Microprocessors and Microsystems 73 (2020) 102994

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Towards resource-frugal deep convolutional neural networks for hyperspectral image segmentation

Jakub Nalepa[a,b,*], Marek Antoniak[a], Michal Myller[a], Pablo Ribalta Lorenzo[b], Michal Marcinkiewicz[c]

[a] KP Labs Konarskiego 18C, Gliwice 44-100, Poland
[b] Silesian University of Technology, Akademicka 16, Gliwice 44-100, Poland
[c] Netguru Wojskowa 6, Poznan 60-792, Poland

ARTICLE INFO

Article history:
Received 31 May 2019
Revised 18 December 2019
Accepted 3 January 2020
Available online 7 January 2020

Keywords:
Hyperspectral imaging
Deep neural network
Convolutional neural network
Quantization
Segmentation
Classification

ABSTRACT

Hyperspectral image analysis has been gaining research attention thanks to the current advances in sensor design which have made acquiring such imagery much more affordable. Although there exist various approaches for segmenting hyperspectral images, deep learning has become the mainstream. However, such large-capacity learners are characterized by significant memory footprints. This is a serious obstacle in employing deep neural networks on board a satellite for Earth observation. In this paper, we introduce resource-frugal quantized convolutional neural networks, and greatly reduce their size without adversely affecting the classification capability. Our experiments performed over two hyperspectral benchmarks showed that the quantization process can be seamlessly applied during the training, and it leads to much smaller and still well-generalizing deep models.

1. Introduction

Hyperspectral imaging (HSI) is being continuously applied in a variety of fields, including biochemisty, biology, mineralogy, and remote sensing [90]. It captures a wide spectrum of light, and forms an array of usually more than a hundred of reflectance values acquired for every pixel in the image. Such amount of information can be effectively used to classify each pixel to a specific class, and to find the boundaries of objects within a scene imaged using a hyperspectral sensor in the process of HSI segmentation[1] [62]. The remote sensing community currently struggles with applying hyperspectral segmentation engines in constrained hardware settings, in the context of on-board Earth observation. It is in contrast to the post processing of such imagery which is performed back on Earth, after transferring images from a satellite equipped with a hyperspectral camera. This data transfer is extremely costly and time-consuming, and it is not feasible in the majority of Earth observation use cases where short re-visit times that can be seen as the temporal resolution of hyperspectral data, and rapid response to the events captured within a scene are critical practical issues. Disaster prevention, monitoring, and post-crisis operation alongside precision agriculture are the most notable examples of such applications [80].

Deep learning has achieved unprecedented success and established the state of the art in a plethora of pattern recognition tasks, including medical image segmentation and analysis [29,49], object detection [95], speech recognition [92], text processing [27,86], time-series analysis [21], autonomous driving [79], and many more [25,30,43]. HSI analysis is not an exception here, and there exist numerous deep learning-powered techniques towards accurate segmentation of such data [6,42]. However, the memory requirements of deep neural networks are commonly fairly large due to the number of their trainable parameters which can easily reach millions [74]. It makes them infeasible to be deployed in embedded hardware environments, as the available memory in such systems is very constrained. To deal with this issue, *quantization* of deep neural networks can be exploited—quantized representations store weights or activations using more compact formats such as integers or even binary numbers [28]. The main objective of

* Corresponding author at Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland.
E-mail addresses: jnalepa@ieee.org (J. Nalepa), mantoniak@kplabs.pl (M. Antoniak), mmyller@kplabs.pl (M. Myller), pribalta@ieee.org (P. Ribalta Lorenzo), michal.r.marcinkiewicz@gmail.com (M. Marcinkiewicz).

[1] By *classification* we mean assigning a label to a pixel, and by *segmentation*—finding the boundaries of objects belonging to different classes in HSI. Therefore, segmentation often involves classification of separate pixels within an input HSI.

https://doi.org/10.1016/j.micpro.2020.102994

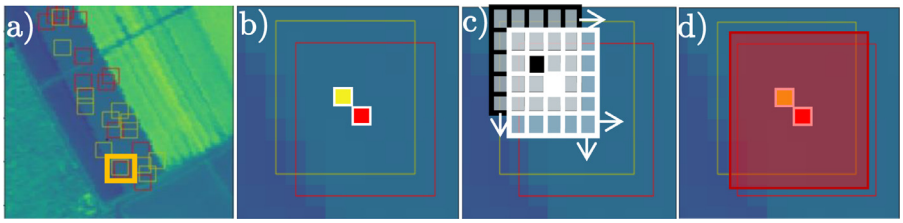

Fig. 1. Once (a) the training (yellow) and test (red) pixels, alongside their neighboring pixels (see a zoomed part of an HSI in (b) are drawn to the training and test sets in the Monte-Carlo cross-validation fashion, they are exploited during training (the black kernel in c) and final validation (the while kernel in c) of the spectral-spatial convolutional neural network, in which kernels move in both spectral and spatial dimensions, hence (d) the overlapping pixels (in red-shadowed area) are "leaked" across these sets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

quantization is to fit deep models into the extreme execution environments, and to ensure that the quantized models are memory- and energy-frugal. Additionally, quantization may help speed up the inference process which can be extremely useful in real-time applications, when the deep models are deployed on mobile or hardware-constrained devices [41].

In this work, we tackle the problem of large memory requirements of deep convolutional neural networks in the context of HSI classification and segmentation, and introduce resource-frugal convolutional neural networks for this task. We focus on *spectral* deep models which utilize exclusively the spectral information about the pixel that is being classified. Such models do not suffer from the training-test information leak problem, in which the training examples may be easily included in the test sets, if the experiments are performed in the Monte-Carlo cross-validation fashion over a single hyperspectral scene. It is in contrast to *spectral-spatial* models exploiting both spectral and spatial (neighborhood) information about the pixel (see an example rendered in Fig. 1). In our recent work [62], we showed that such spectral-spatial models should be validated differently, to avoid obtaining over-optimistic classification results. In practically all papers from the literature, the HSI classification and segmentation algorithms are validated over a number of benchmark hyperspectral images, but the training and test pixels are sampled from *the same* image. The training-test information leak would not occur if a supervised learner was trained and tested over *different* hyperspectral data, i.e., sampled from different images. This is, however, very difficult to achieve in practice because there are no manually-annotated hyperspectral images of the same scene, acquired e.g., in different time points using the same sensor. As different HSI benchmarks capture different classes of objects within a scene, a model trained over an image A cannot be straightforwardly applied to classify pixels from an image B (it would require e.g., applying transfer learning [61]).

In the work reported here, we decided to focus on quantizing spectral models to make the experimental results easier to understand and dependent only on the quantization process, not the validation approach—the differences between quantized and non-quantized models might be more difficult to compare due to non-trivial dataset splits. Noteworthy, the current efforts in the machine learning community are aimed at ensuring that data management does not cause any data leakage [2,37]. However, it is important to acknowledge that the quantization-aware training may be applied to spectral-spatial models too.

Overall, the contribution of our work is multifold:

- We exploit quantization-aware training to build a resource-frugal spectral convolutional neural network for segmenting hyperspectral images.
- We undertake an extensive experimental study over two popular hyperspectral benchmarks to understand the impact of the quantization process on the segmentation abilities of our spectral convolutional neural network. The analysis is backed up with various statistical tests to verify the importance of the obtained results.
- We show that the quantization-aware training process greatly reduces the memory requirements of our deep model without significantly affecting its segmentation performance.
- We investigate which classes are most often confused by our deep learners during the classification by investigating their average spectral curves.

This paper is organized as follows. In Section 2, we review the state of the art in HSI segmentation and quantization of deep neural networks. Section 3 presents our spectral convolutional neural network, and discusses the quantization-aware training that has been exploited for building a resource-frugal deep convolutional model. In Section 4, we discuss our experimental study performed over two popular hyperspectral benchmarks. Section 5 concludes the paper, and highlights the directions of our future work.

2. Related literature

In this section, we review the current advances in satellite hyperspectral image segmentation (Section 2.1). In Section 2.2, we briefly discuss the current approaches towards building resource-frugal deep neural nets through quantization,[2] and provide an overview of quantized deep networks applied to various real-life tasks in Section 2.3.

2.1. Segmentation of satellite hyperspectral images

HSI segmentation[3] techniques can be divided into two groups, including *conventional* machine learning algorithms, and *deep learning* (DL)-powered approaches. The former algorithms require manual feature engineering, being the process of extracting and selecting features from HSI to build classification and segmentation engines. Such features (i.e., quantifiable measures) can be extracted within (i) the spectral or (ii) spatial HSI dimension, or (iii) they can couple both spectral and spatial information for better generalization performance of trained models.

In *supervised* conventional machine learning techniques, the models are built in a training process in which the ground-truth HSI data—a manually-annotated HSI, where the pixels are assigned specific class labels—is used to train a model from the acquired *experience*. Such approaches include support vector machines [31], various boosting and sampling methods [23], evolution of cellular automata [66], hybrid techniques coupling Markov random fields

[2] For a more detailed discussion on the current state-of-the-art advances in deep neural network quantization, we refer to an excellent survey by Guo [28].

[3] As mentioned previously, *segmentation* of HSI is a process of determining the boundaries of objects within the image which belong to a given class, whereas *classification* is a procedure of assigning a class label to a separate pixel within the image.

and multinomial logistic regression [45], spectral-spatial classification based on affinity scoring [14], and many more [18,26,44,71,82]. On the other hand, in *unsupervised* techniques, the ground-truth data is not exploited while training a classifier, and an input HSI is grouped into coherent regions of pixels sharing similar feature characteristics in the elaborated feature space [3,35,78]. As pointed out by Yu et al. [89], there are two main problems affecting the classification performance of HSI segmentation techniques: (i) hand-crafted features extracted from HSI, for which the additional dimensionality reduction might have been applied, often cannot characterize the most important properties of the materials, and (ii) the amount of ground-truth data is very limited and may be difficult to interpret. The latter issue is also manifested in a fairly small number of hyperspectral benchmarks. In our recent paper [62], we investigated 17 state-of-the-art segmentation techniques—virtually all of them have been tested using up to three HSI benchmarks, with the Pavia University, Indian Pines, and Salinas Valley images being the most popular. These benchmarks were exploited in 15 (Pavia University), 8 (Indian Pines), and 5 (Salinas Valley) out of 17 approaches inspected in the aforementioned paper.

