# Estimating Optimal Number of Compressively Sensed Bands for Hyperspectral Classification via Feature Selection

C. J. Della Porta, *Member, IEEE*, and Chein-I Chang, *Life Fellow, IEEE*

***Abstract*—Compressive sensing (CS) has received considerable interest in hyperspectral sensing. Recent articles have also exploited the benefits of CS in hyperspectral image classification (HSIC) in the compressively sensed band domain (CSBD). However, on many occasions, the requirement of full bands is not necessary for HSIC to perform well. So, a great challenge arises in determining the minimum number of compressively sensed bands (CSBs), $n_{\mathrm{CSB}}$, needed to achieve full-band performance. Practically, the value of $n_{\mathrm{CSB}}$ varies with the complexity of an imaged scene. Although virtual dimensionality (VD) has been used to estimate the number of bands to be selected, $n_{\mathrm{BS}}$, it is not applicable to CSBD because a CSB is actually a mixture of $n_{\mathrm{CSB}}$ bands sensed by a random sensing matrix, while VD is used to estimate $n_{\mathrm{BS}}$ which is the number of single bands to be selected. As expected, $n_{\mathrm{CSB}}$ will be generally smaller than $n_{\mathrm{BS}}$. To estimate an optimal value of $n_{\mathrm{CSB}}$, two feature selection approaches, filter and wrapper methods, are proposed to extract scene features that can be used to estimate the minimum value of $n_{\mathrm{CSB}}$ required to maximize performance with minimum redundancy. Specifically, these methods are fully automated by leveraging optimal partitioning schemes which enable classification to further reduce storage requirements in CSBD. Finally, a set of experiments are conducted using real-world hyperspectral images to demonstrate the viability of the proposed approach.**

***Index Terms*—Compressively sensed bands (CSBs), compressively sensed bands domain (CSBD), compressive sensing (CS).**

## I. Introduction

HYPERSPECTRAL sensing technologies continue to find great success in many new applications. Advances in sensor design have also allowed for new, smaller sensors to be implemented on remote and autonomous platforms. The wealth of information provided by hundreds of contiguous spectral bands also comes with challenges of more stringent requirements on size, weight, and power, on-board storage, and computation. Such restrictions certainly limit what data processing can be accomplished remotely as well as in real time.

To resolve these issues, one general approach is data reduction (DR) which reduces a high-dimensional hyperspectral data cube to a low-dimensional dataset via various dimensionality reduction techniques [1]. One commonly used DR is the use of data transformations, referred to as spectral-transformed DR (STDR), which represents the original data cube in a lower dimensional data space specified by a particularly designed transformation as a finite number of components such as principal components analysis [1], maximum noise fraction [1], independent component analysis [2], and nonlinear transformation [3]. In particular, many deep learning (DL) methods have been applied to hyperspectral DR [4]. For example, DL methods use an approach of error reconstruction of estimating the best components in hyperspectral data such as autoencoder [4]. Since these structures are complicated and requires many training data, Makantasis *et al.* introduced methods of using tensors where the main principal components are identified as a reconstruction error between the test data and the data learned by the network model [5] and later also proposed a common mode patterns method by taking the information of labels into account to ensure that tensor objects that belong to different classes do not share [6]. However, DL has issues regarding explainability, since the whole model is treated as a black box and also requires tuning many model parameters empirically without providing constructive guidelines. In addition, the main issue resulting from STDR is that the original spectral band information will be lost and compromised. For hyperspectral imaging, preserving original spectral information is crucial and STDR has mixed spectral information by transformations.

Another DR approach is band selection (BS) which selects a set of appropriate bands to represent the original data cube with spectral data fidelity, while discarding all unselected spectral bands. As a result, BS does not use the information provided by all spectral bands as STDR does. Due to the fact that BS retains the data integrity of each of selected bands, BS is generally preferred to STDR in hyperspectral data exploitation. Accordingly, many BS methods have been developed and reported in the literature. In general, to best utilize BS, an application must

Manuscript received June 30, 2021; revised August 23, 2021 and September 28, 2021; accepted November 8, 2021. Date of publication November 16, 2021; date of current version December 1, 2021. The work of Chein-I Chang was supported by the Fundamental Research Funds for Central Universities under Grant 3132019341. (*Corresponding author: Chein-I Chang*.)

C. J. Della Porta is with the University of Maryland Baltimore County, Baltimore, MD 21250 USA (e-mail: dellac1@umbc.edu).

Chein-I Chang is with the Center for Hyperspectral Imaging in Remote Sensing Information and Technology College, Dalian Maritime University, Dalian 116026, China, and also with the Remote Sensing Signal and Image Processing Laboratory, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore, MD 21250 USA, and also with the Department of Computer Science and Information Management, Providence University, Taichung 02912, Taiwan (e-mail: cchang@umbc.edu).

Digital Object Identifier 10.1109/JSTARS.2021.3128288

be identified so that a specific BS strategy can be designed, such as the classification problem considered in this article. Many BS methods have been proposed for classification in recent years. For band clustering, Wang *et al.* [7] proposed a general framework for hyperspectral BS which adopts an adaptive subspace partition strategy to partition a hyperspectral image cube into multiple subcubes by maximizing the ratio of interclass distance to intraclass distance. Wang *et al.* [8] further proposed a fast neighborhood grouping method for hyperspectral BS to partition a hyperspectral image cube into several groups using coarse-fine strategy and then obtain the most relevant and informative bands simultaneously as a subset in accordance with two factors, such as local density and information entropy. Using mutual information (MI), Torres *et al.* [9] developed a spatial spectral MI BS method that utilizes a spatial feature extraction technique as a preprocessing step, followed by the clustering of the MI of spectral bands for enhancing the efficiency of the BS. Chang *et al.* [10] recently derived a new approach, referred to as self-MI-based BS for hyperspectral image classification (HSIC). For graph-based representation, Yuan *et al.* [11] developed a multigraph determinantal point process approach. Cai *et al.* [12] proposed an efficient graph convolutional self-representation BS method by incorporating graph convolution into the self-representation model. Sui *et al.* [13] proposed a robust unsupervised BS method to reveal bandwise representativeness based on the comprehensive interband neighborhood structure. For DL network, Feng *et al.* [14] considered hyperspectral BS as a reinforcement learning problem where a semi-supervised convolutional neural network, called EvaluateNet, is constructed by adding the intraclass compactness constraint of both limited labeled and sufficient unlabeled samples. For band subset selection (BSS), Yu *et al.* [15] investigated a BSS approach to HSIC which utilized a linearly constrained minimum variance to select multiple bands simultaneously as a band subset, referred to as simultaneous multiple BS rather than one band at a time sequentially as most traditional BS methods do. Yu *et al.* [16] later developed a novel approach, called class signature-constrained background suppression approach to BS for HSIC where class signatures can be obtained either by *a priori* or *a posteriori* knowledge or training samples and BKG suppression can be accomplished by the sample correlation matrix $\mathbf{R}$. Unfortunately, similar to STDR, BS also faces two challenging issues, how to determine the number of spectral bands to be selected, $n_{\mathrm{BS}}$, and how to select appropriate spectral bands once $n_{\mathrm{BS}}$ is known.

A third approach is compressive sensing (CS) that was recently introduced in [20]. It is CS-based DR with the concept of CS reviewed in [18]–[20]. In particular, the universality model was explored for HSIC in [21]–[25] via the compressively sensed band domain (CSBD). Several other applications were also investigated, restricted entropy property [26], [27], endmember finding [28], [29], unmixing [30], BS [31], [32], and target detection [33]. Interestingly, CSBD offers a new look into DR. It can be considered as a hybrid of STDR and BS in the sense that it works as STDR without specifying any particular spectral transformation and also as BS without specifying a BS strategy to randomly selected bands.

Motivated by recent articles which took advantage of the university model in [20] to develop CBSD, this article explores the potential of CBSD by integrating STDR and BS into one approach which combines both advantages.

As recalled, DR makes use of transformations to reduce the entire set of spectral bands to a number of reduced dimensions by a small number of components, $n_{\mathrm{SD}}$ to represent the original data cube, while BS only selects a desired set of appropriate spectral bands to represent the entire spectral bands. Therefore, theoretically speaking, $n_{\mathrm{SD}}$ should not be greater than $n_{\mathrm{BS}}$ selected by BS. On the other hand, since CSBs take advantage of its particular properties, maximum incoherence and sparse representation to randomly sense $m_{\mathrm{opt}}$ bands from the full band set to perform data conversion, the number of CSBs, $m_{\mathrm{opt}}$ is expected to be no greater than $n_{\mathrm{STDR}}$. Furthermore, CSBs are random mixtures of $m_{\mathrm{opt}}$ sensed bands. Therefore, $m_{\mathrm{opt}}$ is expected to be smaller than $n_{\mathrm{BS}}$. Accordingly, from a DR point of view, CSBs are randomly generated by sensing $m_{\mathrm{opt}}$ mixtures of full spectral bands. On the other hand, from a BS point of view, CSBs only uses $m_{\mathrm{opt}}$ bands randomly sensed from and mixed by the entire full band set. Most importantly, CSBs are sensed and obtained during data acquisition, which can be particularly implemented in a designed sensing hardware that can greatly reduce the computational requirements, while still maintaining a high level of algorithm performance.

There are several distinct differences of CSBD from STDR and BS. First of all, CSBD randomly senses spectral bands via a Gaussian or Bernoulli random matrix compared to BS which selects deterministic bands. From this point of view, CSBD can be regarded as a dynamic BS as opposed to BS to be considered as a static BS. Second of all, each CSB is actually a mixture of full spectral bands via randomly sensed spectral bands. In other words, CSBD offers a new insight into DR and can be considered as a hybrid of STDR and BS in the sense that it works as STDR without specifying any particular spectral transformation as well as works as BS without specifying a BS strategy to randomly selected bands. Finally, both STDR and BS need to select components by a specific transformation or spectral bands by a custom-designed BS method, but CBSD does not. This is because CSBD generates the sensing matrix based on the two key CS properties, maximum inherence and sparsity that can achieve sampling rate lower than the Nyquist rate. As anticipated, CSBD generally requires fewer bands than STDR and BS do to achieve the same level of performance.

