Article

Classification of Compressed Remote Sensing Multispectral Images via Convolutional Neural Networks

Michalis Giannopoulos [1,2,*,†] **, Anastasia Aidini** [1,2,†] **, Anastasia Pentari** [1,2] **,**
Konstantina Fotiadou [1,2] **and Panagiotis Tsakalides** [1,2]

[1]   Signal Processing Lab (SPL), Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH), 70013 Crete, Greece; aidini@csd.uoc.gr (A.A.); anpentari@gmail.com (A.P.); kfot@ics.forth.gr (K.F.); tsakalid@ics.forth.gr (P.T.)
[2]   Computer Science Department, University of Crete, 70013 Crete, Greece
*   Correspondence: mgiannop@ics.forth.gr
†   These authors contributed equally to this work.

check for
updates

Abstract: Multispectral sensors constitute a core Earth observation image technology generating massive high-dimensional observations. To address the communication and storage constraints of remote sensing platforms, lossy data compression becomes necessary, but it unavoidably introduces unwanted artifacts. In this work, we consider the encoding of multispectral observations into high-order tensor structures which can naturally capture multi-dimensional dependencies and correlations, and we propose a resource-efficient compression scheme based on quantized low-rank tensor completion. The proposed method is also applicable to the case of missing observations due to environmental conditions, such as cloud cover. To quantify the performance of compression, we consider both typical image quality metrics as well as the impact on state-of-the-art deep learning-based land-cover classification schemes. Experimental analysis on observations from the ESA Sentinel-2 satellite reveals that even minimal compression can have negative effects on classification performance which can be efficiently addressed by our proposed recovery scheme.

1. Introduction

The domain of *Remote Sensing* (RS) focuses on monitoring the physical characteristics of a target area by measuring the reflected and emitted radiation from a distance, in contrast to in situ observations which require on-site measuring devices. To achieve this objective, airborne and spaceborne *Multispectral* (MS) and *Hyperspectral* (HS) sensors, capable of observing an extended region of the electromagnetic spectrum, are used in order to detect and classify objects on Earth, exploiting the unique spectral profiles of different materials [1]. While the acquisition of a large number of observations is critical for Earth observation, as it provides valuable spatial and spectral information, the copious amounts of data collected from high-resolution sensors over extended time periods that can be formulated by means of high-dimensional data structures, known as tensors, introduce considerable challenges in terms of data storage and data transfer [2]. Especially in RS cases, MS images that are modeled as a third-order tensor defined by two spatial and one spectral variable, are collected on satellites or unmanned aerial vehicles and need to be transferred to the ground-based stations. For that purpose, compression of the acquired MS images is mandatory, in order to reduce bandwidth and increase the system lifetime [3].

To that end, a number of lossless and lossy compression approaches have been proposed, including transform coding methods, such as the *Discrete Cosine Transform* and the *Discrete Wavelet Transform* (DWT) [4]. While lossless compression schemes are designed so that no information is lost during compression, they typically offer moderate compression rates, in the range of 1.5:1 to 3:1. On the other hand, lossy compression techniques can achieve much higher compression rates, in excess of 20 : 1, primarily by mapping a range of values to values from discrete sets-a process known as quantization-which can be applied to the original pixels or their representation in some transform domain.

Once the massive amounts of observations acquired by modern instruments, including satellites such as ESA's Sentinel 2 and NASA's Landsat, are transferred to ground-station, methodologies that will allow the automation of classification tasks are necessary. To that end, several *Machine Learning* (ML) algorithms have been employed by the RS community, including *Support Vector Machines* (SVMs) [5] and *Random Forests* (RFs) [6]. In the past five years, *Deep Learning* (DL) and in particular architectures such as *Convolutional Neural Networks* (CNN) have attracted considerable attention due to their significant success in various RS analysis tasks, including multi-label classification [7], land cover classification [8], object detection [9] and image retrieval [10], among others. Specifically, CNNs pass filters, widely known as kernels, over the raw image data aiming to exploit the underlying correlations among existing pixels to design as representative and compact features as possible. Most interestingly, CNNs are not fixed but instead learned through the training process of the network [11].

Since satellite imagery data depend directly on spacecraft operations, the *Launch and Early Orbit Phase* (LEOP) of space missions is always treated with great concern and caution. During LEOP, operations staff have to activate, monitor and control the various subsystems of the satellite, including the deployment of any satellite appendages (e.g., antennas, solar array, reflector, etc.), and undertake critical orbit and attitude control maneuvers. Engineering calibration (e.g., platform, instrument, and processor calibration) and geophysical validation (e.g., collection, quality control and archiving of correlative data) are crucial activities associated with this stage. Unfortunately, even though the launch phase is designed to provide high-quality reference data for the training of the ML algorithms, during test phase the online monitoring data are in most cases noise-infected (e.g., compressed) versions of the reference ones. This has an adverse effect on the ML task as the models are tested with data with significantly different statistical distributions than those during training. Motivated by this real-world problem, we conducted the present study with the intention of designing and providing a system capable of mitigating the effects of the quantization and subsampling operations.

In this paper, the effect of compression on the classification task of MS images is tackled via the notion of CNNs using a tensor recovery algorithm. More specifically, a recently released MS dataset is used to train and evaluate the performance of the employed CNN with respect to the crucial parameter of the samples it was trained with. Furthermore, a recovery algorithm of the real-valued measurements on high-dimensional data from its quantized and possibly subsampled values-due to noise or communication failures-is proposed in order to further process more efficiently the observed measurements. In particular, we consider a constrained maximum likelihood estimation of a low-rank tensor, in combination with a quantization and statistical measurement model, taking into account the quantization bin boundaries and the existence of possible noise. Subsequently, we quantify the effects of the aforementioned quantization, subsampling, and recovery processes, solely and in conjunction, in the evaluation process of the network in order to investigate its response under hostile real-world regimes. Experimental results on real satellite MS images demonstrate that when trained with a decent number of samples, CNNs can perform quite well in the MS image classification task. Moreover, direct processing with the quantized or subsampled measurements, rather than recovering their real values, ends up with a significantly higher error, indicating the efficacy of the proposed method.

All in all, the key contributions of this paper can be summarized as follows:

- Usage of CNNs (pre-trained and from-scratch models) to tackle the (quantized vs. original) MS image classification task of a recently released dataset.

- Investigation of the dataset size with which a CNN should be trained to efficiently classify real satellite MS images.
- Exploration of the effect of the quantization and subsampling processes on the MS image classification task.
- Provision of a recovery algorithm of the real-valued measurements on high-dimensional data from their quantized and possibly corrupted observations.
- Quantification of the classification scheme's performance on real quantized & subsampled recovered satellite MS images, highlighting the clear merits when it operates on the recovered images vis-à-vis their quantized counterparts.

2. Related Work

In this section, we review the state-of-the-art in terms of quantifying and optimizing the impact of MS/HS compression as well as sampling on subsequent analysis tasks such as classification.

A geometric approach for image classification presented in [12] can be used to group image patches from different source images into smooth, stochastic and dominant orientation patches, based on the similarity of image geometric structure, such as edge and sharp line information. In addition, different ML methods have also been employed for the analysis of HS observations, such as SVMs and RFs, which have been considered for problems like object detection [13] and scene classification [14,15]. In the past 3–4 years, however, this field has been dominated by deep learning methods including CNNs [16]. Towards this direction, a recently released, eloquent survey-study concerning cutting-edge DL approaches in RS applications, can be found in [17].

An interesting work employing patch-based training of CNNs was conducted in [18], whereas a multi-scale CNN approach was proposed in [19] for HS image classification purposes. A pioneering work on the field was that in [20], with specific improvements concerning the model architecture being proposed in [21]. An early attempt of classifying 3D MS imagery was explored by [22], where features from each spectral band where interdependently extracted and fused together at a higher level. Extending the notion of convolution from 2 to 3 dimensions, in [23] is proposed a 3D-CNN for RS image classification, whereas, to address the challenges associated with the limited availability of training examples, works based on the notion of *Transfer Learning* were presented in [24,25]. Additionally, two quite informative reports on land-use and land-cover classification can be found in [16,26].

The above-mentioned cases involve the use of state-of-the-art ML classifiers applied on high-quality noise-free observations. However, due to several reasons, in real-world applications observations are affected by different types of noise, including noise due to the quantization of real-valued measurements, as well as missing observations. The probabilistic model that describes this type of observations was first introduced in [27], for the case of binary or 1-bit measurements. Under the assumption that the measurement matrix is low-rank, in [27] is proposed a convex program using maximum likelihood estimation and a nuclear norm to promote a low-rank solution. A similar approach is considered in [28], but with the max-norm in place of the nuclear norm. Finally, a constrained maximum likelihood estimation based on low-rank factorization is presented in [29].

An extension to multi-level observations is introduced in [30,31], using the same observation model and a constrained maximum likelihood estimator, either calculating the set of quantization bin boundaries from the observed data using an alternating optimization procedure or assuming that the quantization bin boundaries are known. In contrast to the previous observation model, which involves just one matrix, the observation model proposed in [32] involves a vector of $K-1$ underlying matrices for K level observations. Moreover, a maximum likelihood estimation from multi-level quantized observations based on matrix factorization is presented in [33].

Concerning the case of missing observations, the application of matrix and tensor completion of HS data is considered in [34] where the authors consider HS observation from mass spectral imaging and remote sensing and conclude that for cases of limited spectral resolution, exploiting the tensor structure of the data is advantageous while for large spectral resolution, matrix completion is

preferable. The application of tensor completion for the recovery of missing values in imaging data is also considered in [35], where the authors proposed three optimization schemes for the recovery of missing values.

An interesting comparison of matrix and tensor-based methods for the recovery of missing environmental data takes place in [36], where the authors propose the exploitation of higher-order structural information, in favor of matrix-oriented techniques, when randomly distributed measurements are missing. In addition, in [37] the authors further examine the impact of completing higher-order structural information by matrix and tensor completion methods, both in the task of missing HS measurements' reconstruction, as well as in that of subsequent classification.

Although numerous algorithms have been presented for the recovery of the real-valued entries of a matrix from its quantized measurements, no prior work has been conducted for the recovery of a tensor from multi-level quantized observations. However, in the case of binary measurements, a maximum likelihood estimator using a nuclear norm constraint on the different matricizations of the underlying tensor is proposed in [38]. A tensor recovery method from binary measurements is also presented in [39], which minimizes a loss function that incorporates a generalized linear model with the tensor nuclear norm as a regularization term, using an alternating direction method of multipliers-based algorithm to find the solution. Finally, a tensor-based method that uses the decomposition of the tensor in order to retain its structure, rather than matricization, is proposed in [40], in which a certain atomic M-norm as a convex proxy for rank is used to approximate low-rank tensors from binary measurements.

The concepts of matrix and tensor completion have also been considered for the efficient on-board compression of HS imagery. Matrix completion is employed for HS compression, targeting the on-board processing of observations in [41], where the authors propose a new method for the enhancement of the spatio-spectral resolution of hypercubes acquired by snapshot spectral imagers. In [42] the authors explore the potential of matrix and tensor completion as a compression mechanism with dramatically fewer computational requirements compared to more traditional transform coding schemes like JPEG2000. To fully exploit the spatio-spectral correlations of HS imagery, they consider the group non-overlap sampling scheme and adaptive rank threshold selection, and demonstrate that the proposed scheme offers comparable performance to 3D-JPEG2000 on AVIRIS imagery using less computational resources.

Trying to capitalize on the inherent high dimensionality of HS data, tensor decompositions have also been proven to be a valuable HS image compression tool. In this flavor, the ability to express a 3D tensor encoding both spatial and spectral measurements in more compact forms through tensor decompositions is explored in [43]. Tensor decomposition of HS image is also explored in [44], where the authors consider the CANDECOMP/PARAFAC approach in order to express the 3D HS cube as the sum of rank-1 tensors and exploit the sparsity of this representation for compression.

Finally, the impact of compression in the classification task of remote sensing images considered in the proposed scheme has also been examined with different compression and classification techniques. Specifically, the effects of JPEG2000 lossy image compression on the classification of WorldView-2 satellite images are explored in [45]. The authors consider the *k-Nearest Neighbors* (k-NN) and SVM classification methods, and experimentally demonstrate that even at high compression rates (30:1) there is a minimal impact on classification accuracy and in some cases, the results are even better than the original image classification. The same observation is established in [46] where the impact of lossy compression is investigated on *Principal Component Analysis* (PCA)-based spectral unmixing and SVM-based supervised classification using the CCSDS-ICS, TER and JPEG2000 compression schemes on AVIRIS data. However, these results indicate a strange behavior that cannot be completely trusted since very high compression ratios should have an adverse effect on classification accuracy.

Furthermore, the quantification of the impact of compression on the classification accuracy of satellite images is explored in [47], where a framework based on the *Multiple Kernel Learning* algorithm is employed for the prediction of how compression affects the quality of extracted features

for classification. Specifically, the authors consider the JPEG2000 compression standard and the subsequent classification of each image into five classes (water, forest, bare land, vegetation and residential area) by two classifiers, namely a maximum likelihood and a pixel-level SVM classifier. However, the prediction performance of their method is evaluated on images with only three bands (near-infrared, red and green bands), from GF2, Landsat 8 and ZY-3.

In a similar vein, the impact of compression on synthetic PROBA-V images was investigated in [48], where the authors considered two compression schemes—the TER and the CCSDS blue book standard which evolved to the CCSDS-122 standard for lossless and near-lossless MS and HS image compression—and investigated the impact on land-cover estimation. The authors' conclusion is that compression has a minimal impact on classification, which is more pronounced in regions with multiple boundaries between different land-cover classes. Similar observations were drawn for the case of Landsat 7 Enhanced Thematic Mapper Plus (ETM+) observation compression using JPEG2000, SPIHT and CCSDS 122.0-B-1 and its impact on two important vegetation features, namely *Normalized Difference Vegetation Index* and *Normalized Difference Water Index*, which are reported in [49]. Nevertheless, the above methods cannot handle cases of missing observations, due to communication failures or physical obstacles like clouds, in contrast to our method that can efficiently recover the missing values for the subsequent processing using a quantized tensor completion algorithm.

3. Problem Formulation and Proposed Method

In this section, we initially describe the scenarios under investigation for the classification task of quantized MS images. Subsequently, we provide some background concerning the reasoning of choosing CNNs to tackle the land-cover classification task, as well as the formulation of the problem at hand via application of two-dimensional filters on MS datacubes. Furthermore, basic mathematical details on how CNNs can be used for performing the estimation task are provided, accompanied by the proposed methodology for classifying MS images. In addition, the description of the measurement model, as well as the method for the recovery of the real-valued MS images from several quantized measurements via matricization of the tensor, are provided.

3.1. Quantized Multispectral Imagery Classification

Quantization is an integral part of data acquisition, especially for the RS scenario where full data transmission introduces considerable challenges in terms of data storage and data transfer. Therefore, an important study concerns the effect of quantization on the classification task.

The simplest approach for the classification task is to operate directly on the quantized—to a specific number of bits—MS images, using a trained CNN classifier, as illustrated in Figure 1. However, instead of classifying the quantized images, one can apply a tensor-based recovery algorithm, as the one introduced in this paper, for the reconstruction of the real-valued images. Experimental results on real satellite MS images demonstrate that directly processing using the quantized measurements, rather than recovering their real values, ends up with a significantly higher error-indicating the efficacy of the proposed method.
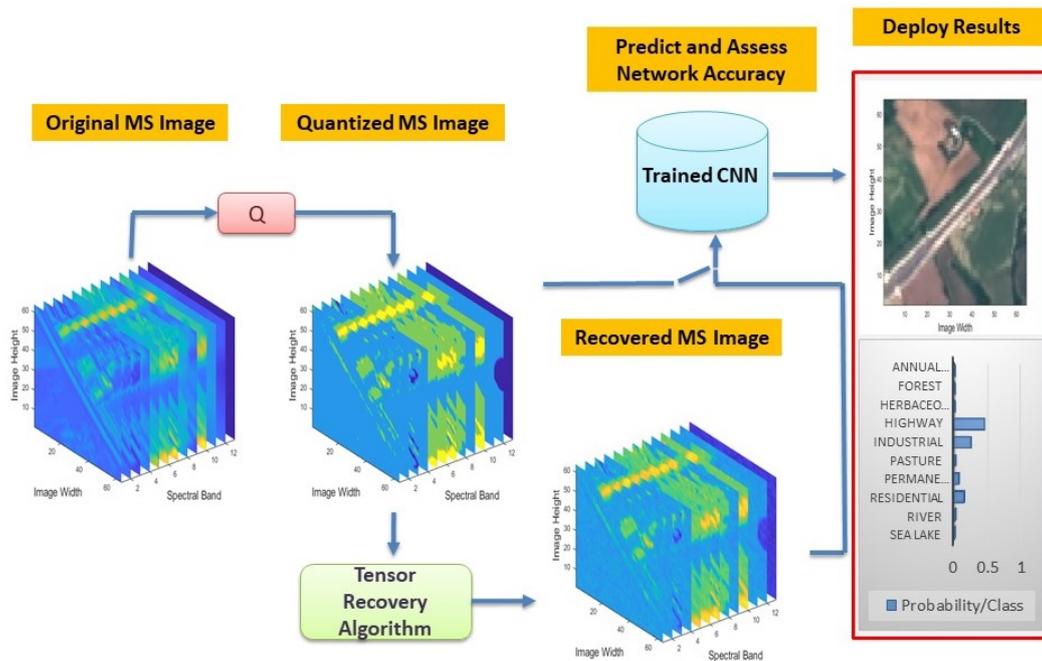
Figure 1. The scenarios under investigation for the classification task of compressed MS images.