DL-powered approaches towards HSI segmentation allow us to conveniently extract *spectral* features (e.g., using deep belief or recurrent networks [50,56,96]) or both *spectral* and *spatial* features (e.g., using convolutional neural networks [CNNs] [10,22,68,70] or deep belief networks [12,13,46]), in an automatic fashion without any user intervention [94,36]. These features are intrinsically elaborated by the deep networks, and can capture data characteristics which are unknown for humans, hence could have been omitted in the feature engineering process [58]. Deep features have been extracted by a variety of deep-network architectures—they can be later utilized in other learners as well. In [89], Yu et al. exploited deep stacked autoencoders for feature extraction, and deployed them in the context of HSI analysis. Other approaches for automated deep feature extraction encompass multi-scale fusion methods [64], alongside multiple variants of CNNs [11,40]. Deep features may undergo further post-processing to determine their most discriminative subset, therefore to reduce the amount of data to be transferred from an airborne or spaceborne hyperspectral sensor [59].

Although deep learning has established the state of the art in HSI analysis [62], we need to combat the important challenges concerning the high HSI dimensionality in both conventional and DL-powered segmentation approaches [51], especially if we are to apply them over real-life HSI which must be efficiently acquired, transferred, and stored. Additionally, very large computational and memory requirements of deep neural networks are the crucial obstacles in deploying such large-capacity learners in on-board Earth observation applications, where the hardware constraints are extremely tight. In this work, we show how to effectively build resource-frugal deep models to decrease their memory footprints without affecting the segmentation performance.

2.2. Quantization of deep convolutional neural networks

Recent advances in deep learning led to designing more sophisticated models for solving increasingly challenging pattern-recognition problems. Although these models can generalize fairly well and provide high-quality performance, their memory requirements are a large obstacle which make them hard to be applied on board a satellite, in a constrained execution environment. To alleviate this problem, deep neural network (DNN) quantization is being actively developed [15,48,52]. There are two main streams of research in this area—the first group of approaches is focused on quantizing trained DNNs (of different types) [8,84,97], whereas the second encompasses methods for quantization-aware DNN training [32]. In this paper, we follow the latter research pathway for reducing the memory requirements of our CNN for HSI segmentation, and making it applicable in a very constrained small-satellite hardware.

2.2.1. Post-training quantization

In the *post-training* quantization, we quantize a trained floating-point deep model. This process may involve quantizing (i) weights, (ii) activations, or (iii) both of them. In general, quantization techniques are divided into *deterministic* and *stochastic* methods—these algorithms can be easily applied to quantize all of the aforementioned aspects of CNNs. The deterministic techniques include simple rounding of real-valued numbers, vector quantization (grouping the weights into clusters, and exploiting centroids which replace the actual weights during inference [24]), and casting the quantization process into various optimization problems [98]. On the other hand, stochastic quantization techniques often employ random rounding, in which the number of mappings of the real values onto a quantized space is larger than one because this mapping is randomized [16], and probabilistic quantization, where the weights are assumed to be discretely distributed [28]. Then, the actual quantized values can be inferred from such distributions. Overall, the post-training quantization leads to reducing the memory footprint of previously trained (in full precision) deep models. Additionally, the inference can be significantly accelerated since the bitwise operations may conveniently replace the dot products in the case of quantized CNNs, as presented in [5,17,81,93]. Therefore, they become memory and computationally-frugal.

2.2.2. Quantization-aware training

In the *quantization-aware training*, the CNN training process is modified in such a way that the resulting CNN is quantized—this is in contrast to the previously-discussed post-training methods which utilize the full-precision training. In the process of quantizing the weights and activations, we produce the quantized, e.g., to 8-bit precision, values of weights and activations, respectively, corresponding to their full-precision counterparts. On the other hand, the gradient quantization emerged only recently [28], and it is aimed at enhancing the data-parallel training. In this stochastic gradient descent training scenario, massively-large CNNs can be trained using multiple processing nodes which compute sub-gradients. The sub-gradients are later broadcast to other nodes to find the weight updates. Since such data transfers may easily become infeasible due to significant data transfer costs, several approaches towards quantizing sub-gradients have been proposed, and the authors reported large speedups when compared to the full-precision distributed training [1].

In an excellent work by Wen et al. [83], the authors reduced the communication cost for synchronizing gradients by the usage of *ternary gradients*—the original gradients are aggressively quantized to ternary levels $\{-1, 0, 1\}$. The experiments showed that the proposed approach not only does not significantly deteriorate the performance of well-known deep architectures (the accuracy loss of GoogLeNet was less than 2% on average), but allows for obtaining up to 3.04 $\times$ speedup for AlexNet on 8 graphics processing units. In [16], Courbariaux et al. utilized the binary weights during the CNN training and achieved the state-of-the-art results in classification over MNIST and CIFAR-10. A similar research pathway has been followed in multiple other deep network architectures, including XNOR-Net [67], ABC-Net [77], and DoReFa-Net [98].

In this work, we utilize a multi-stage quantization-aware training [34], in which the deep model is trained in full-precision at first. Then, it undergoes *fake quantization* and it is trained again before it is quantized to its final low-bit version (Section 3). Deep networks with fake quantization nodes are an intermediate step between the full-precision and quantized models. Inserting such

additional nodes allows us to simulate the quantization of weights and activations in the network (they are not intended to reduce the memory footprint of the network). Finally, such models with fake quantization nodes are transformed into their quantized versions, whose size is smaller than the size of full-precision models, and those containing fake quantization nodes. The quantization-aware training procedures can ultimately lead to higher-quality models (compared to those elaborated with post-training quantization routines), because the weights can be optimized against simulated quantization errors that are injected into a model during the training [55].

2.3. Practical applications of quantized deep neural networks

We have witnessed a tremendous success of various kinds of deep neural networks in practically all fields of computer science and engineering, including computer vision [19], image processing and analysis [20,85], remote sensing [54], prognostics and health management [91], materials science [72], time series classification [33], speech analysis [38], natural language processing [65], and more [25,76]. However, applying such models on embedded, mobile and other hardware-constrained devices is still an open issue. Hence, developing resource-frugal and energy-efficient deep networks has become an important research topic in the machine learning field [5,17,81,93]. In [9], Chen et al. ported deep CNNs onto iOS mobile devices (iPhone 6s and iPad Pro) and showed how data reusability may alleviate the high bandwidth burden in the convolutional layers of such networks. The experiments, performed over high-capacity deep models trained using the ImageNet dataset [69], revealed that they were able to reduce the storage requirements by 34% for a 16-layer CNN without degrading its performance. The problem of image classification with the use of deep models deployed on mobile devices is the most common task in the context of quantizing CNNs, and it was also tackled by multiple research groups (from academia and industry, especially by Qualcomm AI Research [75]), including Wu et al. [84], Yin et al. [88], Li et al. [47], Yin et al. [87], and Louizos et al. [53]. Interestingly, Nagel et al. [57] showed that such quantized CNNs can be easily applied to semantic segmentation and object detection as well. Although there are other real-life applications of quantized CNNs, e.g., small-footprint keyword spotting [55], to the best of our knowledge deploying the deep models on board the imaging satellites has not been extensively investigated in the literature so far. In this paper, we tackle this issue in the context of the on-board hyperspectral image classification and segmentation.

3. Quantized convolutional neural networks for HSI segmentation

Although HSI has been already proven useful in accurate identification of a variety of materials [4], efficient analysis and segmentation of such imagery have become a big issue in practical applications, and it is currently being faced by the machine learning and remote sensing communities. In this section, we introduce our quantized CNN for HSI segmentation, and present the quantization-aware training which was used to train our model.

In this work, we utilize a spectral CNN which has been proposed in our recent work [62], and has been shown to obtain high-quality HSI classification in the Monte-Carlo setting—it exploits the spectral pixel information exclusively (Fig. 2). Since this CNN operates only in one dimension, we refer to it as 1D-CNN. In the feature-extraction part of the 1D-CNN architecture, we use one convolutional layer with $n = 200$ trainable kernels, each kernel is of $k = 5$ size, and it is applied with stride $s = 1$ in the spectral dimension. To reduce the dimensionality of extracted features, we use one max-pooling layer of size $k = 2$ with stride $s = 2$. The classification part of the network consists of two fully-connected layers with 512 and 128 neurons with the ReLU activation function given as $\mathrm{ReLU}(x) = \max(0, x)$, followed by the softmax layer which converts a real-valued score x (e.g., the network output) into a probability value p in the multi-class classification.

To decrease the memory footprint of 1D-CNN, we utilized the simulated quantization technique proposed in [34], and presented in Algorithm 1 (our implementation of the full quantization process is available at https://gitlab.com/jnalepa/hsi_quantization). First, the deep network is trained in full precision (line 2) until the training process reaches the maximal number of epochs (in this work, it was 300 epochs). Then, the quantization error is modeled using *fake quantization* nodes that are inserted into the model in places where tensors will be downcasted to fewer bits—at this point, we obtain an intermediate deep model with fake quantization nodes, Q-CNN' (line 3). Therefore, both forward- and backward passes simulate the quantization of weights and activations in the network, and the model is further trained (line 4) for a given number of additional epochs (here, 50). Importantly, the weights are updated using full precision during the backward pass to enable 1D-CNN handle such adjustments to auxiliary quantization nodes. Finally, the model with fake quantization nodes is transformed into its fully-quantized counterpart, referred to as Q-CNN which unfolds to quantized-CNN (line 5). For each floating-point value r which is being quantized (i.e., weights and activations), we have [34]:

Algorithm 1 Simulated quantization of a deep convolutional neural network $\mathcal{M}$ leads to obtaining a quantized convolutional neural network (Q-CNN).

1: **procedure** SIMULATED QUANTIZATION($\mathcal{M}$)
2: Train $\mathcal{M}$ in full precision
3: Simulate quantization error using *fake quantization* (Q-CNN')
4: Train $\mathcal{M}$ with simulated quantization error
5: Generate fully-quantized $\mathcal{M}$ (Q-CNN)
6: **end procedure**

$$q(r; a, b, n) = \left\lfloor \frac{\mathrm{clamp}(r; a, b) - a}{s(a, b, n)} \right\rceil \cdot s(a, b, n) + a, \tag{1}$$

where

$$\mathrm{clamp}(\mathrm{r}; \mathrm{a}, \mathrm{b}) = \min(\max(r, a), b), \tag{2}$$

and

$$s(a, b, n) = \frac{b - a}{n - 1}, \tag{3}$$

where $[a, b]$ is the quantization range, n denotes the number of quantization levels ($n = 2^8 = 256$ in the case of 8-bit quantization), and $\lfloor \cdot \rceil$ is the operation of rounding the input to the closest integer.

