Fast and Efficient Limited Data Hyperspectral Remote Sensing Image Classification via GMM-Based Synthetic Samples

AmirAbbas Davari, Hasan Can Özkan, Andreas Maier, Christian Riess, *Senior Member, IEEE,*

***Abstract*—In hyperspectral remote sensing, feature data can potentially become very high dimensional. At the same time, manual labeling of that data is an expensive task. As a consequence of these two factors, one of the core challenges is to perform multi-class classification using only relatively few training data points.**

In this work, we investigate the classification performance with limited training data. First, we revisit the optimization of the internal parameters of a classifier in the context of limited training data. Second, we report an interesting alternative to parameter optimization: classification performance can also be considerably increased by adding synthetic GMM data to the feature space while using a classifier with unoptimized parameters. Third, we show that using variational expectation maximization, we can achieve a much faster convergence in fitting the GMM on the data.

In our experiments, we show that addition of synthetic samples leads to comparable, and in some cases even higher classification performance than for a properly tuned classifier on limited training data. One advantage of the proposed framework is that the reported performance improvements are achieved by a quite simple model. Another advantage is that this approach is computationally much more efficient than classifier parameter optimization and conventional expectation maximization.

***Index Terms*—HSRS image classification, limited training data, classifier parameter tuning, synthetic data, variational EM**

I. Introduction

REMOTE sensing (RS) is of high importance for several application fields, including environmental monitoring, urban planning, ecosystem-oriented natural resources management, urban change detection and agricultural region monitoring [1].

The history of spectral RS sensors can be tracked back to the 1960s when Television Infrared Observation Satellite (TIROS-1) was launched with the mission of observing large-scale weather patterns from space [2]. Due to the low spatial resolution of sensors at that time, the recorded images could only be processed based on spectral information. Today's hyperspectral remote sensing (HSRS) sensors record hyperspectral images that also exhibit a high spatial resolution, which leads to much more informative data than before.

The majority of the monitoring and detection applications counted above require the construction of a label map of remotely sensed images in which individual pixels are marked as members of specific classes like water, asphalt, or grass. Automated generation of these label maps is done via classification.

Classification algorithms on high-resolution RS data exploits both the spectral and spatial properties of pixels [3]. It was shown in the literature that jointly exploiting spatial and spectral information considerably enhances the classification performance. Fauvel *et al.* [4] provide a thorough review on recent advances in the spectral-spatial analysis of HSRS images. To this end, morphological profiles (MP) are one of the most popular and powerful approaches to compute such spectral-spatial pixel descriptions. Indeed, they have been studied extensively in the last decade, and their effectiveness has been validated repeatedly [5], [6], [7]. Morphological profiles are particularly suitable for representing the multi-scale variations of image structures, but they are limited by the shape of the structuring elements. To avoid this limitation, several follow-up works lead to the extended multi-attribute profiles (EMAP) [8], [9]. EMAP allows to employ arbitrary region descriptors like shape, color, or texture. In addition, EMAP can be implemented efficiently, for example via max- and min-trees [10] or alpha trees [11].

However, a notorious limitation in RS image classification is the availability of only a limited number of labeled pixels for classifier training, because manual labeling is expensive and time consuming. However, powerful descriptors like EMAP often produce high dimensional features. These two factors together lead to the Hughes phenomenon [12], and make classifier training challenging. Researchers have put considerable effort into developing algorithms to address this challenge, which we categorize into three groups: 1) Development of new or reformulation of existing classifiers to improve performance with limited training data, 2) Dimensionality reduction of the feature vectors, 3) Increase of the feature pool by synthesized feature vectors. There exist several recent works on generative adversarial networks (GANs) for synthetic data generation [13], [14]. However, GANs themselves require significant amounts of data, which in conjunction with the high dimensionality of HS data prevents their use if training data is severely limited.

To address limited training data, Hoffbeck, Tadjudin and Landgrebe proposed Gaussian maximum likelihood for high-dimensional features [15], [16]. In particular, they proposed an estimator for the covariance matrices that requires considerably less labeled samples for generating a well-performing Gaussian maximum likelihood classifier. Similar in spirit, but on the SVM classifier, Chi *et al.* proposed a modification to the

A. Davari, A. Maier and C. Riess are with Friedrich-Alexander University Erlangen-Nürnberg, 91058 Erlangen, Germany (email: amir.davari@fau.de)

SVM classifier that is more robust to limited training data [17]. Also, Bruzzone *et al.* [18] proposed semi-supervised classification by introducing transductive and inductive functions as a controlling unit on the outputs of SVM classifier which are the candidates to be used as semi-labeled training data. Semi-supervised classification has also been proposed by Jackson *et al.* and Vatsavai *et al.* as a remedy to limited training data [19], [20]. Their main idea is to exploit classifier decisions on unlabeled data as semi-labeled data. The classifier is re-trained with that data to increase its performance. To minimize the impact of wrongly classified samples, semi-labeled data is weighted using a maximum likelihood (ML) filter. Recently, Xia *et al.* proposed a novel ensemble approach called rotation-based SVM (RoSVM) [21], using random feature selection to diversify the classifier. Compared to standard SVM, this approach performs better on limited training data, but it is computationally expensive. Li *et al.* proposed a classification framework based on integrating multiple linear and non-linear features, including EMAP [22], into a more effective classifier.

Another family of algorithms to address HS limited training data uses dimensionality reduction (DR). Reducing the number of spectral channels can effectively cure the Hughes phenomenon. Principle component analysis (PCA) and independent component analysis (ICA) are two of the most commonly used DR algorithms in the literature. In a recent work by Kang *et al.*, PCA is used to reduce the dimensionality of edge-preserving filters prior to classification [23]. They showed that the combination of edge-preserving filters and PCA results in a powerful feature vector. Sofolahan *et al.* introduced the summed component analysis, which exploits PCA and principle feature analysis (PFA) for dimensionality reduction [24]. A benefit of PFA over PCA and ICA is that PFA selects a subset of features, and thus its output can be physically further interpreted. However, PFA also causes a loss of information as it simply disregards certain features and dimensions. In contrast to the unsupervised DR techniques mentioned above, there exist also supervised dimensionality reduction algorithms, which are guided by the label information. To this end, non-parametric weighted feature extraction (NWFE) [25], discriminant analysis feature extraction (DAFE) [26] and decision boundary feature extraction (DBFE) [27] are probably the most popular reduction algorithms, which performed strongly in a comparison by Castaings *et al.* [28]. The common idea behind these algorithms is to map the data to another space, to minimize the within-class distance while maximizing the between-class distance in the lower dimensional space. While being conceptually similar to linear discriminant analysis, it was shown that NWFE in particular outperforms LDA on limited training data [29]. Recently, Kianisarkaleh *et al.* proposed nonparametric feature extraction (NFE) [29] for limited training data. It is similar to NWFE, but uses k neighbors in a class to compute the local class mean.

The third family of approaches aims to overcome the limited training data by generating synthetic data that is statistically similar to the available labeled data. To our knowledge, only few methods have been proposed in this direction. Skurichina *et al.* proposed to inject Gaussian noise in the k nearest neighborhood of the training data (k-NN DNI) [30]. Neagoe *et al.* proposed virtual sample generation using the weights of concurrent self-organizing maps (CSOM) [31]. In our previous work [32], we showed that re-sampling from the training data to increase the set population has positive effect on the classifier. However, as the added synthetic data was drawn from the dataset, the improvement was minor. In a later work, we showed that drawing synthetic samples from an accurately estimated distribution is more effective [33]. This is supported by several works in the literature that aim to find a general distribution model for hyperspectral remote sensing images. Specifically, Marden *et al.* proposed to use an elliptically contoured distribution, a more general distribution case of multivariate Gaussian, for generating statistically similar synthetic data [34]. Camps *et al.* propose a kernel-based framework for change detection and classification of multi-temporal and multi-source RS images [35] using a Gaussian mixture model (GMM). Williams *et al.* showed that GMM fitting with variational Bayesian Expectation Maximization works well on limited sample data [36], [37]. They estimate the number of GMM components by evidence maximization [38]. They also showed that a severely imbalanced dataset degrades the classification performance, which is another good application for synthetic data generation.

The first two families of methods, namely to design classifiers for limited amounts of training data and to reduce feature vector dimensionality, seem to be quite challenged by extreme cases when training data is severely limited. As a consequence, we focus on the third direction, and present a framework for generation of synthetic feature vectors to remedy the limited training data problem.

This work is an extension of our work presented in [39] on variational expectation maximization and provides a broader, consolidated view on our previous work on addition of synthetic samples [33]. Specifically, we show that synthetic samples can alleviate the limited data problem with minimal additional knowledge in a way that is computationally extremely efficient. It mitigates the costly traditional parameter tuning of a classifier. Instead, a GMM is fitted to the limited training data with a Variational Bayesian. This GMM is used to generate additional training data, using the common assumption that HS remote sensing samples can be modelled by a GMM [3], [40]. We show that if an off-the-shelf, unoptimized classifier is trained with this data, the resulting performance is comparable to a properly tuned classifier, at a fraction of the computational effort.

The paper is organized as follows. Gaussian mixture models are introduced in Sec. II, and variational expectation maximization in Sec. III. The proposed framework for addition of synthetic samples is presented in Sec. IV. Experimental results are reported in Sec. V, before we conclude our paper in Sec. VI.

II. Gaussian Mixture Model

For a d-dimensional random variable $\mathbf{x}$, the multivariate Gaussian density function is defined as:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Lambda}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Lambda}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{1}$$

where $\boldsymbol{\mu}$ and Λ are mean vector and covariance matrix of the Gaussian model respectively.

Mixture models model the data by a combination of components. A Gaussian Mixture Model (GMM) is a parametric probability density function that can model any other distribution. It is represented as the weighted sum of K Gaussian density components:

$$p(\mathbf{x}|\psi) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i), \tag{2}$$

with parameters $\psi = (\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i)$, where for the i-th component, π_i denotes the mixture weight, $\boldsymbol{\mu}_i$ the mean and $\boldsymbol{\Lambda}_i$ the covariance matrix. The mixture weights is constrained to $\sum_{i=1}^{K} \pi_i = 1$.

To estimate the GMM parameters from the data, iterative algorithms for expectation maximization (EM) or Maximum A Posteriori (MAP) estimation are commonly used [41], [42].

When the training data is severely limited, it is particularly important to consider the number of parameters of a GMM. For d-dimensional training data and K mixture components, the most general formulation of GMM requires a total of $K(1 + d + d^2)$ parameters. However, this number can be reduced by applying simplifying assumptions. For example, the covariance matrix can be set identical for all components, or the covariance is constrained to be diagonal, or diagonal with identical entries per dimension. It is also possible to combine these assumptions, i.e., to share covariances while constraining their content.

In our work, the only constraint is that we assume diagonal covariance matrices. Thus, $\boldsymbol{\Lambda}_i = \mathrm{diag}(\sigma_{i1}^2, \sigma_{i2}^2, \ldots, \sigma_{id}^2)$. This leads to a total of $K(1 + 2d)$ parameters, which is a trade-off between the number of parameters and the model flexibility: for example, a linear combination of diagonal covariance matrices is still able to model correlations between the data dimensions [42].

III. Variational Bayesian Inference for GMM

Variational Bayesian (VB) can be considered as a family of methods that makes the computation of probability distributions tractable. VB methods are an extension of the EM algorithm that maximize a lower bound on model evidence $p(\boldsymbol{X})$, where $\boldsymbol{X}$ denotes the set of observations. Variational methods and EM are both iterative algorithms which alternate between a) determining the probabilities for a data point to belong to a mixture component and b) fitting the mixture to the corresponding data. However, variational methods add regularization by integrating information from prior distributions. A particularly useful property of VB over maximum likelihood GMM is that VB methods avoid over-fitting and singularities [43].

We denote N observations as $\boldsymbol{X} = \{x_1, ..., x_N\}$, and N latent variables as $\boldsymbol{Z} = \{z_1, ..., z_N\}$. Probabilistic formulation of VB becomes easier when the membership of the GMM components is made explicit. To this end, each observation x_i has an associated latent indicator variable z_i. Then, $p(\boldsymbol{X})$ is the marginal distribution of $p(\boldsymbol{X}, \boldsymbol{Z})$, i.e.,

$$p(x) = \sum_{z} p(z)p(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Lambda_k) \ , \tag{3}$$

where we omitted for clarity of notation the dependency on the model parameters $\boldsymbol{\mu}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\pi}$.