The main focus of our article is to predict the minimum number of bands that are required by a compressively sampled system to maintain full-band classification performance. This task, while related, is markedly different from BS and traditional DR.

For CSBD to work effectively, the only remaining issue is how to determine an optimal number of CSBs, $m_{\mathrm{opt}}$. Recently, the virtual dimensionality (VD) developed in [1], [34]–[37] has been used to estimate an appropriate value of $n_{\mathrm{BS}}$. It seems that VD may be also used to estimate $m_{\mathrm{opt}}$. However, it should be noted that VD was originally developed to estimate the number of single spectrally signatures as pure signatures and

a CSB is actually a mixture of full bands sensed by a random sensing matrix. Consequently, it is expected that $m_{\rm opt}$ should be generally smaller than $n_{\rm BS}$ due to the fact that a CSB is sensed from full bands compared to a BS-selected band which is only and simply a single band. With this in mind VD may be applicable to estimating $m_{\rm opt}$ but may not produce a desired value of $m_{\rm opt}$. In this case, instead of appealing for VD, this article develops a feature selection-based approach to determining $m_{\rm opt}$ for HSIC based on a given set of training data.

In [24], HSIC was performed in CSBD and the effects of scene complexity were investigated where the experimental results demonstrated that $m_{\rm opt}$ varied with both individual classes and scene complexity. This observed variability motivated an interesting issue in how an appropriate number of CSBs can be selected. For BS, the data is considered as a whole and selects the same bands for classification of all the classes. However, since a class has its own characteristics and requires a different number of bands to specify its class. Using the same number of bands to specify all of classes is not realistic unless all classes are considered equally significant. To address this issue, Song *et al.* [38] is believed to be the first work to propose an idea of using self-information arising in information to determine the number of bands required to process a class. It took advantage of class features developed in [39] to calculate the significance of each class as a probability from which the self-information of each class can be measured as class information to be used as a guide for selecting its required number of bands. Inspired by this approach, each CSB is considered as a feature. Since CSBs are sensed in the sense of maximum incoherence, each sensed band will be considered equally significant. In this case, only the number of CSBs matters. This is similar to uniform BS widely used in BS which generally performs well provided that $n_{\rm BS}$ is appropriately sufficient. So, the main issue for CSBs is reduced to determination of $m_{\rm opt}$ for each class.

For a compressed hyperspectral classification problem, CSBs can be considered as classification features in which case $m_{\rm opt}$ corresponds to the optimal number of features. As a result, the task of estimating $m_{\rm opt}$ can then be considered as a feature selection problem. More specifically, feature selection is an approach which selects a set of features with the goal of removing redundancy or maximizing relevance. In a typical feature selection problem, there are $n$ disparate classification features that must be chosen from, each of which is independent and has varying, unknown levels of discriminatory power. The challenge is to determine an appropriately sized subset of features which are the most effective for classification. This becomes an NP-complete problem where each of the subsets must be searched exhaustively.

Given that the CSBs are sensed incoherently, all of the features can be considered to have equal discriminatory power. More specifically, we can simply state that the nature of the incoherent sampling will cause all possible subsets of a fixed size to be approximately equivalent. This greatly reduces the problem because it implies that only the number of features needs to be determined rather than a specific combination of features. This is simply due to the fact that CBSs are randomly selected with maximum possible inherence. As a consequence, using CBSs can significantly reduce the problem complexity and also remove the computational burdens that would otherwise limit the practicality of an exhaustive technique.

There are three general categories of supervised feature selection algorithms: filter methods, wrapper methods, and embedded methods. A detailed review of such feature selection algorithms can be found in [40]. One one end, filter methods are independent of a particular classifier and based on characteristics of the used dataset. They are typically fast because classifiers do not need to be trained on various sets of features. On another end wrapper methods are a brute force approach in which a particular classifier is trained using various sets of features and performance metrics such as classification accuracy or precision is used as an optimal feature set. This approach is typically very slow for general feature selection. Embedded methods are a combination of filter and wrapper methods, where data characteristics are used to limit the selection of features and then wrapper methods are applied to ensure that maximum performance is achieved. Embedded methods are often applied to achieve the accuracy of wrapper methods but with a reduced search time.

Additionally, each pixel may also require a different $n_{\rm CSB}$. In this case, a new CS metric for hyperspectral images, referred to as compressively sensed pixel number ratio (CSPNR), is defined and shown to be a convenient representation of compression performance. This metric facilitates the comparison of compression performance across different images and is easily relatable to file sizes and hardware storage.

Several contributions made in this article are summarized as follows:

1) Two feature extraction algorithms are proposed for estimating $m_{\rm opt}$ needed to achieve full classification performance. One is classifier independent and is based solely on characteristics of the training data. The other is classifier dependent and directly leverages the training accuracy of a trained classifier.
2) The proposed algorithms are fully automated by adopting an optimal partitioning approach. Such algorithm automation allows users to estimate individual minimum number of CSBs for each class.
3) A new pixel-based metric, CSPNR, is introduced to measure pixel performance in terms of $n_{\rm CBS}$.

## II. HSIC in Compressively Sensed Band Domain

CS has emerged as a new technique in communications/signal processing. Since its inception in early 2000s, CS has found its great success in many different disciplines. The central goal of CS is to sample data under the Nyquist rate, while still being able to recover a continuous signal that can be reconstructed without distortion. In order to achieve this goal, it must satisfy two crucial and important properties: maximum incoherence and sparsity. Maximum incoherence enables the signals to be sampled randomly under the Nyquist rate with no signal aliasing, while sparsity allows signals to be represented by only a few nonzero components. In other words, CS allows signals to be

directly acquired in a compressed form and then later uncompressed using a sparse recovery process. Due to its potential in reducing hardware costs, on-board storage, computational costs, and communications bandwidths, CS draws considerable interest in many applications. Most recently, CS has also found many applications in hyperspectral data exploitation.

### A. CSBD

A hyperspectral image, $\mathrm{HSI} \in \Re^{N_x \times N_y \times L}$ consists of two spatial dimensions, $N_x$ and $N_y$ and a spectral dimension $L$. A CSBD image, $\mathrm{HSI}_{\mathrm{CSBD}} \in \Re^{N_x \times N_y \times m}$ can be formed by sensing $m$ linear mixtures of the original spectral bands. The compressed acquisition of a single hyperspectral pixel vector can be mathematically represented using the following model:

$$\mathbf{y} = \mathbf{\Phi \Psi r} + \mathbf{n} \tag{1}$$

where $\mathbf{y} \in \Re^{m \times 1}$ is the compressed measurement vector, $\mathbf{\Phi} \in \Re^{m \times L}$ is a sensing matrix, $\mathbf{\Psi} \in \Re^{L \times L}$ is the sparse representation basis, $\mathbf{r} \in \Re^{L \times 1}$ is a hyperspectral pixel vector, and $\mathbf{n} \in \Re^{m \times 1}$ is a noise term. The model in (1) requires that the signals are sparse and sensed incoherently [20]. The challenge in designing a CS system is choosing $\Phi$ and $\Psi$ to satisfy these requirements for a given application. Previous work [21]–[33] showed that a hyperspectral model, well-suited for analysis in the compressed domain, can be developed by using random sensing matrices. More specifically, $\Phi$, is generated from any random distribution that satisfies the concentration of measure inequality [20]. Such random sensing matrices have been shown to be incoherent with any representation basis, $\Psi$, and also satisfy the restricted isometry property (RIP) that is often desired in CS applications [17]–[20], [26], [27]. RIP guarantees that relative measures of distance are preserved in the compressively sensed space, effectively making the sensing matrix an orthonormal projector $\mathbf{\Phi \Phi}^T \approx \mathbf{I}$. According to the "*universality*" property of random sensing described in [20] the sparse basis, $\Psi$ is only needed for signal recovery. Since HSIC does not require data reconstruction, this guarantees that all the needed information is captured in CSBD provided that a sufficient number of CSBs is ensured. Consequently, the sensing matrix $\Phi$ is the main consideration in the CS model design. In other words, the commonly used CS model in (1) can be further simplified to

$$\mathbf{y} = \mathbf{\Phi}_m \mathbf{r} + \mathbf{n} \tag{2}$$

where $\Phi_m$ represents a random sampling matrix using $m$ CSBs. To be more precise, (2) can be re-expressed as

$$\mathbf{y}_{\mathbf{\Omega}_{\mathrm{CSBD}}} = \mathbf{\Phi}_m \mathbf{r}_{\mathbf{\Omega}_{\mathrm{B}}} + \mathbf{n}_{\mathbf{\Omega}_{\mathrm{CSBD}}} \tag{3}$$

where $\mathbf{y}$ and $\mathbf{r}$ are defined on $\Omega_{\mathrm{CSBD}}$ and the original data cube with full band set $\Omega_{\mathrm{B}}$, respectively.

### B. Classification in CSBD

A classifier is basically a class membership assignment function $f(\mathbf{r})$ that assigns an HSI pixel $\mathbf{r}$ to one of $P$ possible classes, $\{C_p\}_{p=1}^{P}$, resulting in a classification map, $\mathrm{CMap} \in \Re^{N_x \times N_y}$. Based on the simplified CS model (3), Della Porta *et al.* [24] showed that it was possible to derive a CSBD-based classifier, $\delta_{\mathrm{CSBD}}(\mathbf{r})$ which would converge to full band performance for a given sufficient number of CSBs. Specifically, the support vector machine (SVM) was chosen as the spectral classifier and a guided edge-preserving filter (EPF) was used to incorporate the spatial context of the image. The EPF-based method in [41] is adopted as the classifier to be studied in this article with specific details referred to the CSBD adaptation performed in [24], where SVM was shown both analytically and empirically to converge to full band performance.