Let us assume that we want to quantize a floating-point value of $r = 14.76$, where $a = 0$, $b = 255$, and we use 8-bit quantization ($n = 256$). Then, we will obtain $\mathrm{clamp}(14.76; 0, 255) = \min(\max(14.76, 0), 255) = 14.76$, and s becomes $s(0, 255, 256) = \frac{255-0}{256-1} = 1$. Therefore, $q(14.76; 0, 255, 256) = \lfloor \frac{14.76-0}{1} \rceil \cdot 1 + 0 = 15$. On the other hand, if the quantization range was e.g., $[-244, 11]$, we would have $s(-244, 11, 256) = \frac{11+244}{255} = 1$, and $q(14.76; -244, 11, 256) = \lfloor \frac{11+244}{1} \rceil \cdot 1 + (-244) = 255 - 244 = 11$.

Typically, the quantization ranges are calculated differently for weights and activations. In the former case, the range spans across all weight values within the network, hence a and b are the minimal and maximal weight in the network, respectively. The range utilized for quantizing the activations is estimated during the training process using the exponential moving averages with

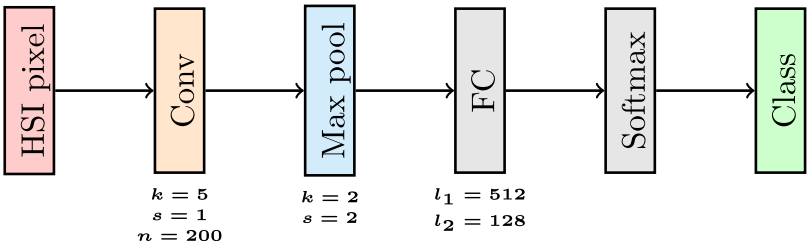

Fig. 2. Our spectral CNN (1D-CNN) with n kernels in the convolutional layer (applied with the s stride) and l_1 and l_2 neurons in the fully-connected (FC) layers.

the smoothing parameter commonly close to 1, so that the estimated range is smoothed across a large number of training steps, as it depends on the input data [34].

4. Experiments

In this section, we present the results of our experimental study which was aimed at verifying the impact of the quantization process on the segmentation abilities of our spectral CNN. We focused on two multi-class HSI benchmark sets (Salinas Valley and Pavia University) which are discussed in more detail in Section 4.2. The experiments have been backed up with various visualizations and statistical tests to understand the importance of the results and to check if quantization affects the model performance in a statistically-important way.

4.1. Experimental setup

Our 1D-CNN and the training-aware quantization were implemented in Python (TensorFlow). The network was trained using the ADAM optimizer [39], with the learning rate of 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, and the batch size was 64. The full-precision training finishes after 300 epochs, whereas the post-fake-quantization training was executed for 50 epochs (for details, see Section 3). Each dataset was randomly divided into the non-overlapping training ($\boldsymbol{T}$), validation ($\boldsymbol{V}$), and test (Ψ) sets. The validation set is used to calculate the loss during the training process, whereas the test set is never used while training a deep model, and it is utilized to quantify the generalization abilities of our CNNs. For both datasets, we followed the data splits reported in [22]—for Salinas Valley, we have 300 examples from each class in $\boldsymbol{T}$, and 30 examples from each class in $\boldsymbol{V}$ (hence, both sets are balanced), whereas for Pavia University, we sample 250 and 25 examples from each class to $\boldsymbol{T}$ and $\boldsymbol{V}$, respectively. We performed Monte-Carlo cross-validation, and executed 30 independent Monte-Carlo runs for each configuration (i.e., for each dataset, with and without quantization). Finally, we report per-class accuracies, alongside the average (AA) and overall (OA) accuracy, obtained for the test sets Ψ, and averaged across all executions.

4.2. Datasets

The Salinas Valley dataset has been acquired using the NASA Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, whereas Pavia University has been acquired using the Reflective Optics System Imaging Spectrometer (ROSIS) sensor. AVIRIS registers 224 contiguous bands with wavelengths in a 400 to 2450 nm range (visible to near-infrared), with 10 nm bandwidth, and it is calibrated to within 1 nm. The ROSIS sensor acquires the spectral radiance data in 115 bands in a 430 to 850 nm range (4 nm nominal bandwidth). Both sets are imbalanced (see the numbers of each-class examples in Tables 1–2), and contain under-represented classes, e.g., class 13 (C13) in the Salinas Valley scene, or class 9 (C9) in the case of Pavia University. These HSI benchmarks are discussed in more detail in the following subsections.

Table 1
The number of examples from each class in the Salinas Valley dataset.

Class	Description	No. of examples
1	Broccoli green weeds 1	2,009
2	Broccoli green weeds 2	3726
3	Fallow	1976
4	Fallow rough plow	1394
5	Fallow smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes untrained	11,271
9	Soil vineyard green weeds	6203
10	Corn senescent green weeds	3278
11	Lettuce romaine 4 week	1068
12	Lettuce romaine 5 week	1927
13	Lettuce romaine 6 week	916
14	Lettuce romaine 7 week	1070
15	Vineyard untrained	7268
16	Vineyard vertical trellis	1807
–	Total number of examples	54,129

Table 2
The number of examples from each class in the Pavia University dataset.

Class	Description	No. of examples
1	Asphalt	6631
2	Meadows	18,649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare soil	5029
7	Bitumen	1330
8	Self-blocking bricks	3682
9	Shadows	947
–	Total number of examples	42,776

4.2.1. Salinas valley

This HSI benchmark (of 217 × 512 pixels) was captured over Salinas Valley in California, USA, with a spatial resolution of 3.7 m. The image presents different sorts of vegetation, and contains 16 classes of objects (Fig. 3). The original data encompasses 224 bands, however 20 bands were removed by the authors of this set due to either atmospheric absorption or noise contamination (the final HSI contains 204 bands[4]). In Table 1, we gathered the number of each-class examples within the Salinas Valley set, and visualized their spectral profiles (averaged across all examples in each class) in Fig. 4. These visualizations show that there are parts of the spectrum which cannot be effectively used for classification (e.g., the band 150 and above), as the classes are indistinguishable in such

[4] For more details concerning HSI benchmarks, see http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes ; last access: May 29, 2019.

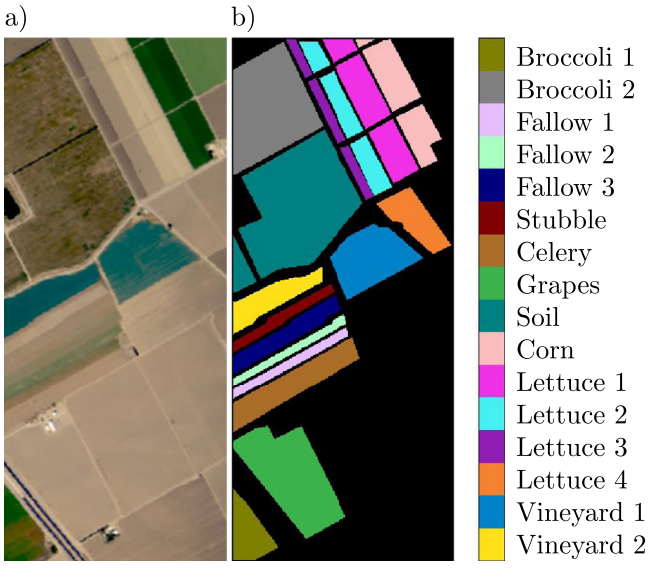

Fig. 3. The (a) false-color Salinas Valley scene with its (b) ground-truth segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

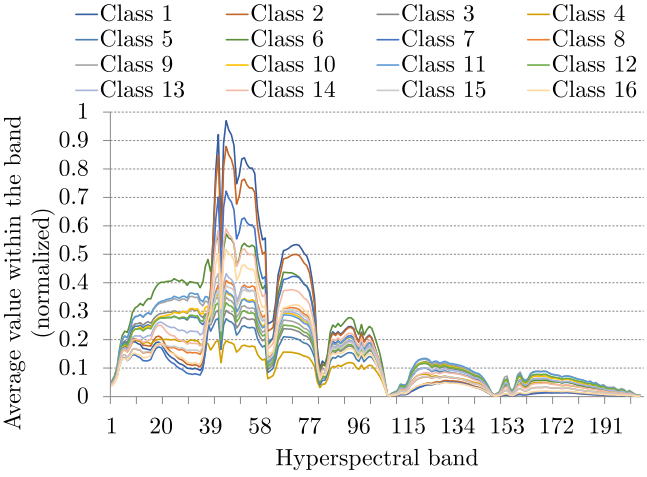

Fig. 4. The spectral profiles of all classes (averaged across all examples) within the Salinas Valley dataset.

wavelength ranges. Although we do not benefit from these observations in this work, as we exploit the entire spectral information, they can be useful to reduce the dimensionality of HSI by removing the parts of the spectrum which do not convey any useful information concerning the scanned objects. Such analysis should be performed for each use case separately—different bands will likely be important for discriminating different materials.

4.2.2. Pavia university

This HSI benchmark (340 × 610 pixels) was acquired over Pavia University in Lombardy, Italy, with a spatial resolution of 1.3 m. It presents an urban scenery with nine classes (Fig. 5). The set contains 103 bands, as 12 most noisy bands (out of 115) were removed by the authors of this benchmark. The number of examples in each class in this dataset is collected in Table 2, whereas the spectral profiles of all classes are presented in Fig. 6.

4.3. The results

In this section, we report the experimental results obtained for Salinas Valley and Pavia University, elaborated using our original

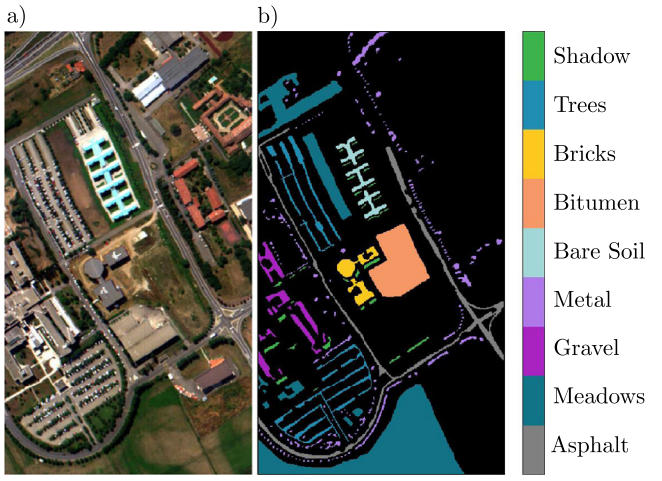

Fig. 5. The (a) false-color Pavia University scene with its (b) ground-truth segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

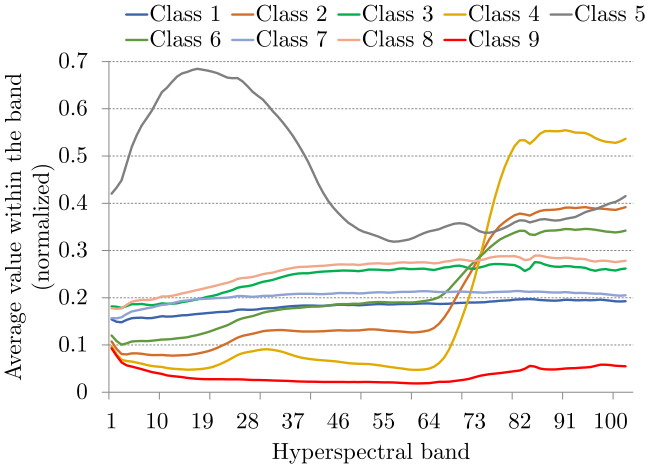

Fig. 6. The spectral profiles of all classes (averaged across all examples) within the Pavia University dataset.