3.2. Deep Neural Networks for Land-Cover Classification

When applying DL techniques for predictive land-cover purposes, the charged statistical learning model can be naturally compared to that of human beings: learn by (training) examples. In this context, training data (e.g., RGB/MS/HS images) is used to derive a prediction concerning the class to which a specific image scene belongs to. In order for the model to have as high representational power as possible, this prediction ability should generalize well on new (unseen-test) data, on which there is no ground truth. To do so, models are trained by using a large set of labeled data and *Neural Network* (NN) architectures that learn features directly from the data without the need for manual feature extraction.

Putting things into a stricter mathematical perspective, predictive modeling through NNs can be described as the goal of approximating a (in general, non-linear) mapping function from input variables to output variables. In the case described in this paper, the input variables correspond to image pixel values, whereas the output variable is the (discrete) label value of the predicted class corresponding to the current piece of image content being considered. The reasoning behind such an expressive power (and the outstanding classification accuracy results obtained, consequently) of NNs is explained by their characterization as *universal approximators* [50,51]. That means that NNs can approximate arbitrarily well any given function, with the error of approximation being inversely proportional to the system's order (i.e., the number of neurons). As a consequence, compensating for the non-linear optimization nature of the problem (hence the possibility of converging to local minima), NNs can achieve remarkable performance vis-à-vis conventional predictive modeling methods, and therefore are employed in the present experimental study.

3.2.1. Spatial Feature Learning with Convolutional Neural Networks

CNNs constitute one of the most popular types of *Deep Neural Networks*. A CNN convolves learned features with input data, and uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images. A CNN can have tens or hundreds of layers, each one of whom learns to detect different features of an image. Filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input to the next layer. The filters can start as very simple features, such as brightness and edges, and increase in complexity to features that uniquely define the object.

The proposed method fits within classification-flavored supervised learning, employing second-order CNNs for efficient spatial feature learning of the image content. Unlike conventional NNs, CNNs employ the notion of local receptive fields to effectively extract features from raw data. More specifically, each locally connected input subset of the input neuron is mapped to a single output neuron, a process which is performed in a stacked manner throughout convolutional layers, in order to capture as many representative features as possible. The connection between input and output neurons is performed via convolutions by means of trainable kernels, namely filters with specific filter coefficients. The number of such filters can be trimmed using pooling layers to avoid over-fitting issues.

Formally, the value of a convolved output neuron at position (k, l) in the j-th feature map of the i-th layer can be expressed as follows:

$$y_{i,j}^{k,l} = f\left(\sum_{i=0}^{H-1} \sum_{j=0}^{W-1} w_{i,j} x_{(k+i)(l+j)} + b_{i,j} \right), \tag{1}$$

where $f(.)$ is an activation function, $w_{i,j}$ stands for the value of the kernel connected to the current feature map at the position (i, j), $x_{(k+i)(l+j)}$ represents the value of the input neuron, $b_{i,j}$ is the bias of the computed feature map, and H and W are the height and width of the kernel respectively.

3.2.2. Multispectral Image Prediction Modeling

Creating a network from scratch with the aforementioned procedure, translates to clear and total determination of the network configuration by the designer. This approach provides the most control over the network and can end up with impressive results, but, at the same time, requires an understanding of the structure of a neural network and the many options for layer types and configuration.

On the other hand, fruitful intuition can be gained by several popular pre-trained network architectures commonly used for image processing applications, such as AlexNet [52] and GoogleNet [53]. The majority of the pre-trained networks are trained on a subset of the ImageNet database (http://www.image-net.org/), which is used in the *ImageNet Large-Scale Visual Recognition Challenge* (ILSVRC) [54]. These networks have been trained on more than a million images and can classify images into 1000 object categories, such as keyboard, coffee mug, pencil, and many animals. The reason for focusing on these architectures is that by learning a rich set of features they behave remarkably well in practice for a *similar* task to the problem at hand (RGB instead of MS image classification). Due to the different nature of the problem though, intuition can be gained by obtaining only their architecture and not their weights as well.

Based on the aforementioned remark, the architecture of the CNN used throughout this paper is the one proposed in [55], inspired by an innovative work in image recognition. Therein, the authors build up on the framework of *Residual Learning* to argue about the reasoning of deeper architectures vis-à-vis swallow ones. To do so, each layer is charged not to learn unreferenced functions-mappings, but residual functions with reference to its inputs. Such a task is implemented by introducing *Shortcut Connections* (i.e., connections that skip 1 or more network layers), right after every stack of at least 2 convolutional layers, as shown in Figure 2.
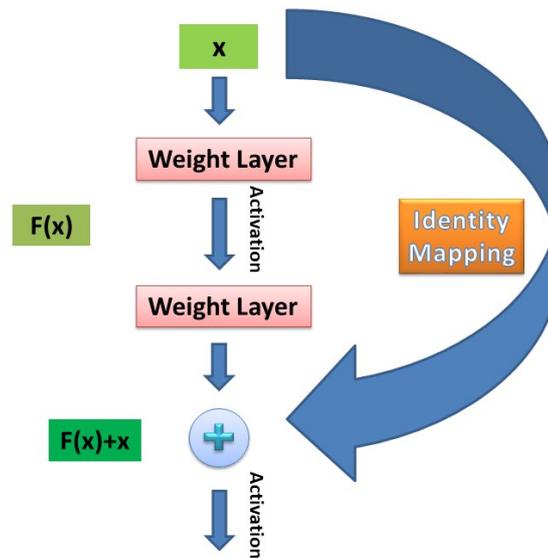
Figure 2. Residual learning framework with skip connections proposed in [55]. Instead of learning an unreferenced mapping (e.g., $\mathcal{H}(x)$), the network learns a residual mapping (e.g., $\mathcal{F}(x) + x$, with $\mathcal{F}(x) = \mathcal{H}(x) - x$). In this simple case, shortcut connections perform *identity* mapping, and subsequently add their output to the convolutional layers' stack outputs.

From a theoretical point of view, our choice of using a specific version of residual networks as the appropriate model in the present study is based on the argument proven in [55], stating that deeper residual networks address more effectively the problem of *degradation* than their shallow counterparts. In other words, residual networks do not suffer from accuracy saturation and subsequent degradation as the depth of the network increases.

We chose the ResNet-50 model as the residual architecture. The network is 50 layers deep, comprising of approximately 25.6 million parameters (i.e., weights). As the whole ResNet-50 network architecture is quite extended, in order to meet space limitations, we depict in Figure 3 a small part of the network from which it becomes evident that layers can have inputs from multiple layers and outputs to multiple layers, as well.

The pattern "family of operations" described in Figure 3 is repeated several times, one right after the other, in the whole network architecture in order to formulate the final model. Although the layers comprising the network architecture can be considered to be conventional (i.e., conv-convolutional, bn_conv-batch normalization, activation, max-pooling), it is of interest to note the flow-split in multiple branches operating on par and adding up their processing results to be propagated to the next "family of operations". We selected the ResNet-50 model over a shallower network of the same family (e.g., ResNet-18) because the deeper the architecture, the lower the classification error obtained [55].

Apart from the residual learning-based pre-trained models, several others have been proposed in the bibliography for addressing the ILSVRC and imaging tasks in general. When choosing which one to select for a specific task, one should take into consideration characteristics such as network speed, accuracy, and size (apart from task-specific issues). All of these play a crucial role in the selection (as long as a fast and highly accurate model is always desired), but usually there exists a trade-off between them due to "priority-and-resource" reasons. Our choice of ResNet-50 rested upon choosing a model with a good accuracy performance and a manageable computational burden.
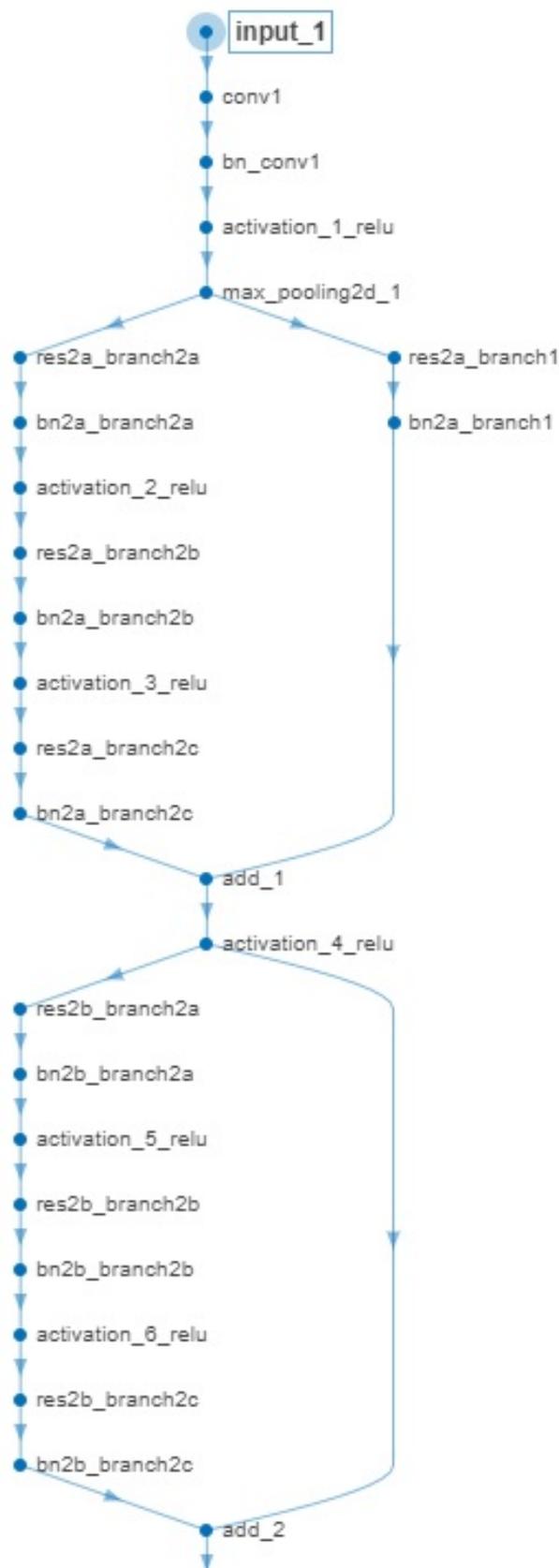
Figure 3. ResNet-50 layer graph. The employed network comprises several convolutional, activation, batch-normalization layers, connected via the notion of shortcut connections.

In terms of the training process followed, as long as our intention is to classify MS images (instead of RGB ones) into significantly fewer than 1000 classes (for which ResNet-50 model is originally trained), we adopted the following steps:

1. Load the pre-trained ResNet-50 network, which has been trained for a task of similar flavor (i.e., RGB image classification) to that at hand (i.e., MS image classification).
2. Replace the classification layers (i.e., 1000 different classes in the ImageNet database) for the new MS image classification task.
3. Train the network on the available dataset for the MS image classification task.
4. Test accuracy of the trained network.

Regarding the optimization process followed during the training of the network, instead of the standard *Stochastic Gradient Descent* (SGD) optimizer, we employed Adagrad [56] with a learning rate of 0.001. The latter one was selected due to its adaptive selection of reducing the learning rate of parameters with high gradients and, conversely, increasing the learning rate of parameters which have small or infrequent updates. In order to recast the network's outputs as probabilities, categorical cross-entropy was used as the loss function, as long as the cross-entropy can be interpreted as the log-likelihood function of the training samples.

3.3. Tensor Recovery from Quantized Measurements

In this section, we present our proposed method for the recovery of the real values of a MS image from its partial noisy and quantized measurements, in combination with a quantization and statistical model of our observations.

3.3.1. Quantization and Statistical Model

Let $\mathcal{M} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ be the N-th order tensor that models a high-dimensional signal, like a MS image, whereby the order of the tensor is defined as the number of its dimensions, also known as ways or modes. If $\Omega \subseteq \{1, ..., I_1\} \times \cdots \times \{1, ..., I_N\}$ is the index set of observed entries, $\mathcal{P}_\Omega$ is a random sampling operator which keeps the entries in Ω and zeros out others, retaining only several entries from the tensor, i.e.,

$$\mathcal{P}_\Omega(\mathcal{M}) = \begin{cases} \mathcal{M}_{i_1...i_N} & \text{if } (i_1, ..., i_N) \in \Omega \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

The quantized and corrupted observations are related to the underlying tensor $\mathcal{M}$ via a probabilistic model, as follows. Given the tensor $\mathcal{M}$ and a twice differentiable function $f_l : \mathbb{R} \to [0, 1]$, with $l \in \{1, ..., K\}$, $K \geq 2$, we observe

$$\mathcal{Y}_{i_1...i_N} = l \text{ with probability } f_l(\mathcal{M}_{i_1...i_N}), (i_1, ..., i_N) \in \Omega, \tag{3}$$

where $\sum_{l=1}^{K} f_l(\mathcal{M}_{i_1...i_N}) = 1$. This model can be considered to be the K-level quantization of noisy measurements $\mathcal{M}_{i_1...i_N} + \mathcal{E}_{i_1...i_N}$, where $\mathcal{Y}_{i_1...i_N}$ is given by

$$\mathcal{Y}_{i_1...i_N} = Q(\mathcal{M}_{i_1...i_N} + \mathcal{E}_{i_1...i_N}), (i_1, ..., i_N) \in \Omega. \tag{4}$$

The noise tensor $\mathcal{E}$ has i.i.d. entries with *Cumulative Distribution Function* (CDF) $\Phi(z)$, and the function $\mathcal{Q} : \mathbb{R} \to \{1, ..., K\}$ corresponds to a uniform quantizer that maps a real number to a set of K ordered labels according to

$$\mathcal{Q}(x) = l \text{ if } w_{l-1} < x \leq w_l, l \in \{1, ..., K\}, \tag{5}$$

where $\{w_0, w_1, ..., w_K\}$ represents the set of quantization bin boundaries of all measurements, which satisfies $w_0 \leq w_1 \leq \cdots \leq w_K$. We will assume that the set of quantization bin boundaries is known a priori.

In terms of the likelihood of the observations $\mathcal{Y}_{i_1...i_N}$, the model in (4) can be written equivalently as

$$f_l(\mathcal{M}_{i_1...i_N}) = P(\mathcal{Y}_{i_1..i_N} = l | \mathcal{M}_{i_1..i_N}) = \Phi(\mathcal{U}_{i_1..i_N} - \mathcal{M}_{i_1..i_N}) - \Phi(\mathcal{L}_{i_1..i_N} - \mathcal{M}_{i_1..i_N}), \tag{6}$$

where the $I_1 \times ... \times I_N$ tensors $\mathcal{U}$ and $\mathcal{L}$ contain the upper and lower bin boundaries corresponding to the measurements $\mathcal{Y}_{i_1...i_N}$.

We consider two common choices for $\mathcal{E}$ and its associated CDFs:

- The logistic model (logistic noise), which is common in statistics, with $\mathcal{E}_{i_1...i_N}$ i.i.d. according to the logistic distribution with zero mean and unit scale, and $\Phi_{\log}(x) = \frac{1}{1+e^{-x}}$.
- The probit model (standard normal noise) with $\mathcal{E}_{i_1...i_N}$ i.i.d. according to the standard normal distribution $\mathcal{N}(0,1)$, and $\Phi_{\mathrm{pro}}(x) = \int_{-\infty}^{x} \mathcal{N}(s \mid 0, 1) \, ds$.

The proposed algorithm can be formulated for both noise models.

3.3.2. Quantized Tensor Recovery

A common framework for tensor computations is to turn the tensor into a matrix, as it is easier to handle matrices instead of third- or higher-order tensors. The process of reordering its elements into a matrix is called matricization or unfolding of the tensor. Specifically, given the measurement tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times ... \times I_N}$, its mode-$n$ matricization is denoted as $\mathrm{unfold}_n(\mathcal{Y}) = \mathbf{Y}_{(n)} \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$, and corresponds to a matrix with columns being the vectors obtained by fixing all indices of $\mathcal{Y}$ except the n-th index [57]. The reverse process, which is the folding of a matrix to a tensor along mode-n, is denoted as $\mathrm{fold}_n(\mathbf{Y}_{(n)}) = \mathcal{Y}$.

Fortunately, the high-dimensional data often lie in a low-dimensional subspace by removing potential noise, which can be translated into the low-rank assumption on the recovered tensor. However, there is no straightforward algorithm to determine the rank of a specific given tensor, as it is an NP-hard problem [58]. Therefore, we can apply a nuclear norm constraint on each unfolding of the tensor, which is a convex relaxation of the low-rank constraint [59].

More, formally, in order to recover the real values of the low-rank tensor $\mathcal{M} \in \mathbb{R}^{I_1 \times ... \times I_N}$ from partial quantized and possibly corrupted observations, we unfold the measurement tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times ... \times I_N}$ into N matrices and for each of them, we apply the following algorithm. Then, the estimated tensors of each unfolding, $\mathcal{Z}_n = \mathrm{fold}_n(\mathbf{Z}_{(n)}), n = 1, ..., N$, are produced by folding each of the recovered matrices $\mathbf{Z}_{(n)}$, and synthesize the recovered tensor such that:

$$\mathcal{M} \approx \sum_{n=1}^{N} a_n \cdot \mathcal{Z}_n \tag{7}$$

where $a_n, \ n = 1, ..., N$, are weights, which depend on the fitting error, and satisfy $\sum_n a_n = 1$.