Consider a variational distribution which factorizes into latent variables and model parameters as

$$q(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{Z})q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \ . \tag{4}$$

This factorization is the only assumption required in order to acquire a tractable and useful result for the mixture model. With the expectation maximization (EM) algorithm, $q(\boldsymbol{Z})$ is estimated in the expectation and $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ in the maximization step. Both can be determined automatically by optimizing the variational distribution. For the full theoretical derivation we refer to [43, Chap. 10] due to space constraints. The EM update equations are presented in the Appendix of the paper.

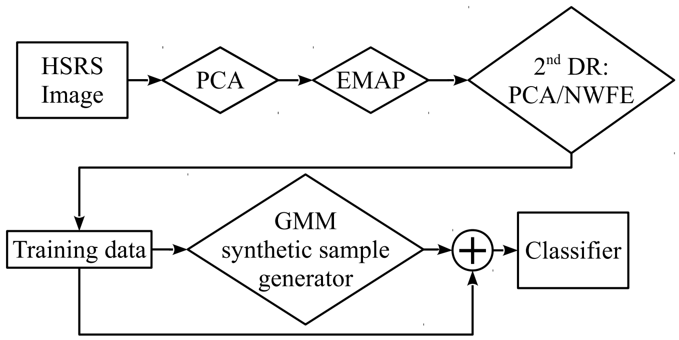
IV. GMM-Based Synthetic Data Generation: Overview and Benefits

In this work, we consolidate our earlier works on synthetic data generation for hyperspectral remote sensing (HSRS) images [33], [39], evaluate the performance on two new datasets, and show additional experiments on the benefit of added synthetic samples including a neural network classifier. In our previous work, we have proposed the generation of GMM-based synthetic samples as a remedy for limited availability of training samples in HSRS image classification [33]. The generated synthetic samples are a considerably faster alternative for tuning the classifiers' parameters. We further proposed to substitute the classical expectation maximization (EM) with the variational EM to gain a faster convergence in our GMMs [39]. In this section, we explain each part of this pipeline in detail.

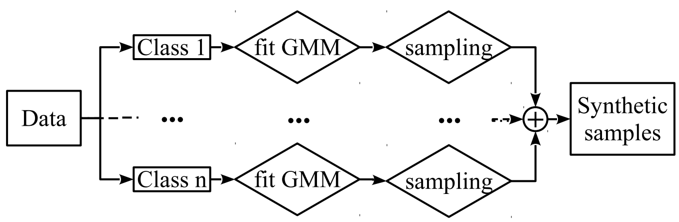
A. GMM-Based Synthetic Data Generation

We show the effectivity of synthetic sample addition on a standard classification pipeline that is based on dimensionality reduction. Our pipeline is shown in Fig. 1. First, the dimensionality of the hyperspectral image is reduced via PCA. Then, extended multi-attribute profiles (EMAP) [9] are computed as the feature vector for every pixel. The EMAP feature vectors can be further reduced in their dimensionality via PCA or non-parametric weighted feature extraction (NWFE). We then estimate the probability density function (PDF) of each class in the dataset by fitting a GMM on the training data. The approach we took for handling the issues for GMM estimation is as follows:

a) Number of Gaussian components: the distribution of a class label is typically not a clean Gaussian, and hence a GMM typically requires more than one component to model the distribution. Thus, for each class we construct GMMs with 1 to 4 components, to compromise between the model flexibility and the number of model parameters. To find the best model for representing the class, the well-known Akaike Information criterion (AIC) is used.



(a) Our proposed classification pipeline.



(b) Proposed GMM-based synthetic sample generation block.

Fig. 1: The (a) classification pipeline and (b) GMM-based synthetic sample generator block. Rectangle represents a data, diamond indicates a function (operation), and the plus sign shows the data concatenation operation.

b) Initialization: The iterative EM algorithm is highly sensitive to its initial values. To start the iteration from a reasonably good solution, we initialize the GMM components with the cluster centers of a k-means clustering to the data, where k is identical to the number of GMM components [43, Sec. 9.1, p. 427].

c) Constraints on the covariance matrices: We use diagonal covariance matrices as a trade-off between representational power and the feasibility of fitting such a model to the few training data samples. Linear combinations of diagonal matrices can model the correlation between the dimensions [42], such that full covariance matrices are not necessarily needed. On the other hand, when estimating full covariance matrices on very few samples, EM may not converge. Thus, diagonal covariance matrices have much fewer parameters, which makes the estimation feasible and much more efficient [42].

After the construction of the feature vector and the GMMs from the training data, we draw an equal number n of synthetic samples per class by sampling from the GMMs and add them to the original training data for classifier training.

B. GMM-based Synthetic Data; An Alternative to Classifier Parameter Tuning

When using, e.g., the support vector machine classifier (SVM), it is widely known that parameter selection is a critical preparatory step towards obtaining competitive results. This is the reason why for example the SVM parameter selection is hardwired into the popular SVM implementation `libSVM`. However, other classification frameworks do not necessarily include a parameter selection submodule. One notable example is classification with a random forest. Several works [8], [32], [44], [45], [46], [47], [48] rely on the default settings of 100 trees with a tree depth equal to the square root of the feature dimensionality, $\sqrt{d}$, as originally proposed by Breiman [49]. However, these parameters have been proposed based on training on a relatively large dataset. In the case of classification on severely limited training data, such default parameters yield suboptimal classification performance [33].

GMMs only roughly approximate the true underlying distribution of hyperspectral data [34]. Nevertheless, synthetic data can enrich the feature space with additional, similar features to compensate challenges that a non-optimized classifier has on severely limited samples. This is illustrated with an example on simulated data in Fig. 2.

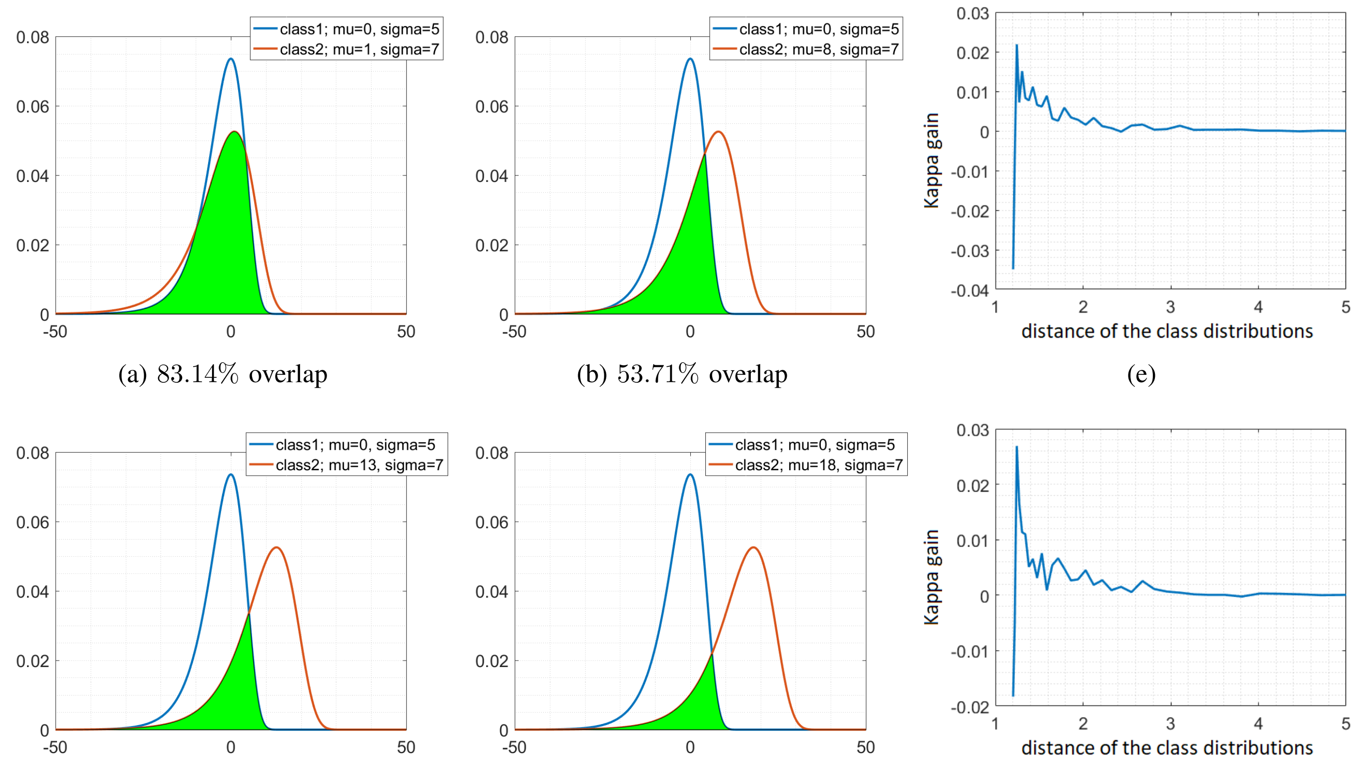
In this simulation, we generate a 2-class dataset. The classes have the "Extreme Value" distribution, which is parameterized by location parameter μ and scale parameter $\sigma > 0$,

$$f(x|\mu,\sigma) = \sigma^{-1} \exp\left(\frac{x-\mu}{\sigma}\right) \exp\left(-\exp\left(\frac{x-\mu}{\sigma}\right)\right). \tag{5}$$

We show the interplay of a classifier with unoptimized parameters, the amount of overlap between the classes' underlying distributions, and the addition of synthetic data. For the first class, 2000 points are sampled from an extreme distribution with parameters (μ,σ)=(0,5). For the second class, 2000 data points are sampled from an extreme distribution with $\sigma = 7$ and $\mu \in [1, 20]$ with steps of 0.5. In this way, different class distributions with different amount of overlap are produced. Examples are shown on the left of Fig. 2. The overlap between the classes is plotted in green. We randomly sample 13 training examples from each class to create a 2-class-classification problem. We then fit a GMM to each group of training samples, draw 500 additional samples and train on that data an otherwise unoptimized random forest with 2 and 10 trees. The results are plotted on the right of Fig. 2. On the x-axis, the distance of the two distributions is shown. This distance is defined as the inverse of the distributions' overlap area. The y-axis shows the kappa difference between unoptimized classifiers when using additional samples or not. For moderate to high amounts of class overlap, addition of the synthetic samples improves the classification. Although the GMM does not exactly match the extreme distributions, the classification performance is considerably improved by the added samples. This shows that the mis-parameterization of the classifier is a major performance bottleneck that can be met by (gently) adapting the feature space to the classifier. When the class distributions get far from each other, the positive benefit of synthetic samples decreases as the classifier can easily distinguish these classes from the initial 13 samples. Moreover, the further the classifier's parameters are from the optimal values, the higher the effect of the synthetic data augmentation will be.

C. Faster and More Efficient GMM fitting via Variational EM

The estimation of GMM parameters usually is carried out using the expectation maximization (EM) algorithm. It was shown in [36], [37] that the GMM model estimated



(a) 83.14% overlap (b) 53.71% overlap (e)

(c) 33.86% overlap (d) 20.07% overlap (f)

Fig. 2: Simulation example on model accuracy gain versus the overlap of the classes distributions. The left plots show the distributions of a two class data. The class overlap for each example is stated under each plot. Plots (e) and (f) show the Kappa gain after adding 500 synthetic samples to the limited training data (13 samples per class). The x-axis shows the inverse of the overlapping area of the two class distributions. The classifier is a random forest. Number of variables is square root of the number of data features and number of trees in (e) and (f) are chosen to be 2 and 10, respectively.

by their proposed optimization technique, i.e. variational Bayesian expectation maximization (VB-EM), performs better, particularly when dealing with a small number of training samples. Furthermore, they addressed another problematic factor, the determination of the number of GMM components when estimating a probability density function (PDF). They showed that using VB-EM instead of EM, the importance of this factor will be lessened. Conventionally, determining the number of components in a GMM is carried out by fitting different models a posteriori and the best model is being chosen via an algorithm, e.g. Akaike information criterion, Bayesian information criterion, etc. However, the number of GMM components can be exactly determined by evidence maximization [38] in conjunction with the VB-EM algorithm. The main advantage of using VB-EM is that there is no need to generate multiple models a posteriori. Therefore, in order to speed up the GMM computation and make it memory efficient, we use Variational Bayesian method to determine the number of GMM components automatically.