## III. Feature Section for CSBs

In compressed hyperspectral classification problems, CSBs can be considered as classification features in which case $m_{\mathrm{opt}}$ corresponds to the optimal number of features. As a result, the task of estimating $m_{\mathrm{opt}}$ can then be considered as a feature selection problem. Feature selection is an approach which selects a set of features with the goal of removing redundancy or maximizing relevance. In a typical feature selection problem, there are $n$ disparate classification features that must be chosen from, each of which is independent and has varying, unknown levels of discriminatory power. The challenge is to determine an appropriately sized subset of features which are the most effective for classification. This becomes an NP-complete problem where each of the subsets must be searched exhaustively.

Given that the CSBs are sensed incoherently, all of the features can be considered to have equal discriminatory power. More specifically, we can simply state that the nature of the incoherent sampling will cause all possible subsets of a fixed size to be approximately equivalent. This greatly reduces the problem because it implies that only the number of features needs to be determined rather than a specific combination of features. This is simply due to the fact that CBSs are randomly selected with maximum possible incoherence. As a consequence, using CBSs can significantly reduce the problem complexity and also remove the computational burdens that would otherwise limit the practicality of an exhaustive technique.

There are three general categories of supervised feature selection algorithms: filter methods, wrapper methods, and embedded methods [40]. Filter methods are independent of a particular classifier and based on characteristics of the used dataset. They are typically fast because classifiers do not need to be trained on various sets of features. Wrapper methods are a brute force approach in which a particular classifier is trained using various sets of features and performance metrics such as classification accuracy or precision is used to select an optimal feature set. This approach is typically very slow for general feature selection. Embedded methods are a combination of filter and wrapper methods, where data characteristics are used to limit the selection of features and then wrapper methods are applied to ensure that maximum performance is achieved. Embedded methods can be considered as a compromise between these two and are often applied to achieve the accuracy of wrapper methods but with a reduced search time.

In this section, filter and wrapper methods are studied for estimating $m_{\mathrm{opt}}$.

### A. Filter Method

Filter methods are based on calculating measures of effectiveness for classification features, without directly applying them to a classifier. This benefit offers their potential in extracting features that are universally effective for any type of a classifier. However, this filter-based approach also raises a challenge of identifying an appropriate measure of effectiveness for each feature. The choice of such a measure is often dependent upon the type of data and also its discriminatory power needed in classification which is not always straightforward.

In general, filter methods can be further classified into univariate and multivariate. In the univariate approach, each feature is considered one at a time, making implementation simple and also reduces the overall search space. A typical univariate filter approach consists of individually scoring each feature and then choosing the highest score-ranked features to be used for classification. A disadvantage to the univariate approach is that it is incapable of identifying the combined discriminatory power of multiple features since they are only considered one at a time. Additionally, a univariate approach also overlooks the correlation among features and is unable to recognize redundancy between the features.

A multivariate approach considers features in batches giving it the ability to solve the two challenges faced by a univariate approach. Multivariate approaches consider both the number and specific combination of features being considered jointly for classification. The task of estimating the optimal number of CSBs, $m_{\text{opt}}$, can be considered as a special case of a multivariate filter approach, where $m_{\text{opt}}$ is a batch size. Furthermore, since each CSB is sensed incoherently, all CBSs will be considered to have approximately the same discriminatory power (i.e., one CSB is no more informative than another), meaning that the priority of CSBs can be ignored. The training dataset to be used for filter methods must be grouped into individual classes, denoted as $S^p_{\text{train}} = \{\mathbf{r}^p_k\}^{n_p}_{k=1}$. A representative vector, $\mathbf{z}_p$ can be calculated for each class and selected to form a class representation matrix $\mathbf{Z} = [\mathbf{z}_1\mathbf{z}_2, \ldots, \mathbf{z}_P]$. Such representative vectors can be generated in many ways. However, an obvious and meaningful approach is to simply calculate the mean for each class

$$\mathbf{z}_p = \frac{1}{n_p}\sum\nolimits_{k=1}^{n_p} \mathbf{r}^p_k \tag{4}$$

where $n_p$ is the number of samples in class $p$. The representative matrix, $\mathbf{Z}_m = \mathbf{\Phi}_m\mathbf{Z}$ with $m$ CSBs can be generated. A distance measure $\zeta$ is then used to calculate the normalized distance between any pair of class representative pixel vectors among $P$ classes to produce a distance matrix:

$$\begin{bmatrix} \varsigma_{11} & \varsigma_{12} & \cdots & \varsigma_{1P} \\ \varsigma_{11} & \varsigma_{11} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \varsigma_{(P-1)P} \\ \varsigma_{P1} & \cdots & \varsigma_{P(P-1)} & \varsigma_{PP} \end{bmatrix} \tag{5}$$

where

$$\varsigma(\mathbf{z}_k, \mathbf{z}_l) = \varsigma_{kl} \tag{6}$$

**Algorithm 1:** Filter-Based Feature Selection.

**Input**: Training data set, $S_{\text{train}} = \cup_{p=1}^{P} S^p_{\text{train}}$.

1. Summarize each class into $P$ representative vectors, $\mathbf{z}_p$ one for each class.
2. Choose an initial number of CSBs, $m = m_{\min}$ and a CSB step size $\Delta$.
3. Randomly generate a compressive sampling matrix, $\mathbf{\Phi} \in \Re^{m\times L}$.
4. Compressively sample the summary statistic vectors, $\mathbf{Z}_m = \mathbf{\Phi}\mathbf{Z}^T$
5. Calculate a measure of distance to form a distance matrix, $[\varsigma_{kl}]^{P,P}_{k=1,l=1}$ (4) between all pairs of class representative pixel vectors.
6. Set $m = m + \Delta_m$ and repeat steps 3 through 5 until $m \geq L$.
7. Chose the value of $m$ where the distance measure converges.

**Output**: The estimated minimum number of CSBs: $m_{\text{opt}}$ and a set of trained classifiers, $\{\delta_{m+\Delta}(\mathbf{r}), \delta_{m+2\Delta}(\mathbf{r}), \cdots, \delta_{m_{\max}}(\mathbf{r})\}$

denotes the normalized distance between two vectors $\mathbf{z}_k$ and $\mathbf{z}_l$. This process can then be repeated for all values of $m$ so that $m_{\text{opt}}$ can then be selected by observing the point at which adding additional features (i.e., CSBs) no longer significantly affects the distance measure. A general filter-based approach feature extraction algorithm is summarized in Algorithm 1.

Once each class has been reduced to a representative vector, a distance measure defined by (6) can be used to calculate closeness between any pair of two individual classes. The diagonal of the distance matrix in (5) will have to be zero, since a pixel vector will always be perfectly similar to itself. Furthermore, the distance metric is generally symmetric, i.e., $\varsigma_{kl} = \varsigma_{lk}$, then the matrix in (4) is always symmetric, resulting in only $P(P-1)/2$ unique distance values.

To further reduce the distance metric to a single per-class metric, we can define the average class distance for class $p$

$$\bar{\varsigma}_p = \frac{1}{P-1}\sum\nolimits_{k=1}^{P} \varsigma_{kp} \text{ with } \varsigma_{pp} = 0 \tag{7}$$

which provides a simple way of observing the effect of varying the number of CSBs.

Since various hyperspectral distance metrics are used to quantify the impact of increasing the number of CSBs, it is imperative that these metrics be normalized by the total number of used CBSs to allow for a fair comparison. Five different distance metrics are of particular interest: normalized Euclidean distance (NED), normalized variance distance (NVD), spectral variance angle (SVA), Pearson variance coefficient (PVC), and spectral information divergence (SID) [42].

The Euclidean distance between class means is a suitable choice for a distance metric since it provides an easily understood measure of class separation. Unfortunately, the standard Euclidean distance is unbounded and will potentially grow monotonically as the number of dimensions is increased. To

consider the Euclidean distance as a tractable metric, it must be normalized to ensure bounded outputs. This gives rise to the NED given by

$$\mathrm{NED}(\mathbf{z}_k, \mathbf{z}_l) = \frac{||\mathbf{z}_k - \mathbf{z}_l||^2}{||\mathbf{z}_k||^2 + ||\mathbf{z}_l||^2} \leq 1. \tag{8}$$

In addition, we can also modify (8) by subtracting its mean from each pixel vector $\mathbf{r}$ to yield an NVD defined by

$$\mathrm{NVD}(\mathbf{r}_k, \mathbf{r}_l) = \frac{||\left(\mathbf{r}_k - \mu_k \mathbf{1}\right) - \left(\mathbf{r}_l - \mu_l \mathbf{1}\right)||^2}{||\mathbf{r}_k - \mu_k \mathbf{1}||^2 + ||\mathbf{r}_l - \mu_l \mathbf{1}||^2} \leq 1 \tag{9}$$

where $\mathbf{r}_k$ and $\mathbf{r}_l$ are pixel vectors with their corresponding means, $\mu_k = \frac{1}{L}\sum_{j=1}^{L} r_{kj}$ and $\mu_l = \frac{1}{L}\sum_{j=1}^{L} r_{lj}$, subtracted and $\mathbf{1} = (1, 1, \ldots, 1)^T$ is an $L$-dimensional unity vector with all ones in its components.