The proposed algorithm can be regarded as an extension of the quantized matrix recovery [30] to the quantized tensor recovery. In particular, in order to recover the low-rank mode-n matricization $\mathbf{M}_{(n)}$ from a subset of its quantized measurements, we use a constrained maximum likelihood approach. We assume $\mathcal{Y}$ is related to $\mathcal{M}$ via the probabilistic model given in (3)–(6). Then, the negative log-likelihood function is given by

$$F_{\mathbf{Y}_{(n)}}(\mathbf{X}) = - \sum_{j,k:(j,k)\in\Omega_n} \left[\sum_{l=1}^{K} \log(f_l(\mathbf{X}_{jk})) \mathbf{1}_{[\mathbf{Y}_{(n)jk}=l]} \right], \tag{8}$$

where Ω_n is the index set of observed entries of $\mathbf{M}_{(n)}$. The function $F_{\mathbf{Y}_{(n)}}$ is convex in $\mathbf{X}$ when the function f_l is log-concave in $\mathbf{X}_{jk}$. Both choices for the function f_l are log-concave. Therefore, we seek to solve the following constrained optimization problem:

$$
\begin{aligned}
\text{minimize}_{\mathbf{M}_{(n)}} \quad & - \sum_{j,k:(j,k)\in\Omega_n} \log P(\mathbf{Y}_{(n)jk} \mid \mathbf{M}_{(n)jk}) \\
\text{subject to} \quad & \|\mathbf{M}_{(n)}\|_* \leq \lambda.
\end{aligned}
\tag{9}
$$

The nuclear norm constraint $\|\mathbf{M}_{(n)}\|_* \leq \lambda$ promotes low-rankness of $\mathbf{M}_{(n)}$ [59] and the parameter $\lambda > 0$ is used to control its rank.

Since the gradient of the negative log-likelihood of the inverse logit and probit link functions are convex in $\mathbf{M}_{(n)}$ when keeping the quantization bin boundaries $w_0, w_1, ..., w_K$ fixed, the optimization problem in (9) can be solved efficiently. Starting with an initialization of the estimated matrix $\mathbf{Z}_{(n)}$ as a random matrix $\mathbf{Z}_{(n)}^1$ with entries between the corresponding quantization bin boundaries, the algorithm performs two steps at each iteration $l = 1, 2, ...$ Both steps are repeated until a maximum number of iterations $l_{\max}$ is reached or the change in $\mathbf{Z}_{(n)}$ between consecutive iterations is below a given threshold.

The first step aims at reducing the objective function $F_{\mathbf{Y}_{(n)}}(\mathbf{Z}_{(n)})$ of (9) and is given by

$$
\hat{\mathbf{Z}}_{(n)}^{l+1} \leftarrow \mathbf{Z}_{(n)}^l - \frac{s_l}{\sqrt{l}} \cdot \nabla F_{\mathbf{Y}_{(n)}},
\tag{10}
$$

where s_l is the step-size at iteration l. For simplicity, we use a constant step-size $s_l = \frac{1}{L}$, where L is the Lipschitz constant, which is given by $L_{\log} = \frac{1}{4}$ for the logistic model and $L_{\text{pro}} = 1$ for the probit model. The gradient of the objective function $F_{\mathbf{Y}_{(n)}}(\mathbf{Z}_{(n)})$ of each measurement, with respect to $\mathbf{Z}_{(n)}$, is given by

$$
[\nabla F_{\mathbf{Y}_{(n)}}]_{jk} = \begin{cases} \dfrac{\Phi'(\mathbf{L}_{(n)jk} - \mathbf{Z}_{(n)jk}) - \Phi'(\mathbf{U}_{(n)jk} - \mathbf{Z}_{(n)jk})}{\Phi(\mathbf{U}_{(n)jk} - \mathbf{Z}_{(n)jk}) - \Phi(\mathbf{L}_{(n)jk} - \mathbf{Z}_{(n)jk})} & \text{if } (j,k) \in \Omega_n \\ 0 & \text{otherwise} \end{cases},
\tag{11}
$$

where $\mathbf{L}_{(n)}$ and $\mathbf{U}_{(n)}$ are the mode-n matricizations of $\mathcal{L}$ and $\mathcal{U}$ and contain the lower and the upper bin boundaries of the observations $\mathbf{Y}_{(n)jk}$ respectively, i.e., $\mathbf{L}_{(n)} = \text{unfold}_n(\mathcal{L})$ and $\mathbf{U}_{(n)} = \text{unfold}_n(\mathcal{U})$. The derivative of the function $\Phi(x)$ can be calculated as $\Phi'_{\log}(x) = \frac{1}{2+e^{-x}+e^x}$ and $\Phi'_{\text{pro}}(x) = \mathcal{N}(x \mid 0, 1)$ for each model.

The second step aims to impose low-rankness on $\mathbf{Z}_{(n)}$ to make the solution satisfy the constraint $\|\mathbf{Z}_{(n)}\|_* \leq \lambda$. To achieve this, we take the SVD, $\tilde{\mathbf{U}}\mathbf{S}\tilde{\mathbf{V}}^T$, of the matrix $\hat{\mathbf{Z}}_{(n)}^{l+1}$ of the previous step, and we hold some of its singular values $\mathbf{s} = \text{diag}(\mathbf{S})$, depending on the parameter λ. In our experiments, we keep 97% of the information of the singular values.

3.3.3. Dynamic Weights

The weights $a_1, ..., a_N$ in (7) can be fixed as $a_n = \frac{1}{N}$, $n = 1, .., N$, indicating that the recovered tensor of each unfolding equally contributes to the recovery. However, in some cases, the recovery in one unfolding may be better than others. Therefore, instead of fixed weights, we use dynamic weights which depend on the fitting error

$$
\text{fit}_n(\mathbf{Z}_{(n)}) = \|\mathcal{P}_\Omega(\text{fold}_n(\mathbf{Z}_{(n)}) - \mathcal{Y})\|_F,
\tag{12}
$$

where $\|\mathcal{X}\|_F = \sqrt{<\mathcal{X}, \mathcal{X}>}$ is the Frobenius norm of $\mathcal{X}$ (with $<\mathcal{X}, \mathcal{Y}> = \sum_{i_1=1}^{I_1} .. \sum_{i_N=1}^{I_N} x_{i_1..i_N} y_{i_1..i_N}$ denoted to be the inner product of $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times .. \times I_N}$). The smaller $\text{fit}_n(\mathbf{Z}_{(n)})$ is, the larger a_n should be. Specifically, we set

$$a_n = \frac{[\text{fit}_n(\mathbf{Z}_{(n)})]^{-1}}{\sum_{i=1}^{N} [\text{fit}_i(\mathbf{Z}_{(i)})]^{-1}}, \ n = 1, ..., N. \tag{13}$$

As demonstrated below, the dynamic weights a_n can improve the recovery quality of the recovered tensor.

4. Experimental Evaluation

In this section, the dataset on which upcoming experiments are performed is described, as well as the experimental setup followed. Results of the adopted approach under different experimental scenarios, using several configurations in order to quantify the effects of various parameters in the whole MS image classification pipeline, are presented and discussed, to illustrate the performance of the methodologies introduced in this paper.

4.1. Dataset Description

In designing DL solutions and training ML models, the definition of a suitable training and test set is a critical task. Bearing into mind that to achieve decent classification performance, the number of samples comprising the dataset must be sufficiently large, most current land-use and land-cover classification datasets did not meet these expectations. Furthermore, the "distribution" of samples (i.e., images) inside the dataset should be as close to uniform as possible (in order to avoid class imbalance problems), with each class consisting of several thousands of samples to be fed to the NN model.

Fortunately, quite recently, a large MS image dataset (containing RGB images as well) meeting the above requirements was released [60,61], where Sentinel 2A MS image data are employed to address land-use and land-cover classification tasks. The dataset comprises image patches measuring 64×64 pixels, each one of them corresponding to spatial resolution of 10 meters/pixel, across 13 different spectral bands (in the 443 nm–2190 nm wavelength range). In total, several thousands of sample satellite images are gathered, across 34 different European countries (hence, the EUROSAT dataset name), ending up with 10 different classes.

To have a better sense of the EUROSAT dataset, in Figure 4 we depict RGB sample images for each one of the dataset's different classes, by employing spectral bands 04 (Red-665 nm), 03 (Green-560 nm) and 02 (Blue-490 nm). Since in this study we deal with a MS image classification task, in Figure 5 we demonstrate the various spectral bands (MS datacubes) of the sample images depicted in Figure 4.
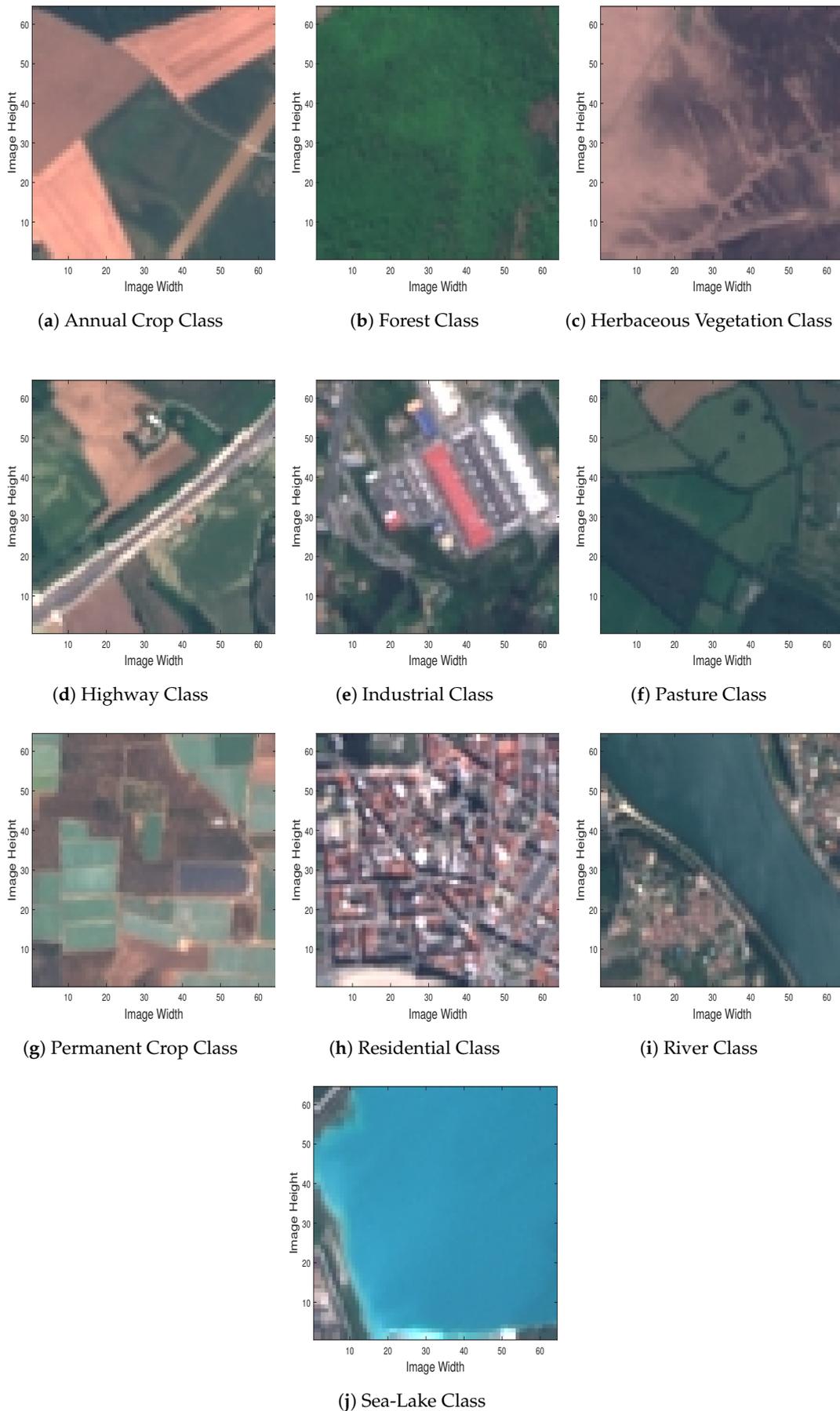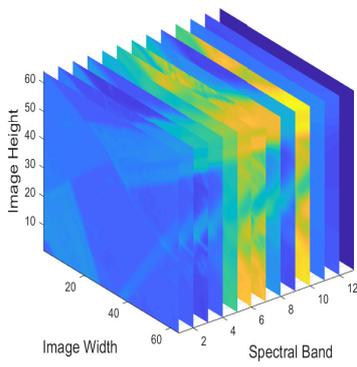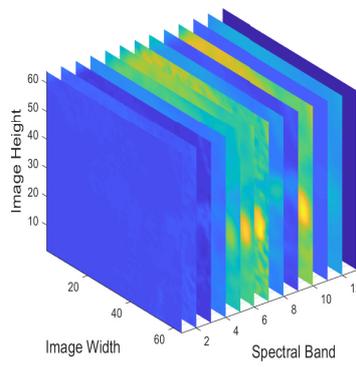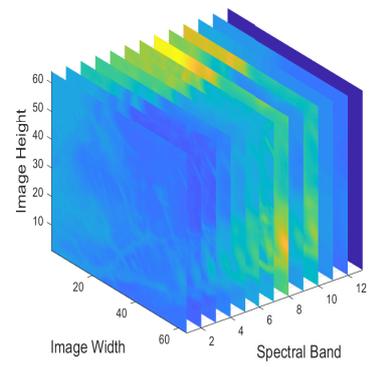
(**a**) Annual Crop Class

(**b**) Forest Class

(**c**) Herbaceous Vegetation Class

(**d**) Highway Class

(**e**) Industrial Class

(**f**) Pasture Class

(**g**) Permanent Crop Class

(**h**) Residential Class

(**i**) River Class

(**j**) Sea-Lake Class

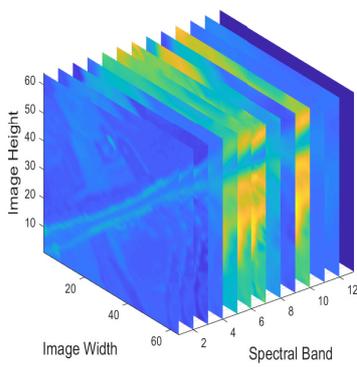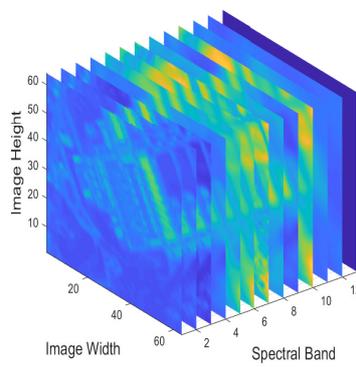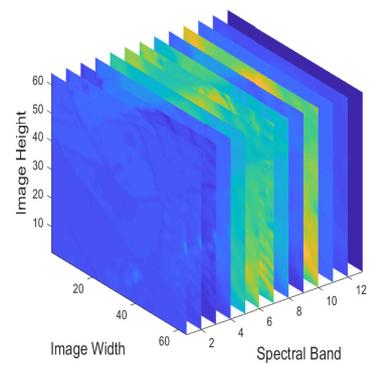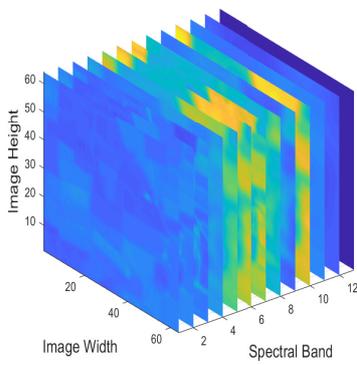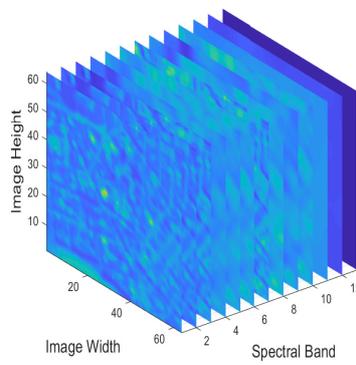**Figure 4.** Sample RGB images of all different classes included in the EUROSAT dataset.

(**a**) Annual Crop Class　　　　　(**b**) Forest Class　　　　　(**c**) Herbaceous Vegetation Class
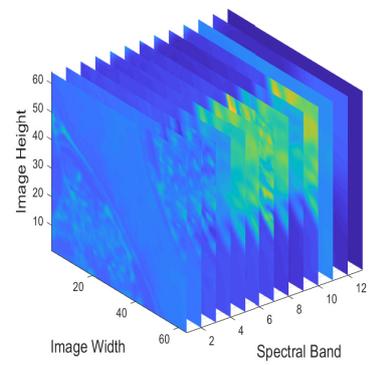
(**d**) Highway Class　　　　　(**e**) Industrial Class　　　　　(**f**) Pasture Class
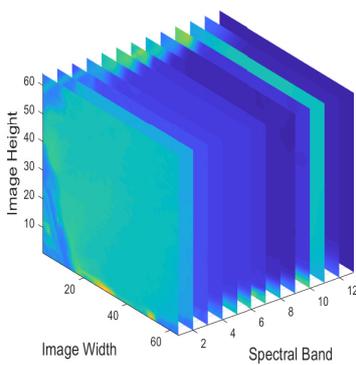
(**g**) Permanent Crop Class　　　　　(**h**) Residential Class　　　　　(**i**) River Class

(**j**) Sea-Lake Class

Figure 5. MS datacubes of sample images of all different classes included in the EUROSAT dataset.