V. Evaluation

A. Datasets Description

We use, for the evaluation, the Salinas, SalinasA, Botswana and Pavia Centre datasets. All these datasets are publicly available via [50], [51]. Salinas dataset is a 512×217 pixels image with a geometrical resolution of $3.7\,\mathrm{m}$. It was acquired by the AVIRIS sensor in 224 spectral bands over Salinas Valley, California. 20 bands were discarded due to water absorption and the remaining 204 bands are used in this work. Its ground truth contains 16 classes, including different types of vegetation, fields and soil, with 54129 labeled pixels. SalinasA is a 86×83 pixels subset of the Salinas dataset which is commonly used as a benchmark in the community [52], [53], [54], [55], [56], [57] and contains 6 classes. The number of available labeled pixels in this dataset is 5348.

The Botswana dataset was acquired by NASA EO-1 satellite using the Hyperion sensor in 242 bands in the wavelength range of 400-2500 nm. After removing the noisy bands, 145 spectral channels were used in this work. This dataset contains 14 classes, including different swamps and woodlands. The number of available labeled pixels are 3248.

The Pavia Centre dataset has been acquired by the ROSIS sensor in 115 spectral bands over Pavia, northern Italy. 13 of these bands are removed due to noise and therefore 102 bands are used in this work. The scene image is 1096×715 pixels with a geometrical resolution of $1.3\,\mathrm{m}$. This dataset contains 148152 labeled pixels in 9 classes.

B. Classification Pipeline

To demonstrate the effect of adding synthetic samples, we use a standard classification pipeline that is based on dimensionality reduction. The algorithm variants are shown in Fig. 1. First, PCA is performed on the input data to preserve 99% of the total spectral variance. On these PCA components, extended multi-attribute profile (EMAP) features are computed. We followed the literature by using four attributes and four thresholds λ per attribute [9], [58]. More specifically, the thresholds for area of connected components are chosen as $\lambda_a = [100, 500, 1000, 5000]$, and the thresholds for length of the diagonal of the bounding box fitted around the connected components λ_d are chosen as $\lambda_d = [10, 25, 50, 100]$. The thresholds for standard deviation of the gray values of the connected components λ_s and the moment of inertia λ_i are chosen differently per dataset [9], [58], i.e., for Salinas and SalinasA $\lambda_s = [20, 30, 40, 50]$ and $\lambda_i = [0.1, 0.15, 0.2, 0.25]$, and $\lambda_s = [20, 30, 40, 50]$ and $\lambda_i = [0.2, 0.3, 0.4, 0.5]$ for the Botswana dataset. For the Pavia Centre dataset, the threshold values of the attributes are similar to [9].
For the second dimensionality reduction, we use in one variant the unsupervised PCA, and in another variant the supervised non-parametric weighted feature extraction (NWFE) [25], [28] to preserve 99% of the feature variance. In our experiments, we use abbreviations to specify the used pipeline configuration. We use either EMAP, EMAP-PCA, or EMAP-NWFE to distinguish the use of no secondary dimensionality reduction, PCA, or NWFE, respectively. Classification is performed with random forest classifier.

C. Feature Set Augmentation via Synthetic Samples

To quantitatively evaluate the difference between an unoptimized and an optimized classifier, we use the random forest default parameters as proposed by Breiman [49], with 100 trees, $H = 100$, and number of parameters to be the square root of number of feature dimensions, $D = \sqrt{d}$. The optimized random forest parameters are found via leave-one-sample-out cross validation. On average, the kappa value for the classification grows by 5.47% after optimizing the classifier, with a standard deviation of 3.06%. The parameter optimization for SalinasA and Botswana datasets takes in average 48.21 seconds and 77.74 seconds, respectively. Representative example results are shown in Tab. I. In this table, we show the average accuracy (AA), overall accuracy (OA) and the Cohen's Kappa [59] for a random forest classification on the Botswana and the SalinasA datasets. We showed the results on the Pavia Centre and Salinas datasets in a previous work [33], which agree with the experiments shown here.

In a second experiment, we add synthetic samples to the feature space. A first result is shown in Fig. 3. Here, we performed classification on 13 (left) and 40 (right) training samples per class, respectively. We used a random forest with unoptimized parameters on EMAP-NWFE computed on Salinas dataset, and report Kappa for different numbers of up to 5000 added synthetic samples. It turns out that adding only a few synthetic samples leads to a jump in classification performance, e.g. from about 0.78 to about 0.86 if 13 training samples per class are used (Fig. 3, left plot). This performance gain is quite stable with respect to the exact number of added samples, i.e., it does not make much difference whether 500 or 5000 samples are added.

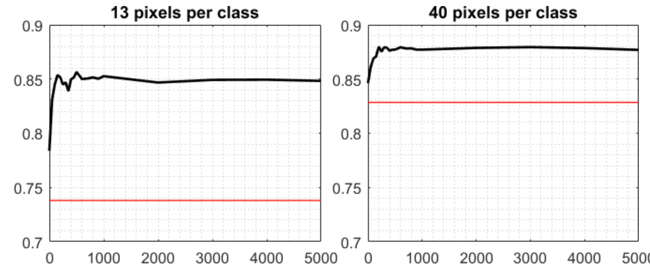


Fig. 3: Unoptimized random forest's classification performance (kappa) versus the number of synthetic samples added to the original training set. Classification is performed on EMAP-NWFE computed over Salinas dataset. Red line represents the classification performance of raw EMAP without any synthetic sample addition.

A full quantitative evaluation is performed with the same feature variants EMAP, EMAP-PCA, EMAP-NWFE, and the same experimental protocol as explained earlier on the SalinasA and Botswana datasets. Since we require a low feature dimensionality to fit the GMM model to very few samples, synthetic samples are only added to the dimensionality-reduced variants of EMAP feature, i.e. EMAP-PCA and EMAP-NWFE, but not to the very high-dimensional EMAP space. Representative example results are shown in Tab. I. In every case, the variants using synthetic samples improve the classification performance over an unoptimized classifier. The average improvement of the kappa value jumps 5.84% up after adding synthetic samples to the training set, with a standard deviation of 3.18% [33].

Two observations can be made when comparing the results of synthetically augmented data using unoptimized classifier versus an optimized classifier in Tab. I. First, the addition of synthetic samples performs comparable and sometimes even slightly higher than an optimized classifier. Second, a dimensionality-reduced EMAP (EMAP-PCA or EMAP-NWFE) with synthetic samples performs comparably or in some cases even better than using full EMAP feature vector with a properly tuned classifier. Both observations indicate the positive impact of adding synthetic samples and show that it is an interesting alternative to classifier optimization.

The class-wise classification performance for the SalinasA and Botswana datasets are presented in Tab. II and Tab. III, respectively. Analogously to the summary results in Tab. I, the results on an optimized classifier and on an unoptimized classifier with added synthetic samples are comparable. Additionally, it is interesting to further investigate the relationship of the unoptimized classifier without and with synthetic samples. A subset of classes achieves low accuracy when using just the original limited training data. These classes are challenging for the classifier, and reduce the overall classifier accuracy. Adding the additional synthetic samples greatly boosts the performance on those classes, and thereby improves the overall

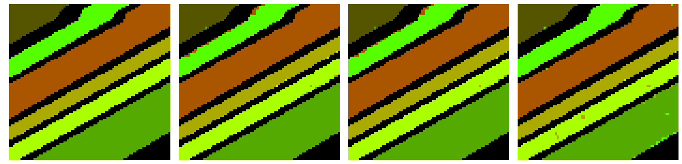
TABLE I: Random forest performance computed over SalinasA and Botswana. H and D represent the forest parameters, where "-" indicates an unoptimized forest. $|S|$ denotes the number of added synthetic samples per class.

Algorithm	H	D	$\|S\|$	AA% ($\pm$SD)	OA% ($\pm$SD)	Kappa ($\pm$SD)
SalinasA						
13 pix/class						
HS raw	-		0	85.42 ($\pm$3.34)	79.85 ($\pm$5.21)	0.7546 ($\pm$0.0612)
HS raw	5	4	0	**95.40** ($\pm$0.95)	**94.90** ($\pm$1.25)	**0.9363** ($\pm$0.0155)
EMAP	-		0	94.88 ($\pm$4.06)	93.38 ($\pm$6.06)	0.9186 ($\pm$0.0736)
EMAP	10	4	0	**99.04** ($\pm$0.45)	**99.15** ($\pm$0.34)	**0.9893** ($\pm$0.0043)
EMAP-PCA	-		0	93.21 ($\pm$2.73)	91.58 ($\pm$4.01)	0.8958 ($\pm$0.0484)
EMAP-PCA	5	4	0	**98.53** ($\pm$1.15)	**98.59** ($\pm$1.09)	**0.9824** ($\pm$0.0136)
EMAP-PCA	-		500	**99.04** ($\pm$0.29)	**99.22** ($\pm$0.23)	**0.9903** ($\pm$0.0028)
EMAP-NWFE	-		0	93.13 ($\pm$2.12)	90.34 ($\pm$3.13)	0.8808 ($\pm$0.0384)
EMAP-NWFE	10	4	0	**99.11** ($\pm$0.23)	**99.17** ($\pm$0.15)	**0.9896** ($\pm$0.0019)
EMAP-NWFE	-		500	**99.10** ($\pm$0.28)	**99.00** ($\pm$0.54)	**0.9874** ($\pm$0.0067)
40 pix/class						
HS raw	-		0	92.43 ($\pm$1.13)	90.02 ($\pm$1.70)	0.8764 ($\pm$0.0207)
HS raw	5	4	0	**97.39** ($\pm$0.40)	**96.96** ($\pm$0.56)	**0.9620** ($\pm$0.0070)
EMAP	-		0	98.46 ($\pm$0.43)	98.52 ($\pm$0.64)	0.9815 ($\pm$0.0080)
EMAP	10	10	0	**99.61** ($\pm$0.32)	**99.67** ($\pm$0.23)	**0.9959** ($\pm$0.0029)
EMAP-PCA	-		0	97.38 ($\pm$0.87)	97.40 ($\pm$1.17)	0.9675 ($\pm$0.0146)
EMAP-PCA	10	6	0	**99.15** ($\pm$0.25)	**99.31** ($\pm$0.23)	**0.9914** ($\pm$0.0029)
EMAP-PCA	-		500	**99.12** ($\pm$0.22)	**99.28** ($\pm$0.26)	**0.9909** ($\pm$0.0032)
EMAP-NWFE	-		0	95.83 ($\pm$1.04)	94.30 ($\pm$1.53)	0.9293 ($\pm$0.0188)
EMAP-NWFE	20	2	0	**99.40** ($\pm$0.19)	**99.45** ($\pm$0.19)	**0.9932** ($\pm$0.0023)
EMAP-NWFE	-		500	**99.47** ($\pm$0.15)	**99.49** ($\pm$0.16)	**0.9936** ($\pm$0.0020)
Botswana						
13 pix/class						
HS raw	-		0	70.39 ($\pm$2.42)	67.32 ($\pm$2.69)	0.6473 ($\pm$0.0287)
HS raw	10	8	0	**81.60** ($\pm$1.06)	**79.84** ($\pm$0.90)	**0.7819** ($\pm$0.0098)
EMAP	-		0	89.83 ($\pm$1.94)	88.79 ($\pm$2.17)	0.8786 ($\pm$0.0235)
EMAP	10	10	0	**94.69** ($\pm$0.72)	**94.04** ($\pm$0.87)	**0.9354** ($\pm$0.0094)
EMAP-PCA	-		0	83.45 ($\pm$2.13)	83.05 ($\pm$2.18)	0.8165 ($\pm$0.0236)
EMAP-PCA	20	8	0	**91.64** ($\pm$0.73)	**91.00** ($\pm$0.83)	**0.9025** ($\pm$0.0090)
EMAP-PCA	-		500	**93.35** ($\pm$0.55)	**92.72** ($\pm$0.49)	**0.9212** ($\pm$0.0053)
EMAP-NWFE	-		0	87.38 ($\pm$1.96)	87.12 ($\pm$1.51)	0.8605 ($\pm$0.0163)
EMAP-NWFE	10	4	0	**92.31** ($\pm$1.07)	**91.59** ($\pm$1.11)	**0.9089** ($\pm$0.0120)
EMAP-NWFE	-		500	**93.73** ($\pm$0.77)	**93.17** ($\pm$0.77)	**0.9260** ($\pm$0.0083)
40 pix/class						
HS raw	-		0	82.58 ($\pm$0.36)	80.39 ($\pm$0.52)	0.7880 ($\pm$0.0056)
HS raw	5	8	0	**88.29** ($\pm$0.33)	**86.77** ($\pm$0.46)	**0.8568** ($\pm$0.0049)
EMAP	-		0	95.61 ($\pm$0.34)	94.90 ($\pm$0.38)	0.9448 ($\pm$0.0041)
EMAP	10	10	0	**97.35** ($\pm$0.36)	**96.98** ($\pm$0.40)	**0.9673** ($\pm$0.0043)
EMAP-PCA	-		0	93.25 ($\pm$0.92)	92.39 ($\pm$1.23)	0.9176 ($\pm$0.0133)
EMAP-PCA	20	4	0	**94.91** ($\pm$0.48)	**94.38** ($\pm$0.51)	**0.9391** ($\pm$0.0055)
EMAP-PCA	-		500	**95.04** ($\pm$0.44)	**94.49** ($\pm$0.42)	**0.9403** ($\pm$0.0046)
EMAP-NWFE	-		0	93.63 ($\pm$0.42)	92.76 ($\pm$0.55)	0.9217 ($\pm$0.0060)
EMAP-NWFE	10	4	0	**95.28** ($\pm$0.46)	**94.73** ($\pm$0.54)	**0.9429** ($\pm$0.0058)
EMAP-NWFE	-		500	**95.33** ($\pm$0.32)	**94.66** ($\pm$0.37)	**0.9421** ($\pm$0.0040)