Spectral angle is also a metric that can be used to measure the distance between the spectral signature of two variance pixel vectors $\mathbf{r}_k - \mu_k \mathbf{1}$ and $\mathbf{r}_l - \mu_l \mathbf{1}$, called SVA defined by

$$\mathrm{SVA}(\mathbf{r}_k, \mathbf{r}_l) = \cos^{-1}\left(\frac{\left(\mathbf{r}_k - \mu_k \mathbf{1}\right)^T \left(\mathbf{r}_l - \mu_l \mathbf{1}\right)}{||\mathbf{r}_k - \mu_k \mathbf{1}||||\mathbf{r}_l - \mu_l \mathbf{1}||}\right). \tag{10}$$

It should be noted that the inherent normalization of SVA in (10), lends itself straightforwardly as a multivariate filter metric. Another spectral metric, called PVC, can be defined by

$$\mathrm{PVC}(\mathbf{r}_k, \mathbf{r}_l) = \frac{\left(\mathbf{r}_k - \mu_k \mathbf{1}\right)^T \left(\mathbf{r}_l - \mu_l \mathbf{1}\right)}{||\mathbf{r}_k - \mu_k \mathbf{1}||||\mathbf{r}_l - \mu_l \mathbf{1}||}. \tag{11}$$

In addition to NED, NVD, SVA, and PVC, an information distance measure, called SID, is introduced to quantify spectral similarity. Unlike the NED or SVA which treat the pixel vectors as points in an $L$-dimensional space, SID views each pixel vector as a random variable and estimates differences between the distributions of the spectral bands. A normalized band probability is defined for a pixel vector, $\mathbf{p} = (p_1, p_2, \ldots, p_L)^T$,with $p_j = \frac{r_j}{\sum_{i=1}^{L} r_i}$ for $1 \leq j \leq L$. Furthermore, the Kullback–Leibler (KL) divergence between two arbitrary pixel vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ is defined as

$$\mathrm{SID}(\mathbf{r}_k, \mathbf{r}_l) = \sum\nolimits_{k=1}^{L} p_k \log \frac{p_k}{q_k} + \sum\nolimits_{l=1}^{L} q_l \log \frac{q_l}{p_l} \tag{12}$$

where $p_j = \frac{r_{kj}}{\sum_{i=1}^{L} r_{ki}}$ and $q_j = \frac{r_{lj}}{\sum_{i=1}^{L} r_{li}}$.

In the original form, SID cannot be readily used as a distance metric, due to the unnormalized output of the KL-divergence. A simple adjustment can be made by normalizing by the number of bands, $L$

$$\mathrm{NSID}(\mathbf{r}_k, \mathbf{r}_l) = \frac{\mathrm{SID}(\mathbf{r}_k, \mathbf{r}_l)}{L} \tag{13}$$

to account for the total number of bands. The resulting metric will be an average estimate of how correlated high probability events are between both pixel vectors, and more importantly, will provide a normalized upper bound on the metric.

### B. Wrapper Method

Given that only the number of compressed bands needs to be determined rather than finding a specific combination of bands, the wrapper feature selection method is well-suited to this problem. A wrapper method often shows the best performance but tends to be intractable since the combinations of features must be exhaustively searched. If we consider all of the features to have approximately equal discriminatory power, then all fixed size feature sets will be equivalent. This premise significantly reduces the parameter space that must be searched for estimating $m_{\mathrm{opt}}$, and therefore reduces the total amount of computational time required for training the classifiers.

A straightforward and effective algorithm for estimating $m_{\mathrm{opt}}$ can be designed by observing the performance of a classifier, as $m$ is increased. As a compressed classifier is trained for a range of CSBs, the classifier performance is evaluated at each step. The minimum number of CSBs can then be determined by observing the point at which the classifier performance asymptotes or reaches a desired minimum performance level. The wrapper feature selection algorithm is summarized in Algorithm 2.

**Algorithm 2:** Wrapper-Based Feature Selection.

**Input**: A hyperspectral training data, $S_{\mathrm{train}} = \cup_{p=1}^{P} S_{\mathrm{train}}^{p}$ sampling matrix $\mathbf{\Phi} \in \Re^{m_{\max} \times L}$

1. Choose an initial number of CSBs, $m = m_{\min}$ and a CSB step size $\Delta$.
2. Form a valid sampling matrix, $\mathbf{\Phi} \in \Re^{m \times L}$ and project all of the training pixels into the CSBD, $\mathbf{r}_{\mathrm{CSBD},k}^{p} = \mathbf{\Phi}\mathbf{r}_{k}^{p}$
3. Train the classifier $\delta_m(\mathbf{r})$ and calculate a performance metric.
4. Increment the number of compressed bands by $\Delta$.
5. Set $m = m + \Delta_m$ and repeat steps 2 through 4 until $m \geq L$
6. Choose the value of $m$ where the performance metric asymptotes or when a desired performance level has been met.

**Output**: The estimated minimum number of CSBs, $m_{\mathrm{opt}}$ and a set of trained classifiers, $\{\delta_{m+\Delta}(\mathbf{r}), \delta_{m+2\Delta}(\mathbf{r}), \cdots, \delta_{m_{\max}}(\mathbf{r})\}$

## IV. Automatic CSB Bound Estimation

In Sections III and IV, two feature selection algorithms were proposed. However, a stopping rule for terminating these algorithms was not specified. In this section, a change-point detection algorithm is proposed for adaptively determining an optimal number of CSBs, $m_{\mathrm{opt}}$. The proposed change-point algorithm can be combined with either the feature selection algorithm described in Algorithm 1 or the wrapper algorithm described in Algorithm 2, to create a fully automated algorithm.

### A. Optimal Partition Change-Point Detection

Change-point detection is a task of determining a point within a data sequence at which the signal characteristics abruptly change. A large number of related works have been reported [43]. The optimal partitioning developed in [44] is a global approach to change-point detection where all possible change

**Algorithm 3:** Optimal Partitioning Change-Point Detection.

**Input**: A data sequence $h(\tau) \in \Re^{1\times T}$

1. Choose a candidate change-point, $\tau$.
2. Divide the data sequence into lower segment, $h_{\text{lower}}(\tau) = h[1, \cdots, \tau]$ and upper segment, $h_{\text{upper}}(\tau) = h[\tau+1, \cdots, T]$.
3. Calculate the summary statistics for the lower and upper data segments.
4. Calculate the total cost, $J(h(\tau)) = J(h_{\text{lower}}(\tau)) + J(h_{\text{upper}}(\tau))$.
5. Repeat steps 1 through 4 for all candidate change-points, $\tau \in [1, T]$.
6. Chose the value of $\tau$ that minimizes $J(h(\tau))$.

**Output**: The optimally partitioned change-point, $\tau_{\text{opt}}$.

points are simultaneously detected by minimizing a single cost function $J(h)$ constructed as a function of a data sequence, $h \in \Re^{1\times T}$. For a given candidate change-point, $\tau \in [1, T]$, the data are partitioned into a lower segment, $h_{\text{lower}}(\tau) = h[1, \ldots, \tau]$ and an upper segment, $h_{\text{upper}}(\tau) = h[\tau+1, \ldots, T]$. That is

$$J(h(\tau)) = J(h_{\text{lower}}(\tau)) + J(h_{\text{upper}}(\tau)) \tag{14}$$

where

$$J(h_{\text{lower}}(\tau)) = \left(\frac{1}{\tau}\sum\nolimits_{t=1}^{\tau}(h_t - \mu_h)^2\right)^{1/2} \tag{15}$$

$$J(h_{\text{upper}}(\tau)) = \left(\frac{1}{T-(\tau+1)}\sum\nolimits_{t=\tau}^{T}(h_t - \mu_h)^2\right)^{1/2}. \tag{16}$$

A criterion is calculated for segments to form an associated cost $J(h(\tau))$ for that particular change-point $\tau$. The optimal change-point, $\tau_{\text{opt}}$ is then determined by choosing the value of $\tau$ that minimizes the cost function $J(h(\tau))$. The details of implementing the optimal partitioning algorithm are summarized in Algorithm 3.

In general, a criterion used to find an optimal partitioning value $\tau_{\text{opt}}$ can take any form. Nevertheless, in this article, the partition mean and partition standard deviation are specifically considered. To understand the differences between these approaches, two simple simulations were created to show how the change-point detection works for filter method and wrapper methods. For a filter method, a distance metric always tends to converge quickly; however, the overall variance will be reduced at a slower rate. For the classifier performance observed in [24], there was a lower amount of variance, but the point of convergence spanned a larger range of values.

For both simulations, a logarithmic convergence was modeled. The data sequence, $h(\tau)$, was calculated using the following form:

$$h(\tau) = \min(\log(\beta\tau), \log\tau_{\max}) \tag{17}$$

where the scale factor, $\beta$, can be used to control how quickly the data sequence will converge and $\tau_{\max}$ corresponds to the point of convergence, for a unity scale factor. The true point of convergence, for the general case, can be calculated by multiplying $\tau_{\max}$ by the inverse of $\beta$ as $\tau_{\text{converge}} = \frac{\tau_{\max}}{\beta}$.

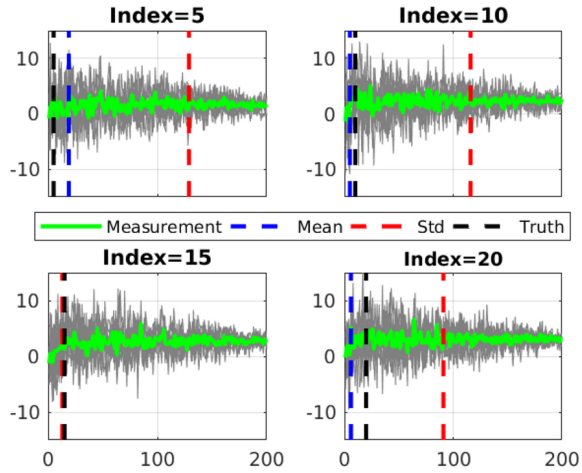



Fig. 1. Optimal partitioning simulation for the filter approach.

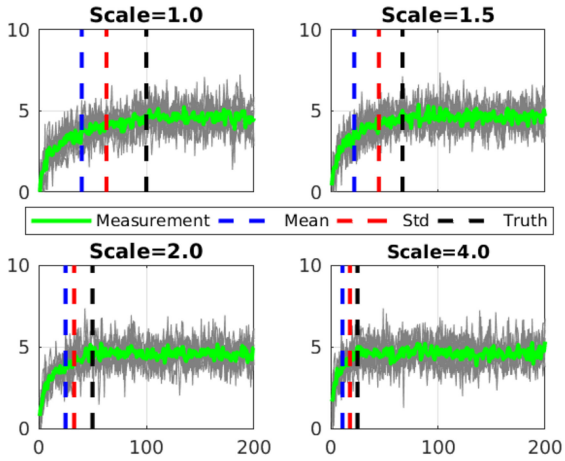



Fig. 2. Optimal partitioning simulation for the Wrapper approach.

