assign a label to an input pixel, whereas in segmentation, we are focused on finding the boundaries of objects within an image.[1] Due to the increased availability of hyperspectral sensors, HSI analysis has become an important topic tackled by machine learning, remote sensing, and pattern recognition communities. Such imagery has multiple applications in a plethora of fields, including biochemistry, medicine, geosciences, military defense, and more [1]. HSI is an indispensable tool in Earth observation, as it captures Earth peculiarities that are useful in precision agriculture, managing environmental disasters, soil monitoring, or prediction of environmental events.

HSI classification and segmentation techniques can be divided into conventional machine learning algorithms, requiring feature engineering, i.e., feature extraction and selection [2], [3], and deep learning-powered approaches, in which the appropriate representation is learned during the training [4]–[10]. To deploy deep models in practice, we need large and representative ground-truth training sets. It is a serious limiting factor in the hyperspectral Earth observation analysis, where transferring HSI from an imaging satellite back to Earth is extremely costly. Creating new ground-truth sets is error-prone and requires building a thorough understanding of the materials within a scene. Therefore, it involves acquiring observational ground-sensor data—it is often cost and time inefficient. These difficulties result in a very small number of ground-truth HSIs. We analyzed 17 recent articles in which only *seven* benchmarks were exploited, with only *three* of them being "widely used": Pavia University (15 articles), Indian Pines (8 articles), and Salinas Valley (5 articles) [11]. Xu *et al.* [12] introduced a new partially annotated HSI benchmark and exploited it in the framework of the IEEE Geoscience and Remote Sensing Society Data Fusion Contest.

There are three main approaches to deal with the limited ground-truth hyperspectral sets:1) *data augmentation*, 2) *transfer learning*, and 3) *unsupervised* analysis of HSI, with 1) and 2) being exploited mostly in deep-learning-powered techniques. Data augmentation is a process of generating artificial examples following the original data distribution. Such samples can extend the training sets, or they can be elaborated at the inference time, to build an intrinsic ensemble-like deep model [13]. In transfer learning, we train feature extractors over the *source* training data, and apply them to the *target* data [14]. This approach allows us to benefit from the available data to train efficient extractors—the classification part of a network is later fine-tuned over the target HSI. Both augmentation and transfer learning require the annotated target sets to either input them to an augmentation engine or to utilize them for fine-tuning the deep models. Hence, their usefulness is limited in scenarios where manually analyzed HSIs do not exist and are infeasible to generate.

Unsupervised segmentation offers the possibility of processing HSI *without* any prior class labels. Although the literature in unsupervised HSI segmentation is rather limited, there are approaches which benefit from mean shift filtering [15], diffusion-based dimensionality reduction

Manuscript received July 19, 2019; revised November 26, 2019; accepted December 15, 2019. Date of publication January 1, 2020; date of current version October 27, 2020. This work was supported in part by the European Space Agency (HYPERNET) and in part by the Polish National Centre for Research and Development under Grant POIR.01.01.01-00-0356/17. The work of Jakub Nalepa was supported by the Silesian University of Technology Funds under The Rector's Habilitation Grant 02/020/RGH19/0185. The work of Yasuteru Imai, Ken-Ichi Honda, and Tomomi Takeda was supported by the Hyperspectral Imager SUIte Public Research promoted by the Ministry of Economy, Trade and Industry, Japan. *(Corresponding author: Jakub Nalepa.)*

Jakub Nalepa and Michal Myller are with the Silesian University of Technology, Gliwice, Poland, and also with KP Labs, Gliwice, Poland (e-mail: jnalepa@ieee.org).

Yasuteru Imai and Ken-Ichi Honda are with Kokusai Kogyo, Company, Ltd., Tokyo 102-0085, Japan (e-mail: yasuteru_imai@kk-grp.jp; kenichi_honda@kk-grp.jp).

Tomomi Takeda is with the Japan Space Systems, Tokyo 105-0011, Japan (e-mail: takeda-tomomi@jspacesystems.or.jp).

Marek Antoniak is with KP Labs, Gliwice, Poland.

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

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.2019.2960945

[1]Therefore, segmentation involves the classification of separate pixels.