performance on the whole dataset. It is also interesting to note that the addition of synthetic samples greatly reduces the standard deviation of the overall dataset, and in particular on the classes with low accuracy. Overall, we conclude that addition of GMM-based synthetic samples not only improves the accuracy of the classifier, but also boosts the confidence of the classifier, which is reflected by having lower standard deviation.

Figures 4 and 5 show the qualitative results, i.e. label maps, on SalinasA and Botswana datasets with unoptimized random forest classifier when adding synthetic samples. The synthetic data augmentation improves the classification accuracy and avoids some misclassification.

One interesting question is whether the improvement should be attributed simply to the increased number of samples, or to an improved representation of the underlying distribution. While it is difficult to experimentally factorize these influencing factors, a comparison with other data augmentation schemes (confer Tab. XVI of the supplemental material of [33]) indicates that the representation is at least better suited than that from earlier works [31], [30].



(a) (b) (c) (d)

Fig. 4: Label maps on SalinasA using 13 training samples per class and unoptimized random forest. (a) ground truth; (b) EMAP (OA: 93.38%, Kappa: 0.9186); (c) EMAP-PCA with 500 synthetic samples (OA: 99.22%, Kappa: 0.9903); (d) EMAP-NWFE with 500 synthetic samples (OA: 99.00%, Kappa: 0.9874).

D. Variational Bayes instead of the Conventional Expectation Maximization

Expectation maximization (EM) is commonly used for finding and optimizing GMM parameters. One difficulty, however, is how to determine the number of components in the GMM. In this work, we compared the runtime and classification performance of our synthetic data generation pipeline using variational EM (VEM) with four other algorithms, namely, Akaike information criterion (AIC) [60], Bayesian information criterion (BIC) [61], average silhouette width [62], [63] and gap [64]. What all these algorithms have in common is the fact that they choose the best model, i.e. the most suitable number of components, from the a posteriori generated GMMs with different number of components. In contrast, variational Bayesian (VB), does not require the pre-computation of GMM models with different number of components.

For the GMMs, the covariance matrix is constrained to be diagonal. As for the AIC, BIC, average silhouette width and gap methods, we constrain K between 1 and 4 and let these algorithms choose the best model. We did not fit GMMs with more than 4 components, because the number of parameters would be large enough and considering the limited available training data, EM algorithm fails to converge. Please refer to Section II for more information. As stated in Section III, we prefer to have a large initial number of components (K). Therefore, K is selected to be 25. The number of generated synthetic samples is set to 5000 samples per each class for all the experiments.

The quantitative evaluation results are shown in Table IV, Table V, Table VI and Table VII for the Pavia Centre, Salinas, SalinasA and Botswana data sets, respectively. Considering the classification performance, all variants that make use of VB generate similar, if not better, classification results. The Wilcoxon statistical significance test indicated no significant difference in the classification results that are computed with

TABLE II: Class-wise performance computed over SalinasA dataset using both optimized (o-RF) and unoptimized random forest (u-RF), with 0 and with 500 synthetic samples per class. The bold font numbers represent the classes, which benefit the most from the addition of GMM-based synthetic samples.

		o-RF	u-RF		o-RF	u-RF	
Class	Train/Test	EMAP-PCA	EMAP-PCA	EMAP-PCA-Synth	EMAP-NWFE	EMAP-NWFE	EMAP-NWFE-Synth
Broccoli green weeds 1	13/391	99.23 ± 1.71	99.92 ± 0.12	99.80 ± 0.11	99.83 ± 0.15	99.72 ± 0.08	99.74 ± 0.00
Corn	13/1343	99.18 ± 1.33	74.18 ± 17.76	**99.91 ± 0.15**	99.16 ± 0.50	66.49 ± 11.77	**98.97 ± 0.59**
Lettuce romaine 4wk	13/616	96.59 ± 0.92	91.69 ± 8.16	**96.31 ± 2.07**	97.93 ± 2.74	97.69 ± 2.26	**99.11 ± 1.20**
Lettuce romaine 5wk	13/1525	98.20 ± 2.49	99.91 ± 0.27	99.55 ± 0.92	99.98 ± 0.04	98.54 ± 2.42	98.90 ± 1.97
Lettuce romaine 6wk	13/674	99.24 ± 0.68	99.60 ± 0.20	99.51 ± 0.24	99.55 ± 0.26	99.78 ± 0.08	99.57 ± 0.18
Lettuce romaine 7wk	13/799	98.75 ± 1.40	93.97 ± 5.49	**99.16 ± 0.59**	98.21 ± 0.71	96.55 ± 1.51	**98.27 ± 0.66**
Average Accuracy		98.53 ± 1.15	93.21 ± 2.73	99.04 ± 0.29	99.11 ± 0.23	93.13 ± 2.12	99.10 ± 0.28
Overall Accuracy		98.59 ± 1.09	91.58 ± 4.01	99.22 ± 0.23	99.17 ± 0.15	90.34 ± 3.13	99.00 ± 0.54
Kappa		0.9824 ± 0.0136	0.8958 ± 0.0484	0.9903 ± 0.0028	0.9896 ± 0.0019	0.8808 ± 0.0384	0.9874 ± 0.0067

TABLE III: Class-wise performance computed over Botswana dataset using both optimized (o-RF) and unoptimized random forest (u-RF), with 0 and with 500 synthetic samples per class. The bold font numbers represent the classes, which benefit the most from the addition of GMM-based synthetic samples.

		o-RF	u-RF		o-RF	u-RF	
Class	Train/Test	EMAP-PCA	EMAP-PCA	EMAP-PCA-Synth	EMAP-NWFE	EMAP-NWFE	EMAP-NWFE-Synth
Water	13/270	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Hippo grass	13/101	96.37 ± 4.00	93.56 ± 6.16	94.36 ± 5.50	95.05 ± 4.54	90.69 ± 16.49	97.52 ± 2.96
Floodplain grasses1	13/251	98.27 ± 1.66	94.74 ± 6.57	96.57 ± 2.25	95.48 ± 4.41	97.13 ± 4.92	98.53 ± 1.44
Floodplain grasses2	13/215	93.49 ± 3.05	94.42 ± 6.02	94.84 ± 3.23	93.49 ± 3.63	92.47 ± 7.18	94.47 ± 2.63
Reeds1	13/269	81.41 ± 1.12	65.99 ± 14.94	**83.94 ± 8.85**	78.94 ± 4.65	75.46 ± 11.85	**87.73 ± 4.30**
Riparian	13/269	73.11 ± 5.05	70.52 ± 14.49	**81.64 ± 9.26**	84.26 ± 12.36	68.18 ± 15.52	**79.33 ± 5.18**
Firescare2	13/259	97.55 ± 1.36	97.88 ± 2.66	99.23 ± 0.60	98.71 ± 0.59	95.14 ± 2.43	99.00 ± 0.49
Island interior	13/203	99.84 ± 0.28	99.75 ± 0.48	99.85 ± 0.33	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Acacia woodlands	13/314	95.65 ± 4.64	77.52 ± 5.96	**97.10 ± 1.98**	89.92 ± 8.62	86.75 ± 4.84	**93.47 ± 5.93**
Acacia shrublands	13/248	87.37 ± 3.75	89.68 ± 9.91	86.85 ± 5.34	88.63 ± 6.94	91.61 ± 9.30	90.16 ± 5.19
Acacia grasslands	13/305	84.97 ± 6.40	66.98 ± 11.08	**86.89 ± 5.80**	89.73 ± 5.98	80.30 ± 6.99	**89.67 ± 7.30**
Short mopane	13/181	97.05 ± 0.32	91.49 ± 10.06	96.35 ± 2.57	90.24 ± 2.30	96.46 ± 5.35	95.80 ± 2.00
Mixed mopane	13/268	83.45 ± 3.77	65.78 ± 18.92	**89.85 ± 6.07**	88.19 ± 5.78	74.44 ± 18.42	**90.90 ± 5.51**
Exposed soils	13/95	94.44 ± 3.04	60.00 ± 21.08	**99.47 ± 1.02**	99.65 ± 0.61	74.63 ± 31.17	**95.58 ± 6.54**
Average Accuracy		91.64 ± 0.73	83.45 ± 2.13	93.35 ± 0.55	92.31 ± 1.07	87.38 ± 1.96	93.73 ± 0.77
Overall Accuracy		91.00 ± 0.83	83.05 ± 2.18	92.72 ± 0.49	91.59 ± 1.11	87.12 ± 1.51	93.17 ± 0.77
Kappa		0.9025 ± 0.0090	0.8165 ± 0.0236	0.9212 ± 0.0053	0.9089 ± 0.0120	0.8605 ± 0.0163	0.9260 ± 0.0083

TABLE IV: GMM model selection methods vs. Variational Bayesian on the Pavia Centre dataset.