1D-CNN alongside its quantized version (Q-CNN). Additionally, we present the results for the intermediate model which contains the fake quantization nodes (Q-CNN').

4.3.1. Classification performance of the deep models

In Table 3, we gather the per-class accuracies alongside AA and OA averaged across 30 runs obtained for Salinas Valley. We can appreciate the fact that the segmentation abilities are not significantly deteriorated by the quantization process for the majority of classes, and for some of them the accuracy increased for the quantized CNNs (C1, C5, C15, and C16). To better understand the classification process performed with all deep models, we present the confusion matrices obtained using 1D-CNN in full precision (in this work, using 32-bit floating point numbers) in Table 4, with fake quantization nodes in Table 5, and its fully-quantized version in Table 6. The results confirm that the C8 class is the most challenging for classification, and it is fairly often confused with C15. However, the classification becomes more accurate with quantization which may indicate that the full-precision models started overfitting to $\boldsymbol{T}$ due to their large capacity, and simulating quantization error before the final "fine-tuning" helped deliver better generalization. The average spectral profiles of C8 and C15 given in Fig. 7a show that the examples from these classes indeed follow a very similar (almost overlapping across the entire spectrum)

Table 3
The results (averaged across 30 independent executions) obtained for the Salinas Valley dataset using our original, intermediate, and quantized spectral CNNs.

Class →	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	AA	OA
1D-CNN	99.03	99.36	98.44	99.62	97.72	99.83	99.46	74.52	99.18	93.65	98.65	99.73	98.95	97.41	73.14	98.86	95.47	89.58
Q-CNN'	99.09	99.42	99.03	99.63	97.84	99.79	99.47	75.62	99.37	93.88	98.36	99.65	98.62	97.19	71.57	98.77	95.46	89.66
Q-CNN	99.67	96.49	81.08	99.24	99.01	99.81	99.40	63.26	97.62	90.67	95.87	99.62	98.66	96.95	78.54	98.91	93.42	86.69

Table 4
Confusion matrix presenting the average number of examples classified to a specific class using our non-quantized 1D-CNN in the Salinas Valley dataset. The darker the cell is, the larger number of examples belonging to a class A (corresponding to the row the cell belongs to) were classified as the class B (corresponding to the column the cell belongs to) examples. The examples correctly classified lay on the main diagonal.

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
C1	1,662.63	16.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20
C2	14.20	3,374.13	0.00	0.00	0.00	0.53	0.67	0.07	0.00	0.57	0.00	0.00	0.07	2.13	0.00	3.63
C3	0.00	0.00	1,620.37	0.00	19.60	0.00	0.00	0.00	0.40	3.00	2.63	0.00	0.00	0.00	0.00	0.00
C4	0.00	0.00	0.00	1,059.97	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03
C5	0.00	0.00	28.67	14.33	2,294.50	1.27	0.00	0.37	0.70	2.97	2.37	0.03	0.00	0.17	1.17	1.47
C6	0.00	2.03	0.00	0.00	2.03	3,622.67	0.03	0.03	0.13	0.77	0.30	0.00	0.03	0.67	0.27	0.03
C7	0.13	1.57	0.00	0.00	0.57	0.47	3,231.57	2.60	0.00	0.70	0.20	0.00	0.80	4.00	1.00	5.40
C8	0.30	0.00	0.00	0.00	0.10	1.17	7.50	8,153.47	0.57	297.13	6.93	0.00	0.10	35.80	2,426.17	11.77
C9	0.00	0.00	0.23	0.00	0.00	0.00	0.00	0.83	5,825.03	12.80	25.00	0.57	0.13	8.40	0.00	0.00
C10	0.00	0.20	41.40	0.53	2.37	1.20	0.00	16.77	29.33	2,760.83	50.27	11.57	0.00	9.20	14.07	10.27
C11	0.00	0.00	1.43	0.00	0.47	0.00	0.00	0.00	0.83	3.60	728.07	2.67	0.00	0.00	0.43	0.50
C12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.53	3.37	1,592.63	0.33	0.00	0.00	0.00
C13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	579.87	6.13	0.00	0.00
C14	0.00	0.00	0.00	0.00	0.00	0.07	0.13	3.93	0.27	4.40	0.03	0.10	10.20	720.83	0.00	0.03
C15	0.00	0.00	3.17	0.17	0.23	0.07	0.93	1,736.63	0.00	95.67	5.00	0.00	0.00	1.47	5,074.13	20.53
C16	0.33	2.53	0.00	0.00	0.30	0.00	2.30	0.53	0.00	6.50	0.57	0.00	0.00	0.10	3.60	1,460.23

Table 5
Confusion matrix presenting the average number of examples classified to a specific class using our 1D-CNN with fake quantization nodes (Q-CNN') in the Salinas Valley dataset. The darker the cell is, the larger number of examples belonging to a class A (corresponding to the row the cell belongs to) were classified as the class B (corresponding to the column the cell belongs to) examples. The examples correctly classified lay on the main diagonal.

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
C1	1,663.73	15.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.20
C2	13.57	3,376.17	0.00	0.00	0.00	0.37	1.10	0.00	0.00	0.77	0.00	0.00	0.00	1.37	0.00	2.67
C3	0.00	0.00	1,630.10	0.00	10.97	0.00	0.00	0.00	0.37	3.40	1.10	0.00	0.00	0.00	0.07	0.00
C4	0.00	0.00	0.00	1,060.03	3.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10
C5	0.00	0.00	27.77	13.07	2,297.37	1.20	0.00	0.23	0.70	2.70	2.37	0.00	0.03	0.13	1.23	1.20
C6	0.00	2.17	0.00	0.00	1.57	3,621.27	0.07	2.23	0.17	0.57	0.20	0.00	0.07	0.37	0.30	0.03
C7	0.00	1.63	0.03	0.00	0.70	0.77	3,231.80	2.30	0.00	0.73	0.13	0.00	0.53	3.97	0.83	5.57
C8	0.10	0.00	0.00	0.00	0.17	1.10	5.80	8,273.47	0.57	279.33	6.23	0.00	0.00	28.33	2,339.50	6.40
C9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.53	5,836.03	10.93	17.07	0.57	0.07	7.80	0.00	0.00
C10	0.00	0.13	34.37	0.63	2.67	1.20	0.07	21.13	29.00	2,767.60	47.83	12.23	0.00	9.13	13.33	8.67
C11	0.00	0.00	1.97	0.00	0.53	0.00	0.00	0.00	1.67	3.20	725.87	4.17	0.00	0.00	0.27	0.33
C12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.70	4.17	1,591.43	0.20	0.00	0.00	0.00
C13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	577.90	8.10	0.00	0.00
C14	0.00	0.00	0.00	0.00	0.00	0.10	0.23	4.30	0.33	4.57	0.10	0.03	11.10	719.20	0.03	0.00
C15	0.00	0.00	1.83	0.50	0.47	0.03	0.57	1,863.17	0.03	86.27	4.33	0.00	0.00	0.40	4,965.73	14.67
C16	0.37	2.03	0.00	0.10	0.43	0.00	2.67	0.97	0.00	6.87	0.67	0.00	0.00	0.33	3.73	1,458.83

spectral pattern, therefore can be easily misclassified given a non-representative and limited (in terms of size) training set $\boldsymbol{T}$.

We performed an analogous analysis over the Pavia University dataset, and report the results in Table 7. Similarly, there in no significant classification performance deterioration for quantized CNNs, and both AA and OA remained practically the same as those obtained using the full-precision deep model which indicates that simulating the quantization error by adding fake quantization nodes allows us to train well-generalizing low-precision deep models. The same observation is manifested in the confusion matrices rendered for all models (Tables 8–10). Although all examples (i.e., belonging to all Pavia University classes) were quite accurately classified to the correct class, the C3 and C8 classes were most often confused with each other. Their spectral profiles (Fig. 7b) are similar in shape across the entire spectrum—note that the scale of the Y axis in Fig. 7b is different from the scale of the Y axis in

Table 6
Confusion matrix presenting the average number of examples classified to a specific class using our fully-quantized 1D-CNN (Q-CNN) in the Salinas Valley dataset. The darker the cell is, the larger number of examples belonging to a class A (corresponding to the row the cell belongs to) were classified as the class B (corresponding to the column the cell belongs to) examples. The examples correctly classified lay on the main diagonal.

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
C1	1,673.53	5.10	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33
C2	112.70	3,276.80	0.00	0.00	0.00	0.30	0.80	0.00	0.00	0.63	0.00	0.00	0.00	0.73	0.00	4.03
C3	0.00	0.00	1,334.50	0.37	305.27	0.00	0.00	0.00	0.00	4.40	1.37	0.00	0.00	0.00	0.10	0.00
C4	0.00	0.00	0.07	1,055.97	7.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20
C5	0.00	0.00	4.13	9.87	2,324.70	1.40	0.00	0.17	0.47	2.33	2.00	0.00	0.07	0.07	1.10	1.70
C6	0.00	2.47	0.00	0.00	1.97	3,622.20	0.10	1.00	0.13	0.43	0.10	0.00	0.00	0.23	0.30	0.07
C7	0.10	1.93	0.03	0.00	0.70	0.90	3,229.37	1.97	0.00	1.03	0.07	0.00	0.30	2.60	0.90	9.10
C8	0.07	0.00	0.03	0.00	0.67	2.57	6.37	6,920.97	0.17	196.93	8.73	0.00	0.00	25.37	3,758.67	20.47
C9	0.00	0.00	0.93	0.00	0.00	0.00	0.00	0.53	5,732.97	81.13	44.93	0.70	0.20	11.60	0.00	0.00
C10	0.07	0.07	41.40	0.37	4.13	2.33	0.07	65.73	21.53	2,672.93	45.20	9.10	0.00	9.87	62.17	13.03
C11	0.00	0.00	4.23	0.00	0.83	0.00	0.00	0.00	0.77	14.23	707.50	9.77	0.00	0.00	0.43	0.23
C12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.27	5.13	0.03	1,590.87	0.57	0.13	0.00	0.00
C13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	578.13	7.87	0.00	0.00
C14	0.00	0.03	0.00	0.00	0.00	0.27	0.70	5.53	0.03	3.50	0.10	0.00	11.90	717.47	0.40	0.07
C15	0.00	0.00	2.30	0.50	3.00	0.10	0.83	1,411.53	0.00	46.40	5.83	0.00	0.00	0.33	5,449.17	18.00
C16	0.30	1.67	0.00	0.10	0.60	0.00	2.63	0.47	0.00	4.83	0.87	0.00	0.00	0.10	4.50	1,460.93

Table 7
The results (averaged across 30 independent executions) obtained for the Pavia University dataset using our original, intermediate, and quantized spectral CNNs.