Fig. 1 shows the results for the filter method with $\tau_{\max} \in \{5, 10, 15, 20\}$, $\tau \in [1, 200]$, and $\beta = 1$. To account for the slower reduction in variance, Gaussian noise with zero mean and standard deviation, linearly decreasing from 4 to 0.8, was added directly to the data sequence. Thin gray lines represent 10 individual random trials, and the thick green line is the mean over all trials. The dashed blue and red lines represent the optimal partitioning estimate for the partition mean and the partition standard deviation summary statistics, respectively. The black dashed line represents the true convergence index. For this case, the mean statistic provided a fairly robust estimate of the true convergence point. The standard deviation metric was clearly biased by the increased variance and was unable to provide a robust estimate.

Fig 2 shows the results for a wrapper method simulation with $\tau_{\max} = 100$, $\tau_{\max} \in \{5, 10, 15, 20\}$ , $\tau \in [1, 200]$, and $\beta \in \{1, 1.5, 2, 4\}$. To introduce a level of uncertainty, zero-mean, Gaussian random noise, with a standard deviation of 0.8,

was added directly the sequence. The thin gray lines represent 10 individual random trials, and the thick green line is the mean over all trials. The dashed blue and red lines represent the optimal partitioning estimate for the partition mean and the partition standard deviation summary statistics, respectively. The black dashed line represents the true convergence index.

In general, both statistics underestimated the true convergence; however, they both appeared to perform better for sequences that converged quickly. The standard deviation consistently provided a better estimate than the mean.

Based on the above simple experiments, the optimal partitioning algorithm using mean as the criterion will be employed to automate the filter method approach. Conversely, the optimal partitioning algorithm using the standard deviation as the criterion will be used to automate the wrapper method approach.

## V. Pixel-Based Adaptive Compression

The estimation of an appropriate value of $m_{\text{opt}}$ for CBSs provides an opportunity to further compress the acquired images before storing them to disk. For most hardware implementations, the number of CSBs should be estimated by either on average or worst case of the lower bound among all classes. However, it is possible to further compress some of the pixels in each class based on its own class lower bound on the number of CBSs. In other words, for the number of CSBs of the $p$th class, $m_p$ the $p$th classification map can be combined with the class-specified CSB bound, $m_p$, to further save each pixel using its particularly determined-$m_p$ compressed bands.

To measure the effectiveness of this approach, a new metric must be introduced to account for the number of compressed bands used for each pixel. The compressively sensed band ratio (CSBR), introduced in [24], can be generalized to the CSPNR for this purpose and defined by the average of individual pixels in a class rather than compressed bands for a class as

$$\text{CSPNR} = \frac{1}{LN}\sum_{p=1}^{P} m_p n_p. \tag{18}$$

Furthermore, it can be easily shown that the CSBR is simply a special case of the CSPNR with $m_p = m$ for $1 \le p \le P$. In this case, $\sum_{p=1}^{P} m_p n_p = m\sum_{p=1}^{P} n_p = mN$ and (17) is reduced to

$$\text{CSPNR} = \frac{m}{LN}\sum_{p=1}^{P} n_p = \frac{mN}{LN} = \frac{m}{L} \tag{19}$$

which is exactly the CSBR defined in [24].

Another convenient property of the CSPNR metric in (19) is that it can easily be related to the storage requirement. This can be accomplished by multiplying the number of pixels in a class, $n_p$ with its assigned $m_p$ as $\sum_{p=1}^{P} m_p n_p$. For a specific value of CSBs, $m$ fixed for each pixel, this is reduced to $m \times N$. Similarly, for the case of a full $L$-band image, this becomes $L \times N$. As an example, assume there is a 2-class image with $L = 100$ bands, and $N = 200\,000$ pixels, that is stored with single precision (4 bytes per element). The required number of bytes to store the full band image would be $100 \times 200\,000 \times 4\,\text{bytes} = 80\,\text{MB}$. Further assume that a CSB lower bound was estimated to be $m_1 = 5$ and $m_2 = 10$ and that each of these two classes has an equal number of pixels, $N/2$. Then, the total required number of bytes to store the compressed image would be $(5 \times \frac{200000}{2} + 10 \times \frac{200000}{2}) \times 4\text{bytes} = 6\text{MB}$. The storage savings in this example would simply be $\frac{80-6}{80} = 92.5\%$. Alternatively, this can be directly calculated using the CSPNR, as $1 - \text{CSPNR} = 1 - \frac{7.5}{100} = 92.5\%$. Thus, the CSPNR provides a normalized metric for comparing various images and can be use directly to relate compression performance to storage requirement.

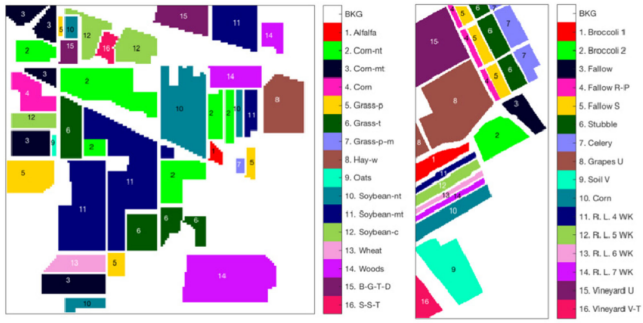



Fig. 3. Indian Pines (left) and Salinas (right) ground truth images.

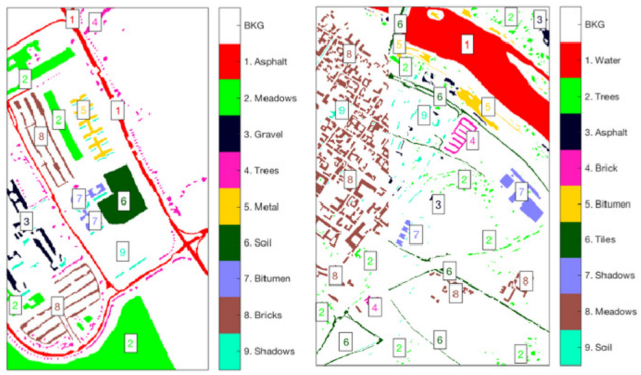



Fig. 4. Pavia University (left) and Pavia Centre (right) ground truth images.

## VI. Experiments

### A. Real-World Hyperspectral Images

Four real-world hyperspectral images were used for the experiments. The first image is of the Indian Pines scene [45] shown in Fig. 3(left), which was collected using the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor [46] and consists of 16 agricultural classes and a background class. The second image, shown in Fig. 3(right), is the Salinas scene [47] which was also collected using the AVIRIS sensor, and consists of 16 agricultural classes and a background class. The third image is of the Pavia University scene [47] shown in Fig. 4(left), which was collected using the Reflective Optics System Imaging Spectrometer (ROSIS) sensor [47] and consists of nine urban classes and a background class. The fourth image is of the Pavia Centre scene [47] shown in Fig. 4(right), which was also collected using the ROSIS sensor, and consists of nine urban classes and a background class.

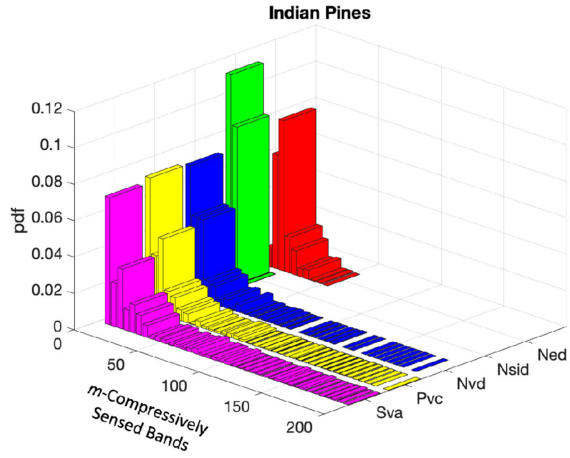



Fig. 5. Indian Pines filter results for the mean-based optimal partition algorithm over 2000 trials.

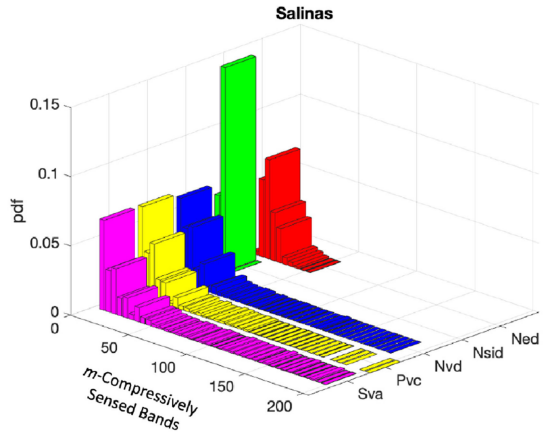



Fig. 6. Salinas filter results for the mean-based optimal partition algorithm over 2000 trials.

### B. *Filter-Based Feature Extraction Experiments*

An experiment was performed to explore the effectiveness of the filter-based algorithm. Given the probabilistic nature of random projections, the experiment was setup in a Monte Carlo fashion, and repeated for 2000 trials to provide representative statistics for each of the proposed similarity measurements. In each trial, the class mean vector was calculated for each class and then projected into the compressed domain following the model specified by (2). The number of CSBs, $m$, was varied from 5 to 55 in a step size of 5 and then from 65 to $L$ in a step size of 10. Average class distances using (6) were calculated at each value of $m$, based on NED, NVD, SVA, PVC, and NSID according to (8), (9), (10), (11), and (13), respectively. Finally, $m_{\text{opt}}$ was estimated by the optimal partitioning algorithm using the mean partition statistic.