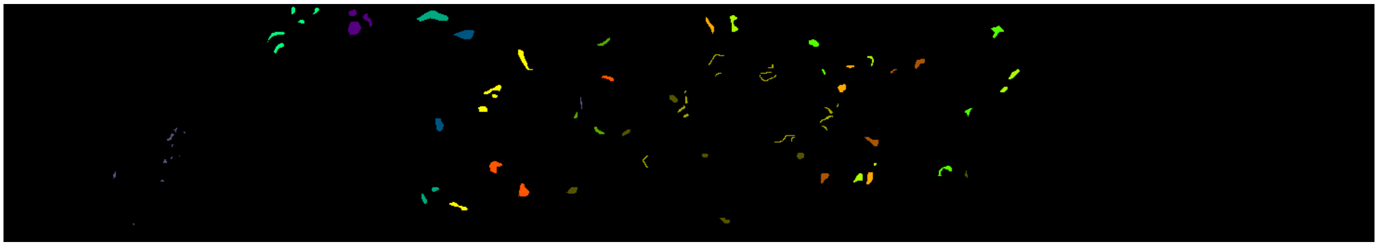
	samp.	AA% (±SD)	OA% (±SD)	Kappa (±SD)	Runtime (s) (±SD)
EMAP-PCA					
AIC	13	85.17 (±1.21)	93.39 (±1.44)	0.9072 (±0.0197)	0.0877 (±0.0115)
	30	88.52 (±0.71)	94.68 (±0.51)	0.9252 (±0.0070)	0.0978 (±0.0054)
BIC	13	85.50 (±1.14)	93.67 (±0.82)	0.9110 (±0.0114)	0.0781 (±0.0017)
	30	88.23 (±1.06)	94.81 (±0.46)	0.9270 (±0.0064)	0.0856 (±0.0025)
sil.	13	85.10 (±1.27)	93.64 (±1.03)	0.9105 (±0.0142)	0.2013 (±0.0082)
	30	87.61 (±1.14)	94.68 (±0.32)	0.9251 (±0.0045)	0.3521 (±0.0223)
gap	13	83.14 (±1.87)	92.23 (±1.00)	0.8911 (±0.0138)	16.4766 (±0.1400)
	30	85.87 (±2.62)	93.54 (±1.03)	0.9091 (±0.0145)	35.4273 (±0.2511)
VB	13	85.60 (±0.62)	93.52 (±0.40)	0.9090 (±0.0055)	**0.0324** (±0.0030)
	30	89.14 (±0.46)	95.15 (±0.42)	0.9317 (±0.0059)	**0.0450** (±0.0029)
EMAP-NWFE					
AIC	13	87.72 (±1.96)	94.75 (±0.96)	0.9260 (±0.0133)	0.0788 (±0.0043)
	30	91.86 (±1.05)	96.41 (±0.53)	0.9493 (±0.0074)	0.0895 (±0.0037)
BIC	13	89.84 (±0.90)	95.65 (±0.66)	0.9387 (±0.0092)	0.0785 (±0.0044)
	30	91.98 (±0.53)	96.50 (±0.45)	0.9506 (±0.0063)	0.0884 (±0.0038)
sil.	13	88.83 (±0.99)	95.00 (±0.61)	0.9297 (±0.0084)	0.2136 (±0.0103)
	30	91.30 (±0.78)	96.28 (±0.64)	0.9476 (±0.0089)	0.3817 (±0.0377)
gap	13	89.08 (±0.83)	95.30 (±0.61)	0.9338 (±0.0084)	17.5375 (±0.1385)
	30	90.75 (±1.16)	96.01 (±0.61)	0.9437 (±0.0084)	39.1785 (±0.5053)
VB	13	89.60 (±1.37)	96.11 (±0.53)	0.9404 (±0.0075)	**0.0328** (±0.0023)
	30	91.55 (±0.62)	96.43 (±0.47)	0.9469 (±0.0065)	**0.0471** (±0.0026)

TABLE V: GMM model selection methods vs. Variational Bayesian on the Salinas dataset.

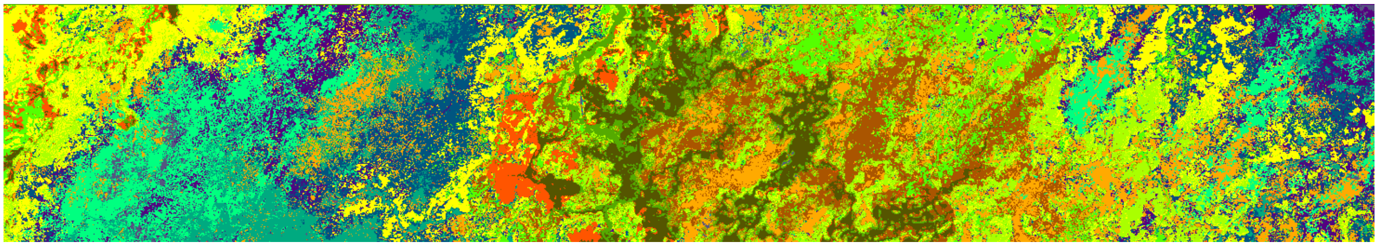
	samp.	AA% (±SD)	OA% (±SD)	Kappa (±SD)	Runtime (s) (±SD)
EMAP-PCA					
AIC	13	91.01 (±0.87)	83.90 (±1.61)	0.8214 (±0.0175)	0.1430 (±0.0060)
	30	92.55 (±0.31)	85.80 (±0.91)	0.8425 (±0.0098)	0.1686 (±0.0069)
BIC	13	90.40 (±0.85)	83.00 (±2.02)	0.8115 (±0.0222)	0.1391 (±0.0071)
	30	92.68 (±0.55)	85.93 (±1.45)	0.8440 (±0.0158)	0.1646 (±0.0065)
sil.	13	90.50 (±0.72)	82.76 (±1.31)	0.8092 (±0.0141)	0.4293 (±0.0159)
	30	92.14 (±0.42)	85.35 (±1.29)	0.8374 (±0.0140)	0.7702 (±0.0418)
gap	13	90.01 (±1.08)	81.66 (±1.69)	0.7973 (±0.0183)	38.9433 (±0.6351)
	30	91.49 (±0.87)	84.27 (±1.82)	0.8258 (±0.0199)	81.9563 (±0.8802)
VB	13	91.02 (±0.87)	84.07 (±1.60)	0.8235 (±0.0175)	**0.0579** (±0.0044)
	30	92.59 (±0.55)	86.00 (±1.01)	0.8447 (±0.0110)	**0.0802** (±0.0051)
EMAP-NWFE					
AIC	13	92.46 (±1.08)	85.93 (±2.33)	0.8435 (±0.0254)	0.1491 (±0.0049)
	30	94.38 (±0.51)	88.42 (±1.30)	0.8715 (±0.0142)	0.1734 (±0.0096)
BIC	13	93.22 (±0.60)	86.99 (±1.28)	0.8557 (±0.0140)	0.1493 (±0.0051)
	30	94.33 (±0.30)	88.90 (±0.64)	0.8767 (±0.0070)	0.1660 (±0.0058)
sil.	13	93.02 (±0.57)	85.84 (±1.96)	0.8430 (±0.0213)	0.3710 (±0.0184)
	30	93.95 (±0.44)	87.30 (±1.65)	0.8591 (±0.0179)	0.6242 (±0.0246)
gap	13	92.98 (±0.62)	86.60 (±1.05)	0.8514 (±0.0115)	30.4463 (±0.5223)
	30	94.05 (±0.42)	87.93 (±1.13)	0.8659 (±0.0124)	65.7990 (±1.5273)
VB	13	93.26 (±0.67)	87.05 (±1.03)	0.8562 (±0.0112)	**0.0610** (±0.0070)
	30	94.09 (±0.55)	88.30 (±1.31)	0.8700 (±0.0144)	**0.0818** (±0.0024)

normal EM and variational EM. Besides, in most cases the standard deviation is generally lower for Variational EM, which indicates the more accurate underlying data distribution approximation by VEM.

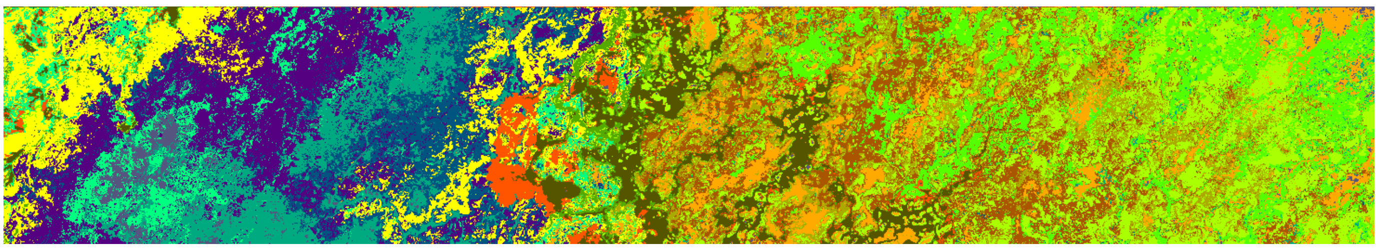
Furthermore, we compared the classification performances which are obtained via our synthetic data generation pipeline versus the ones that are computed via an optimized random forest on the original raw HS images. The results are reported



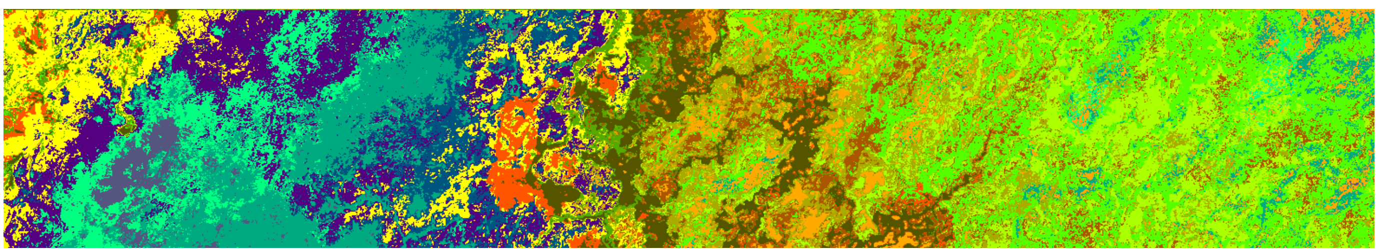
(a)



(b)



(c)



(d)

Fig. 5: Label maps on Botswana using 13 training samples per class and unoptimized random forest. (a) ground truth; (b) EMAP (OA: 88.79%, Kappa: 0.8786); (c) EMAP-PCA with 500 synthetic samples (OA: 92.72%, Kappa: 0.9212); (d) EMAP-NWFE with 500 synthetic samples (OA: 93.17%, Kappa: 0.9260).

in Table VIII. It can be observed that in all cases, our proposed pipeline results in a considerable boost in the performance, comparing to using the raw hyperspectral image.

Focusing on the runtime, it can be observed that VB is in average almost two times faster than the AIC and BIC and eight times faster than the average silhouette width method. These timing differences are visualized in the diagram in Fig. 6. The gap method is by about two orders of magnitude slower than the other methods, and therefore is not shown in the plot.

The AIC, BIC, silhouette and gap methods select among different models, there is a need to create multiple GMMs, which is not the case for VB. This is the main reason for the big runtime advantage of the Variational Bayesian.

E. Synthetic Samples for Data Augmentation in Neural Networks

Neural networks (NNs) are powerful tools in machine learning. They are capable of finding complex linear or non-linear mappings between the input and the output. Despite their power, NNs have many parameters and hence, their training requires a lot of training data. One strategy that is commonly used for increasing the size of the training data is data augmentation [65].

Synthetic sample generation can be viewed as a data augmentation strategy as it enhances the population of the training data with statistically similar samples. In order to investigate the effectiveness of our proposed variational Bayesian GMM synthetic sample generation as data augmentation in deep learning, we designed this set of experiments.

To do so, we generated a rather simple feed forward fully connected neural network with two hidden layers. We used 50 neurons in each hidden layer. In all layers, except the last layer,

TABLE VI: GMM model selection methods vs. Variational Bayesian on the SalinasA dataset.

	samp.	AA% (±SD)	OA% (±SD)	Kappa (±SD)	Runtime (s) (±SD)
EMAP-PCA					
AIC	13	98.84 (±0.32)	98.86 (±0.61)	0.9858 (±0.0075)	0.0588 (±0.0062)
	30	99.14 (±0.25)	99.15 (±0.36)	0.9894 (±0.0045)	0.0762 (±0.0105)
BIC	13	98.78 (±0.63)	98.83 (±1.17)	0.9854 (±0.0146)	0.0659 (±0.0212)
	30	99.41 (±0.25)	99.53 (±0.17)	0.9941 (±0.0021)	0.0834 (±0.0301)
sil.	13	98.64 (±0.33)	98.60 (±0.66)	0.9824 (±0.0082)	0.1695 (±0.0132)
	30	99.22 (±0.20)	99.31 (±0.29)	0.9913 (±0.0036)	0.2770 (±0.0114)
gap	13	99.00 (±0.22)	99.09 (±0.44)	0.9886 (±0.0055)	13.8983 (±0.1091)
	30	99.12 (±0.20)	99.26 (±0.26)	0.9907 (±0.0032)	33.2671 (±0.2481)
VB	13	98.63 (±0.63)	98.67 (±0.74)	0.9834 (±0.0093)	**0.0431** (±0.0194)
	30	99.13 (±0.26)	99.27 (±0.25)	0.9909 (±0.0031)	**0.0716** (±0.0186)
EMAP-NWFE					
AIC	13	98.48 (±0.72)	98.25 (±1.09)	0.9782 (±0.0135)	0.0702 (±0.0379)
	30	99.44 (±0.22)	99.44 (±0.34)	0.9930 (±0.0042)	0.0813 (±0.0202)
BIC	13	98.74 (±1.06)	98.63 (±1.55)	0.9829 (±0.0193)	0.0666 (±0.0134)
	30	99.30 (±0.46)	99.32 (±0.42)	0.9915 (±0.0053)	0.0809 (±0.0176)
sil.	13	99.01 (±0.47)	99.08 (±0.39)	0.9884 (±0.0049)	0.1606 (±0.0128)
	30	99.22 (±0.25)	99.26 (±0.31)	0.9908 (±0.0038)	0.2697 (±0.0282)
gap	13	98.97 (±0.25)	99.08 (±0.40)	0.9885 (±0.0050)	14.0880 (±0.1594)
	30	99.08 (±0.24)	99.09 (±0.38)	0.9887 (±0.0048)	34.0915 (±0.6003)
VB	13	99.38 (±0.30)	99.33 (±0.48)	0.9916 (±0.0060)	**0.0461** (±0.0218)
	30	99.52 (±0.14)	99.55 (±0.16)	0.9944 (±0.0020)	**0.0753** (±0.0221)

TABLE VII: GMM model selection methods vs. Variational Bayesian on the Botswana dataset.