Class →	C1	C2	C3	C4	C5	C6	C7	C8	C9	AA	OA
1D-CNN	89.65	91.47	87.17	96.52	99.66	94.83	94.87	87.44	99.87	93.50	91.83
Q-CNN'	89.70	92.09	87.59	96.31	99.68	94.28	94.74	86.78	99.86	93.45	92.00
Q-CNN	90.28	91.51	88.43	96.02	99.65	94.70	94.26	84.92	99.88	93.29	91.73

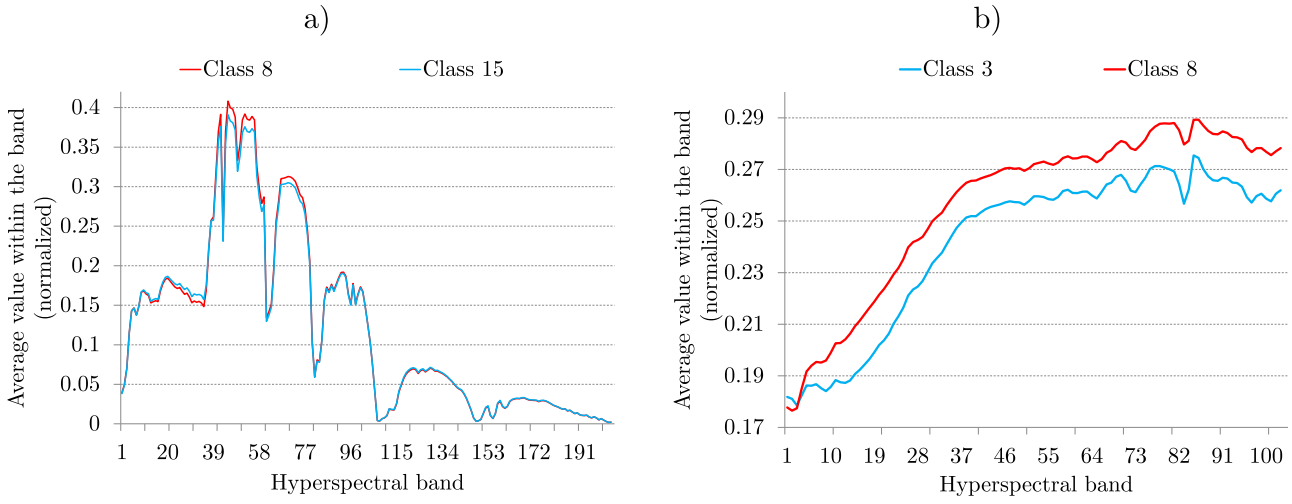

Fig. 7. The spectral profiles of (a) the C8 and C15 classes in Salinas Valley, and (b) the C3 and C8 classes in Pavia University.

Fig. 7a (see also Fig. 6, in which we can see that the spectral profiles of C3 and C8 are the most similar ones in this dataset). Additionally, the confusion matrix shows that C2 was often misclassified as C6 with Q-CNN (Table 6). Since C2 is the most numerous class in Pavia University (Table 2), the misclassification likely indicates the lack of representativeness of sampled training examples belonging to C2, and the inability of Q-CNN to capture subtle characteristics of the pixels belonging to this class. Although the segmentation results (AA and OA) reported for a spectral-spatial network in [22] exceeded 99% for Pavia University, and 98% for Salinas Valley, they may be quite over-optimistic due to the training-test information leak present in the Monte-Carlo cross-validation applied to spectral-spatial CNNs.[5]

In Fig. 8, we render example overall accuracies[6] obtained over the training and validation sets during both phases of training (the full-precision training on the left, and the second phase of train-

[5] For more details on the training-test information leak and its impact on the classification scores, see our recent paper [62].

[6] The overall accuracies obtained over the training and validation sets were very consistent across all executions for both datasets—for more examples, see detailed numerical results available at https://gitlab.com/jnalepa/hsi_quantization.

Table 8
Confusion matrix presenting the average number of examples classified to a specific class using our non-quantized 1D-CNN in the Pavia University dataset. The darker the cell is, the larger number of examples belonging to a class A (corresponding to the row the cell belongs to) were classified as the class B (corresponding to the column the cell belongs to) examples. The examples correctly classified lay on the main diagonal.

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	5,676.00	14.60	85.33	0.90	10.50	37.20	340.87	164.67	0.93
C2	3.00	16,784.27	4.13	497.10	0.30	1,039.23	0.30	20.63	0.03
C3	18.13	5.83	1,568.10	0.07	0.20	0.33	2.27	204.00	0.07
C4	0.17	78.70	0.17	2,667.80	0.67	16.23	0.00	0.00	0.27
C5	0.67	0.67	0.17	0.10	1,041.40	1.60	0.07	0.30	0.03
C6	8.53	200.60	2.13	11.73	6.70	4,484.73	0.10	14.47	0.00
C7	49.20	0.00	1.40	0.00	0.50	0.07	977.13	1.70	0.00
C8	66.67	10.00	320.80	0.10	0.37	18.23	8.63	2,957.20	0.00
C9	0.60	0.07	0.03	0.00	0.03	0.03	0.07	0.00	646.17

Table 9
Confusion matrix presenting the average number of examples classified to a specific class using our 1D-CNN with fake quantization nodes (Q-CNN') in the Pavia University dataset. The darker the cell is, the larger number of examples belonging to a class A (corresponding to the row the cell belongs to) were classified as the class B (corresponding to the column the cell belongs to) examples. The examples correctly classified lay on the main diagonal.

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	5,679.07	15.07	86.20	0.90	9.70	36.83	330.90	171.80	0.53
C2	2.47	16,897.10	4.03	452.57	0.10	972.63	0.37	19.73	0.00
C3	18.60	6.07	1,575.83	0.03	0.20	0.17	1.97	196.10	0.03
C4	0.23	84.63	0.20	2,661.97	0.77	15.93	0.00	0.00	0.27
C5	0.70	0.70	0.10	0.17	1,041.67	1.33	0.07	0.27	0.00
C6	7.30	227.70	1.87	12.77	6.27	4,458.37	0.13	14.60	0.00
C7	49.97	0.00	1.77	0.00	0.50	0.07	975.87	1.83	0.00
C8	64.37	10.07	346.03	0.13	0.20	18.03	8.23	2,934.93	0.00
C9	0.70	0.00	0.03	0.03	0.03	0.07	0.03	0.03	646.07

Table 10
Confusion matrix presenting the average number of examples classified to a specific class using our fully-quantized 1D-CNN (Q-CNN) in the Pavia University dataset. The darker the cell is, the larger number of examples belonging to a class A (corresponding to the row the cell belongs to) were classified as the class B (corresponding to the column the cell belongs to) examples. The examples correctly classified lay on the main diagonal.

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	5,715.40	14.83	81.60	0.77	9.40	36.77	315.30	156.30	0.63
C2	2.83	16,791.97	4.20	423.00	0.13	1,100.87	0.57	25.43	0.00
C3	21.53	6.13	1,590.80	0.03	0.20	0.17	2.07	178.03	0.03
C4	0.90	92.13	0.27	2,653.90	1.43	15.03	0.03	0.00	0.30
C5	0.87	0.70	0.20	0.20	1,041.33	1.33	0.07	0.30	0.00
C6	7.43	207.43	1.90	12.23	6.20	4,478.37	0.20	15.23	0.00
C7	55.50	0.00	1.60	0.00	0.47	0.07	970.90	1.47	0.00
C8	75.77	9.77	396.93	0.13	0.20	18.00	9.10	2,872.10	0.00
C9	0.57	0.00	0.03	0.03	0.03	0.03	0.03	0.03	646.23

ing a deep model with fake quantization nodes which simulate the quantization error on the right). The results show that the full-precision training phase allowed us to converge to high-quality deep models, and the process of fine-tuning intermediate models (with fake quantization nodes) does not deteriorate their classification abilities. It is also worth mentioning that the quantized models generalize well over the unseen test data (as reported in this section), hence the models did not overfit to the training examples during neither full-precision nor 8-bit training phases.