The estimated lower bound from each trial has been summarized into a separate probability distribution for each of the distance metrics. The resulting distributions for Indian Pines are shown in Fig. 5. The NVD, SVA, and PVC distributions all show maximum probability with five CSBs or less, which is in line with the classification results reported in [24]. However, these distributions also show much lower tails that extend through nearly all possible values of $m$. NED shows a peak closer to 15 CSBs, but with tails that do not extend beyond 50 CSBs. Finally, the NSID distribution shows a peak probability between 5 and 10 CSBS and is very tightly bound. The resulting distributions for Salinas are shown in Fig. 6 and are very similar to Indian Pines. Given that the types of image scenes and classes are very similar, this result is not surprising.

Similarly, distributions were also calculated for both of the ROSIS images. The resulting distributions for Pavia University and Pavia Centre are shown in Figs. 7 and 8, respectively. All of the distance metrics show distributions with highest probabilities occurring within the first 15 CSBs. NVD, SVA, and PVC show the extended low tails, and NED and NSID show much tighter distributions. In general, all metrics provide results that are in line with the performance that was observed for nearly all of the classes reported in [24].

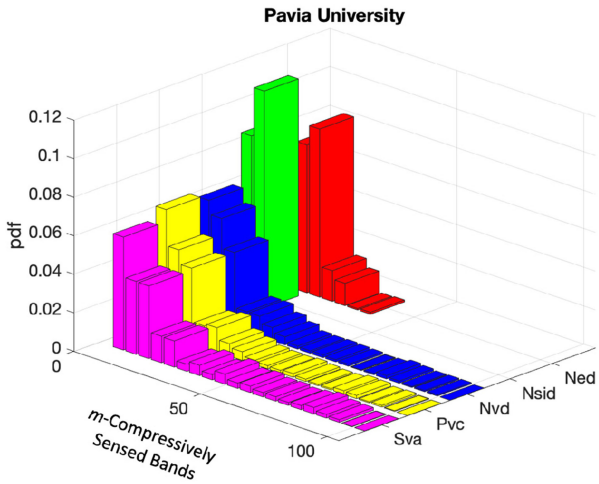



Fig. 7. Pavia University filter results for the mean-based optimal partition algorithm over 2000 trials.

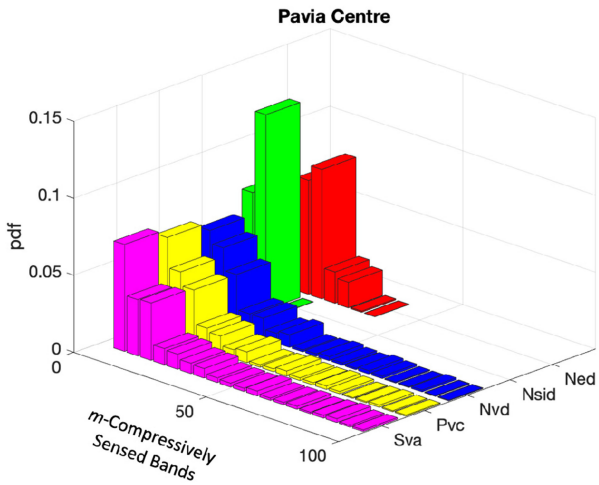



Fig. 8. Pavia Centre filter results for the mean-based optimal partition algorithm over 2000 trials.

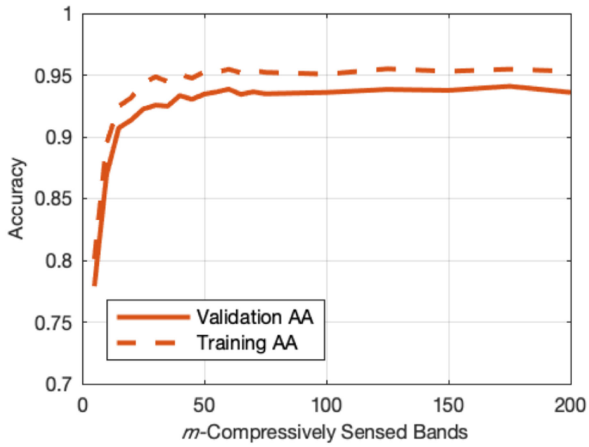



Fig. 9. Comparison of $P_{AA}$ for Indian Pines.

In general, NVD, SVA, and PVC gave comparable results. Each of these metrics provided distributions with peaks near the expected values; however, they showed a lot of variances. The NED and NSID distributions showed lower amounts of variances but were biased to be higher than what was observed in [24]. It is also important to note that these results were all sensitive to the specific change-point algorithm that was implemented, as well as the averaging sizes that were used in the experiment.

### C. Wrapper-Based Feature Extraction Experiments

For the wrapper method experiments and the algorithm comparison, the classifier experiments presented in [24] were repeated and all applicable algorithm parameters were maintained. Specifically, the RBF-based SVM was used as the spectral classifier with the tunable kernel parameter set to 0.72, 0.64, 0.75, and 0.59 for Indian Pines, Salinas, Pavia University, and Pavia Centre, respectively. The edge preserving filtering process was based on a guided filter with a color guidance image, based on the first three principal components, i.e., EPF-G-c. Furthermore, the same approach was used for modeling the CS process. Random Gaussian sensing matrices were used to model the sparse acquisition and the number of CSBs, $m$, was varied from 5 up to full bands, $L$ for each image. Finally, algorithm performance was measured using overall accuracy (OA) and average accuracy (AA) metrics. Training and test samples were the same as that described in [24].

The first experiment was conducted for the wrapper approach to see if the value of $m_{\text{opt}}$ determined by the training data and the test data was consistent. That is, can the training data alone provide a robust estimate of $m_{\text{opt}}$? While it may be thought to potentially be biased by the training dataset compared to the test dataset, it turned out that $m_{\text{opt}}$ was indeed unbiased. This is demonstrated empirically by comparing $P_{AA}$ obtained by training dataset (training $P_{AA}$) to $P_{AA}$ obtained by the test dataset (test $P_{AA}$). In this case, the test samples represent the unobserved pixels that will be encountered in a real-world application.

Fig. 9 shows an example of the training $P_{AA}$ and test $P_{AA}$ for the Indian Pines image scene. Note that there was approximately 2% difference between the training and test $P_{AA}$. Nevertheless, the relative effect from varying the number of CSBs was consistent. Similarly, a strong correlation existed between the training $P_{AA}$ and test $P_{AA}$ for all of test image scenes as well. Although the training $P_{AA}$ and test $P_{AA}$ did show some varying amounts of bias in $P_{AA}$, the relative behavior as a function of the number of CSBs was again consistent.

The correlation between the training and test performance can be easily quantified by using a measure such as the Pearson correlation coefficient (PCC) defined by

$$\text{PCC} = \frac{\text{cov}(a,b)}{\text{SD}(a)\text{SD}(b)} \tag{20}$$

where $a$ and $b$ are arbitrary vectors of equal length, cov($a$,$b$) represents the covariance between $a$ and $b$, and SD($a$/$b$) represents the standard deviation $a$/$b$. The PCC between the training and test performance was equal to 0.9980, 0.9811, 0.9996, and 0.9996 for Indian Pines, Salinas, Pavia University, and Pavia Centre, respectively. It was quite clear for these images that the relationship between the training and test data are highly correlated as a function of the number of CSBs. Accordingly, the training data can be used reliably for the test dataset. This is an important realization because it allows for $m_{\text{opt}}$ to be selected based solely on the training data and ensures that it will be also applicable to the test dataset as an unseen dataset.

### D. Algorithm Comparison

As a final comparison, a single $m_{\text{opt}}(p)$ was estimated for each class, from all of the images. The individual bound was then combined with the classifier results, to determine what the resulting CSPNR in (17) and algorithm efficacy (Eff) defined by

$$\text{P}_{\text{Eff}} = \frac{\text{P}_{\text{CSBD}}}{\text{P}_{\text{Full}}}, \text{ and } \text{P}_{\text{Eff}}(C_p) = \frac{\text{P}_{\text{CSBD}}(C_p)}{\text{P}_{\text{Full}}(C_p)} \tag{21}$$

would be. Here, P can be either AA or OA, $\text{AA}_{\text{Eff}}$ or $\text{OA}_{\text{Eff}}$, and $C_p$ is the $p$th class. For the filter approach, the value of $m_{\text{opt}}$ corresponding to the expected value of the estimated distribution was selected. For the wrapper approach, the value of $m$ was determined by the change-point detection algorithm using the standard deviation as a criterion. For each of the experimental cases, the results have been summarized into a collection of tables. The efficacy of the individual class accuracy is shown for all the five filtering methods, as well as the SVM wrapper method. Additionally, AA, OA, and CSPNR are shown at the bottom of the table. Finally, the results achieved using VD are also included for comparison. Note, that VD only provides a single estimate of $m_{\text{opt}}$ which is assumed to be the same for all classes. The required threshold value for VD has been set to 1e-6 for all image scenes for simplicity since there is not prescribed way of determining this value.