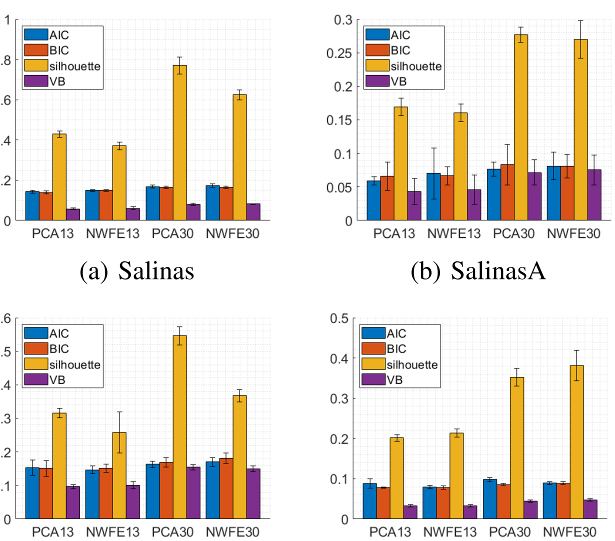
	samp.	AA% (±SD)	OA% (±SD)	Kappa (±SD)	Runtime (s) (±SD)
EMAP-PCA					
AIC	13	93.60 (±0.51)	92.93 (±0.64)	0.9234 (±0.0069)	0.1532 (±0.0234)
	30	95.61 (±0.46)	95.09 (±0.54)	0.9468 (±0.0059)	0.1630 (±0.0097)
BIC	13	93.25 (±0.67)	92.80 (±0.71)	0.9220 (±0.0076)	0.1505 (±0.0231)
	30	95.55 (±0.43)	95.07 (±0.47)	0.9466 (±0.0051)	0.1690 (±0.0139)
sil.	13	93.74 (±0.92)	93.21 (±1.13)	0.9265 (±0.0123)	0.3162 (±0.0143)
	30	95.52 (±0.47)	94.93 (±0.52)	0.9451 (±0.0056)	0.5463 (±0.0271)
gap	13	93.27 (±0.83)	92.56 (±0.95)	0.9194 (±0.0102)	25.6284 (±0.1868)
	30	94.79 (±0.40)	94.16 (±0.39)	0.9367 (±0.0042)	56.2470 (±0.8254)
VB	13	93.35 (±0.81)	92.73 (±0.95)	0.9212 (±0.0102)	**0.0965** (±0.0062)
	30	95.76 (±0.55)	95.25 (±0.62)	0.9486 (±0.0067)	**0.1541** (±0.0074)
EMAP-NWFE					
AIC	13	93.42 (±0.72)	92.82 (±0.88)	0.9222 (±0.0095)	0.1466 (±0.0116)
	30	95.56 (±0.36)	95.01 (±0.34)	0.9459 (±0.0037)	0.1700 (±0.0126)
BIC	13	93.77 (±0.66)	93.27 (±0.59)	0.9271 (±0.0064)	0.1515 (±0.0123)
	30	95.42 (±0.39)	94.79 (±0.44)	0.9436 (±0.0048)	0.1808 (±0.0154)
sil.	13	93.59 (±0.51)	93.05 (±0.54)	0.9247 (±0.0058)	0.2584 (±0.0616)
	30	94.42 (±0.49)	93.79 (±0.52)	0.9328 (±0.0056)	0.3677 (±0.0180)
gap	13	93.45 (±0.79)	92.93 (±0.60)	0.9234 (±0.0065)	25.4886 (±0.6314)
	30	94.93 (±0.21)	94.17 (±0.26)	0.9368 (±0.0028)	56.8894 (±0.2117)
VB	13	93.64 (±0.62)	93.03 (±0.74)	0.9245 (±0.0080)	**0.1008** (±0.0104)
	30	95.64 (±0.33)	95.10 (±0.38)	0.9469 (±0.0041)	**0.1493** (±0.0087)

we used rectified linear units (RELU) as activation functions with a Sigmoid as the activation function of the last layer. As for the regularizer, we used dropout [66] with dropout fraction set to 20%. We used ADAM as the optimizer with learning rate $= 0.001$, and binary cross entropy as the loss function. We trained our model for 75 epochs.

We fed EMAP-PCA and EMAP-NWFE to the network, once without synthetic samples and once after adding 500 synthetic samples. Furthermore, to compare the quality of the synthetic samples that are generated via normal EM and variational Bayesian EM (VBEM), we computed the results using both EM and VBEM. Finally, for the sake of comparison, we report the performance of the neural network on the raw hyperspectral image. The classification results on Pavia Centre, Salinas, SalinasA and Botswana datasets are presented in Tables IX, X, XI and XII, respectively. It can be observed that in all cases, addition of synthetic samples increases the performance. Moreover, in most cases, VBEM outperforms the EM algorithm.

TABLE VIII: Quantitative comparison of the classification performances, obtained by VB in the proposed synthetic sample generation pipeline and the optimized classifier using the raw HS data. The classifier is a random forest (RF). Optimized and unoptimized RF are indicated by "o" and "u", respectively. The training set size is 13 pixels per class. $|S|$ represents the number of added synthetic samples in the case of an unoptimized RF.

Algorithm	RF	$\|S\|$	AA% (±SD)	OA% (±SD)	Kappa (±SD)
Pavia Centre					
HS raw	o	-	83.98±0.81	89.83±1.22	0.8583±0.0163
VB EMAP-PCA	u	500	87.65±1.54	94.50±0.73	0.9225±0.0101
VB EMAP-NWFE	u	500	88.44±2.76	95.74±1.14	0.9358±0.0159
Salinas					
HS raw	o	-	87.93±1.07	80.65±1.57	0.7854±0.0174
VB EMAP-PCA	u	500	92.43±0.78	85.82±1.69	0.8428±0.0185
VB EMAP-NWFE	u	500	93.06±0.56	87.33±0.83	0.8596±0.0093
SalinasA					
HS raw	o	-	95.40±0.95	94.90±1.25	0.9363±0.0155
VB EMAP-PCA	u	500	98.89±0.52	99.09±0.46	0.9886±0.0058
VB EMAP-NWFE	u	500	99.25±0.36	99.25±0.33	0.9906±0.0042
Botswana					
HS raw	o	-	81.60±1.06	79.84±0.90	0.7819±0.0098
VB EMAP-PCA	u	500	93.61±0.52	92.90±0.53	0.9231±0.0058
VB EMAP-NWFE	u	500	93.95±0.82	93.37±0.83	0.9282±0.0090



(a) Salinas (b) SalinasA

(c) Botswana (d) Pavia Centre

Fig. 6: Runtimes in seconds for EMAP-PCA and EMAP-NWFE, computed over (a) Salinas, (b) SalinasA, (c) Botswana, and (d) Pavia Centre datasets, using 13 and 30 samples per class. It can be observed that for all the variants, VB's runtime is less than the other algorithms under study.

The training and validation loss for the aforementioned four datasets for the first 50 epochs are depicted in Fig. 7. As we had limited training data, we did not use a separate validation set during the training and used all the test data as the validation set. In other words, the validation loss in this figure represents the evolution of the network's capability in classifying the test set. It can be observed that in all cases, adding synthetic samples results in a faster decrease of the loss, and a lower final loss. For example, in the case of EMAP-NWFE in the SalinasA dataset, i.e. Fig. 7-(k) and Fig. 7-(l), without the synthetic samples, the loss value reached to around 2 after 50 epochs. However, after adding 500 synthetic samples, the loss value reaches close to zero within almost 20.

TABLE IX: Classification performance of the neural network on the variants of Pavia Centre dataset, with and without adding synthetic samples, and using the conventional EM or variational EM.

Algorithm	EM type	$\|S\|$	AA%±SD	OA%±SD	Kappa±SD
13 pixels per class					
HS raw	-	0	33.48±2.26	33.48±2.26	0.1347±0.0606
EMAP-PCA	-	0	78.57±2.76	78.57±2.76	0.8310±0.0373
EMAP-PCA	EM	500	88.15±1.83	88.15±1.83	0.9273±0.0121
EMAP-PCA	VBEM	500	89.33±1.21	89.33±1.21	0.9341±0.0050
EMAP-NWFE	-	0	16.50±5.84	16.50±5.84	0.0655±0.1295
EMAP-NWFE	EM	500	83.21±0.91	83.21±0.91	0.8038±0.0060
EMAP-NWFE	VBEM	500	82.36±0.79	82.36±0.79	0.7769±0.0201
40 pixels per class					
HS raw	-	0	75.85±5.32	75.85±5.32	0.8153±0.0487
EMAP-PCA	-	0	89.20±1.58	89.20±1.58	0.9323±0.0088
EMAP-PCA	EM	500	90.39±1.05	90.39±1.05	0.9391±0.0087
EMAP-PCA	VBEM	500	94.33±0.65	94.33±0.65	0.9580±0.0021
EMAP-NWFE	-	0	29.84±15.19	29.84±15.19	0.1955±0.1781
EMAP-NWFE	EM	500	85.53±0.80	85.53±0.80	0.8513±0.0172
EMAP-NWFE	VBEM	500	85.82±0.37	85.82±0.37	0.8516±0.0090

TABLE X: Classification performance of the neural network on the variants of Salinas dataset, with and without adding synthetic samples, and using the conventional or variational EM.

Algorithm	EM type	$\|S\|$	AA%±SD	OA%±SD	Kappa±SD
13 pixels per class					
HS raw	-	0	15.15±3.27	15.15±3.27	0.0970±0.0321
EMAP-PCA	-	0	65.64±3.78	65.64±3.78	0.5959±0.0627
EMAP-PCA	EM	500	93.04±0.76	93.04±0.76	0.8539±0.0170
EMAP-PCA	VBEM	500	93.10±0.30	93.10±0.30	0.8598±0.0129
EMAP-NWFE	-	0	26.54±9.05	26.54±9.05	0.2209±0.1263
EMAP-NWFE	EM	500	92.39±0.95	92.39±0.95	0.8416±0.0178
EMAP-NWFE	VBEM	500	92.29±0.53	92.29±0.53	0.8494±0.0069
40 pixels per class					
HS raw	-	0	27.62±3.30	27.62±3.30	0.2039±0.0242
EMAP-PCA	-	0	83.50±3.49	83.50±3.49	0.7649±0.0480
EMAP-PCA	EM	500	94.55±0.58	94.55±0.58	0.8789±0.0160
EMAP-PCA	VBEM	500	94.23±0.61	94.23±0.61	0.8641±0.0188
EMAP-NWFE	-	0	56.62±9.22	56.62±9.22	0.4924±0.0989
EMAP-NWFE	EM	500	93.86±0.43	93.86±0.43	0.8642±0.0110
EMAP-NWFE	VBEM	500	93.81±0.59	93.81±0.59	0.8649±0.0168

VI. Conclusion

A common issue in hyperspectral remote sensing image classification is limited training data. Limited data requires special classifier tuning, which can be done in multiple ways. First, a rather conventional parameter grid search based on cross-validation can be used, which indeed significantly improves the classifier. Second, it is also possible to add synthetic samples to adapt the data to the classifier. These samples are drawn from a GMM that is fitted to the training samples. On the SalinasA and Botswana datasets, results for addition of synthetic samples are comparable or even higher than for an optimized classifier, at a lower computational cost. Furthermore, taking advantage of variational expectation maximization rather than conventional EM in the GMM fitting achieves the aforementioned improvements in a considerably faster and more efficient way.