4.3.2. Analysis of the memory savings in the quantized deep models

To better understand what is the impact of reducing the size of deep models that are to be deployed on board a satellite, let us assume we have a satellite with a hyperspectral sensor which is able to capture 12-bit HSI with 200 bands. If we acquire a 2000×1000 image, that will give us $2000 \cdot 1000 \cdot 200 \cdot 12 = 4800$ megabits (Mb) of data for transmission. Assuming that we have a downlink with 50 Mbps nominal downlink speed and 0.5 downlink efficiency, it would require 3.2 min to send a single HSI scene back

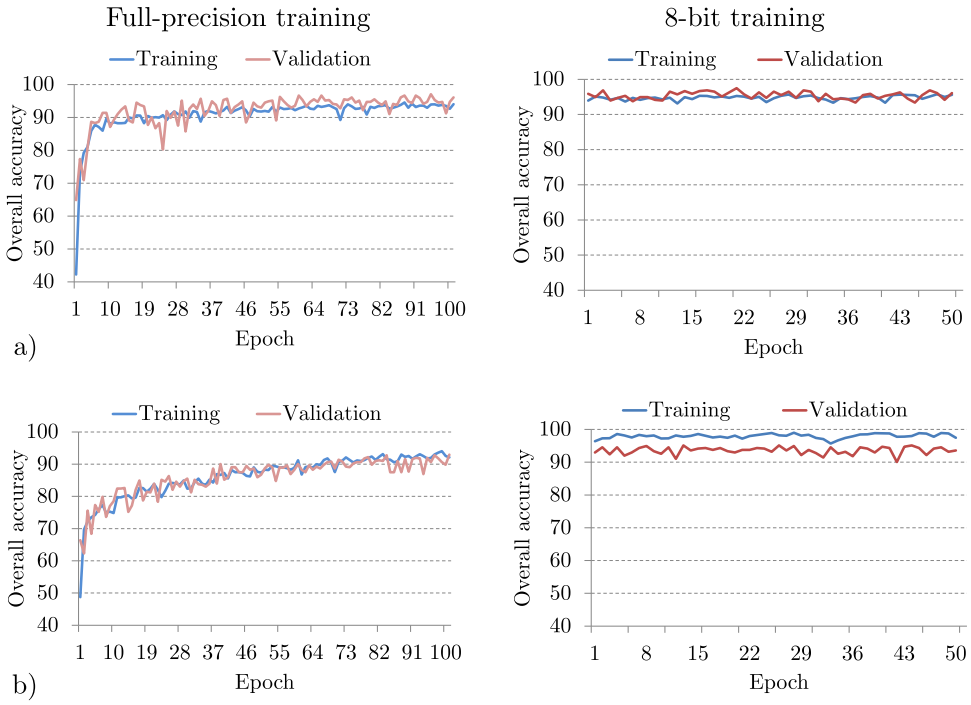

Fig. 8. Example overall accuracy obtained over the training and validation sets during both phases of training (the full-precision training phase on the left, and the 8-bit phase, after adding fake quantization nodes, on the right) for (a) Salinas Valley, and (b) Pavia University. Note that we present only initial 100 epochs for the full-precision training in this example, as the training process converged, and the overall accuracies for neither training nor validation sets fluctuated during the final 200 epochs.

Table 11
The memory footprints (in megabytes) and the uplink times (in minutes) of all models for the Salinas Valley and Pavia University datasets.

Dataset →	Salinas Valley		Pavia University	
Model ↓	Size (MB)	Uplink time (min)	Size (MB)	Uplink time (min)
1D-CNN	41.24	43.99	20.35	21.71
Q-CNN'	41.24	43.99	20.35	21.71
Q-CNN	10.31	11.00	5.09	5.43

to Earth. If this raw data was analyzed on board using a CNN, we would decrease the amount of data for transfer up to 200 ×, as the resulting (segmented) image would be the only one to send. On the other hand, if we wanted to replace a deep model on a working satellite, perhaps to a new one trained over updated training data or due to a new use case that is being tackled with a new model, we would have to exploit a much slower uplink connection (let us assume 0.25 Mbps with the efficiency of 0.5). Following the same reasoning, we can calculate the estimated uplink times for the trained models.

In Table 11, we can observe that the memory requirements of the quantized models (Q-CNNs) have been decreased four times when compared with their full-precision counterparts which led to elaborating resource-frugal CNNs. Note that adding fake quantization nodes, as being an intermediate step during the quantization process, does not affect the size of the original models. However, reducing the size of the models significantly decreased the estimated uplink times. Assuming that a satellite has an orbital altitude of 600 km (sun-synchronous orbit), we may easily end up having a communication window of approximately 4 hours per month, hence decreasing the downlink and uplink times is a critical issue for efficient operational phase of a satellite (in both duplex and half-duplex communication modes) [73]. Therefore, minimizing the memory footprints of deep models trained on Earth can not only allow us to optimize the time necessary to enter the satellite operational phase with a new model deployed on board, but also to test up to 4 × (Table 11) new deep models in the operational environment. It, in turn, can lead to better Earth observation services delivered in much shorter times, hence to making them applicable in new real-life use cases, including environmental monitoring and disaster detection.

Table 12
The results (p-values) of two-tailed Wilcoxon's tests show that the differences between the per-class accuracies obtained using quantized and non-quantized CNNs are not statistically important.

	Q-CNN'	Q-CNN
CNN	> 0.2	> 0.05
Q-CNN'	–	> 0.1

4.3.3. Statistical analysis

To verify the statistical significance of the obtained results and to check if the quantization process affected the capabilities of our spectral CNN in a statistically-important way, we executed two-tailed Wilcoxon's tests over the averaged per-class accuracies. The p-values reported in Table 12 show that quantization allowed us to obtain the deep models which generalize as good as their full-precision counterparts (all p-values are $p > 0.1$). It indicates that we can deliver high-quality segmentation of HSI using models with significantly reduced memory footprints.

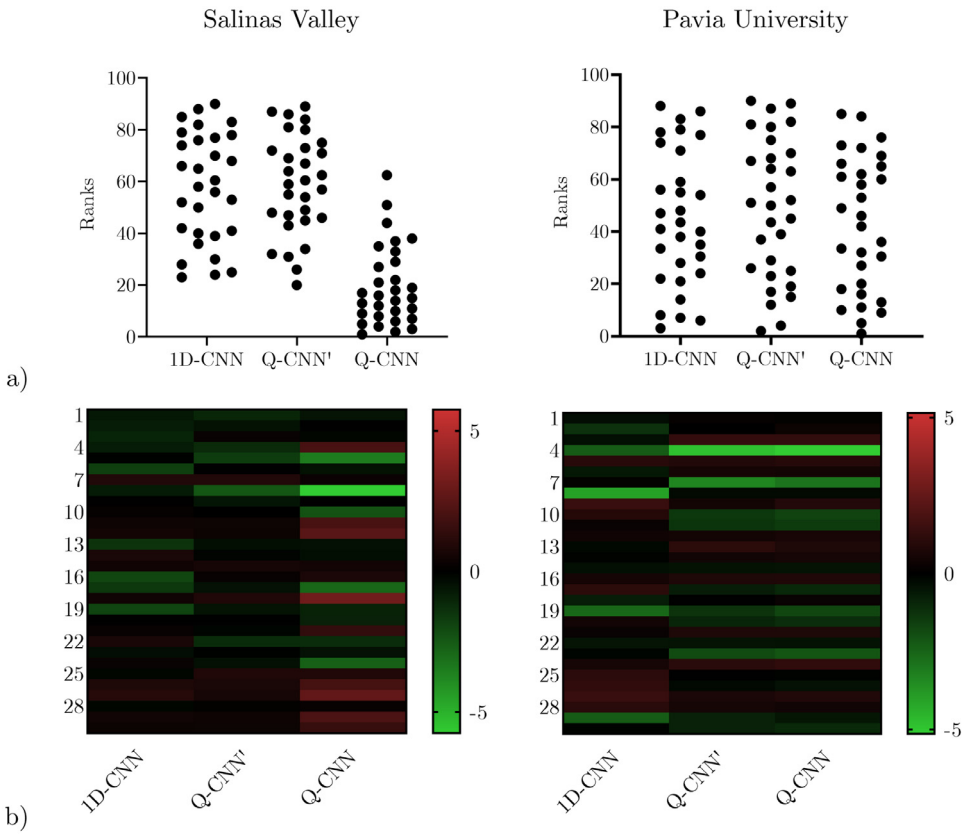

Fig. 9. The results of the Kruskal-Wallis tests (performed over the overall accuracy obtained for Salinas Valley and Pavia University): (a) the ranks obtained for each model, and (b) the differences between the OA values obtained for each execution and the median OA. If the differences are close to zero, then the deep model is more "stable" and leads to obtaining the OA values close to their median value in every run.

The results of the Kruskal-Wallis non-parametric tests with the Dunn's multiple comparison test (over OA obtained for Salinas Valley and Pavia University) show that there is no statistical difference in OA obtained using full-precision and quantized CNNs for Pavia University (for 1D-CNN vs. Q-CNN', Q-CNN' vs. Q-CNN, and 1D-CNN vs. Q-CNN), whereas these differences are statistically important for Salinas Valley (for 1D-CNN vs. Q-CNN at $p < 0.05$, the other differences are not statistically significant). In Fig. 9, we visualize the ranks alongside the differences between the obtained OA and the median OA values. We can appreciate the fact that introducing fake quantization nodes does not affect OA of the models—it is also manifested in both ranks obtained for all models (which are evenly distributed for 1D-CNN and Q-CNN'), and the differences between the OA values and their median for 1D-CNN and Q-CNN' (they are close to zero). Although quantization led to deteriorating the final classification accuracy of Q-CNN for Salinas Valley, the models performed much worse than their full-precision counterparts in the case of only two classes: C3 and C8 (Table 3). This decrease in accuracy might be potentially addressed by using additional training- and/or test-time data augmentation to increase the representativeness and size of training sets, especially for these classes [60]—note that C8 is the most numerous class in this dataset, hence sampling a small subset of all examples could easily lead to non-representative training data.

5. Conclusions and future work

We introduced resource-frugal spectral convolutional neural networks for segmenting hyperspectral satellite images, and investigated their performance and memory footprints using two most-popular hyperspectral benchmarks. To decrease the memory requirements of our models, we exploited the quantization-aware training which starts with the full-precision training, followed by additional fine-tuning performed for a model augmented with auxiliary fake quantization nodes that simulate quantization errors. The experimental study, coupled with various visualizations and statistical tests, revealed that the quantization process does not deteriorate the performance of the deep models (the results are the same in the statistical sense when compared with those obtained for full-precision CNNs), and allowed us to obtain the models which are four times smaller than their original counterparts. Such resource-frugal CNNs are much easier to deploy in hardware-constrained execution environments [7,63], e.g., on board an imaging satellite in real-life Earth observation scenarios.

In this paper, we focused on investigating the memory savings of quantized CNNs, alongside their classification abilities. Since such resource-frugal models are often required in hardware-constrained or mobile environments, our current research focus includes verifying the impact of the quantization process on the inference time of the corresponding models, and comparing them with other state-of-the-art classification techniques. Also, we work on quantizing spectral-spatial deep neural networks, and on validating their performance over the patch-based splits of hyperspectral benchmark sets [62]. Finally, varying the number of quantization levels could shed more light on the abilities of deep models deployed over different hardware architectures.

Declaration of Competing Interest

None.

Acknowledgements

This work was supported by the Polish National Centre for Research and Development under Grant POIR.01.01.01-00-0356/17. JN, MA, and MMy were partially supported by European Space Agency (HYPERNET project). JN was supported by the Silesian University of Technology funds (02/020/BKM19/0183 and The Rector's Habilitation Grant No. 02/020/RGH19/0185).

The authors are grateful to the anonymous Reviewers for their constructive and valuable comments that helped improve the paper.

References

[1] D. Alistarh, D. Grubic, J. Li, R. Tomioka, M. Vojnovic, QSGD: communication-efficient SGD via Gradient Quantization and Encoding, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 1709–1720.