4.2. Experimental Setup

Since the classification task lies in the heart of the present study, two non-overlapping training-test sets had to be defined a priori. In order to avoid class imbalance problems, we employed 20,000 MS sample images from the EUROSAT dataset assuming that each class consists of 2000 samples. With this setup, we split each class into 90%/5% training/test sets (i.e., 1800/100 training/test examples per class, and 18,000/1000 training/test examples in total, respectively), while at the same time another 5% of each class was used for validation purposes during the training process of the network. Training was performed using a constant number of up to 100 epochs with the intention of observing how the prediction accuracy changes by increasing the number of epochs the network is trained for.

To make more clear the above data distribution among the training, validation, and test sets, we summarize in Table 1 the aforementioned splits for each of the employed classes as well as in total.

Table 1. Dataset split among training-validation-test sets. Each class is split into 90%-5%-5% training-validation-test sets respectively, ending up with a training set of 18, 000 samples and validation & test sets of 1000 samples each.

Class Name	Available Samples	Training Set Samples	Validation Set Samples	Test Set Samples
Annual Crop	2000	1800	100	100
Forest	2000	1800	100	100
Herbaceous Vegetation	2000	1800	100	100
Highway	2000	1800	100	100
Industrial	2000	1800	100	100
Pasture	2000	1800	100	100
Permanent Crop	2000	1800	100	100
Residential	2000	1800	100	100
River	2000	1800	100	100
Sea Lake	2000	1800	100	100
Total	**20,000**	**18,000**	**1000**	**1000**

To quantify the performance of the proposed system from a ML perspective, we report *Classification Accuracy* and *Classification Loss* metrics, accompanied by the required *Computational Time* as well as the respective *Confusion Matrix*. Furthermore, *Precision* and *Recall* metrics are also computed. However, in order to examine the effect of the quantization on the classification task, we quantized the MS images of our dataset to 2, 4, 6, 8 and 10 bits, as the original MS images use 12 bits per pixel. Then, we compared the classification performance given the original, the quantized and the recovered images via the proposed recovery algorithm. Moreover, we examined the recovery of the real values of the images given the quantized measurements.

The reconstruction quality is evaluated by computing the *Peak-Signal-to-Noise-Ratio* (PSNR) between the original and the estimated MS images. Higher PSNR values correspond to a better-quality recovered image. Specifically, PSNR is computed using the expression

$$PSNR = 10 \cdot \log_{10}(\frac{R^2}{MSE}), \tag{14}$$

where *R* is the maximum value of the input image and MSE is the *Mean Square Error*, i.e., the average of the squares of the differences between the original and the estimated signal.

Since the training of a DL model is a computational expensive procedure, the employed CNN architectures were developed in Python programming platforms-exploiting the TensorFlow (https://www.tensorflow.org/) and Keras (https://keras.io/) libraries. There are two reasons behind such a choice: First, TensorFlow is an open-source ML framework, which when used as the back-end engine for the higher-level DL-specific Keras library, provides support and updates on most state-of-the-art DL models and algorithms. Secondly, and most importantly, both libraries can perform calculations on a GPU, dramatically decreasing the computational time of the training process. In our experiments, we used NVIDIA's GPU model, Quadro P4000.

On the other hand, since the compression algorithms described earlier do not need a GPU usage for their computations, they were implemented in MATLAB R2019a version.

4.3. Effect of the Training Set Size on the Classification Performance

In a first set of experiments, our objective is to assess the performance of the employed CNN relative to the size of the training set. A random selection of up to 200 training examples per class was initially considered, a number which was augmented by 200 at each step until containing all available training examples of each class (i.e., 2000 per class). The number of test examples was 100 per class (i.e., 1000 in total), as mentioned earlier.
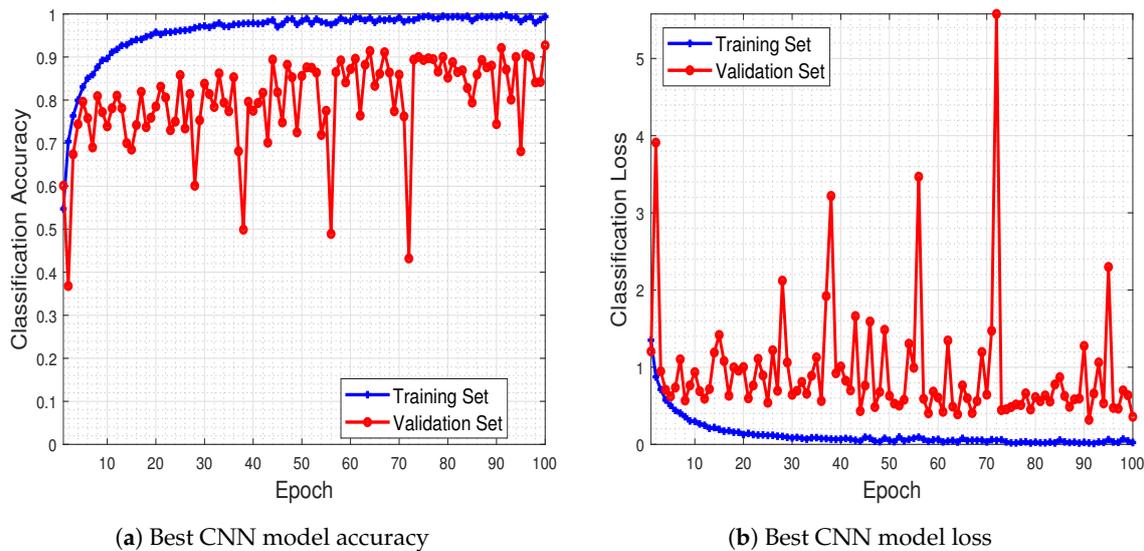
Figure 6 illustrates the performance of the CNN measured as a function of the number of the training examples. The results in Figure 6a demonstrate that increasing the number of training examples has a positive effect on the generalization capacity of the CNN, as dictated by the theoretical underpinnings. The network has a stable performance when trained with at least 8000 training examples. It should be noted that the CNN classifier achieves the best performance when trained with about 1200 image samples per class (i.e., 12,000 image samples in total), ending up with a classification accuracy equal to 92.80%.

(**a**) Classification accuracy (**b**) Computational time
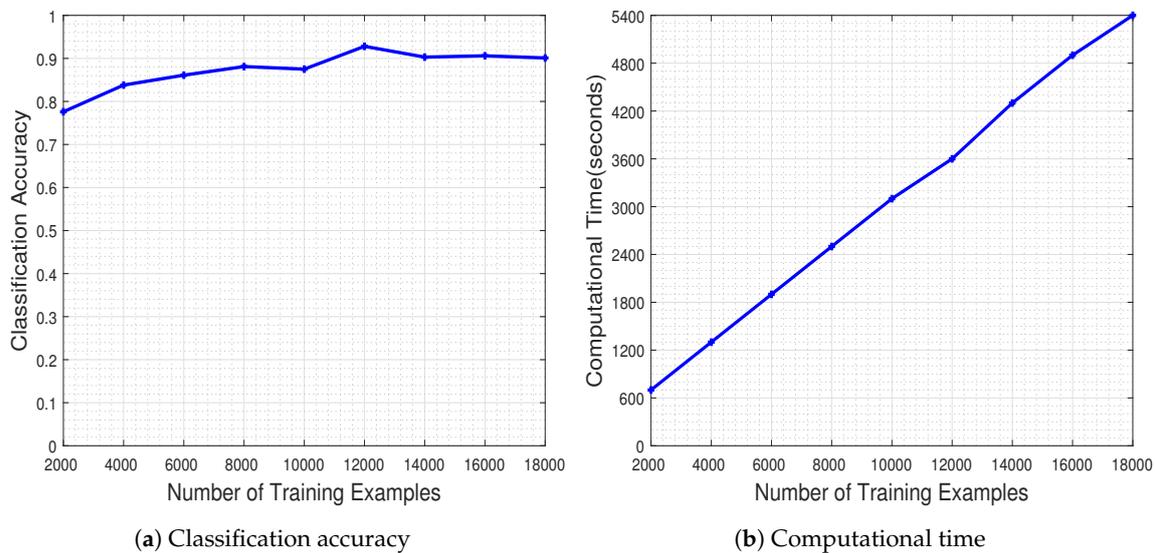
Figure 6. Classification accuracy and respective computational time regarding the number of training examples. The more training examples are used, the better the classification accuracy, translating to a more computational thirsty network.

Naturally, the improved classification performance of the CNN model does not come for free, as demonstrated by the execution time plots in Figure 6b. However, one should highlight the fact that on the one hand the time increases by a factor of 600 s as the number of training samples is doubled, while on the other hand the time needed for achieving the best performance at 12,000 training examples, namely 3600 s (i.e., 1 h), is far from being characterized as prohibitive considering typical DL experimental regimes. Overall, the results clearly demonstrate the merits and learning potential of the employed CNN modeling.

To have a better sense of the network's behavior during the training process, in Figure 7 we provide the best model accuracy as well as its loss (i.e., for 12,000 training examples), for the training and validation sets, with respect to the number of epochs up to which the network was trained.

(**a**) Best CNN model accuracy　　　　　　　　　　(**b**) Best CNN model loss

Figure 7. Best CNN model's classification accuracy and loss regarding training epochs. As the number of epochs increases, so does the performance of the CNN model, both in terms of classification accuracy as well as in classification loss.

The results of this experiment show that the trained CNN model improves its performance in terms of classification accuracy, both in the training as well as in the validation set, as the number of epochs increases. Furthermore, the classification loss in the validation set keeps track, in general, towards that of the training set, showing that the model is capable of generalizing well during the test phase (which is indeed the case, as mentioned earlier in Figure 6, with the 92.80% classification accuracy in the test set).

Useful information concerning the classification task at hand can be extracted from the confusion matrix shown in Figure 8, where we see that the trained CNN achieves quite remarkable performance in predicting correctly nearly all actual classes in the test set with a success rate of over 90%. A special notice should be given to the classes that appear to be more challenging for correct prediction, namely:

- Class *Highway* is most frequently mistaken with classes *Industrial* and *Residential*
- Class *Permanent Crop* is most frequently mistaken with classes *Herbaceous Vegetation*, *Annual Crop* and *Pasture*

Even for these difficult to classify classes in Figure 4, the trained CNN model achieves a success rate of over 80%, indicating the efficacy of the proposed approach.

4.4. Effect of the Compression Ratio on the Recovery

As the quantization process can be used for compression purposes, we examined the efficacy of our method for different compression ratios on the MS images of each class, with the results presented in Table 2. Specifically, in order to compress the data, a lossless coding algorithm, namely Huffman coding, was employed on the quantized MS images, using a dictionary obtained from the images of the training set. The corresponding *Compression Ratio* is calculated as the fraction of the number of bits of the uncompressed image over the number of bits of the compressed one, i.e.,

$$\text{Compression Ratio} = \frac{\text{Bits of uncompressed image}}{\text{Bits of compressed image}}. \tag{15}$$
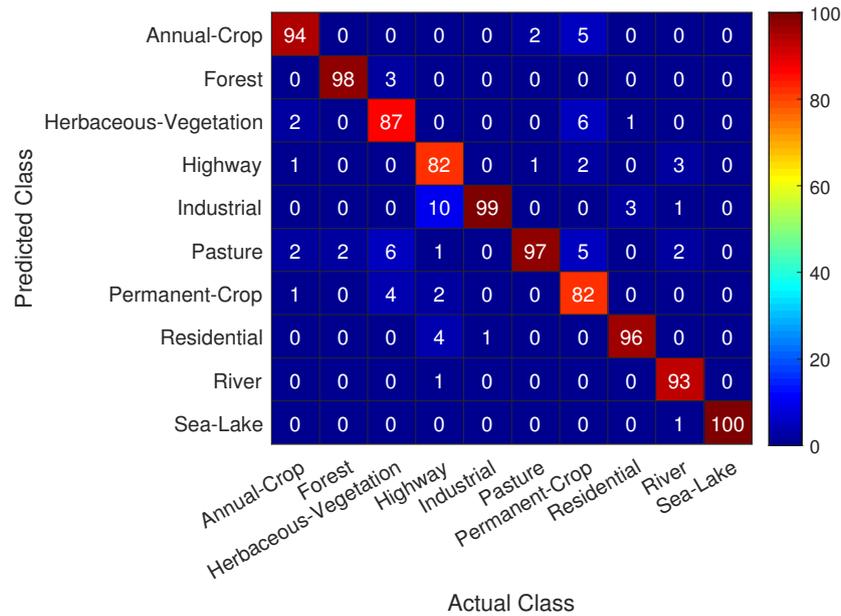
Figure 8. CNN model confusion matrix. For most classes in the test set, the trained CNN model correctly predicts the actual class with a success rate of over 90%.

As expected, the smaller the compression ratio, the better the performance of our algorithm. However, efficient recovery with a high compression ratio can still be achieved by our technique.

Table 2. Recovery error for different compression ratios on the MS images of each class, using the logistic model.

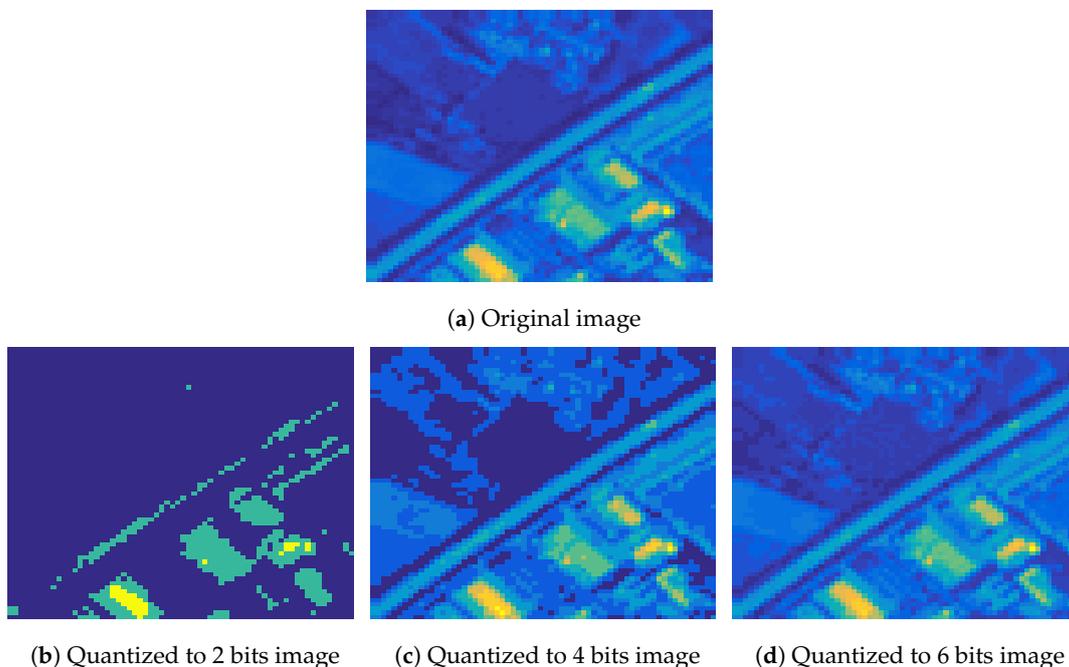| PSNR (dB)	Compression Ratio			
	×6.9	×3.4	×2.25	×1.67
Annual Crop	18.14	29.60	41.25	47.34
Forest	17.59	29.68	41.34	49.09
Herbaceous Vegetation	18.04	29.50	41.37	48.81
Highway	17.76	29.48	41.02	47.82
Industrial	17.56	29.40	40.68	47.11
Pasture	18.09	29.75	41.22	47.73
Permanent Crop	18.08	29.55	41.19	47.82
Residential	17.87	29.46	40.90	48.04
River	18.04	29.61	41.02	48.15
Sea Lake	21.74	30.84	40.62	52.80

4.5. Effect of the Number of Quantization Bits on the Recovery

An important issue that must be examined deals with the effect of the number of quantization bits. To study this, we run the proposed recovery algorithm for several quantization bit values and for various classes, using the logistic model, and we report the recovery error in Figure 9. As it was expected, a bigger number of quantization bits results to a better quality for the recovered images. Similar performance is observed for all classes. Moreover, the recovery improvement in the large number of bits region (8 and 10) is smaller compared to the improvement achieved in the small number of bits region (2 and 4), as more information is retained using more bits for the representation of the signal.
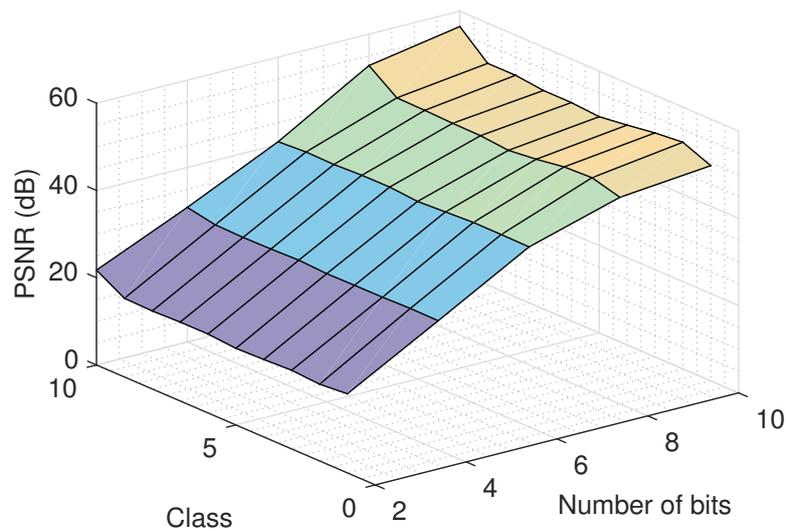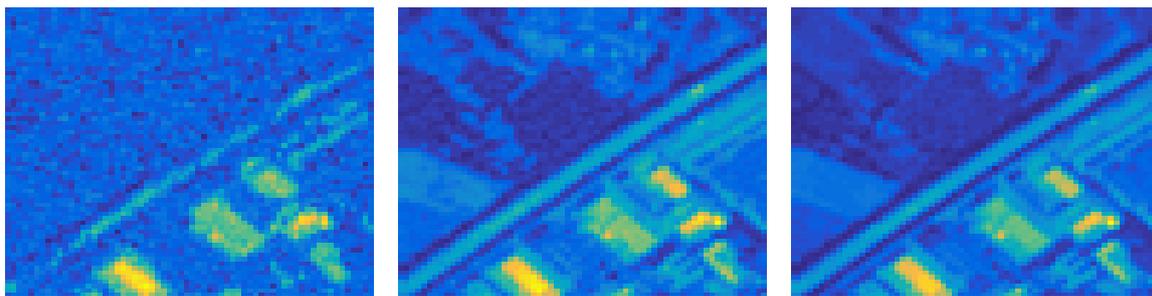
Figure 9. Recovery error as a function of the number of quantization bits and for different classes, using the logistic model.