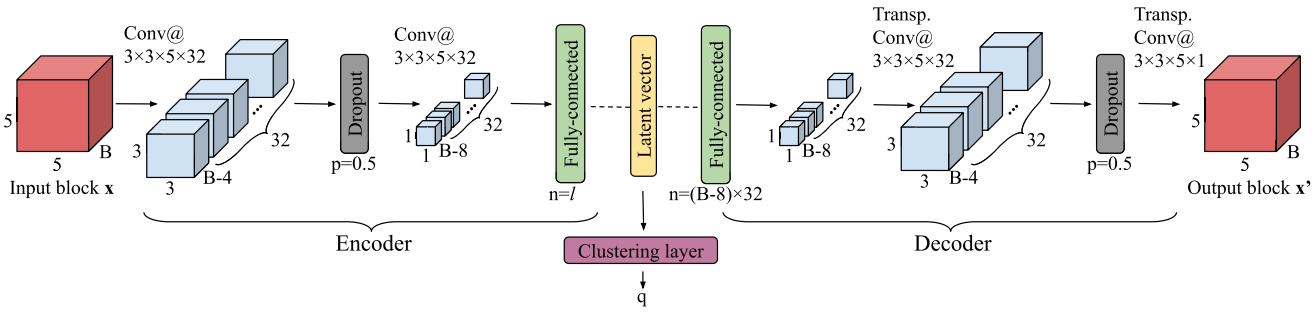

Fig. 1. Our 3D-CAE coupled with a clustering layer is trained in two stages—first, we learn a latent data representation (the clustering layer is not used in this stage, and the loss reflects the data reconstruction abilities of the 3D-CAE), and then we focus on clustering while still allowing for improvements in the latent representations by incorporating the clustering loss into the loss function.

followed by clustering [16], and phase-correlation analysis [17]. Mou *et al.* [18] used a fully convolution–deconvolution network for unsupervised spectral–spatial feature learning. Then, the convolutional subnetwork was used as a generic feature extractor over the target data. A similar technique of extracting deep features using stacked sparse autoencoders, and later embedding them into linear support vector machines have been proposed in [19].

In this letter, we tackle the problem of limited ground-truth hyperspectral sets and propose a deep learning technique for unsupervised HSI segmentation. Inspired by a recent work by Guo *et al.* [20], we introduce a 3-D convolutional autoencoder (3D-CAE) architecture[2] to learn embedded features, which later undergo clustering (Section II). This clustering is performed *during* the network training with a clustering-oriented loss; therefore, our method delivers end-to-end unsupervised HSI segmentation. To the best of our knowledge, such approaches have not been investigated in the HSI literature so far. We performed a multifaceted experimental study—over the benchmark and real-life hyperspectral data—to understand the abilities of the proposed technique. It showed that our method offers high-quality and consistent segmentation and does not require any prior class labels to effectively segment HSI (Section III).

II. Unsupervised HSI Segmentation Using 3-D Convolutional Autoencoders

In our approach (3D-CAE), we exploit 3D-CAEsto extract deep features which later undergo clustering (Fig. 1). In the encoding part of the network, we capture both spectral and spatial features within an input 3-D patch $\boldsymbol{x}$ of size $5 \times 5 \times \mathrm{B}$, where B is the number of bands, and the patches are extracted with unit stride. We use two convolutional layers denoted as Conv@$h_k \times w_k \times d_k \times k$, for which we define the height (h_k), width (w_k), and depth (d_k) of the kernels. Here, we utilize $k = 32$ unit-stride kernels for all convolution/transposed convolution layers. These layers are interleaved with one dropout layer with the dropout probability of $p = 0.5$, acting as a regularizer. The central-pixel features in the patch are reshaped to form a 1-D vector which becomes an input to a fully connected (FC) (*embedding*) layer with l neurons, where l is a hyperparameter of 3D-CAE. The output of this layer is the latent vector. These embedded features are transformed back to the original 3-D patch to get the output 3-D patch $\boldsymbol{x'}$ in the decoding part of a CAE. It is a mirrored version of the encoder with the transposed convolutions applied for upsampling. The CAE is learned in the first training stage with the following reconstruction loss:

$$L_r = \frac{1}{p}\sum_{i=1}^{p} \left|\left|\boldsymbol{x}_i - \boldsymbol{x'}_i\right|\right|_2^2 \tag{1}$$

where p is the number of 3-D *patches* in a batch. This stage, in which we do *not* use the clustering layer, runs until reaching convergence or the stopping condition. In this letter, the optimization terminates if the difference between two consecutive reconstruction loss values is less than $\epsilon = 10^{-6}$.