[2] P. Amer, G. Banos, Implications of avoiding overlap between training and testing data sets when evaluating genomic predictions of genetic merit, J. Dairy Sci. 93 (7) (2010) 3320–3330.

[3] G. Bilgin, S. Erturk, T. Yildirim, Segmentation of hyperspectral images via subtractive clustering and cluster validation using one-class SVMs, IEEE TGRS 49 (8) (2011) 2936–2944.

[4] J.M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, J. Chanussot, Hyperspectral remote sensing data analysis and future challenges, IEEE Geosci. Remote Sens. Mag. 1 (2) (2013) 6–36, doi:10.1109/MGRS.2013.2244672.

[5] M. Blott, T.B. Preußer, N.J. Fraser, G. Gambardella, K. O'Brien, Y. Umuroglu, M. Leeser, K.A. Vissers, Finn-*R*: an end-to-end deep-learning framework for fast exploration of quantized neural networks, TRETS 11 (3) (2018) 16:1–16:23.

[6] L. Cao, C. Wang, J. Li, Vehicle detection from highway satellite images via transfer learning, Inf. Sci. 366 (2016) 177–187.

[7] W. Chang, D. Goswami, S. Chakraborty, L. Ju, C.J. Xue, S. Andalam, Memory-aware embedded control systems design, IEEE TCAD PICS 36 (4) (2017) 586–599, doi:10.1109/TCAD.2016.2613933.

[8] A. Chatterjee, L.R. Varshney, Towards optimal quantization of neural networks, in: Proc. IEEE ISIT, 2017, pp. 1162–1166.

[9] C. Chen, G.G. Lee, V. Sritapan, C. Lin, Deep convolutional neural network on iOS mobile devices, in: 2016 IEEE International Workshop on Signal Processing Systems (SiPS), 2016, pp. 130–135, doi:10.1109/SiPS.2016.31.

[10] Y. Chen, H. Jiang, C. Li, X. Jia, P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, IEEE TGRS 54 (10) (2016) 6232–6251.

[11] Y. Chen, H. Jiang, C. Li, X. Jia, P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, IEEE TGRS 54 (10) (2016) 6232–6251, doi:10.1109/TGRS.2016.2584107.

[12] Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. Gu, Deep learning-based classification of hyperspectral data, IEEE J-STARS 7 (6) (2014) 2094–2107.

[13] Y. Chen, X. Zhao, X. Jia, Spectral-spatial classification of hyperspectral data based on deep belief network, IEEE J-STARS 8 (6) (2015) 2381–2392.

[14] Z. Chen, B. Wang, Spectral-spatial classification based on affinity scoring for hyperspectral imagery, IEEE J-STARS 9 (6) (2016) 2305–2320, doi:10.1109/JSTARS.2016.2536761.

[15] Y. Choukroun, E. Kravchik, P. Kisilev, Low-bit quantization of neural networks for efficient inference, CoRR (2019) 1–10. http://arxiv.org/abs/1902.06822.

[16] M. Courbariaux, Y. Bengio, J.-P. David, Binaryconnect: training deep neural networks with binary weights during propagations, in: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28, Curran Associates, Inc., 2015, pp. 3123–3131.

[17] R. Ding, Z. Liu, R.D.S. Blanton, D. Marculescu, Quantized deep neural networks for energy efficient hardware-based inference, in: 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), 2018, pp. 1–8.

[18] T. Dundar, T. Ince, Sparse representation-based hyperspectral image classification using multiscale superpixels and guided filter, IEEE GRSL (2018) 1–5, doi:10.1109/LGRS.2018.2871273.

[19] X. Feng, Y. Jiang, X. Yang, M. Du, X. Li, Computer vision algorithms and hardware implementations: a survey, Integration 69 (2019) 309–320.

[20] A. Fourcade, R. Khonsari, Deep learning in medical image analysis: a third eye for doctors, J. Stomatol. Oral Maxillofac.Surg. 120 (4) (2019) 279–288.

[21] J.C.B. Gamboa, Deep learning for time-series analysis, CoRR (2017) 1–13. http://arxiv.org/abs/1701.01887.

[22] Q. Gao, S. Lim, X. Jia, Hyperspectral image classification using convolutional neural networks and multiple feature learning, Remote Sens. 10 (2) (2018) 299.

[23] A. Garcí-Pedrero, C. Gonzalo-Martí-n, M. Lillo-Saavedra, A machine learning approach for agricultural parcel delineation through agglomerative segmentation, Int. J. Remote Sens. 38 (7) (2017) 1809–1819.

[24] Y. Gong, L. Liu, M. Yang, L.D. Bourdev, Compressing deep convolutional networks using vector quantization, CoRR (2014) 1–10. http://arxiv.org/abs/1412.6115.

[25] I.J. Goodfellow, Y. Bengio, A.C. Courville, Deep Learning, Adaptive Computation and Machine Learning, MIT Press, 2016. http://www.deeplearningbook.org/.

[26] N. Gorretta, G. Rabatel, C. Fiorio, C. Lelong, J. Roger, An iterative hyperspectral image segmentation method using a cross analysis of spectral and spatial information, Chemometr. Intell. Lab. Syst. 117 (2012) 213–223.

[27] Ç. Gülçehre, S. Ahn, R. Nallapati, B. Zhou, Y. Bengio, Pointing the unknown words, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers, 2016.

[28] Y. Guo, A survey on methods and theories of quantized neural networks, CoRR (2018) 1–17. http://arxiv.org/abs/1808.04752.

[29] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, H. Larochelle, Brain tumor segmentation with deep neural networks, Med. Image Anal. 35 (2017) 18–31.

[30] S. Haykin, S. Wright, Y. Bengio, Big data: theoretical aspects, Proc. IEEE 104 (1) (2016) 8–10, doi:10.1109/JPROC.2015.2507658.

[31] Z. He, Y. Shen, M. Zhang, Q. Wang, Y. Wang, R. Yu, Spectral-spatial hyperspectral image classification via SVM and superpixel segmentation, in: 2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014, pp. 422–427, doi:10.1109/I2MTC.2014.6860780.

[32] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio, Quantized neural networks: training neural networks with low precision weights and activations, J. Mach. Learn. Res. 18 (2017) 187:1–187:30.

[33] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: a review, Data Min. Knowl. Discov. 33 (4) (2019) 917–963.

[34] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A.G. Howard, H. Adam, D. Kalenichenko, Quantization and training of neural networks for efficient integer-arithmetic-only inference, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, 2018, pp. 2704–2713, doi:10.1109/CVPR.2018.00286.

[35] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, L. Wang, SuperPCA: a superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery, IEEE TGRS 56 (8) (2018) 4581–4593, doi:10.1109/TGRS.2018.2828029.

[36] C.S.C. John, E. Ball, D.T. Anderson, Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community, Journal of Applied Remote Sensing 11 (4) (2017). 1–54–54

[37] S. Kaufman, S. Rosset, C. Perlich, Leakage in data mining: formulation, detection, and avoidance, in: KDD, ACM, 2011, pp. 556–563.

[38] R.A. Khalil, E. Jones, M.I. Babar, T. Jan, M.H. Zafar, T. Alhussain, Speech emotion recognition using deep learning techniques: a review, IEEE Access 7 (2019) 117327–117345, doi:10.1109/ACCESS.2019.2936124.

[39] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015.

[40] Y. Kong, X. Wang, Y. Cheng, Spectral-spatial feature extraction for HSI classification based on supervised hypergraph and sample expanded CNN, IEEE J-STARS 11 (11) (2018) 4128–4140, doi:10.1109/JSTARS.2018.2869210.

[41] R. Krishnamoorthi, Quantizing deep convolutional networks for efficient inference: A whitepaper, CoRR (2018) 1–36. http://arxiv.org/abs/1806.08342.

[42] M. Längkvist, A. Kiselev, M. Alirezaie, A. Loutfi, Classification and segmentation of satellite orthoimagery using convolutional neural networks, Remote Sens. 8 (4) (2016) 329.

[43] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature (521) (2016) 436–555, doi:10.1038/nature14539.

[44] F. Li, D.A. Clausi, L. Xu, A. Wong, ST-IRGS: a region-based self-training algorithm applied to hyperspectral image classification and segmentation, IEEE TGRS 56 (1) (2018) 3–16.

[45] J. Li, J.M. Bioucas-Dias, A. Plaza, Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields, IEEE TGRS 50 (3) (2012) 809–823, doi:10.1109/TGRS.2011.2162649.

[46] T. Li, J. Zhang, Y. Zhang, Classification of hyperspectral image based on deep belief nets, in: Proc. IEEE ICIP, 2014, pp. 5132–5136.

[47] X. Li, S. Zhang, B. Jiang, Y. Qi, M.C. Chuah, N. Bi, DAC: data-free automatic acceleration of convolutional networks, in: IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Waikoloa Village, HI, USA, January 7–11, 2019, 2019, pp. 1598–1606.

[48] D.D. Lin, S.S. Talathi, V.S. Annapureddy, Fixed point quantization of deep convolutional networks, in: Proc. ICML, JMLR.org, 2016, pp. 2849–2858.

[49] G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A. van der Laak, B. van Ginneken, C.I. Sánchez, A survey on deep learning in medical image analysis, Med. Image Anal. 42 (2017) 60–88.

[50] P. Liu, H. Zhang, K.B. Eom, Active deep learning for classification of hyperspectral images, IEEE J-STARS 10 (2) (2017) 712–724.

[51] P.R. Lorenzo, L. Tulczyjew, M. Marcinkiewicz, J. Nalepa, Band selection from hyperspectral images using attention-based convolutional neural networks, CoRR (2018) 1–7. http://arxiv.org/abs/1811.02667.

[52] D.M. Loroch, F.-J. Pfreundt, N. Wehn, J. Keuper, Tensorquant: a simulation toolbox for deep neural network quantization, in: Proc. MLHPC, ACM, New York, NY, USA, 2017, pp. 1:1–1:8.

[53] C. Louizos, M. Reisser, T. Blankevoort, E. Gavves, M. Welling, Relaxed quantization for discretized neural networks, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019, 2019.