The results for both algorithms run on all four of the images are shown in Tables I –IV where best cases are boldfaced. First, we see that VD did reasonably well at estimating $m_{\text{opt}}$ and produces AA and OA results similar to the filter approaches. However, since it only produces a single number for $m_{\text{opt}}$, it is not possible to make per-class comparisons with the

TABLE I
COMPARISONS AMONG NED, NVD, SVA, PVC, NSID, AND WRAPPER METHOD FOR PURDUE'S INDIAN PINES DATA

| | NED | NVD | SVA | PVC | NSID | Wrapper | VD |
|---|---|---|---|---|---|---|---|
| | $P_{Eff}(C_p)$ | | | | | | |
| Alfalfa | 0.99 | 0.99 | **1.00** | **1.00** | 0.99 | **1.00** | – |
| Corn-n | 0.85 | 0.85 | 0.93 | 0.91 | 0.72 | **0.95** | – |
| Corn-m | 0.84 | 0.84 | 0.92 | 0.87 | 0.82 | **0.95** | – |
| Corn | **1.00** | **1.00** | **1.00** | 0.99 | **1.00** | **1.00** | – |
| G-pasture | **0.99** | **0.99** | **0.99** | **0.99** | 0.98 | **0.99** | – |
| G-trees | **1.00** | **1.00** | **1.00** | **1.00** | 0.99 | **1.00** | – |
| G-mowed | 0.98 | 0.98 | **0.99** | **0.99** | 0.94 | 0.98 | – |
| Hay-W | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Oats | 0.92 | 0.92 | **1.00** | 0.96 | 0.73 | 0.94 | – |
| Soy-n | 0.94 | 0.94 | 0.96 | 0.95 | 0.92 | **0.97** | – |
| Soy-m | 0.91 | 0.91 | 0.92 | 0.92 | 0.89 | **0.98** | – |
| Soy-c | 0.94 | 0.94 | **0.98** | 0.96 | 0.83 | **0.98** | – |
| Wheat | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Woods | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Buildings | 0.94 | 0.94 | **0.98** | 0.96 | 0.87 | **0.98** | – |
| SS Towers | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| $AA_{Eff}$ | 0.96 | 0.96 | 0.98 | 0.97 | 0.92 | 0.98 | **0.99** |
| $OA_{Eff}$ | 0.93 | 0.93 | 0.96 | 0.95 | 0.89 | **0.98** | **0.98** |
| CSPNR | 0.08 | 0.08 | 0.13 | 0.1 | **0.05** | 0.24 | 0.18 |

TABLE II
COMPARISONS AMONG NED, NVD, SVA, PVC, NSID, AND WRAPPER METHOD FOR SALINAS

| | NED | NVD | SVA | PVC | NSID | Wrapper | VD |
|---|---|---|---|---|---|---|---|
| | $P_{Eff}(C_p)$ | | | | | | |
| Broccoli 1 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Broccoli 2 | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Fallow | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Fallow Rough Plow | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Fallow Smooth | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Stubble | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Celery | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Grapes Untrained | **0.99** | **0.99** | 0.98 | **0.99** | 0.97 | **0.99** | – |
| Soil Vineyard | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Corn | 0.99 | 0.99 | **1.00** | **1.00** | 0.99 | **1.00** | – |
| Lettuce 4 Week | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Lettuce 5 Week | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Lettuce 6 Week | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Lettuce 7 Week | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Vineyard Untrained | **1.00** | **1.00** | 0.99 | **1.00** | **1.00** | **1.00** | – |
| Vineyard VT | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| $AA_{Eff}$ | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| $OA_{Eff}$ | **1.00** | **1.00** | **1.00** | **1.00** | 0.99 | **1.00** | **1.00** |
| CSPNR | 0.09 | 0.09 | 0.13 | 0.1 | **0.05** | 0.31 | 0.11 |

TABLE III
COMPARISONS AMONG NED, NVD, SVA, PVC, NSID, AND WRAPPER METHOD FOR PAVIA UNIVERSITY

| | NED | NVD | SVA | PVC | NSID | Wrapper | VD |
|---|---|---|---|---|---|---|---|
| | $P_{Eff}(C_p)$ | | | | | | |
| Asphalt | 0.97 | 0.98 | 0.98 | 0.98 | 0.95 | **1.00** | – |
| Meadows | 0.99 | 0.99 | 0.99 | 0.99 | 0.96 | **1.00** | – |
| Gravel | 0.99 | 0.99 | 0.99 | 0.99 | **1.00** | 0.99 | – |
| Trees | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Painted Metal Sheets | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Bare Soil | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Bitumen | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Bricks | **1.00** | **1.00** | **1.00** | **1.00** | 0.98 | **1.00** | – |
| Shadows | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| $AA_{Eff}$ | 0.99 | **1.00** | **1.00** | **1.00** | 0.99 | 0.99 | 0.99 |
| $OA_{Eff}$ | **0.99** | **0.99** | **0.99** | **0.99** | 0.97 | **0.99** | 0.98 |
| CSPNR | 0.15 | 0.17 | 0.22 | 0.18 | **0.1** | 0.15 | 0.12 |

TABLE IV
COMPARISONS AMONG NED, NVD, SVA, PVC, NSID, AND WRAPPER METHOD FOR PAVIA CENTRE

| | NED | NVD | SVA | PVC | NSID | Wrapper | VD |
|---|---|---|---|---|---|---|---|
| | $P_{Eff}(C_p)$ | | | | | | |
| Water | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Trees | 0.99 | 0.99 | **1.00** | 0.99 | 0.98 | **1.00** | – |
| Asphalt | 0.99 | 0.99 | 0.99 | 0.99 | 0.98 | **1.00** | – |
| Self-blocking Brick | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Bitumen | **1.00** | **1.00** | **1.00** | **1.00** | 0.99 | **1.00** | – |
| Tiles | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Shadows | 0.99 | **1.00** | **1.00** | **1.00** | 0.98 | 0.99 | – |
| Meadows | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| Bare Soil | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | – |
| $AA_{Eff}$ | **1.00** | **1.00** | **1.00** | **1.00** | 0.99 | **1.00** | **1.00** |
| $OA_{Eff}$ | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| CSPNR | 0.16 | 0.2 | 0.21 | 0.2 | **0.1** | 0.47 | 0.19 |

other methods. Furthermore, this approach is somewhat arbitrary since the VD input threshold could be increased which would result in lower $m_{\text{opt}}$ estimates. For these reasons, VD should be used at most as an initial guess.

As expected, the wrapper method was able to select a value of $m$ that produced the highest efficacy for most cases since it is specifically tuned to the SVM classifier. The wrapper method achieved the highest $AA_{Eff}$ on three of the four images, and the highest $OA_{Eff}$ on all four images. While this approach provided nearly the best efficacy, it also produced the highest estimates of $m$, resulting in the highest CSPNR (i.e., the lowest amount of compression).

The filter method also performed reasonably well on estimating reasonable lower bounds for all of the images. Despite the fact that there was some variability between the different variants of the filter methods, they all nearly provided lower estimates of $m$ than the wrapper method. Of the filter variants, SVA provided the highest efficacy with NSID producing the lowest CSPNR, and NED, NVD and PVC all provided very similar results between these two extremes.

The Indian Pines scene showed the most variability between all of the algorithms. This image showed the largest differences in both $AA_{Eff}$ and $OA_{Eff}$, with the NSID filter resulting in values of 0.92 and 0.89, while the wrapper method provided a 0.98 efficacy for both AA and OA. For the remaining three images, all of the algorithms were able to achieve efficacies 1.0 in nearly all cases.

The CSPNR showed much more variability than the efficacies and these differences were apparent in three of the four images. For Indian Pines and Pavia Centre, the NSID filter method was able to achieve approximately a factor of five times more compression than the wrapper method. Similarly, in the Salinas image, NSID achieved a factor six times more compression than the Wrapper method.

It is generally difficult to select a single method as the best choice. Nevertheless, there are a few general guidelines that can be constructed from these results. First, the wrapper method was likely to give the best possible efficacy at expense of much lower amounts of compression. So this method should be used in the case when achieving the highest efficacy is the priority and the increased computational costs of the wrapper method can be tolerated. Second, NSID consistently provided the best CSPNR. So it should be selected for cases when achieving the most compression is the priority. Finally, based on the four test images, SVA appeared to provide the most balanced performance in terms of providing efficacies nearly as high as the wrapper methods, but with consistently lower CSPNRs.

## VII. Conclusion

HSIC in CSBD was a recently developed new approach [24]. However, the issue of determining the minimum number of CSBs, $n_{\mathrm{CSB}}$, to achieve full band performance was not investigated in [24]. In real applications, $n_{\mathrm{CSB}}$ should vary with the complexity of the imaged scene to be studied. Despite that VD has been used to estimate the number of bands to be selected, $n_{\mathrm{BS}}$, its use of estimating $n_{\mathrm{CSB}}$ seems not applicable because a CSB is actually a mixture of full bands sensed by a random sensing matrix, not a single original band. So it requires a new way to estimate $n_{\mathrm{CSB}}$.

As also noted above, DL will certainly offer many advantages, but many considerations/alterations must be made to the algorithms designed for them to be applied directly in the compressed domain. As far as we know, this has not happened. According to the best of our knowledge, very little work on DL methods has been proposed for compressive hyperspectral processing thus far.

One of main contributions in this article is two supervised feature selection-based approaches developed to find an optimal $n_{\mathrm{CSB}}$ so as to maximize classifier performance, while minimizing redundancy. The first approach is a filter method based on observing the behavior of an average class distance measure, as a function of the number of CSBs. The filter-based approach results in a probability distribution of possible values of $m_{\mathrm{opt}}$. A hard estimate can then be derived by selecting a characteristic of the distribution such as the maximum probability or the expected value. The second approach is a wrapper-based method based on observing the behavior of classifier performance on the training data. A hard bound is directly estimated by choosing the value of $m$ when the performance improvement is saturated.

Another main contribution is that both methods can be automated by a change-point detection algorithm via optimal partitioning and their fully automated versions are tested using the classification results presented in [24]. Accordingly, an individual $m_{\mathrm{opt}}$ can be estimated for each class and then used to perform class specific compression. Specifically, both algorithms successfully estimate adequate bounds for $n_{\mathrm{CSB}}$ and are demonstrated to be valid methods that can be used to predict the minimum number of CSBs needed to achieve desirable levels of compressed classifier performance and overall band reduction.

Since for each $p$th class, the number of CSBs of the $p$th classification map can be combined with its class-specified CSB bound, $m_p$, to further save each pixel using its particularly determined-$m_p$ compressed bands. A third main contribution is to generalize the CSBR introduced in [24] to the CSPNR to account for the number of compressed bands used for each pixel. Compared to the filter method, the wrapper method estimates bounds more consistently with higher efficacy at the expense of a CSPNR up to six times larger than that estimated by the filter method. Finally, some guidelines were provided as to which algorithms might be used depending on the priorities of the use case.