In the second training stage, we modify the loss function and "switch ON" the clustering layer—it is connected to the embedded layer of CAE which outputs the latent vector z_i for the ith patch. The embedded features are assigned a soft label q_i in the clustering layer. As proposed in [20], this layer maintains the cluster centers μ_j, where $j = 1, 2, \ldots J$, and J is the number of clusters, as trainable weights. The probability of assigning an input 3-D patch $\boldsymbol{x}_i$ to each jth cluster is generated using the student's t-distribution with one degree of freedom

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2)^{-1}}{\sum_{j=1}^{J}(1 + ||z_i - \mu_j||^2)}. \tag{2}$$

As shown in [21], the Student's t-distribution is an infinite mixture of Gaussians, and the density of a point under this distribution is much faster to evaluate when compared to the Gaussian distribution, as it does not involve an exponential. In addition, the representation of joint probabilities is invariant to changes in the scale for the points in the feature space which are far apart. Therefore, large clusters of points that are far away in the space interact in the same way as individual points, and thus, the optimization progresses in the same way for all but the finest scales [21]. Finally, the clustering loss is

$$L_c = \mathrm{KL}(\mathcal{T}||q) = \sum_{i=1}^{p'}\sum_{j=1}^{J} t_{ij} \log \frac{t_{ij}}{q_{ij}} \tag{3}$$

where p' is the number of *pixels* in the batches, KL is the Kullback–Leibler divergence, and $\mathcal{T}$ is the target distribution

$$t_{ij} = \frac{q_{ij}^2 / \sum_{i=1}^{p'} q_{ij}}{\sum_{j=1}^{J}\left(q_{ij}^2 / \sum_{i=1}^{p'} q_{ij}\right)}. \tag{4}$$

The clustering loss is incorporated into the total loss function L used in this training stage, and it becomes

$$L = L_r + \alpha L_c \tag{5}$$

[2]Guo *et al.* [20] used much deeper architectures without dropout over full input images (with 2-D kernels only) in the context of image classification.

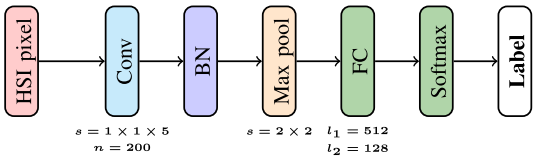

Fig. 2. 1D-CNN with n kernels in the convolutional layer (s stride) and l_1 and l_2 neurons in the FC layers. BN is batch normalization.

where $0 < \alpha < 1$, and it is a loss weighting coefficient (we used $\alpha = 0.1$). This stage continues until the convergence or the termination condition is met—we restrict it to 25 epochs.

III. Experiments

The objectives of our experiments are multifold. We verify the abilities of our unsupervised classification technique and compare it with other clustering methods: k-means, where k equals the number of target classes in the benchmarks, and Gaussian mixture (GM) modelling, being a generalization of k-means which incorporates information about the covariance structure of the data and the centers of latent Gaussians. These methods are applied over the original (full) and reduced HSI. In the latter case, we reduce the dimensionality of the input HSI to match the size of our latent vector using: 1) principal component analysis (PCA); 2) independent component analysis (ICA); 3) our sliding-window algorithm for simulating wider bands from hyperspectral images, in which we generate the averaged band within a nonoverlapping sliding window simulated multispectral image (S-MSI) [22]; and 4) our CAE. The size of the latent vector l is a hyperparameter of 3D-CAE (Fig. 1). To compare 3D-CAE with other dimensionality reduction techniques, we set $l = 25$ and kept it constant across the experiments—as shown in our recent work, aggressive HSI dimensionality reduction may lead to lower quality classification [22]. In addition, we apply our CAE over reduced HSI and check the impact of the corresponding dimensionality reduction on its abilities—in this case, CAEs *do not* perform dimensionality reduction, and the latent vector is of the same size as the input vector. Finally, in our sensitivity analysis, we verify how different latent-vector sizes ($l \in \{5, 10, \ldots, 50\}$) influence the segmentation quality of 3D-CAE. In addition, we compare unsupervised segmentation with our 1D-CNN [11] (Fig. 2) trained in a supervised manner over original and reduced benchmarks. Our study was divided into two experiments, over the available benchmarks (Section III-A), and a real-life HSI for which the ground-truth segmentation does not exist (Section III-B).