The above observation can be visually verified in Figure 10 that depicts the second spectral band of a MS image from the *Highway* class and the corresponding quantized images using 2, 4 and 6 bits, as well the recovered images for each case, using the logistic model. As we observe, the quality of the recovered images from quantized measurements using 4 and 6 bits is better than the one using 2 bits. In addition, we can observe that using fewer bits, more information of the image is lost, which can be recovered using the proposed algorithm.

(**a**) Original image

(**b**) Quantized to 2 bits image   (**c**) Quantized to 4 bits image   (**d**) Quantized to 6 bits image

Figure 10. *Cont*.

(**e**) Recovered image from 2 bits measurements

(**f**) Recovered image from 4 bits measurements

(**g**) Recovered image from 6 bits measurements

Figure 10. The second spectral band of a MS image from the Highway class, and the corresponding quantized to 2, 4 and 6 bits images, as well the recovered images for each case, using the logistic model.

Please note that the recovery error for each class and a specific number of bits is calculated as the mean value of the error on the test images for all the experiments. In addition, we observed from our experiments that the performance of the method for the two noise models used was similar, so we kept the logistic model for the following experiments.

4.6. Effect of the Tensor Unfolding and Dynamic Weights on the Recovery

An important issue relates to the quality of the recovery for each unfolding of the tensor data and the impact of the dynamic weights on the recovery. Figure 11 depicts the recovery error as a function of the number of quantization bits for the different unfoldings of the tensor, using the logistic model and the MS images of the Annual Crop class.
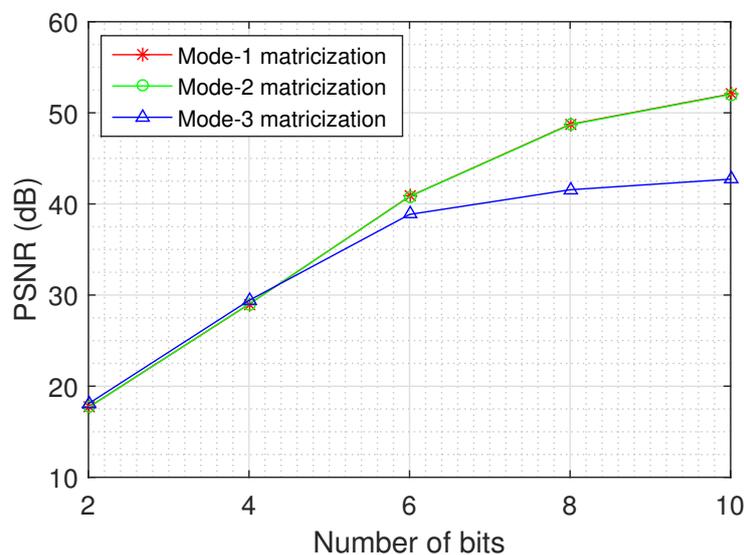

Figure 11. Recovery error as a function of the number of quantization bits on different unfoldings of the tensor, using the logistic model and the MS images of the Annual Crop class that indicate the impact of the dynamic weights on the recovery.

The results demonstrate that the dynamic weights improve the quality of the recovered tensor, as the mode-1 and mode-2 matricizations exhibit a better performance than the mode-3 matricization, especially for large values of quantization bits (8 and 10). In addition, we can deduce from the results that the recovery performance becomes better when the dimensions of the unfolding matrix are more balanced, as the mode-1 and mode-2 matricizations of each MS image have dimensions 64×832, while the mode-3 matricization has dimensions 13×4096.

4.7. Effect of the Quantization on the Classification Performance

In this set of experiments, we investigate the detrimental effect that the quantization process has in the classification performance of a system. For that purpose, the trained CNN model is evaluated to a test set that has been previously quantized to a specific number of bits, in order to obtain intuition about the system response when faced with corrupted image samples.

Figure 12 shows the performance with respect to the number of training examples and for different quantization bit values imposed on the test set image samples. The results clearly demonstrate that whatever the quantization level, the classification performance clearly suffers even if the system is trained with the maximum permitted number of training examples. Even in the favorable case of 11-bit test images (recall that the nominal bits corresponding to each MS image are 12) the network's classification performance barely reaches 40%.
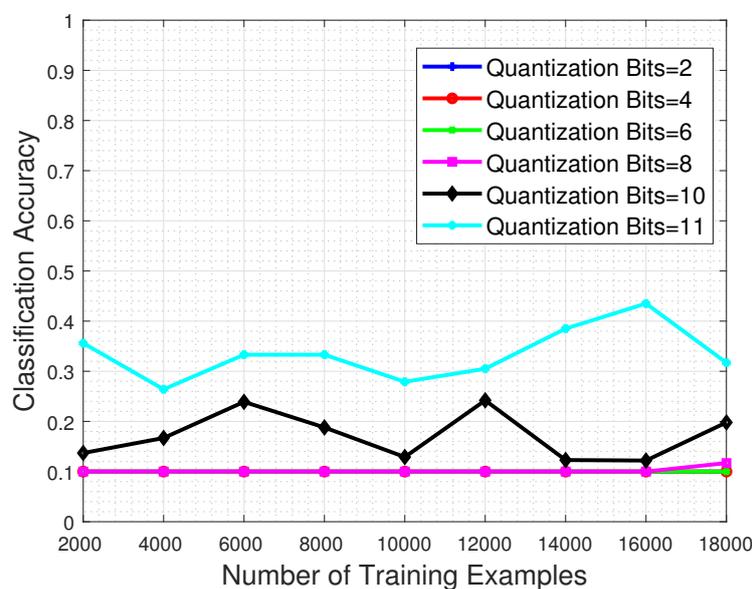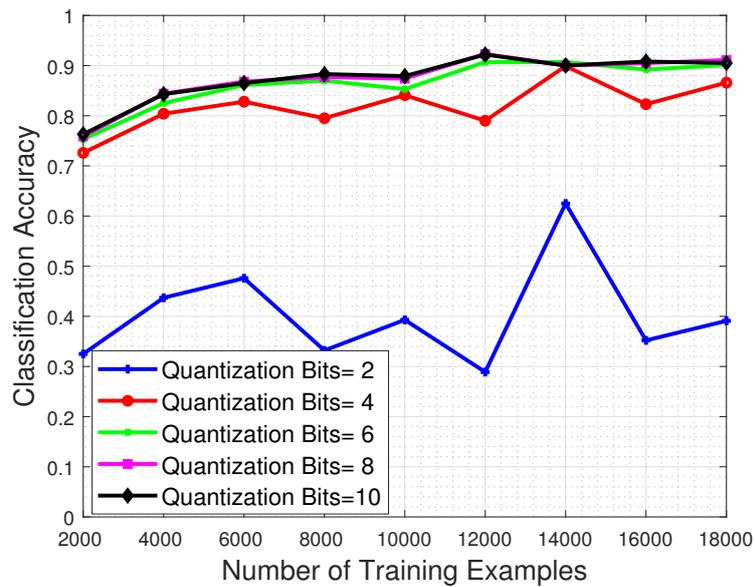

Figure 12. Classification accuracy regarding the number of training examples, for several quantization levels. The classification strength of the system clearly suffers even when the quantization process is performed to only 1 bit less than the nominal case.

4.8. Effect of the Quantization and the Recovery on the Classification Performance

To explore the ability of our CNN model to classify compression-processed MS images, we further investigate the enhancing effect that our proposed recovery process—of previously quantized MS images—has in the classification performance of a system. Towards that goal, the trained CNN model is evaluated using a recovered by our approach test set, which was previously quantized to a specific number of bits.

In Figure 13 we show the system's performance as a function of the number of training examples, and for test images quantized to 2, 4, 6, 8, and 10 bits and subsequently recovered using our proposed method. The results clearly show our recovery approach manages to boost the classification accuracy of the system, even when operating with recovered images that were previously quantized using as few as 4 bits.
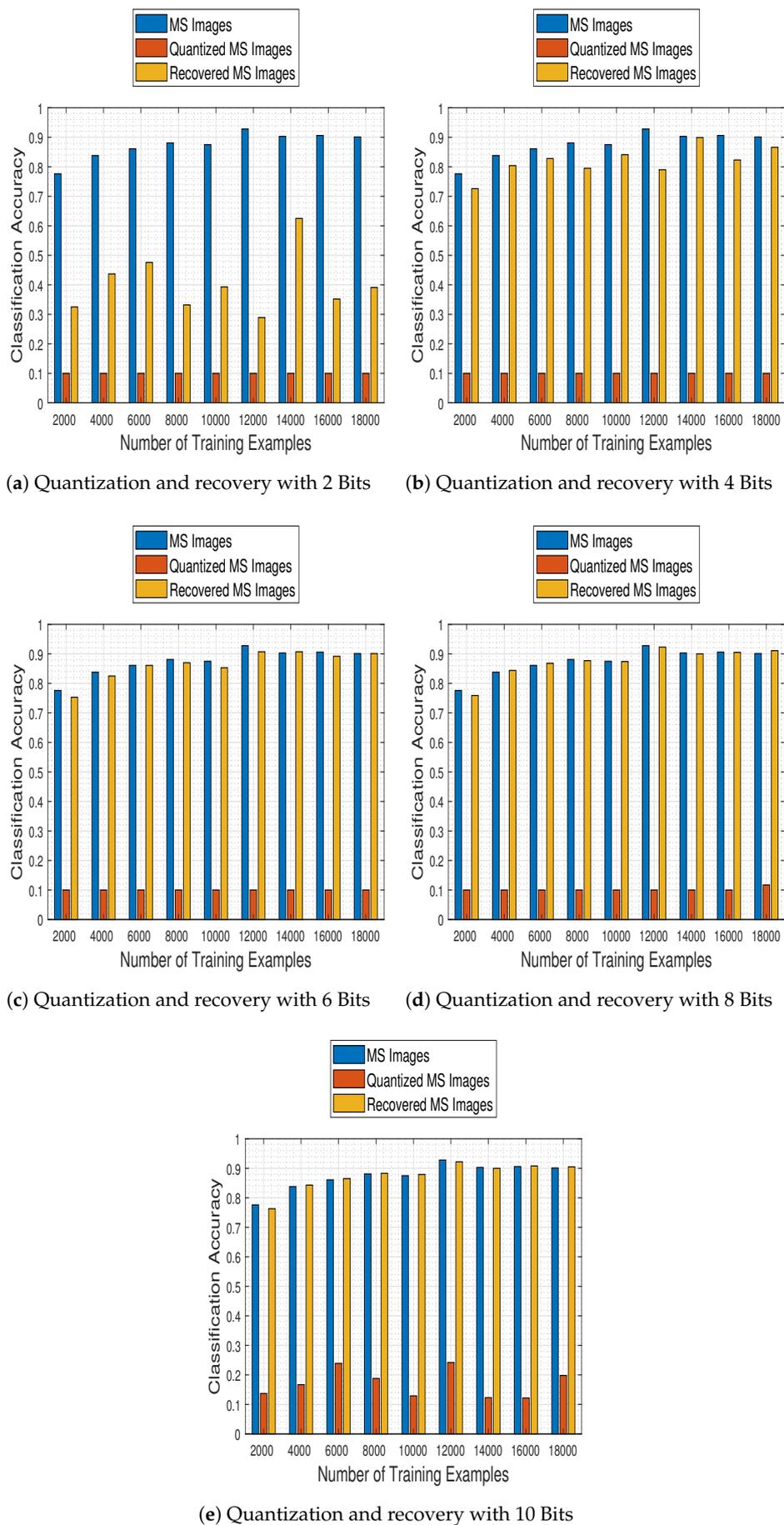
Figure 13. Classification accuracy regarding the number of training examples, for various levels of quantized images subsequently recovered using the proposed method. Our recovery approach clearly improves classification performance even when operating with images quantized with as few as 4 bits.

4.9. Joint Effects of the Quantization and the Recovery on the Classification Performance

In this set of experiments we make a general comparison of the aforementioned quantization and recovery processes, in order to quantify their effects in the MS image classification task, with respect to the "original" case where none of them is present in the classification pipeline.

Figure 14 contains classification accuracy plot bars for the employed CNN trained with 2000:2000:18,000 samples on original test MS images, their quantized versions (using 2, 4, 6, 8, and 10 bits) and their recovered by our approach estimates. The results make clear that the proposed recovery method not only remedies the effects of quantization, but most interestingly it empowers the CNN model to exhibit a classification performance on par to that obtained when working on the original MS images, nearly from a compression level when the images are quantized to half the bits of their nominal available ones (6 bits).

(**a**) Quantization and recovery with 2 Bits　　(**b**) Quantization and recovery with 4 Bits

(**c**) Quantization and recovery with 6 Bits　　(**d**) Quantization and recovery with 8 Bits

(**e**) Quantization and recovery with 10 Bits

Figure 14. General comparison of the quantization and the recovery processes on the classification performance.
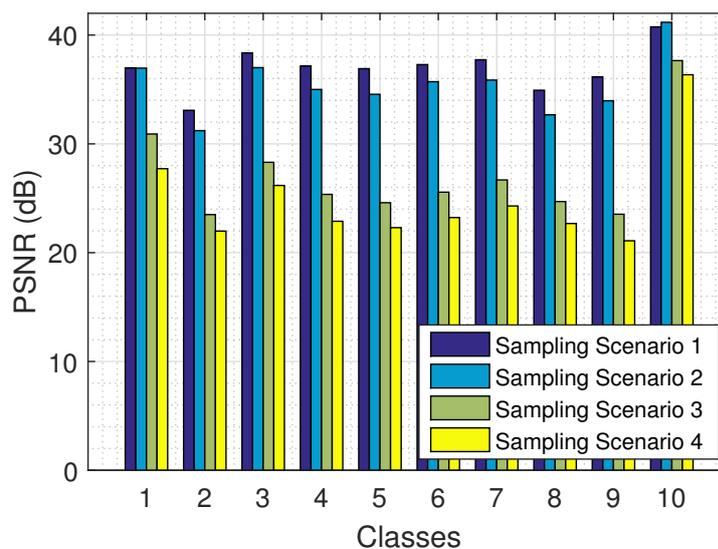
4.10. Effect of Missing Values on the Recovery

In many cases, noise or other issues in the transmission and acquisition of a signal, lead to unobserved or lost measurements. One such case could be caused by the appearance of clouds in the acquired images. Hence, the issue we address concerns missing pixel patches, within all image spectral bands, a problem more challenging than the incoherent condition usually assumed in tensor completion, where the missing elements must be randomly distributed in the tensor.

To examine the effect of missing values in the data, we randomly selected patches of each test image, imposing missing elements across all spectral bands. In particular, we considered the following sampling scenarios:

1. 10 patches of size $3{\times}3{\times}13$ pixels
2. 20 patches of size $3{\times}3{\times}13$ pixels
3. 10 patches of size $7{\times}7{\times}13$ pixels
4. 20 patches of size $7{\times}7{\times}13$ pixels

The results on the recovery of a tensor from its partial observations are presented in Figure 15 for the 4 sampling scenarios and all 10 classes, using the logistic model. As expected, the larger the amount of the observed values (e.g., in sampling scenario 1), the better the performance of our algorithm. The results are fairly consistent across all 10 classes.


Figure 15. Recovery error for the four missing value scenarios across the ten classes using the logistic model.

4.11. Effect of Missing Values and the Recovery on the Classification Performance

In this set of experiments we are concerned with the scenario where missing measurements occur in environmental sensing data, and we therein explore the beneficial impact of our proposed approach for completing higher-order structural information via tensor modeling on the ensuing classification process. In order to do so, we consider the 4 different missing values scenarios described previously, and we quantify the performance of the trained CNN model in each one of them, in an attempt to illuminate the effect of missing measurements features (e.g., nominal number, spatial size) in the system generalization capability under real-world imperfections.

Figure 16 shows the system's classification performance, in the presence of the missing values (Figure 16a) and when these missing values are recovered by means of our proposed method

(Figure 16b). The results demonstrate that the CNN's classification performance improves dramatically when our missing data recovery algorithm is employed for all four scenarios.