[54] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, B.A. Johnson, Deep learning in remote sensing applications: ameta-analysis and review, ISPRS J. Photogramm. Remote Sens. 152 (2019) 166–177.
[55] Y. Mishchenko, Y. Goren, M. Sun, C. Beauchene, S. Matsoukas, O. Rybakov, S.N.P. Vitaladevuni, Low-bit quantization and quantization-aware training for small-footprint keyword spotting, OpenReview (2019) 1–5. https://openreview.net/pdf?id=rJxVxiiDoX.
[56] L. Mou, P. Ghamisi, X.X. Zhu, Deep recurrent nets for hyperspectral classification, IEEE TGRS 55 (7) (2017) 3639–3655.
[57] M. Nagel, M. van Baalen, T. Blankevoort, M. Welling, Data-free quantization through weight equalization and bias correction, CoRR (2019). http://arxiv.org/abs/1906.04721.
[58] J. Nalepa, M. Kawulok, Selecting training sets for support vector machines: a review, Artif. Intell. Rev. 52 (2) (2019) 857–900, doi:10.1007/s10462-017-9611-1.
[59] J. Nalepa, G. Mrukwa, M. Kawulok, Evolvable deep features, in: K. Sim, P. Kaufmann (Eds.), Applications of Evolutionary Computation, Springer International Publishing, Cham, 2018, pp. 497–505.
[60] J. Nalepa, M. Myller, M. Kawulok, Training and test-time data augmentation for hyperspectral image segmentation, IEEE GRSL (2019) 1–5, doi:10.1109/LGRS.2019.2921011.
[61] J. Nalepa, M. Myller, M. Kawulok, Transfer learning for segmenting dimensionally reduced hyperspectral images, IEEE GRSL (2019) 1–5, doi:10.1109/LGRS.2019.2942832.
[62] J. Nalepa, M. Myller, M. Kawulok, Validating hyperspectral image segmentation, IEEE GRSL 16 (8) (2019) 1264–1268, doi:10.1109/LGRS.2019.2895697.
[63] M.A. Neggaz, I. Alouani, P.R. Lorenzo, S. Niar, A reliability study on CNNs for critical embedded systems, in: Proc. IEEE ICCD, 2018, pp. 476–479, doi:10.1109/ICCD.2018.00077.
[64] Z. Niu, W. Liu, J. Zhao, G. Jiang, Deeplab-based spatial feature extraction for hyperspectral image classification, IEEE GRSL 16 (2) (2019) 251–255, doi:10.1109/LGRS.2018.2871507.
[65] D.W. Otter, J.R. Medina, J.K. Kalita, A survey of the usages of deep learning in natural language processing, CoRR (2018) 1–22. http://arxiv.org/abs/1807.10854.
[66] B. Priego, D. Souto, F. Bellas, R.J. Duro, Hyperspectral image segmentation through evolved cellular automata, Pattern Recognit. Lett. 34 (14) (2013) 1648–1658.
[67] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, XNOR-Net: imagenet classification using binary convolutional neural networks, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV, 2016, pp. 525–542, doi:10.1007/978-3-319-46493-0_32.
[68] P. Ribalta, M. Marcinkiewicz, J. Nalepa, Segmentation of hyperspectral images using quantized convolutional neural networks, in: Proc. IEEE DSD, 2018, pp. 260–267.
[69] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[70] A. Santara, K. Mani, P. Hatwar, et al., BASS Net: band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification, IEEE TGRS 55 (9) (2017) 5293–5301.
[71] S.S. Sawant, M. Prabukumar, A review on graph-based semi-supervised learning methods for hyperspectral image classification, TheEgyptian Journal of Remote Sensing and Space Science (2018). (in press)
[72] J. Schmidt, M.R.G. Marques, S. Botti, M.A.L. Marques, Recent advances and applications of machine learning in solid-state materials science, NPJ Comput. Mater. 5 (1) (2019) 83.
[73] Y. Seyedi, S.M. Safavi, On the analysis of random coverage time in mobile LEO satellite communications, IEEE Commun. Lett. 16 (5) (2012) 612–615, doi:10.1109/LCOMM.2012.031912.112323.
[74] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q.V. Le, G.E. Hinton, J. Dean, Outrageously large neural networks: the sparsely-gated mixture-of-experts layer, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings, 2017.
[75] T. Sheng, C. Feng, S. Zhuo, X. Zhang, L. Shen, M. Aleksic, A quantization-friendly separable convolution for MobileNets, CoRR (2018). http://arxiv.org/abs/1803.08607.
[76] A. Shrestha, A. Mahmood, Review of deep learning algorithms and architectures, IEEE Access 7 (2019) 53040–53065, doi:10.1109/ACCESS.2019.2912200.
[77] W. Tang, G. Hua, L. Wang, How to train a compact binary neural network with high accuracy? in: Proc. AAAI Conference on Artificial Intelligence, 2017. https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14619.
[78] Y. Tarabalka, J. Chanussot, J. Benediktsson, Segmentation and classification of hyperspectral images using watershed transformation, Pattern Recognit. 43 (7) (2010) 2367–2379.
[79] Y. Tian, K. Pei, S. Jana, B. Ray, Deeptest: automated testing of deep-neural-network-driven autonomous cars, in: Proc. ICSE, ACM, New York, NY, USA, 2018, pp. 303–314.
[80] J. Transon, R. d'Andrimont, A. Maugnard, P. Defourny, Survey of hyperspectral earth observation applications from space in the sentinel-2 context, Remote Sens. 10 (2) (2018) 157, doi:10.3390/rs10020157.
[81] Y. Umuroglu, M. Jahre, Towards efficient quantized neural network inference on mobile devices: work-in-progress, in: Proceedings of the 2017 International Conference on Compilers, Architectures and Synthesis for Embedded Systems, CASES 2017, Seoul, Republic of Korea, October 15–20, 2017, 2017, pp. 18:1–18:2.
[82] M.A. Veganzones, G. Tochon, M. Dalla-Mura, A.J. Plaza, J. Chanussot, Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation, IEEE TIP 23 (8) (2014) 3574–3589, doi:10.1109/TIP.2014.2329767.
[83] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, H. Li, Terngrad: ternary Gradients to Reduce Communication in Distributed Deep Learning, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 1509–1519.
[84] J. Wu, C. Leng, Y. Wang, Q. Hu, J. Cheng, Quantized convolutional neural networks for mobile devices, in: Proc. CVPR, 2016, pp. 4820–4828, doi:10.1109/CVPR.2016.521.
[85] F. Xing, Y. Xie, H. Su, F. Liu, L. Yang, Deep learning in microscopy image analysis: asurvey, IEEE Trans. Neural Netw. Learn. Syst. 29 (10) (2018) 4550–4568, doi:10.1109/TNNLS.2017.2766168.
[86] W. Yang, L. Jin, M. Liu, Chinese character-level writer identification using path signature feature, dropstroke and deep CNN, in: Proc. IDCAR, 2015, pp. 546–550.
[87] P. Yin, J. Lyu, S. Zhang, S.J. Osher, Y. Qi, J. Xin, Understanding straight-through estimator in training activation quantized neural nets, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019. https://openreview.net/forum?id=Skh4jRcKQ.
[88] P. Yin, S. Zhang, J. Lyu, S. Osher, Y. Qi, J. Xin, Blended coarse gradient descent for full quantization of deep neural networks, Res. Math. Sci. 6 (1) (2019) 14.
[89] S. Yu, S. Jia, C. Xu, Convolutional neural networks for hyperspectral image classification, Neurocomputing 219 (2017) 88–98.
[90] P.W. Yuen, M. Richardson, An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition, Imaging Sci. J. 58 (5) (2010) 241–253, doi:10.1179/174313110X12771950995716.
[91] L. Zhang, J. Lin, B. Liu, Z. Zhang, X. Yan, M. Wei, A review on deep learning applications in prognostics and health management, IEEE Access 7 (2019) 162415–162438, doi:10.1109/ACCESS.2019.2950985.
[92] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, A.C. Courville, Towards end-to-end speech recognition with deep convolutional neural networks, in: Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8–12, 2016, 2016, pp. 410–414, doi:10.21437/Interspeech.2016-1446.
[93] D.-D. Zhao, F. Li, K. Sharif, G.-M. Xia, Y. Wang, Space efficient quantization for deep convolutional neural networks, J. Comput. Sci. Technol. 34 (2) (2019) 305–317.
[94] W. Zhao, S. Du, Spectral-spatial feature extraction for hyperspectral image classification, IEEE TGRS 54 (8) (2016) 4544–4554.
[95] Z. Zhao, P. Zheng, S. Xu, X. Wu, Object detection with deep learning: A review, CoRR (2018). http://arxiv.org/abs/1807.05511.
[96] P. Zhong, Z. Gong, S. Li, C. Schönlieb, Learning to diversify deep belief networks for hyperspectral image classification, IEEE TGRS 55 (6) (2017) 3516–3530.
[97] S.-C. Zhou, Y.-Z. Wang, H. Wen, Q.-Y. He, Y.-H. Zou, Balanced quantization: an effective and efficient approach to quantized neural networks, J. Comput. Sci. Technol. 32 (4) (2017) 667–682.
[98] Y. Zhou, S. Moosavi-Dezfooli, N. Cheung, P. Frossard, Adaptive quantization for deep neural network, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018, 2018, pp. 4596–4604.

Jakub Nalepa received the M.Sc. (2011) and Ph.D. (2016) degrees, both with distinction in computer science from the Silesian University of Technology, Gliwice, Poland, where he is currently an Assistant Professor. His research interests encompass machine learning, deep learning, evolutionary algorithms, pattern recognition, medical and satellite imaging, and inter disciplinary applications of these methods. He has been involved in several projects related to the above-mentioned domains|in both academia and industry. So far, he has published more than 90 papers in these fields.

Marek Antoniak received the M.Sc. degree (2010) in computer science from the Silesian University of Technology, Gliwice, Poland. He currently works as a Software Engineer at KP Labs where he focuses on efficient data analysis and programming on-board satellite computers. He has been involved in the PW-Sat2 and Intuition-1 missions.

Michal Myller is a student of Silesian University of Technology, Gliwice, Poland, currently finishing his M.Sc. degree in Data Science. Michal is interested in machine learning, deep learning, pattern and object recognition, and conduct many personal projects in these fields. Presently he is working as a Machine Learning Specialist at KP Labs, Gliwice, Poland, developing deep learning solutions for satellite imaging, focusing on supervised and unsupervised hyperspectral image segmentation.

Pablo Ribalta Lorenzo is an experienced engineer and researcher specialized in the design and implementation of state of the art technology, built around Machine Learning and AI. He currently works as a Deep Learning Engineer at NVIDIA.

Michal Marcinkiewicz holds a master's degree in physics from the University of Warsaw, followed by a PhD in physics at fundamental research in Montpellier, France. Currently he conducts research in the domain of deep learning, mainly on computer vision, representation learning, optimization, and model compression. He worked on automatic segmentation and processing of medical images, systems for efficient analysis of hyperspectral satellite data, models for fast and accurate classification of natural images, efficient image style transfer, and audio processing and classification. He authored and co-authored 19 publications and conference papers across both physics and computer science.