We exploited three most popular HSI benchmarks from the literature (http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes), alongside a new hyperspectral scene released at the 2018 IEEE GRSS Data Fusion Challenge [12]: 1) Salinas Valley (Sa), USA (217×512 pixels, AVIRIS sensor) showing different sorts of vegetation (16 classes, 224 bands, and 3.7-m spatial resolution); 2) Indian Pines (IP), USA (145×145, AVIRIS)—agriculture and forest (16 classes, 200 bands, and 20 m), 3) Pavia University (PU), Italy (340×610, ROSIS)—urban scenery (9 classes, 103 bands, and 1.3 m); 3) University of Houston (Houston), USA (4172×1202, ITRES CASI 1500)—20 urban land-cover/land-use classes, with 28.5% labeled pixels (20 classes, 48 bands, and 1 m). We also utilize the aerial hyperspectral observations acquired using the HyMap airborne sensor (7982×512, HyVista Corporation Pty, Ltd., Baulkham Hills, NSW, Australia, 126 bands with a wavelength resolution of 20 nm, and 4.2 m) on October 29, 2009. The study area was located in Mullewa, WA, Australia (480 km^2), and it is mainly used for the wheat, canola, and lupin production. Although there were 30 test fields in which *in situ* measurements had been performed, such data are not suitable for verifying segmentation algorithms, because we know the class label of an extremely small subset of all pixels. The ground-truth measurements were captured 1-m above a head of the wheat with eight measurements at each point—the measurement points are rendered in violet in Fig. 1 in the Supplementary Material, also available at https://gitlab.com/jnalepa/3d-cae. Hence, for Mullewa, we focus on qualitative analysis.

We use two clustering-quality measures to quantify the performance of the unsupervised techniques: normalized mutual information (NMI) and adjusted rand score (ARS). NMI is

$$\mathrm{NMI} = \frac{\mathrm{MI}(A, B)}{[H(A) + H(B)]/2} \tag{6}$$

where $\mathrm{MI}(A, B) = H(A) - H(A|B)$ is the mutual information index quantifying the value of information shared between two random variables A and B, $H(\cdot)$ denotes entropy, and $H(A|B)$ is the conditional entropy between clusterings. The entropy $H(A)$ of clusters $a \in A$ in an example clustering A is

$$H(A) = -\sum_{a \in A} p(a) \log p(a) \tag{7}$$

where $p(a)$ is the value of the probability distribution at a. The conditional entropy of A given B becomes

$$H(A|B) = -\sum_{a \in A} \sum_{b \in B} p(a, b) \log p(a|b) \tag{8}$$

where $p(a, b)$ and $p(a|b)$ are their joint and conditional probability distributions, respectively. The ARS metric is

$$\mathrm{ARS} = \frac{\binom{n}{2}(a+d) - [(a+b)\cdot(a+c) + (c+d)\cdot(b+d)]}{\binom{n}{2}^2 - [(a+b)\cdot(a+c) + (c+d)\cdot(b+d)]} \tag{9}$$

where n is the number of objects (pixels) subjected to clustering, and a, b, c, and d denote the number of data points placed: in the same group (cluster) in A and B (a), the same groups in A and in different groups in B (b), the same groups in B and in different groups in A (c), and in different groups in A and in different groups in B (d) [23]. Both NMI and ARS range from 0 to 1, where 1 means perfect score. In addition, we trained 1D-CNN in a supervised manner using our balanced division into the training and validation sets, as presented in [11] (for Houston, we utilize the labeled subset of pixels in this experiment). In the supervised scenario, we report the average accuracy (AA), overall accuracy (OA), and the kappa scores $\kappa = 1 - (1 - p_o)/(1 - p_e)$, where p_o and p_e are the relative observed agreement and hypothetical probability of chance agreement, respectively, and $-1 \leq \kappa \leq 1$ ($\kappa = 1$ is the perfect score). These scores were obtained over all *labeled* input pixels to make them comparable with the unsupervised segmentation performed over the entire HSI, and were averaged across 30 runs. In both cases, however, we exclude the background pixels for which the class label is unknown. Since the test set for 1D-CNN includes the training and validation examples,