(**a**) Sampled MS data   (**b**) Sampled and recovered MS data
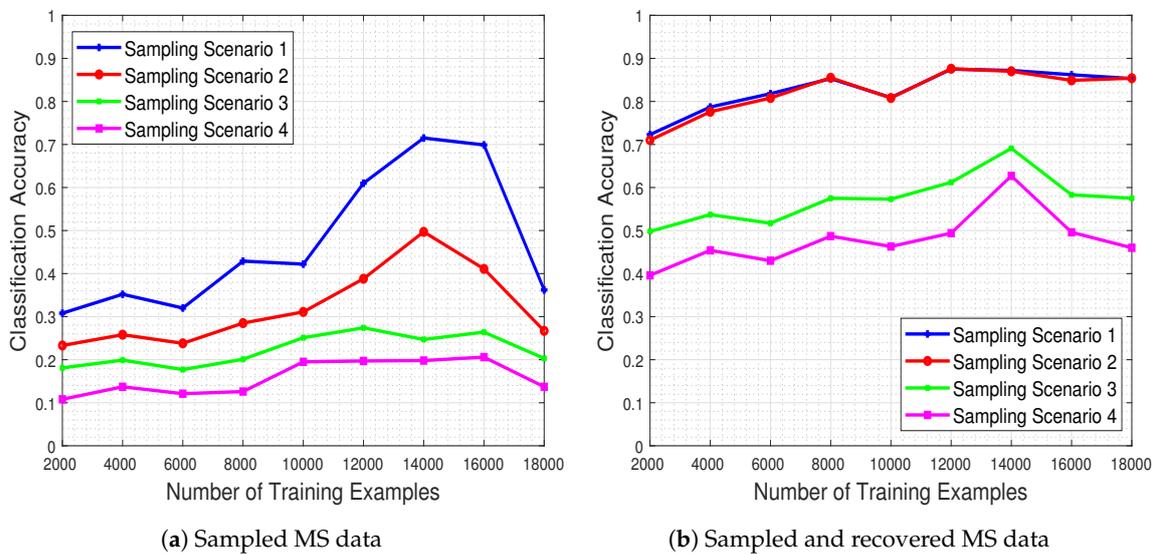
Figure 16. Classification accuracy as a function of the number of training examples for the case of images with missing values and their completed counterparts via our technique. Our recovery algorithm leads to higher levels of classification accuracy for all sampling scenarios.

### 4.12. Effect of Missing Values and Quantization on the Recovery

An interesting issue concerns the recovery of the real values of an image from its partial quantized observations. Therefore, in order to examine the effects of both missing values and quantization in the test images, we applied our method for the four missing value scenarios and for two quantization levels, namely for 8 and 11 bits. The results are presented in Figure 17, for classes *Pasture* and *Permanent Crop*, using the logistic model. As expected, as the missing values increase, the recovery error increases for both quantization levels. Of course, the more the bits of quantization, the better the performance of our recovery.
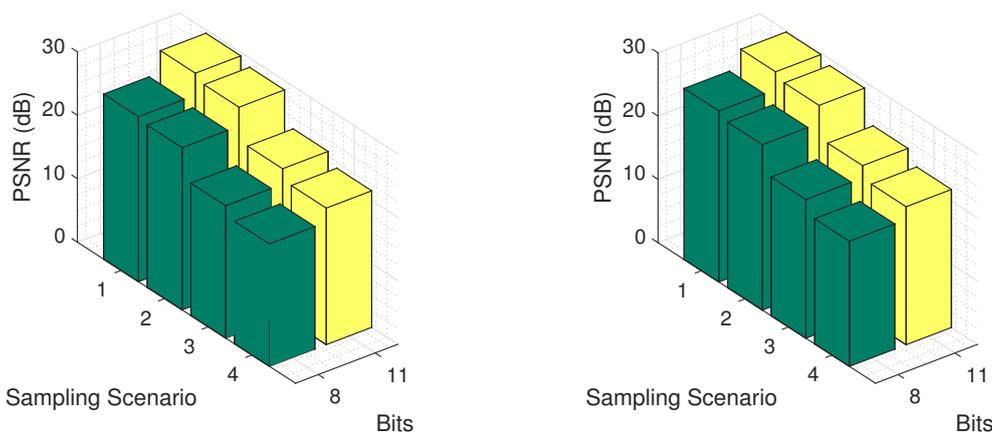
Figure 17. Recovery error for two quantization levels and the four missing value scenarios on classes Pasture (**left**) and Permanent Crop (**right**), using the logistic model.

In Figure 18 we show the second spectral band of a MS image from the *Industrial* class, and the corresponding quantized and subsampled image using 8 bits of quantization and sampling scenario 1, as well the recovered image using the logistic model. As we observe, the original MS image can

be recovered with high fidelity from partial quantized measurements using our proposed algorithm, despite that the missing elements are not randomly distributed in the image.

(**a**) Original image　　　　　　(**b**) Quantized-Subsampled image　　　　　　(**c**) Recovered image
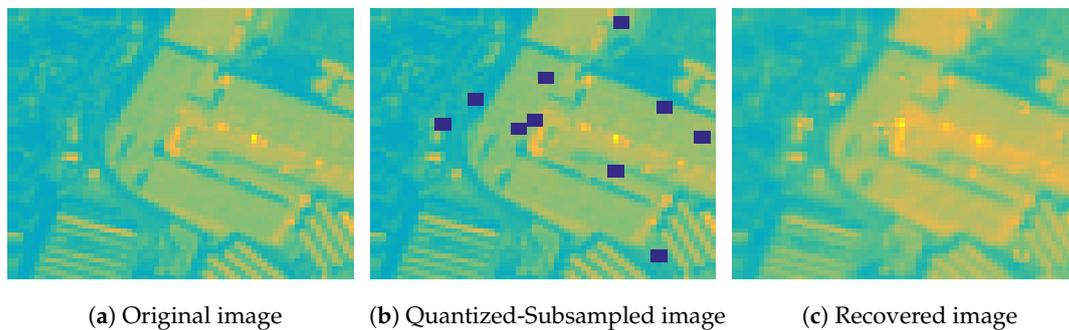
Figure 18. The second spectral band of a MS image from the Industrial class, and the corresponding quantized and subsampled image using 8 bits and sampling scenario 1, as well the recovered image, using the logistic model.

4.13. Effect of the Quantization & Missing Values on the Classification Performance

To test the system's performance in the presence of corrupted MS data (i.e., quantized and subsampled), we evaluated the trained CNN model to a test set that has been quantized to a specific number of bits and subsequently subsampled as well.

In Figure 19a,b, we depict the obtained performance as a function of the number of training examples, for two quantization levels (8 and 11 bits) and for the two most extreme missing value scenarios (i.e., scenarios 1 & 4). The results clearly show that the system's classification accuracy is severely degraded due to the quantization and missing value effects. As expected, when more bits are used for the quantization process the performance improves, while increasing the number of missing pixel patches as well as their spatial size adversely affects the classifier.

(**a**) Quantized & sampled MS data-Scenario 1　　　　　　(**b**) Quantized & sampled MS data-Scenario 4
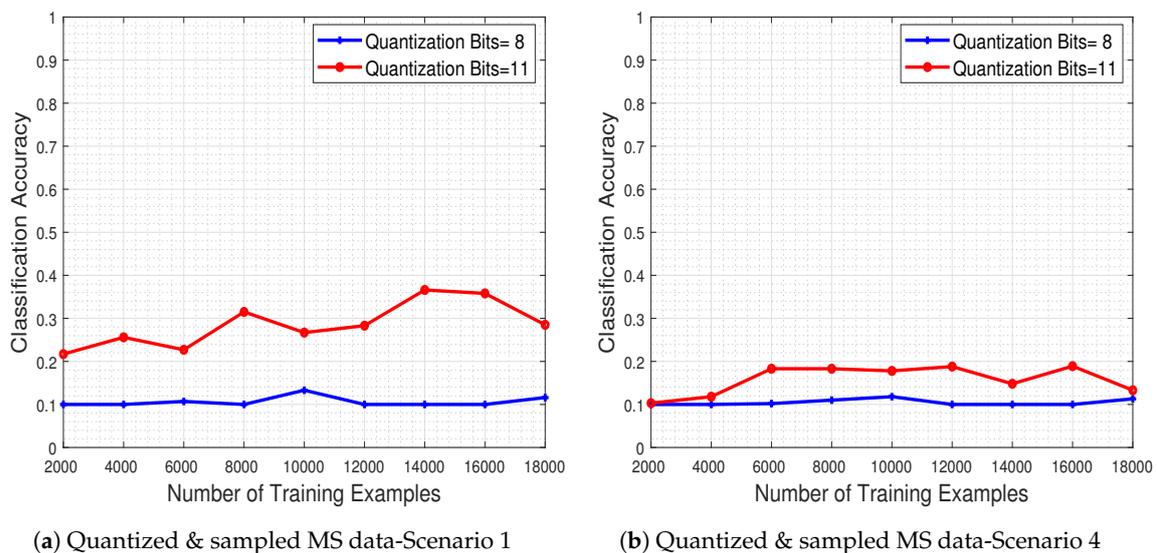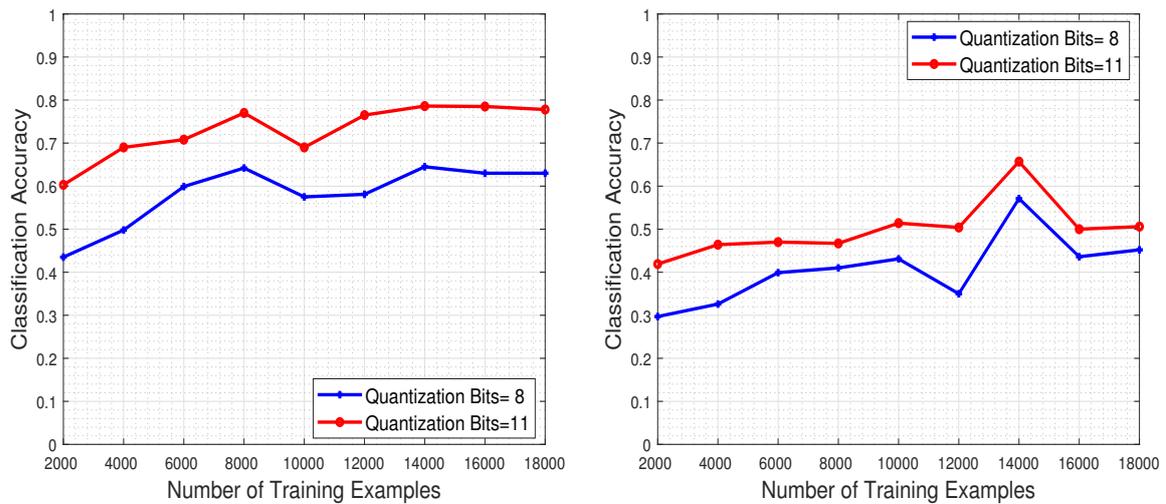
Figure 19. *Cont.*

(**c**) Quantized & sampled and recovered MS data-Scenario 1

(**d**) Quantized & sampled and recovered MS data-Scenario 4

Figure 19. Classification accuracy regarding the number of training examples for indicative number of quantization bits employed in the test set image samples and each different sampling scenario (**a,b**), recovered right after the quantization process (**c,d**). The system's performance is heavily affected by the various confronted types of signal degradation, but augmenting the number of bits used for quantization—as well employing the proposed recovery scheme—clearly ameliorates the obtained performance, even when the spatial size and the number of missing measurements is large.

4.14. Effect of the Quantization & Missing Values and the Recovery on the Classification Performance

To demonstrate the ability of the CNN system to classify noisy-infected MS images, we further investigate the effect that our recovery process has on the classification of quantized and subsampled MS images. For that purpose, the trained CNN model is evaluated to a test set that has been recovered via our approach from a quantized and subsampled original image.

Figure 19c,d depict the system's classification accuracy as a function of the number of training examples, for the recovered by our method versions of test images that were quantized in two levels (with 8 and 11 bits) and for missing value scenarios 1 and 4. The results indicate that adopting a quantization scheme with more bits in conjunction with our proposed recovery process leads to enhanced classification accuracy. Our recovery process clearly improves classification accuracy even in the extreme case where the number of missing pixel patches as well as their spatial size is large.

4.15. Joint Effects of Quantization & Missing Values and Recovery on the Classification Performance

In the final set of experiments, we conduct a general assessment of our proposed recovery process in order to quantify its effects on the MS image classification task operating on the original images and on the recovered images that were previously quantized and corrupted. Figure 20 depicts the trained CNN performance as a function of the number of image samples it was trained with, when faced with all different kinds of test images (namely original MS, quantized & sampled MS, and recovered MS images), for two quantization levels (8 and 11 bits) and for the two most extreme missing value cases (i.e., scenarios 1 & 4). The results support our claim that the proposed recovery method clearly improves the classification accuracy of the network in all cases, as compared to the CNN being applied directly to the quantized and subsampled images.
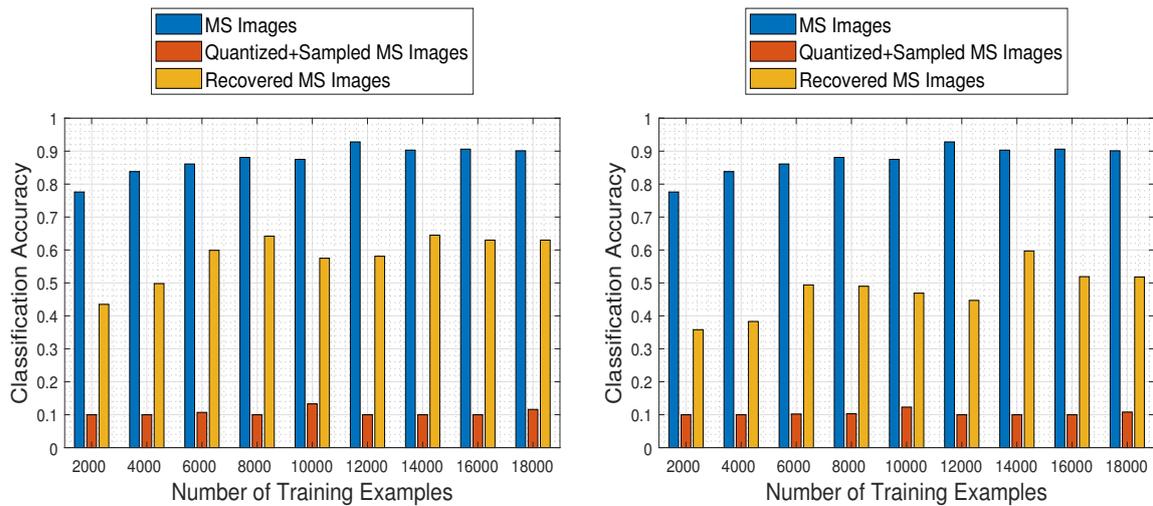
(**a**) Quantization with 8 Bits & sampling and recovery-Scenario 1

(**b**) Quantization with 8 Bits & sampling and recovery-Scenario 4
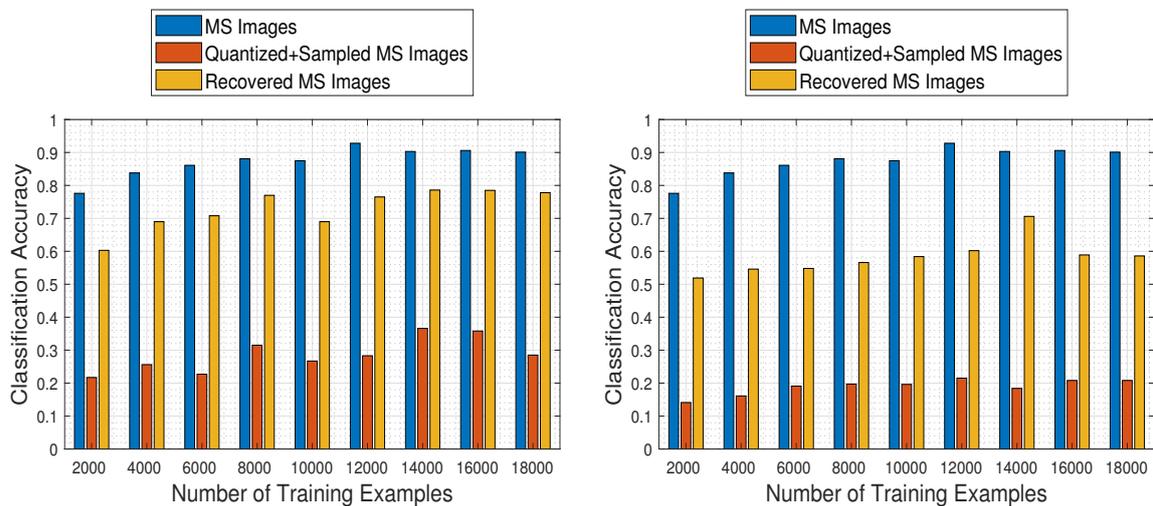
(**c**) Quantization with 11 Bits & sampling and recovery-Scenario 1

(**d**) Quantization with 11 Bits & sampling and recovery-Scenario 4

Figure 20. General comparison of quantization with 8–11 Bits &, missing values, and recovery processes on the classification performance.