TABLE XI: Classification performance of the neural network on the variants of SalinasA dataset, with and without adding synthetic samples, and using the conventional or variational EM.

Algorithm	EM type	$\|S\|$	AA%±SD	OA%±SD	Kappa±SD
13 pixels per class					
HS raw	-	0	37.53±4.53	37.53±4.53	0.1289±0.0471
EMAP-PCA	-	0	90.96±8.61	90.96±8.61	0.9357±0.0475
EMAP-PCA	EM	500	98.78±0.41	98.78±0.41	0.9862±0.0038
EMAP-PCA	VBEM	500	98.78±0.25	98.78±0.25	0.9850±0.0052
EMAP-NWFE	-	0	61.12±16.50	61.12±16.50	0.4580±0.1803
EMAP-NWFE	EM	500	98.85±0.32	98.85±0.32	0.9849±0.0034
EMAP-NWFE	VBEM	500	98.78±0.30	98.78±0.30	0.9862±0.0040
40 pixels per class					
HS raw	-	0	66.36±0.45	66.36±0.45	0.4090±0.0538
EMAP-PCA	-	0	98.66±0.23	98.66±0.23	0.9847±0.0034
EMAP-PCA	EM	500	98.91±0.19	98.91±0.19	0.9857±0.0044
EMAP-PCA	VBEM	500	98.87±0.14	98.87±0.14	0.9849±0.0032
EMAP-NWFE	-	0	94.99±1.02	94.99±1.02	0.9150±0.0195
EMAP-NWFE	EM	500	99.13±0.20	99.13±0.20	0.9892±0.0024
EMAP-NWFE	VBEM	500	99.01±0.14	99.01±0.14	0.9904±0.0011

TABLE XII: Classification performance of the neural network on the variants of Botswana dataset, with and without adding synthetic samples, and using the conventional or variational EM.

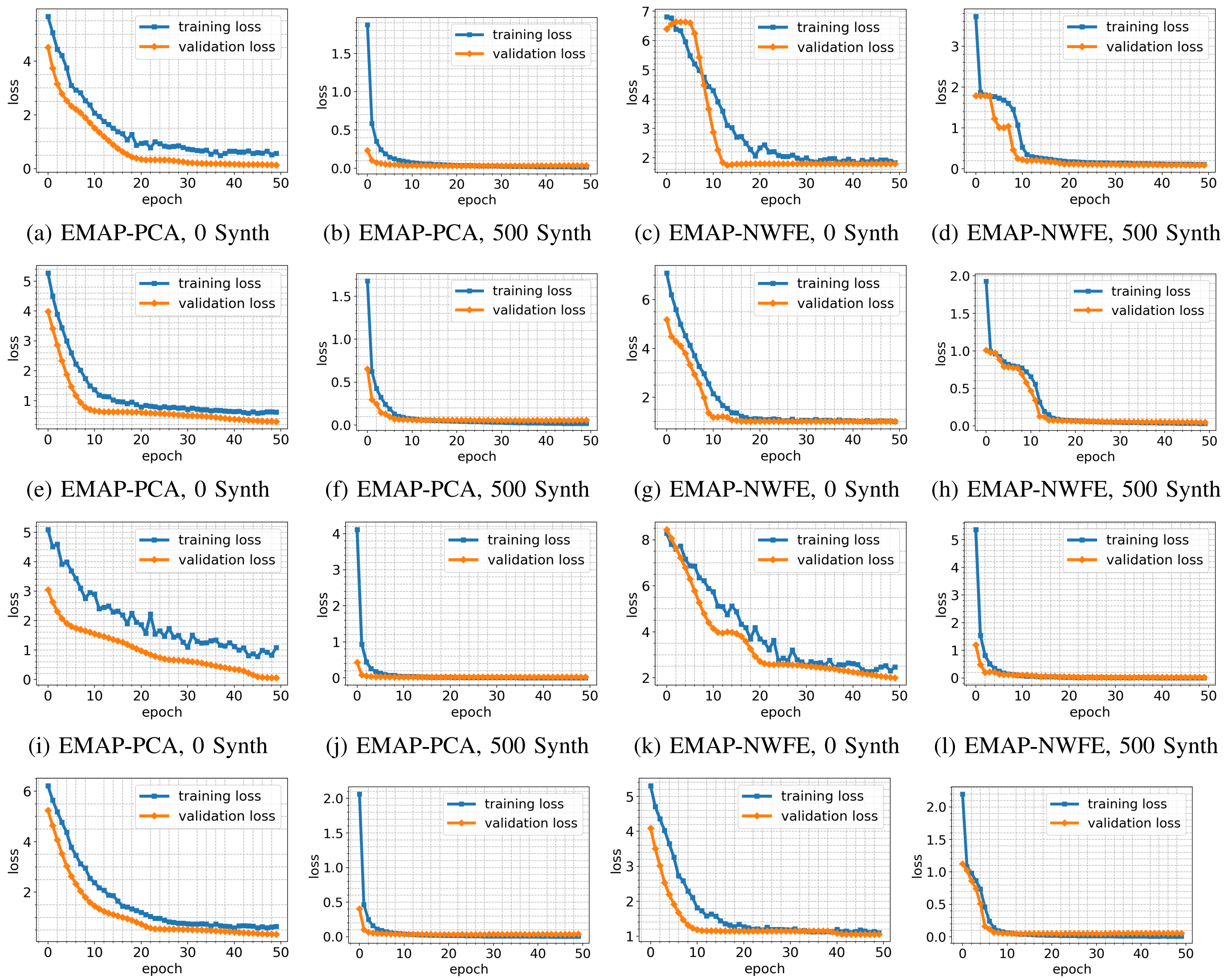
Algorithm	EM type	$\|S\|$	AA%±SD	OA%±SD	Kappa±SD
13 pixels per class					
HS raw	-	0	7.14 ±0.00	7.14 ±0.00	-0.0005±0.0010
EMAP-PCA	-	0	77.05±3.75	77.05±3.75	0.7459±0.0320
EMAP-PCA	EM	500	93.34±0.91	93.34±0.91	0.9194±0.0083
EMAP-PCA	VBEM	500	94.86±0.49	94.86±0.49	0.9377±0.0050
EMAP-NWFE	-	0	26.18±7.36	26.18±7.36	0.1800±0.0790
EMAP-NWFE	EM	500	94.16±0.23	94.16±0.23	0.9278±0.0042
EMAP-NWFE	VBEM	500	92.22±2.40	92.22±2.40	0.9142±0.0089
40 pixels per class					
HS raw	-	0	19.93±7.03	19.93±7.03	0.1237±0.0610
EMAP-PCA	-	0	93.21±0.50	93.21±0.50	0.9154±0.0046
EMAP-PCA	EM	500	96.11±0.43	96.11±0.43	0.9505±0.0055
EMAP-PCA	VBEM	500	96.45±0.37	96.45±0.37	0.9555±0.0047
EMAP-NWFE	-	0	86.50±5.15	86.50±5.15	0.8287±0.0659
EMAP-NWFE	EM	500	95.40±0.16	95.40±0.16	0.9413±0.0018
EMAP-NWFE	VBEM	500	95.78±0.21	95.78±0.21	0.9463±0.0032

Appendix

We present here the update equations for the Expectation-Maximization algorithm. For the expectation, the update is

$$q^*(\mathbf{Z}) = \mathbb{E}[\mathbf{z_{nk}}] = \mathbf{r_{nk}} \quad , \tag{6}$$

where r_{nk} denotes the “responsibility” of component k to sample n, which will be defined in Eqn. 17 further below.



(a) EMAP-PCA, 0 Synth (b) EMAP-PCA, 500 Synth (c) EMAP-NWFE, 0 Synth (d) EMAP-NWFE, 500 Synth

(e) EMAP-PCA, 0 Synth (f) EMAP-PCA, 500 Synth (g) EMAP-NWFE, 0 Synth (h) EMAP-NWFE, 500 Synth

(i) EMAP-PCA, 0 Synth (j) EMAP-PCA, 500 Synth (k) EMAP-NWFE, 0 Synth (l) EMAP-NWFE, 500 Synth

(m) EMAP-PCA, 0 Synth (n) EMAP-PCA, 500 Synth (o) EMAP-NWFE, 0 Synth (p) EMAP-NWFE, 500 Synth

Fig. 7: Training loss and validation loss of the neural network versus the number of epochs for different datasets. Each row represents one dataset. Rows one to four represent Pavia Centre, Salinas, SalinasA and Botswana datasets, respectively. It can be observed that in all the cases, adding synthetic samples helps the network to converge faster and the loss to get smaller.

Let furthermore

$$N_k = \sum_{n=1}^{N} r_{nk} \tag{7}$$

$$\bar{x}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n \tag{8}$$

$$S_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_n - \bar{x}_k)(x_n - \bar{x}_k)^T \tag{9}$$

denote three auxiliary statistics derived from r_{nk}, namely the number of assigned samples, average and covariance. The update equations for the maximization step are based on the factorization

$$q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\pi}) \prod_{k=1}^{K} q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \ . \tag{10}$$

The individual terms are

$$q^{\star}(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \ , \tag{11}$$

where **Dir** denotes the Dirichlet distribution as a prior for the mixture weights, and $\alpha_k = \alpha_0 + N_k$, where α_0 is a hyperparameter, which we heuristically set to 1.

The second factor of Eqn. 10 is represented as a product of a Gaussian distribution $\mathcal{N}$ and a Wishart distribution $\mathcal{W}$,

$$q^{*}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k \boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_k, \boldsymbol{\nu}_k) \tag{12}$$

where

$$\beta_k = \beta_0 + N_k \tag{13}$$

$$\boldsymbol{m}_k = \frac{1}{\beta_k}(\beta_0 \boldsymbol{m}_0 + N_k \bar{\boldsymbol{x}}_k) \tag{14}$$

$$\boldsymbol{W}_k^{-1} = \boldsymbol{W}_0^{-1} + N_k \boldsymbol{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{\boldsymbol{x}}_k - \boldsymbol{m}_0)(\bar{\boldsymbol{x}}_k - \boldsymbol{m}_0)^T \tag{15}$$

$$\nu_k = \nu_0 + N_k \tag{16}$$

denote the remaining parameters for the maximization step, where again ν_0 and β_0 are hyperparameters to the distribution that we heuristically set to 1.

Finally, the responsibilities r_{nk} are computed as

$$r_{nk} \propto \widetilde{\pi}_k \widetilde{\Lambda}_k^{1/2} \exp\{-\frac{D}{2\beta_k} - \frac{\nu_k}{2}(\boldsymbol{x_n} - \boldsymbol{m}_k)^T \boldsymbol{W}_k (\boldsymbol{x_n} - \boldsymbol{m}_k)\} \ , \tag{17}$$

where D denotes the feature dimensionality. Eqn. 17 makes use of the expectation

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n - \mu_k)^{\mathrm{T}} \boldsymbol{\lambda}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)] \\ &= D\beta_k^{-1} + \nu_k (\boldsymbol{x}_n - \boldsymbol{m}_k)^{\mathrm{T}} \boldsymbol{W}_k (\boldsymbol{x}_n - \boldsymbol{m}_k) \end{aligned} \tag{18}$$

and the expectations

$$\ln \widetilde{\Lambda}_k \equiv \mathbb{E}[ln\,|\boldsymbol{\Lambda}_k|] \tag{19}$$

$$\ln \widetilde{\pi}_k \equiv \mathbb{E}[ln\,\pi_k] \tag{20}$$

with

$$\ln \widetilde{\Lambda}_k = \sum_{i=1}^{D} \psi\left(\frac{\nu_k + 1 - i}{2}\right) + D\ln 2 + \ln|\boldsymbol{W}_k| \tag{21}$$

$$\ln \widetilde{\pi}_k = \psi(\alpha_k) - \psi(\sum_k (\alpha_k)) \ , \tag{22}$$

where $\psi(\cdot)$ denotes the digamma function. The EM equations are iteratively evaluated analogously to the standard EM algorithm [43].