TABLE I

RESULTS OBTAINED OVER ALL BENCHMARKS

Set→	Sa		IP		PU		Houston	
Algorithm↓	NMI	ARS	NMI	ARS	NMI	ARS	NMI	ARS
1D-CNN*	0.885	0.725	0.705	0.586	0.786	0.771	0.586	0.569
1D-CNN* (PCA)	0.860	0.685	0.640	0.493	0.360	0.215	0.470	0.354
1D-CNN* (ICA)	0.873	0.723	0.641	0.502	0.616	0.556	0.484	0.440
1D-CNN* (S-MSI)	0.880	0.723	0.718	0.609	0.784	0.759	0.564	0.546
1D-CNN* (CAE)	0.886	0.738	0.663	0.496	0.748	0.658	0.567	0.551
GM	0.819	0.642	0.445	0.229	0.514	0.290	0.343	0.088
GM (PCA)	0.830	0.654	0.443	0.235	0.530	0.404	**0.349**	**0.099**
GM (ICA)	0.838	0.665	0.436	0.212	0.522	0.396	0.054	0.000
GM (S-MSI)	**0.848**	**0.673**	0.456	0.248	0.532	0.407	0.336	0.081
GM (CAE)	0.628	0.475	0.435	0.289	0.480	0.459	0.332	0.082
k-means	0.732	0.538	0.437	0.211	0.546	0.350	0.252	0.043
k-means (PCA)	0.724	0.524	0.430	0.204	0.545	0.324	0.251	0.043
k-means (ICA)	0.730	0.535	0.381	0.178	0.477	0.263	0.247	0.035
k-means (S-MSI)	0.712	0.496	0.430	0.208	0.546	0.325	0.256	0.043
k-means (CAE)	0.710	0.503	0.451	**0.297**	0.539	0.336	0.246	0.031
3D-CAE	0.714	0.533	0.431	0.231	0.553	0.339	0.277	0.047
3D-CAE (PCA)	0.746	0.527	0.467	0.263	**0.639**	**0.546**	0.260	0.038
3D-CAE (ICA)	0.839	0.644	**0.504**	0.278	0.538	0.316	0.293	0.064
3D-CAE (S-MSI)	0.728	0.531	0.442	0.241	0.601	0.450	0.268	0.045

Supervised segmentation measures

Set→	Sa			IP			PU			Houston		
Algorithm↓	AA	OA	κ	AA	OA	κ	AA	OA	κ	AA	OA	κ
1D-CNN	0.946	0.887	0.875	0.828	0.777	0.749	0.894	0.872	0.835	0.859	0.728	0.667
1D-CNN (PCA)	0.873	0.820	0.802	0.766	0.691	0.655	0.451	0.398	0.326	0.728	0.555	0.483
1D-CNN (ICA)	0.953	0.904	0.893	0.803	0.736	0.702	0.771	0.713	0.645	0.759	0.620	0.545
1D-CNN (S-MSI)	0.943	0.887	0.874	0.832	0.790	0.762	0.875	0.839	0.796	0.841	0.705	0.640
1D-CNN (CAE)	0.946	0.895	0.875	0.812	0.735	0.650	0.876	0.822	0.764	0.849	0.714	0.599

How to read this table: The *globally* best *unsupervised* method is boldfaced.
The background of the *globally* worst *unsupervised* method is red.
For each method, we annotate its best and worst variant (green and gray background).
*For the sake of completeness, we report the *unsupervised* measures obtained using 1D-CNN trained in a *supervised* setting.