4.16. Comparison of the Proposed Scheme with Existing Methods

In this section, we compare our approach with other methods which examine the impact of compression on the classification task [45–47]. Existing methods apply JPEG2000 on each spectral band, or JPEG2000 followed by DWT or PCA for spectral decorrelation to compress the data. Subsequently, the decompressed-recovered images are fed to standard k-NN or SVM classifiers to perform the supervised ML task. In our system, we compress the MS images by quantization, and we classify the recovered MS images obtained by the proposed recovery algorithm using the pre-trained ResNet-50 model described above.

At the compression stage of our method, we compared the recovered MS images via the proposed tensor recovery algorithm with the decompressed images via JPEG2000 on each spectral band and via JPEG2000 followed by PCA for spectral decorrelation. The recovery error as a function of *bits per pixel per band* (bpppb) for the class *Forest* presented in Figure 21a, indicates the efficacy of our method since

the PSNR increases as the bpppb value gets higher in contrast to the other compression methods that exhibit a fairly constant PSNR behavior even for large values of bpppb.
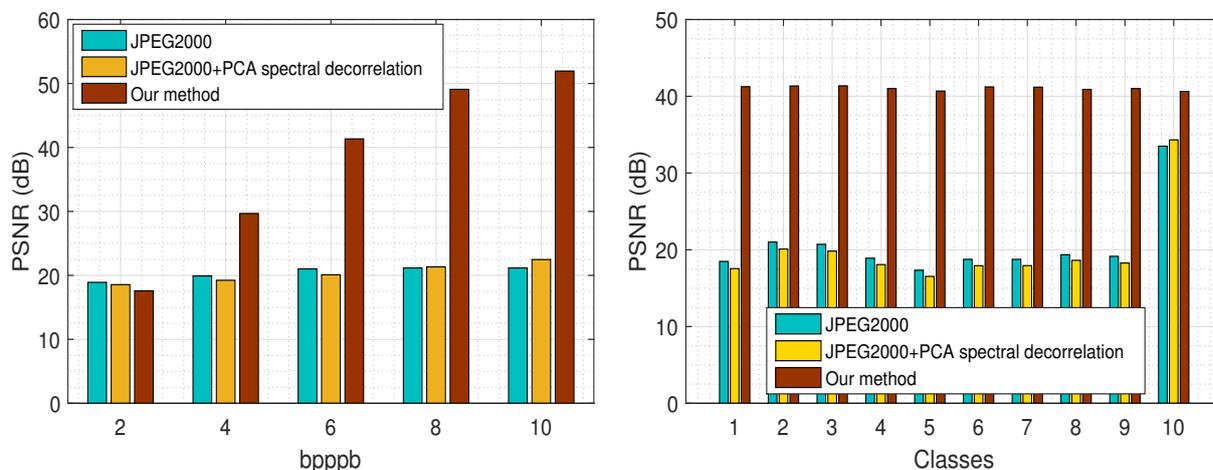
(**a**) Reconstruction quality for different bpppb for the class *Forest*.

(**b**) Reconstruction quality for different classes using 6 bpppb.

Figure 21. Comparison of our approach with existing compression methods.

In addition, we can observe from the results that our approach outperforms existing compression methods, especially for large values of bpppb, as it is demonstrated in Figure 21b for the HS images of each class, using 6 bpppb.

The classification stage is implemented in two levels: one operating on the original MS images and one operating on the recovered MS images. In order to perform a comparison between the adopted ResNet-50 model architecture at both levels, a competing ML model must be selected at first.

Although existing methods studying the compression impact on classification [45–47] adopt specific classifiers such as k-NN or SVM, the examined problem therein is of a different nature: pixel-level classification of a MS image (e.g., for object detection purposes) in contrast to scene classification in our case (i.e., the whole image is classified). Furthermore, following the traditional ML pipeline, feature extraction and feature selection stages must take place before the classification stage. If this is not the case, and one simply treats the MS imagery at hand as "long" feature vectors (of pixels) to be fed to the classifier (e.g., an SVM), the whole task is automatically put in jeopardy mainly for two reasons: possible correlations among the data are not exploited as long as no descriptive features are extracted from them, while the computational burden of such a process may be intractable (these classifiers do not perform computations on GPUs as DL-based CNNs).

Based on the aforementioned reasoning, we did not compare the adopted network with standard ML classifiers, but rather with a CNN model trained from scratch. Apart from the ease of GPU computations, adopting such an approach passes the feature extraction stage directly to the model through its training process.

Concerning the architecture of the competing CNN, we adapt a solution involving 3D-CNNs (for tackling human action recognition [62] and video quality assessment [63] problems) to their 2D counterparts for our task at hand. More specifically, the network architecture comprises multiple stacks of convolutional, max-pooling, normalization, and activation (i.e., ReLU) layers. The number of stacks of layers employed was 5, where each convolutional layer is always followed by a max-pooling layer, a batch-normalization layer and an activation layer in that order. These layers are then followed by 2 fully connected layers, and a final soft-max layer for the prediction task.

Based on the findings of the aforementioned reference papers (as well as on our own cross-validation results) we employed small receptive fields (e.g., 3 × 3) for the convolutional filters, rather than larger ones (e.g., 5 × 5). As far as the number of filters is concerned, we adopted a

"doubling-depth" strategy where the filters are doubled from a previous layer to the next one, starting from 32 filters in the first layer and ending up with 512 in the last layer in that way. All max-pooling layers have a size of 2×2, which corresponds to reducing the input data by a factor of 4, while the fully connected layers have 512 and 256 units respectively. For fairness of comparison, the Adagrad optimizer was selected along with the categorical cross-entropy as the loss function.

Being aligned with the results presented in Figure 6, in Figure 22 we depict the respective results obtained by the "*comparison-CNN*" model described above.

(**a**) Classification accuracy   (**b**) Computational time
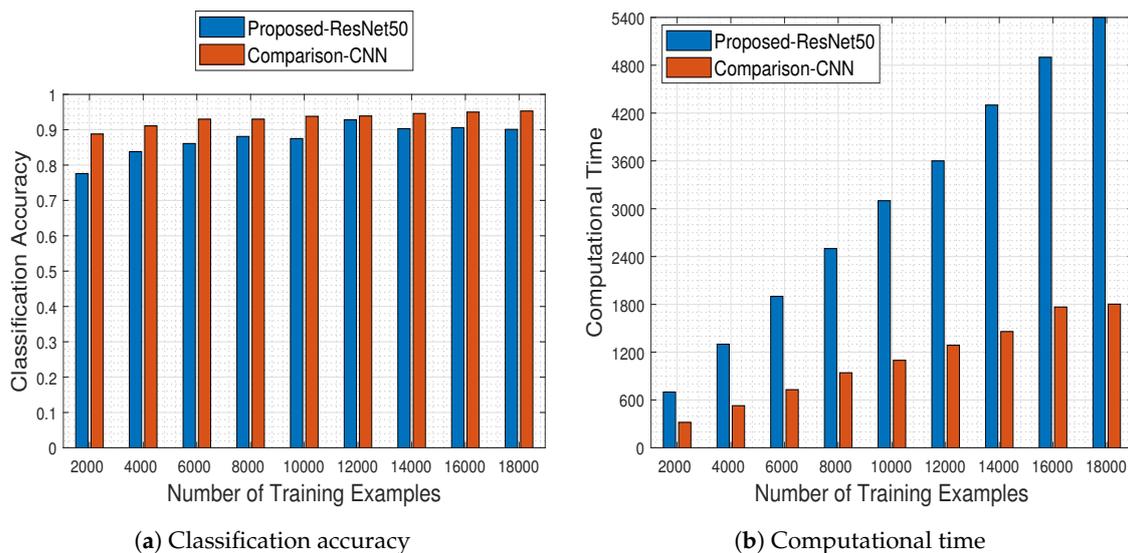
Figure 22. Classification accuracy and respective computational time regarding the number of training examples, for competing classification models. The "lighter" comparison-CNN model outperforms the pre-trained ResNet-50 model by an accuracy margin of up to 2.5% in the best case, being faster at the same time.

As can be seen in Figure 22a,b, the comparison-CNN model outperforms the pre-trained ResNet-50 model for all cases of training example sizes, a fact that can be attributed mainly to 2 reasons:
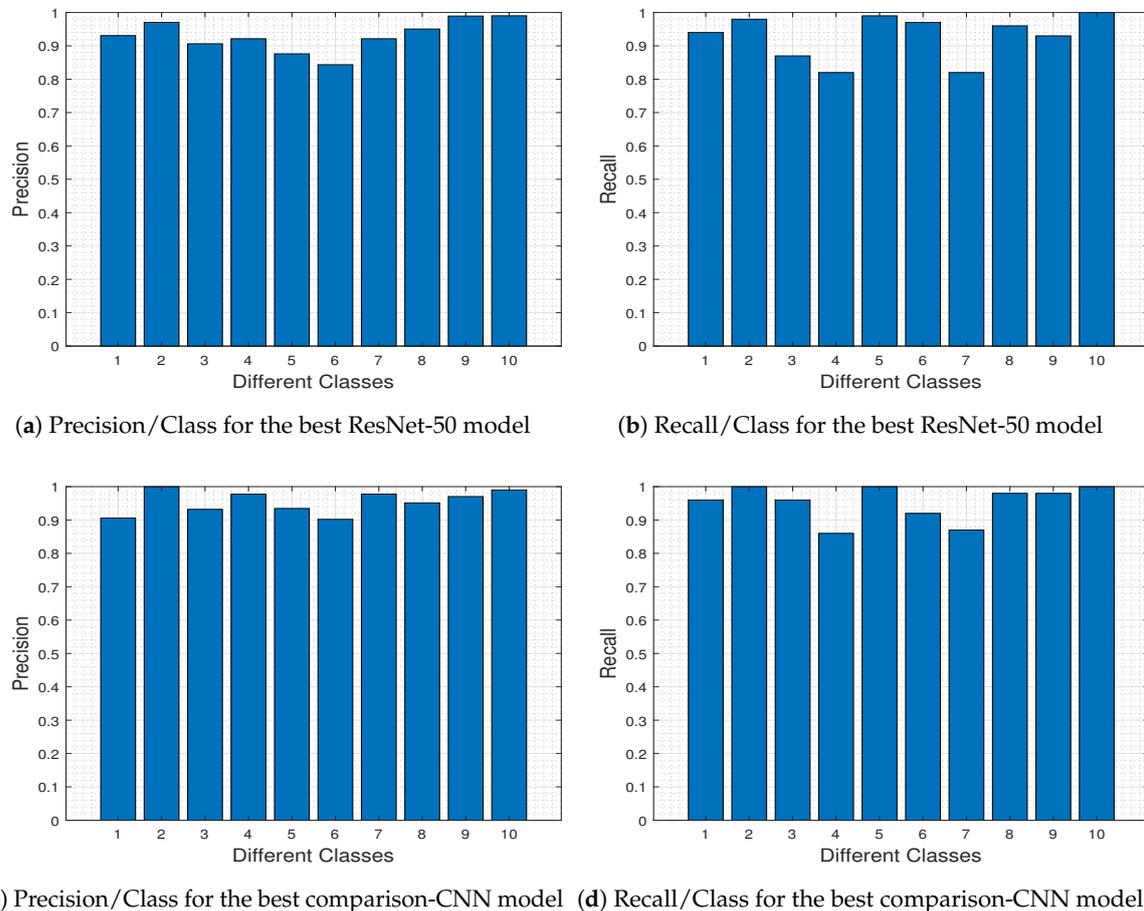
- The pre-trained ResNet-50 model was originally designed for RGB image recognition (and not for MS one), where its success rate was over 98.5% (i.e., in the EUROSAT RGB dataset). Adapting it appropriately led to a logical performance drop to 92.8%, which was to be expected since the problem is of a *similar* flavor and not exactly the same. In contrast, the comparison-CNN reaches a success rate of up to 95.3%, indicating that models designed for video processing purposes can scale back well to image processing tasks.
- The comparison-CNN model is quite faster than the pre-trained ResNet-50 one, a fact which can be attributed to the number of trainable parameters of each architecture. To that end, in Table 3 we present the parameters that must be learned by each model.

Table 3. Number of trainable parameters of the competing DL models. A more complex network architecture clearly leads to a more computational thirsty model.

Network Name	Number of Trainable Parameters
ResNet-50	23,586,442
Comparison-CNN	1,969,994

Of course, as long as the comparison-CNN model was approximately 12 times "lighter" than the ResNet-50 one, the computational time needed for its training was expected to be less, as shown in Figure 22b.
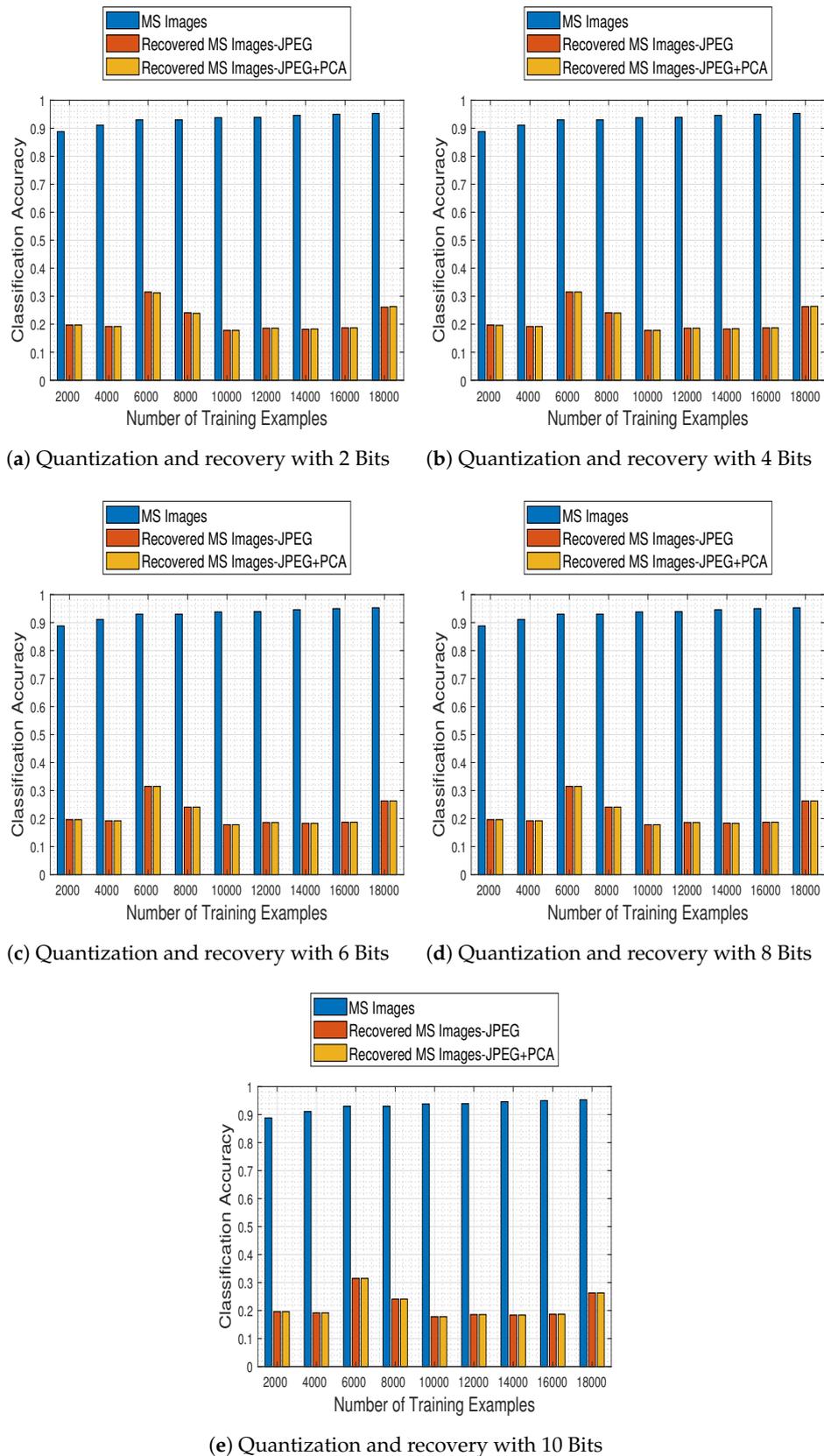
To further investigate the specific differences between the two competing DL models, in Figure 23 we depict the precision-recall metrics per class obtained by the ResNet-50 model (Figure 23a,b) and the comparison-CNN model (Figure 23c,d).

(**a**) Precision/Class for the best ResNet-50 model   (**b**) Recall/Class for the best ResNet-50 model

(**c**) Precision/Class for the best comparison-CNN model   (**d**) Recall/Class for the best comparison-CNN model

Figure 23. Precision and recall plots per each different class for the pre-trained ResNet-50 (**top**) and comparison-CNN (**bottom**).

In ML and *Receiver Operating Characteristics* (ROC) analysis [64], *Recall/Sensitivity/True Positive Rate* refers to the *proportion of Real Positive cases that are correctly Predicted Positive*. In contrast, *Precision/Confidence/Positive Predicted Value* denotes the *proportion of Predicted Positive cases that are correctly Real Positives*. Comparing Figure 23a–c and 23b–d, we 0 outperforms ResNet-50 for classes 3, 4, and 6 (i.e., Herbaceous Vegetation, Highway and Permanent Crop, respectively). These classes are the most difficult to predict using the pre-trained ResNet-50 model (as dictated from Figure 8, and discussed therein), and this difficulty explains the fact that the comparison-CNN exhibits an increased accuracy.