REFERENCES

[1] S. Valero, P. Salembier, and J. Chanussot, "Hyperspectral image representation and processing with binary partition trees," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1430–1443, 2013.

[2] G. A. Shaw and H.-h. K. Burke, "Spectral imaging for remote sensing," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 3–28, 2003.

[3] D. A. Landgrebe, *Signal theory methods in multispectral remote sensing*, vol. 29. John Wiley & Sons, 2005.

[4] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2013.

[5] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 46, no. 11, pp. 3804–3814, 2008.

[6] J. A. Benediktsson, M. Pesaresi, and K. Amason, "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 41, no. 9, pp. 1940–1949, 2003.

[7] K. Tan, E. Li, Q. Du, and P. Du, "Hyperspectral image classification using band selection and morphological profiles," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 7, no. 1, pp. 40–48, 2014.

[8] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 10, pp. 3747–3762, 2010.

[9] M. Dalla Mura, J. Atli Benediktsson, B. Waske, and L. Bruzzone, "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," *International Journal of Remote Sensing*, vol. 31, no. 22, pp. 5975–5991, 2010.

[10] P. Salembier, A. Oliveras, and L. Garrido, "Antiextensive connected operators for image and sequence processing," *IEEE Transactions on Image Processing*, vol. 7, no. 4, pp. 555–570, 1998.

[11] P. Soille, "Constrained connectivity for hierarchical image partitioning and simplification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1132–1145, 2008.

[12] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[14] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2242–2251, July 2017.

[15] J. P. Hoffbeck and D. A. Landgrebe, "Covariance matrix estimation and classification with limited training data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 763–767, 1996.

[16] S. Tadjudin and D. A. Landgrebe, "Covariance estimation for limited training samples," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 5, pp. 2688–2690, 1998.

[17] M. Chi, R. Feng, and L. Bruzzone, "Classification of hyperspectral remote-sensing data with primal SVM for small-sized training dataset problem," *Advances in Space Research*, vol. 41, no. 11, pp. 1793–1799, 2008.

[18] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3363–3373, 2006.

[19] Q. Jackson and D. A. Landgrebe, "An adaptive classifier design for high-dimensional data analysis with a limited training data set," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 12, pp. 2664–2679, 2001.

[20] R. R. Vatsavai, S. Shekhar, and T. E. Burk, "A semi-supervised learning method for remote sensing data mining," in *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*, pp. 5–pp, IEEE, 2005.

[21] J. Xia, J. Chanussot, P. Du, and X. He, "Rotation-based support vector machine ensemble in classification of hyperspectral data with limited training samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1519–1531, 2016.

[22] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. A. Benediktsson, and A. Plaza, "Multiple feature learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1592–1606, 2015.

[23] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "PCA-based edge-preserving features for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7140–7151, 2017.

[24] M. Sofolahan and O. Ersoy, "Summed component analysis for dimensionality reduction and classification," Tech. Rep. 445, Purdue University, 2013.

[25] B.-C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 5, pp. 1096–1105, 2004.

[26] K. Fukunaga, *Introduction to Statistical Pattern Recognition (2Nd Ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[27] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, pp. 388–400, 1993.

[28] T. Castaings, B. Waske, J. Atli Benediktsson, and J. Chanussot, "On the influence of feature reduction for the classification of hyperspectral images based on the extended morphological profile," *International Journal of Remote Sensing*, vol. 31, no. 22, pp. 5921–5939, 2010.

[29] A. Kianisarkaleh and H. Ghassemian, "Nonparametric feature extraction for classification of hyperspectral images with limited training samples," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 64–78, 2016.

[30] M. Skurichina, S. Raudys, and R. P. Duin, "K-nearest neighbors directed noise injection in multilayer perceptron training," *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 504–511, 2000.

[31] V. E. Neagoe and A. D. Ciotec, "A new approach for accurate classification of hyperspectral images using virtual sample generation by concurrent self-organizing maps," in *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, pp. 1031–1034, July 2013.

[32] A. A. Davari, E. Aptoula, and B. Yanikoglu, "On the effect of synthetic morphological feature vectors on hyperspectral image classification performance," in *2015 23nd Signal Processing and Communications Applications Conference (SIU)*, pp. 653–656, May 2015.

[33] A. Davari, E. Aptoula, B. Yanikoglu, A. Maier, and C. Riess, "GMM-based synthetic samples for classification of hyperspectral images with limited training data," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 6, pp. 492–496, 2018.

[34] D. B. Marden and D. G. Manolakis, "Using elliptically contoured distributions to model hyperspectral imaging data and generate statistically similar synthetic data," in *Defense and Security*, pp. 558–572, International Society for Optics and Photonics, 2004.

[35] G. Camps-Valls, L. Gómez-Chova, J. Muñoz-Marí, J. L. Rojo-Álvarez, and M. Martínez-Ramón, "Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 46, no. 6, pp. 1822–1835, 2008.

[36] D. P. Williams, *Classification and data acquisition with incomplete data*. PhD thesis, Duke University, 2006.

[37] D. Williams, X. Liao, Y. Xue, L. Carin, and B. Krishnapuram, "On classification with incomplete data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 3, pp. 427–436, 2007.

[38] D. Mackay, "The evidence framework applied to classification networks," *Neural computation*, vol. 4, no. 5, pp. 720–736, 1992.

[39] A. Davari, H. C. Özkan, A. Maier, and C. Riess, "Fast sample generation with variational Bayesian for limited data hyperspectral image classification," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 6159–6162, IEEE, 2018.

[40] M. Fauvel, *Spectral and spatial methods for the classification of urban remote sensing data*. PhD thesis, Institut National Polytechnique de Grenoble-INPG; Université d'Islande, 2007.

[41] G. McLachlan and D. Peel, *Finite Mixture Models*. John Wiley & Sons, 2004.

[42] D. Reynolds, "Gaussian Mixture Models," in *Encyclopedia of Biometrics* (S. Z. Li and A. Jain, eds.), pp. 659–663, Boston, MA: Springer US, 2009.

[43] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[44] L. Guo, N. Chehata, C. Mallet, and S. Boukir, "Relevance of airborne lidar and multispectral image data for urban scene classification using random forests," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 1, pp. 56–66, 2011.

[45] E. Aptoula, "Hyperspectral image classification with multidimensional attribute profiles," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 10, pp. 2031–2035, 2015.

[46] E. Aptoula, "The impact of multivariate quasi-flat zones on the morphological description of hyperspectral images," *International Journal of Remote Sensing*, vol. 35, no. 10, pp. 3482–3498, 2014.

[47] A. Davari, V. Christlein, S. Vesal, A. Maier, and C. Riess, "GMM supervectors for limited training data in hyperspectral remote sensing image classification," in *International Conference on Computer Analysis of Images and Patterns*, pp. 296–306, Springer, 2017.

[48] E. Aptoula, M. Dalla Mura, and S. Lefèvre, "Vector attribute profiles for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3208–3220, 2016.

[49] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[50] https://goo.gl/rWsYPg.

[51] J. M. noz Marí, E. Izquierdo-Verdiguier, M. Campos-Taberner, A. Pérez-Suay, L. Gómez-Chova, G. Mateo-García, A. B. Ruescas, V. Laparra, J. A. Padrón, J. Amorós, and G. Camps-Valls, "Hyperlabelme: a web platform for benchmarking remote sensing image classifiers," 2017. V1.0.

[52] L. Tian, Q. Du, I. Kopriva, and N. Younan, "Spatial-spectral based multi-view low-rank sparse sbuspace clustering for hyperspectral imagery," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 8488–8491, IEEE, 2018.

[53] B. Pan, Z. Shi, and X. Xu, "Multiobjective-based sparse representation classifier for hyperspectral imagery using limited samples," *IEEE Transactions on Geoscience and Remote Sensing*, no. 99, pp. 1–11, 2018.

[54] F. Deng, S. Pu, X. Chen, Y. Shi, T. Yuan, and S. Pu, "Hyperspectral image classification with capsule network using limited training samples," *Sensors*, vol. 18, no. 9, p. 3153, 2018.

[55] M. Hamouda, K. S. Ettabaa, and M. S. Bouhlel, "Modified convolutional neural network based on adaptive patch extraction for hyperspectral image classification," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–7, IEEE, 2018.

[56] F. Poorahangaryan and H. Ghassemian, "A multiscale modified minimum spanning forest method for spatial-spectral hyperspectral images classification," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 71, 2017.

[57] V. Menon, Q. Du, and J. E. Fowler, "Random-projection-based non-negative least squares for hyperspectral image unmixing," in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2016 8th Workshop on*, pp. 1–5, IEEE, 2016.

[58] T. Liu, Y. Gu, X. Jia, J. A. Benediktsson, and J. Chanussot, "Class-Specific Sparse Multiple Kernel Learning for Spectral-Spatial Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 7351–7365, Dec 2016.

[59] J. Cohen, "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement*, vol. 20, pp. 37–46, Apr. 1960.

[60] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*, pp. 199–213, Springer, 1998.

[61] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[62] T. M. Kodinariya and P. R. Makwana, "Review on determining number of cluster in k-means clustering," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, no. 6, pp. 90–95, 2013.

[63] P. J. Rousseeuw and L. Kaufman, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Online Library, 1990.

[64] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[65] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[66] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, 2016.

AmirAbbas Davari received the B.Sc. degree in Electrical Engineering from the University of Tehran, Tehran, Iran, in 2011. In 2013 He received the M.Sc. degree, also in Electrical Engineering, from Özyeğin university, Istanbul, Turkey. From 2013 to 2015, he was a research assistant in Sabancı University, Istanbul, Turkey. Since 2015, he is pursuing the Ph.D. degree at the Pattern Recognition Laboratory, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany. His research interests include image processing, pattern recognition and deep learning with a focus on remote sensing and hyperspecral image analysis.

Hasan Can Özkan received the B.Sc. degree in Biomedical Engineering from Yeditepe University in 2014 and M.S degree in Medical Engineering from Friedrich-Alexander Universität Erlangen-Nürnberg. Currently, he is working as a consulting engineer. His research interests include image-processing, machine learning and computer vision.

Andreas Maier was born on 26th of November 1980 in Erlangen. He studied Computer Science, graduated in 2005, and received his PhD in 2009. From 2005 to 2009 he was working at the Pattern Recognition Lab at the Computer Science Department of the University of Erlangen-Nuremberg. His major research subject was medical signal processing in speech data. In this period, he developed the first online speech intelligibility assessment tool - PEAKS - that has been used to analyze over 4.000 patient and control subjects so far.

From 2009 to 2010, he started working on flat-panel C-arm CT as post-doctoral fellow at the Radiological Sciences Laboratory in the Department of Radiology at the Stanford University. From 2011 to 2012 he joined Siemens Healthcare as innovation project manager and was responsible for reconstruction topics in the Angiography and X-ray business unit.

In 2012, he returned the University of Erlangen-Nuremberg as head of the Medical Reconstruction Group at the Pattern Recognition lab. In 2015 he became professor and head of the Pattern Recognition Lab. Since 2016, he is member of the steering committee of the European Time Machine Consortium. In 2018, he was awarded an ERC Synergy Grant "4D nanoscope". Current research interests focuses on medical imaging, image and audio processing, digital humanities, and interpretable machine learning and the use of known operators.

Christian Riess received the Ph.D. degree in computer science from the Friedrich-Alexander University Erlangen-Nürnberg (FAU), Erlangen, Germany, in 2012. From 2013 to 2015, he was a Postdoc at the Radiological Sciences Laboratory, Stanford University, Stanford, CA, USA. Since 2015, he is the head of the Phase-Contrast X-ray Group at the Pattern Recognition Laboratory at FAU. Since 2016, he is senior researcher and head of the Multimedia Security Group at the IT Infrastructures Lab at FAU. He is currently a member of the IEEE Information Forensics and Security Technical Committee. His research interests include all aspects of image processing and imaging, particularly with applications in image and video forensics, X-ray phase contrast, color image processing, and computer vision.