TABLE II

RANKING (AVERAGED ACROSS ALL BENCHMARKS) AND THE EXECUTION TIME OF ALL METHODS

	Ranking		Time (min)				
Algorithm↓	NMI	ARS	Sa	IP	PU	Houston	Mu
GM	6.00	7.25	11.83	1.63	2.89	132.71	91.49
GM (PCA)	5.25	4.00	1.35	0.09	1.65	124.74	32.97
GM (ICA)	9.25	8.00	0.87	0.29	0.50	15.08	5.40
GM (S-MSI)	4.00	3.50	1.03	0.15	1.41	118.00	21.33
GM (CAE)	10.25	5.25	0.98	0.18	0.92	122.55	18.42
k-means	7.38	8.25	1.12	0.18	0.79	65.78	52.01
k-means (PCA)	9.88	11.00	0.28	0.07	0.37	45.04	16.96
k-means (ICA)	12.00	11.75	0.35	0.04	0.99	27.94	13.84
k-means (S-MSI)	9.50	11.00	0.31	0.06	0.34	60.80	19.75
k-means (CAE)	9.25	8.75	0.26	0.08	0.44	47.44	19.82
3D-CAE	7.75	7.50	91.31	14.75	102.12	1153.88	1994.20
3D-CAE (PCA)	4.25	6.50	20.15	7.18	36.28	951.83	640.45
3D-CAE (ICA)	4.00	6.00	16.88	5.37	27.99	1023.46	573.52
3D-CAE (S-MSI)	6.25	6.25	20.33	5.36	50.60	822.24	553.55

TABLE III

OUR SENSITIVITY ANALYSIS SHOWED THAT 3D-CAE OBTAINS HIGH-QUALITY SEGMENTATION OF ALL HYPERSPECTRAL SCENES FOR VARIOUS SIZES OF THE LATENT VECTORS, REPORTED IN PARENTHESES. THE BEST RESULTS IN EACH COLUMN ARE BOLDFACED

Set/Ranking→	Sa		IP		PU		Houston		Ranking	
Algorithm↓	NMI	ARS	NMI	ARS	NMI	ARS	NMI	ARS	NMI	ARS
3D-CAE (5)	0.698	0.505	0.409	**0.252**	0.555	0.365	0.269	0.056	7.13	6.38
3D-CAE (10)	0.718	0.525	0.400	0.246	0.569	0.399	0.270	0.056	5.25	5.38
3D-CAE (15)	0.716	0.519	0.407	0.206	0.589	0.427	0.264	0.057	5.63	6.63
3D-CAE (20)	**0.721**	0.530	0.430	0.238	0.602	**0.492**	0.270	**0.060**	**2.50**	**2.63**
3D-CAE (25)	0.714	0.533	**0.431**	0.231	0.553	0.339	**0.277**	0.047	3.75	6.88
3D-CAE (30)	0.702	0.510	0.388	0.216	**0.622**	0.476	0.272	**0.060**	5.00	4.50
3D-CAE (35)	0.704	0.510	0.398	0.213	0.585	0.468	0.264	0.058	7.13	6.00
3D-CAE (40)	0.720	**0.540**	0.381	0.193	0.530	0.309	0.270	0.059	6.50	6.00
3D-CAE (45)	0.712	0.533	0.414	0.243	0.577	0.472	0.271	0.058	4.50	3.25
3D-CAE (50)	0.679	0.502	0.419	0.233	0.525	0.343	0.269	0.057	7.63	7.38

the results can be considered overoptimistic [11]. The deep networks were coded in `Python 3.6`, and the supervised training of 1D-CNN (ADAM, learning rate of 10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$) terminated, if after 25 epochs OA over the validation set (random subset of the training set) does not change. The experiments ran on NVIDIA GeForce RTX 2080.

A. Experiment 1: Benchmark Data

We compare 3D-CAE with other techniques over four HSI data sets—each unsupervised approach was executed exactly *once*, in order to understand its real-life applicability, where running algorithms multiple times is often infeasible. In addition, we performed Monte-Carlo cross-validation (repeated 30×) with balanced training and validation sets [11] and analyzed the average supervised measures (AA, OA, and κ) obtained using 1D-CNN. For the sake of completeness, we report the unsupervised segmentation measures (NMI and ARS) for 1D-CNN—all pixels for which the class labels are available were classified. Since the test set includes both training and validation sets, there is a "training-test information leak," therefore, NMI and ARS may be overoptimistic for 1D-CNN.