Although the comparison results described above concern the first level of the classification stage, the most important ones concern the second level, in which the trained model is used in conjunction with the compression stage method adopted. To that end, we selected the best model of the comparison-CNN described above (i.e., the one obtained when trained with 18,000 training examples and reaching 95.3% accuracy) and investigated the effect of compression and recovery via the two schemes mentioned earlier: JPEG2000 and JPEG2000 followed by PCA across the spectral dimension. In Figure 24 we depict the obtained results for both methods (JPEG stands for JPEG2000), in contrast to those obtained for the compression-free situation portrayed earlier.

(**a**) Quantization and recovery with 2 Bits

(**b**) Quantization and recovery with 4 Bits

(**c**) Quantization and recovery with 6 Bits

(**d**) Quantization and recovery with 8 Bits

(**e**) Quantization and recovery with 10 Bits

Figure 24. Comparison of the quantization and the recovery processes on the classification performance, when competing quantization schemes (JPEG, JPEG+PCA) are adopted.

The results demonstrate that the alternative quantization schemes have a devastating effect in the system's performance, which—irrespective to the number of bits to which the images are quantized to—remains nearly fixed to a maximum accuracy of up to 31.5%.

Based on that remark, we subsequently performed the final and most crucial system-to-system comparison, between our proposed one (i.e., ResNet-50 model and tensor recovery) and the competing ones (comparison-CNN and JPEG200 & JPEG200+PCA).

Figure 25 contains classification accuracy results obtained by our system as well as the competing ones, with respect to the number of training examples with which the classification schemes were trained with and the number of bits to which the compression schemes quantized the MS images.
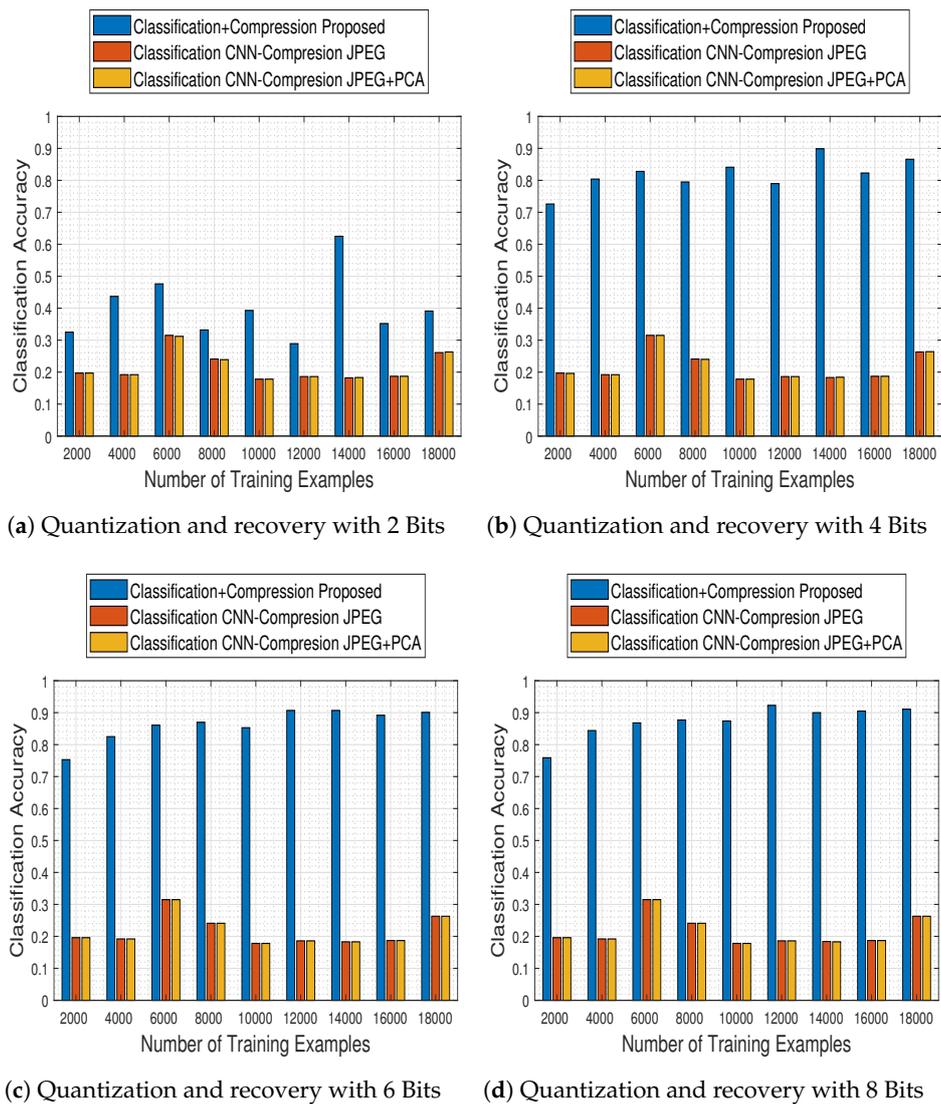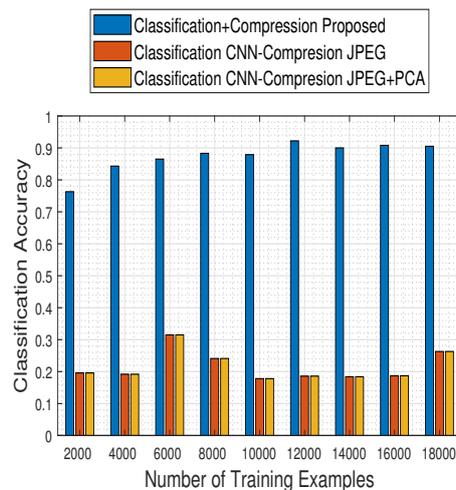

(**a**) Quantization and recovery with 2 Bits   (**b**) Quantization and recovery with 4 Bits

(**c**) Quantization and recovery with 6 Bits   (**d**) Quantization and recovery with 8 Bits

Figure 25. *Cont.*

(**e**) Quantization and recovery with 10 Bits

Figure 25. General system comparison of the quantization-recovery-classification processes. The proposed system clearly outperforms both competing ones in every examined quantization bits scenario.

The results clearly imply that the proposed system not only outperforms the competing ones in every examined case, but its performance steadily ameliorates as well. In contrast, its competitors at the same time obtain poor performance results in every case, indicating in that way the efficacy of the proposed approach.

5. Conclusions

In this work, we proposed a DL architecture for the problem of classifying MS image data. The approach is based on residual networks, successfully used for the respective RGB task for learning efficient spatial feature representations. Furthermore, we designed, tested, and evaluated with actual data two novel techniques to address the real-life scenarios of dealing with quantized data and missing values in the images. A constrained maximum likelihood estimation on the tensor unfoldings is introduced for the recovery of the real values, taking into account the quantization bin boundaries and the existence of possible noise. Based on our experimental findings on a recently released—sufficiently large—MS image dataset, we demonstrated that improved classification accuracy is feasible even in the presence of only half of available data observations using state-of-the-art CNNs. Clearly, adopting a tensor recovery algorithm proved to be a good strategy for efficient classification of previously quantized, as well as subsampled MS data demonstrating the efficacy of the proposed approach.

Abbreviations

The following abbreviations are used in this manuscript:

RS	Remote Sensing
MS	Multispectral
HS	Hyperspectral
DWT	Discrete Wavelet Transform
ML	Machine Learning
SVM	Support Vector Machine
RF	Random Forest
DL	Deep Learning
CNN	Convolutional Neural Network
LEOP	Launch and Early Orbit Phase
k-NN	k-Nearest Neighbors
PCA	Principal Component Analysis
NN	Neural Network
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
SGD	Stochastic Gradient Descent
CDF	Cumulative Distribution Function
SVD	Singular Value Decomposition
PSNR	Peak-Signal-to-Noise-Ratio
MSE	Mean Square Error
bpppb	bits per pixel per band
ROC	Receiver Operating Characteristics

References

1. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.; Chanussot, J. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36.

2. Sudmanns, M.; Tiede, D.; Lang, S.; Bergstedt, H.; Trost, G.; Augustin, H.; Baraldi, A.; Blaschke, T. Big Earth data: Disruptive changes in Earth observation data management and analysis? *Int. J. Digit. Earth* **2019**, 1–19, doi:10.1080/17538947.2019.1585976.

3. Huang, B. *Satellite Data Compression*; Springer Science & Business Media: Berlin, Germany, 2011 .

4. Zhou, S.; Deng, C.; Zhao, B.; Xia, Y.; Li, Q.; Chen, Z. Remote sensing image compression: A review. In Proceedings of the 2015 IEEE International Conference on Multimedia Big Data, Beijing, China, 20–22 April 2015; pp. 406–410.

5. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 247–259.

6. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31.

7. Stivaktakis, R.; Tsagkatakis, G.; Tsakalides, P. Deep Learning for Multilabel Land Cover Scene Categorization Using Data Augmentation. *IEEE Geosci. Remote. Sens. Lett.* **2019**, *16*, 1031–1035.

8. Castelluccio, M.; Poggi, G.; Sansone, C.; Verdoliva, L. Land use classification in remote sensing images by convolutional neural networks. *arXiv* **2015**, arXiv:1508.00092.

9. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498.

10. Zhou, W.; Newsam, S.; Li, C.; Shao, Z. Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval. *Remote Sens.* **2017**, *9*, 489.

11. Theodoridis, S.; Chatzis, S. *A Tour to Deep Learning: From the Origins to Cutting Edge Research and Open Challenges*; IEEE Signal Procesing Society: Piscataway, NJ, USA, 2018 .

12. Zhu, Z.; Qi, G.; Chai, Y.; Li, P. A geometric dictionary learning based approach for fluorescence spectroscopy image fusion. *Appl. Sci.* **2017**, *7*, 161.

13. Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28.

14. Pal, M. Random forest classifier for remote sensing classification. *Int. J. Remote Sens.* **2005**, *26*, 217–222.

15. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790.

16. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40.

17. Tsagkatakis, G.; Aidini, A.; Fotiadou, K.; Giannopoulos, M.; Pentari, A.; Tsakalides, P. Survey of Deep-Learning Approaches for Remote Sensing Observation Enhancement. *Sensors* **2019**, *19*, 3929.

18. Sharma, A.; Liu, X.; Yang, X.; Shi, D. A patch-based convolutional neural network for remote sensing image classification. *Neural Netw.* **2017**, *95*, 19–28.

19. Zhao, W.; Du, S. Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 155–165.

20. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 645–657.

21. Marcos, D.; Volpi, M.; Kellenberger, B.; Tuia, D. Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 96–107.

22. Fotiadou, K.; Tsagkatakis, G.; Tsakalides, P. Deep convolutional neural networks for the classification of snapshot mosaic hyperspectral imagery. *Electron. Imaging* **2017**, *2017*, 185–190.

23. Hamida, A.B.; Benoit, A.; Lambert, P.; Amar, C.B. 3-D Deep learning approach for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4420–4434.

24. Lyu, H.; Lu, H.; Mou, L.; Li, W.; Wright, J.; Li, X.; Li, X.; Zhu, X.; Wang, J.; Yu, L.; et al. Long-term annual mapping of four cities on different continents by applying a deep information learning method to Landsat data. *Remote Sens.* **2018**, *10*, 471.

25. Shin, H.C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med Imaging* **2016**, *35*, 1285–1298.

26. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 166–177.

27. Davenport, M.A.; Plan, Y.; Van Den Berg, E.; Wootters, M. 1-bit matrix completion. *Inf. Inference: A J. IMA* **2014**, *3*, 189–223.

28. Cai, T.; Zhou, W.X. A max-norm constrained minimization approach to 1-bit matrix completion. *J. Mach. Learn. Res.* **2013**, *14*, 3619–3647.

29. Bhaskar, S.A.; Javanmard, A. 1-bit matrix completion under exact low-rank constraint. In Proceedings of the 2015 49th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 18–20 March 2015; pp. 1–6.

30. Lan, A.S.; Studer, C.; Baraniuk, R.G. Matrix recovery from quantized and corrupted measurements. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 4973–4977.

31. Lan, A.S.; Studer, C.; Baraniuk, R.G. Quantized matrix completion for personalized learning. *arXiv* **2014**, arXiv:1412.5968.

32. Lafond, J.; Klopp, O.; Moulines, E.; Salmon, J. Probabilistic low-rank matrix completion on finite alphabets. In *Advances in Neural Information Processing Systems, Proceedings of the Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014*; Neural Information Processing Systems Foundation, Inc.: La Jolla, CA, USA , 2014; pp. 1727–1735.

33. Bhaskar, S.A. Probabilistic low-rank matrix completion from quantized measurements. *J. Mach. Learn. Res.* **2016**, *17*, 2131–2164.

34. Signoretto, M.; Van de Plas, R.; De Moor, B.; Suykens, J.A. Tensor versus matrix completion: A comparison with application to spectral data. *IEEE Signal Process. Lett.* **2011**, *18*, 403–406.

35. Liu, J.; Musialski, P.; Wonka, P.; Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 208–220.

36. Giannopoulos, M.; Savvaki, S.; Tsagkatakis, G.; Tsakalides, P. Application of Tensor and Matrix Completion on Environmental Sensing Data. In *ESANN 2017, Bruges, Belgium, 26–28 April 2017*; ESANN: Bruges, Belgium, 2017.

37. Giannopoulos, M.; Tsagkatakis, G.; Tsakalides, P. On the impact of Tensor Completion in the Classification of Undersampled Hyperspectral Imagery. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 1975–1979.

38. Aidini, A.; Tsagkatakis, G.; Tsakalides, P. 1-bit tensor completion. *Electron. Imaging* **2018**, *2018*, 261-1–261-6.

39. Li, B.; Zhang, X.; Li, X.; Lu, H. Tensor completion from one-bit observations. *IEEE Trans. Image Process.* **2018**, *28*, 170–180.

40. Ghadermarzy, N.; Plan, Y.; Yilmaz, O. Learning tensors from partial binary measurements. *IEEE Trans. Signal Process.* **2018**, *67*, 29–40.

41. Tsagkatakis, G.; Amoruso, L.; Sykas, D.; Abbattista, C.; Tsakalides, P. Lightweight Onboard Hyperspectral Compression and Recovery by Matrix Completion. In Proceedings of the 5th International Workshop on On-Board Payload Data Compression (OBPDC 2016), Frascati, Italy, 28–29 September 2016.

42. Li, N.; Li, B. Tensor completion for on-board compression of hyperspectral images. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 517–520.

43. Zhang, L.; Zhang, L.; Tao, D.; Huang, X.; Du, B. Compression of hyperspectral remote sensing images by tensor approach. *Neurocomputing* **2015**, *147*, 358–363.

44. Fang, L.; He, N.; Lin, H. CP tensor-based compression of hyperspectral images. *JOSA A* **2017**, *34*, 252–258.

45. Marsetic, A.; Kokalj, Z.; Ostir, K. The effect of lossy image compression on object based image classification-WorldView-2 case study. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *3819*, 187–192.

46. Garcia-Vilchez, F.; Muñoz-Marí, J.; Zortea, M.; Blanes, I.; González-Ruiz, V.; Camps-Valls, G.; Plaza, A.; Serra-Sagristà, J. On the impact of lossy compression on hyperspectral image classification and unmixing. *IEEE Geosci. Remote Sens. Lett.* **2010**, *8*, 253–257.

47. Chen, Z.; Hu, Y.; Zhang, Y. Effects of Compression on Remote Sensing Image Classification Based on Fractal Analysis. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 4577–4590.

48. Gimona, A.; Poggio, L.; Aalders, I.; Aitkenhead, M. The effect of image compression on synthetic PROBA-V images. *Int. J. Remote Sens.* **2014**, *35*, 2639–2653.

49. Hagag, A.; Fan, X.; El-Samie, F.E.A. The effect of lossy compression on feature extraction applied to satellite Landsat ETM+ images. In Proceedings of the Eighth International Conference on Digital Image Processing (ICDIP 2016), Chengu, China, 20–22 May 2016.

50. Theodoridis, S.; Pikrakis, A.; Koutroumbas, K.; Cavouras, D. *Introduction to Pattern Recognition: A Matlab Approach*; Academic Press: Cambridge, MA, USA, 2010.

51. Theodoridis, S. *Machine Learning: A Bayesian and Optimization Perspective*; Academic Press: Cambridge, MA, USA, 2015.

52. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; NIPS: Lake Tahoe, NV, USA, 2012 ; pp. 1097–1105.

53. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.

54. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252.

55. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.

56. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.

57. Cichocki, A.; Mandic, D.; De Lathauwer, L.; Zhou, G.; Zhao, Q.; Caiafa, C.; Phan, H.A. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.* **2015**, *32*, 145–163.

58. Håstad, J. Tensor rank is NP-complete. *J. Algorithms* **1990**, *11*, 644–654.

59. Fazel, M.; Hindi, H.; Boyd, S.P.; others. A rank minimization heuristic with application to minimum order system approximation. In Proceedings of the American control conference, Arlington, VA, USA, 25–27 June 2001; Citeseer: Princeton, NJ, USA, 2001 ; Volume 6, pp. 4734–4739.

60. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. Introducing EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. In Proceedings of the IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 204–207.

61. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *arXiv* **2017**, arXiv:1709.00029.

62. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.

63. Giannopoulos, M.; Tsagkatakis, G.; Blasi, S.; Toutounchi, F.; Mouchtaris, A.; Tsakalides, P.; Mrak, M.; Izquierdo, E. Convolutional neural networks for video quality assessment. *arXiv* **2018**, arXiv:1809.10117.

64. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63.