In Table I, we gather the experimental results obtained overall sets. They show that 3D-CAE consistently delivers high-quality segmentation in all settings, with and without HSI reduction—in all cases, we decrease the feature dimensionality to 25 to match the number of 3D-CAE embedded features. The dimensionality reduction is beneficial in the unsupervised setting and leads to better clustering. It indicates that only a small portion of the entire spectrum conveys useful information about the captured materials within those HSI—exploiting the full spectrum makes segmentation much harder due to the curse of dimensionality (the best results were obtained using our S-MSI; Wilcoxon test, $p < 0.001$). These observations are confirmed in Table II, where we report the ranking of all methods averaged across the investigated benchmarks.

To verify if the segmentation quality obtained using 3D-CAE can be improved with a varying size of the latent vector l, we performed the sensitivity analysis for various l's (Table III). The results show that, although 3D-CAE is stable and robustly delivers consistent segmentation across all investigated l values, its quality may be improved if the number of deep features is appropriately selected for an incoming hyperspectral scene. However, the best results were obtained for $20 \leq l \leq 30$ for practically all cases (see NMI and ARS in Table III).

The execution time of all unsupervised techniques is reported in Table II. These times reflect *only* segmentation, without feature extraction for the methods, run over reduced HSI—PCA took 1.17, 0.35, 1.95, and 49.18 s for Sa, IP, PU, and Houston, respectively, ICA: 25.04 s, 1.28 s, 56.68 s, 6.36 min, S-MSI: 0.28, 0.04, 0.47, 17.43 s, and 3D-CAE: 47.39, 4.56, 49.60, and 279.12 min. Although 3D-CAE was significantly slower than other algorithms,[3] it retrieved consistently better segmentation (Table I). In addition, we did not exploit early stopping for the clustering phase of 3D-CAE, as it ran always for 25 epochs. This part of the training could have been terminated earlier, and it would greatly reduce its execution time. It, however, requires further investigation.

[3]The execution time of 3D-CAE over reduced HSI was consistent with other deep learning-powered techniques [18]. In addition, changing the size of the latent vector did not significantly affect the 3D-CAE training time.

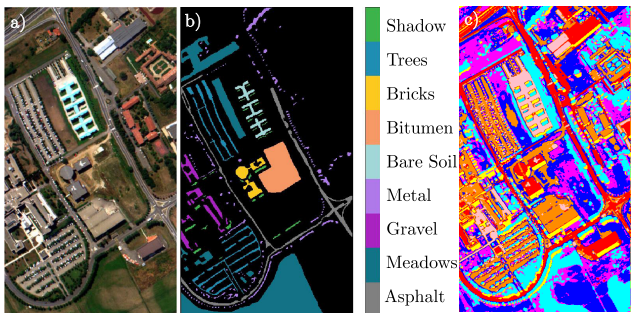

Fig. 3. Unsupervised segmentation offers new possibilities of unrevealing information captured within newly acquired HSI and existent benchmarks. This example shows: 1) the PU false-color scene, 2) its ground truth (black color is "unknown class"), and 2) our full 3D-CAE segmentation which is not only very detailed but also sheds new light on those "unknown" objects.

B. Experiment 2: Real-Life Data

In this experiment, we ran all unsupervised methods over a real-life hyperspectral scene. Since there is no ground-truth segmentation of the Mullewa data set, we qualitatively compare the selected methods in Fig. 2 in the Supplementary Material. We present the segmentations obtained using all unsupervised techniques over: 1) full HSI and 2) reduced HSI. This reduction was performed with the approach, which was the best overall benchmarks for the corresponding segmentation algorithm. The k-means and 3D-CAE techniques give much more detailed segmentation—see example regions annotated with the white and black arrows in the GM visualization. It indicates that those regions are "heterogeneous" and manifest subtle spectral variations. This observation can trigger more detailed *in situ* measurements, and hence, allow us to better understand the scanned regions and their critical characteristics. As previously, the execution time of 3D-CAE was much longer than other methods (Table II)—this issue can be tackled by more aggressive preprocessing (e.g., band selection), parallel GPU training or by applying early stopping conditions.

IV. Conclusion

We proposed a new deep-learning-powered unsupervised HSI segmentation algorithm which exploits 3D-CAEs to learn embedded features, and a clustering layer to segment an input image using the learned representation. Our experimental study, performed over the benchmark and real-life HSI revealed that our approach delivers consistent and high-quality segmentation without any prior class labels. Such unsupervised techniques offer new possibilities to understand the acquired HSI. They can be used to: 1) enable practitioners to generate ground-truth HSI data in affordable time even for very large scenes—unsupervised segmentation of an input HSI would be reviewed and fine-tuned if necessary, 2) perform anomaly detection within a captured region by analyzing unexpected heterogeneous parts of the segmentation map, e.g., a wheat farmland should be moderately homogeneous, and any deviation may be alarming; and to3) see beyond the current ground-truth HSI (Fig. 3). Although our method is computationally expensive, its execution time can be greatly decreased by the initial HSI reduction, applying early stopping conditions in both training phases, performing the parallel training (using multiple GPUs) and optimizing the hyperparameters of the deep network architecture, e.g., decreasing the number of kernels. It constitutes our current work.

References

[1] M. J. Khan *et al.*, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.
[2] G. Bilgin, S. Erturk, and T. Yildirim, "Segmentation of hyperspectral images via subtractive clustering and cluster validation using one-class SVMs," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 2936–2944, Aug. 2011.
[3] T. Dundar and T. Ince, "Sparse representation-based hyperspectral image classification using multiscale superpixels and guided filter," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 246–250, Feb. 2019.
[4] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.
[5] W. Zhao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.
[6] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.
[7] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
[8] A. Santara *et al.*, "BASS net: Band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5293–5301, Sep. 2017.
[9] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.
[10] Q. Gao, S. Lim, and X. Jia, "Hyperspectral image classification using convolutional neural networks and multiple feature learning," *Rem. Sens.*, vol. 10, no. 2, p. 299, 2018.
[11] J. Nalepa, M. Myller, and M. Kawulok, "Validating hyperspectral image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1264–1268, Aug. 2019.
[12] Y. Xu *et al.*, "'Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 IEEE GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1709–1724, Jun. 2019.
[13] J. Nalepa, M. Myller, and M. Kawulok, "Training- and test-time data augmentation for hyperspectral image segmentation," *IEEE Geosci. Remote Sens. Lett.*, to be published.
[14] J. Nalepa, M. Myller, and M. Kawulok, "Transfer learning for segmenting dimensionally reduced hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8864017
[15] S. Lee and C. Lee, "Unsupervised segmentation for hyperspectral images using mean shift segmentation," in *Satellite Data Compression, Communications, and Processing VI*, vol. 7810, B. Huang, A. J. Plaza, J. Serra-Sagristà, C. Lee, Y. Li, and S.-E. Qian, Eds. Bellingham, WA, USA: SPIE, 2010, pp. 271–276, doi: 10.1117/12.862176.
[16] A. Schclar and A. Averbuch, "A diffusion approach to unsupervised segmentation of hyper-spectral images," in *Proc. 9th Int. Joint Conf. Comput. Intell. (IJCCI)*, C. Sabourin, J. J. Merelo, K. Madani, and K. Warwick, Eds. Cham, Switzerland: Springer, 2019, pp. 163–178, doi: 10.1007/978-3-030-16469-0_9.
[17] A. Erturk and S. Erturk, "Unsupervised segmentation of hyperspectral images using modified phase correlation," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 4, pp. 527–531, Oct. 2006.
[18] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral–spatial feature learning via deep residual conv–deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, Jan. 2018.
[19] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, Dec. 2015.
[20] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, and E. S. El-Alfy, Eds. Cham, Switzerland: Springer, 2017, pp. 373–382.
[21] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[22] M. Marcinkiewicz, M. Kawulok, and J. Nalepa, "Segmentation of multispectral data simulated from hyperspectral imagery," in *Proc. IEEE IGARSS*, Jul. 2019, pp. 3336–3339.
[23] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," in *Artificial Neural Networks – ICANN*, C. Alippi, M. Polycarpou, C. Panayiotou, and G. Ellinas, Eds. Berlin, Germany: Springer, 2009, pp. 175–184.