## References


[1] C.-I. Chang, *Hyperspectral Data Processing: Algorithm Design and Analysis*. Hoboken, NJ, USA: Wiley, 2013.

[2] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. Hoboken, NJ, USA: Wiley, 2001.

[3] R. O. Duda, P. E. Hart, and R. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2001, pp. 568–573.

[4] A. Voulodimo, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, 2018.

[5] K. Makantasis, A. D. Doulamis, and N. D. Doulamis, "Tensor-based classification models for hyperspectral data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6884–6898, Dec. 2018.

[6] K. Makantasis, A. D. Doulamis, N. D. Doulamis, and A. Voulodimos, "Common mode patterns for supervised tensor subspace learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 2927–2931.

[7] Q. Wang, Q. Li, and X. Li, "Hyperspectral band selection via adaptive subspace partition strategy," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4840–4950, Dec. 2019.

[8] Q. Wang, Q. Li, and X. Li, "A fast neighborhood grouping method for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 608–619, Jul. 2020.

[9] R. M. Torres, P. W. T. Yuen, C. Yuan, J. Piper, C. McCullough, and P. Godfree, "Spatial spectral band selection for enhanced hyperspectral remote sensing classification applications," *J. Imag.*, vol. 6, no. 7, 2020, Art. no. 87.

[10] C.-I. Chang, Y. M. Kuo, S. Chen, C.-C. Liang, K. Ma, and P. Hu, "Self-mutual information-based band selection for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5979–5997, Jul. 2021.

[11] Y. Yuan, X. Zheng, and X. Lu, "Discovering diverse subset for unsupervised hyperspectral band selection," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 51–64, Jan. 2017.

[12] Y. Cai, Z. Zhang, X. Liu, and Z. Cai, "Efficient graph convolutional self-representation for band selection of hyperspectral image," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4869–4880, 2020.

[13] C. Sui *et al.*, "Unsupervised hyperspectral band selection with multigraph integrated embedding and robust self-contained regression," *IEEE Trans. Geosci. Remote Sens.*, early access 10.1109/TGRS.2021.3068779.

[14] J. Feng *et al.*, "Deep reinforcement learning for semisupervised hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, early access 10.1109/TGRS.2021.3049372.

[15] C. Yu, M. Song, and C.-I. Chang, "Band subset selection for hyperspectral image classification," *Remote Sens.*, vol. 10, no. 1, 2018, Art. no. 1133.

[16] C. Yu, M. Song, and C.-I. Chang, "Class signature-constrained background suppressed approach to band selection for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 14–31, Jan. 2019.

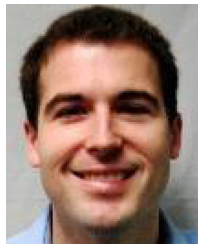
[17] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[18] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[19] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[20] R. Baraniuk, M. Davenport, R. De Vore, and M. B. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximat.*, vol. 28, no. 3, pp. 253–263, 2008.

[21] C. J. Della Porta, A. Bekit, B. Lampe, and C.-I. Chang, "The application of the universality of random compressed sampling for hyperspectral image processing," in *SPIE, Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV*. Baltimore County, MD, USA: SPIE Defense + Commercial Sensing, 2019.
[22] C. J. Della Porta, A. Bekit, B. Lampe, and C.-I. Chang, "A compressed sensing approach to hyperspectral classification," in *SPIE, Big Data: Learning, Analytics, and Applications*. Baltimore County, MD, USA: SPIE Defense + Commercial Sensing, 2019.
[23] C. J. Della Porta, *Hyperspectral Image Classification via Compressive Sensing*. Baltimore County, MD, USA: Dept. of Comput. Sci. and Elect. Eng., Univ. Maryland, 2019.
[24] C. J. Della Porta, B. Lampe, A. Bekit, and C.-I. Chang, "Hyperspectral image classification via compressive sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8290–8303, Oct. 2019.
[25] C. J. Della Porta and C.-I. Chang, "Progressive compressively sensed band processing for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2378–2390, Mar. 2021.
[26] B. Lampe, "Compressive sensing with applications to hyperspectral image processing," Ph.D. dissertation, Dept. Comput. Sci. Elect. Eng., Univ. Maryland, Baltimore County, MD, USA, May 2019.
[27] B. Lampe, C.-I. Chang, A. Bekit, and C. J. Della Porta, "Restricted entropy and spectrum properties for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5642–5652, Aug. 2020.
[28] A. Bekit, *Compressive Sensing for Endmember Finding and Linear Spectral Unmixing for Hyperspectral Imagery*. Baltimore County, MD, USA: Dept. Comput. Sci. Elect. Eng., Univ. Maryland, 2019.
[29] A. Bekit, C.-I. Chang, B. Lampe, C. J. Della Porta, and C. C. Wu, "N-FINDR in a compressively sensed band domain," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1087–1101, Feb. 2020.
[30] A. Bekit and C.-I. Chang, "Unsupervised hyperspectral unmixing using compressive sensing," in *Remotely Sensed Data Compression, Communications, and Processing XII*. Baltimore, MD, USA: SPIE Commercial+Scientific Sensing and Imaging, 2016.
[31] B. Lampe and C.-I. Chang, "Hyperspectral band selection using compressive sensing," in *Remotely Sensed Data Compression, Communications, and Processing XII*. Baltimore, MD, USA: SPIE Commercial+ Scientific Sensing and Imaging, 2016.
[32] B. Lampe, A. Bekit, C. Porta, B. Xue, and C.-I. Chang, "Unsupervised hyperspectral band selection in the compressive sensing domain," in *SPIE, Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV*. Baltimore, MD, USA: SPIE Defense + Commercial Sensing, 2019.
[33] A. Bekit, C. Porta, B. Lampe, B. Xue, and C.-I. Chang, "Unsupervised automatic target generation processing via compressive sensing," in *SPIE, Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV*. Baltimore, MD, USA: SPIE Defense + Commercial Sensing, 2019.
[34] C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral detection and Classification*. New York, NY, USA: Kluwer Academic/Plenum Publishers, 2003.
[35] C.-I. Chang and Q. Du, "Estimation of number of spectrally distinct spectral signal sources in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 608–619, Mar. 2004.
[36] C.-I. Chang, "A review of virtual dimensionality for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1285–1305, Apr. 2018.
[37] C.-I. Chang, *Real-Time Recursive Hyperspectral Sample and Band Processing: Algorithm Architecture and Implementation*. Berlin, Germany: Springer, 2017.
[38] M. Song, X. Shang, Y. Wang, C. Yu, and C.-I. Chang, "Class-information based band selection for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8394–8416, Nov. 2019.
[39] S. Zhong *et al.*, "Class feature weighted hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4728–4745, Dec. 2019.
[40] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," in *Data Classifications: Algorithms and Applications*, vol. 37, Boca Raton, FL, USA: CRC Press, 2014.
[41] X. Kang, S. Li, and J. A. Benediktsson, "Spectral-spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.
[42] C.-I. Chang, "An information theoretic-based approach to spectral variability, similarity and discriminability for hyperspectral image analysis," *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1927–1932, Aug. 2000.
[43] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*, vol. 104. Hoboken, NJ, USA: Prentice Hall, 1993.
[44] M. Lavielle, "Using penalized contrasts for the change-point problem," *Signal Process.*, vol. 85, no. 8, pp. 1501–1510, 2005.
[45] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pines Test Site 3*. New York, NY, USA: Purdue Univ. Research Repository, 2015.
[46] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, "The airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 44, no. 2/3, pp. 127–143, 1993.
[47] "Universidad del pais vasco grupo de inteligencia computacional," Accessed: Jun. 2018, [Online]. Available: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#Salinas



**C. J. Della Porta** (Member, IEEE) received the bachelor's and master's degrees in electrical engineering from the University of Central Florida, Orlando, FL, USA, in 2007 and 2011, respectively, and the Ph.D. degree in electrical engineering from the University of Maryland Baltimore County, Baltimore, MD, USA, in August 2019.

He is currently a Senior Physicist with The Johns Hopkins University Applied Physics Lab, Laurel, MD, USA. His research interests include signal processing, machine learning, compressive sensing, and machine learning techniques in the field of hyperspectral image processing.

**Chein-I Chang** (Life Fellow, IEEE) received the B.S. degree in mathematics from Soochow University, Suzhou, China, in 1973, the M.S. degree in mathematics from the Institute of Mathematics, National Tsing Hua University, Hsinchu, Taiwan, in 1975, the M.A. degree in mathematics from the State University of New York, Stony Brook, NY, USA, in 1977, both the M.S. and M.S.E.E. degrees from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1982, Urbana, IL, USA, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 1987.

He has been with the University of Maryland, Baltimore County, Baltimore, MD, USA, since 1987, where he is a Professor with the Department of Computer Science and Electrical Engineering. He has authored four books, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification* published by Kluwer Academic Publishers in 2003 and *Hyperspectral Data Processing: Algorithm Design and Analysis*, John Wiley & Sons, 2013, *Real Time Progressive Hyperspectral Image Processing*: *Endmember Finding and Anomaly Detection* 2016 by Springer and *Recursive Hyperspectral Sample and Band Processing: Algorithm Architecture and Implementation*, Springer 2017. In addition, he also edited two books, *Recent Advances in Hyperspectral Signal and Image Processing*, 2006 and *Hyperspectral Data Exploitation: Theory and Applications*, John Wiley & Sons, 2007 and coedited with A. Plaza a book on *High Performance Computing in Remote Sensing*, CRC Press, 2007. He is also editing a book, *Advances in Hyperspectral Image Processing Techniques* to be published by Wiley, 2022. His research interests include multispectral/hyperspectral image processing, automatic target recognition, and medical imaging.

Prof. Chang currently serves as an Associate Editor for *Remote Sensing* and *IEEE Transactions on Geoscience and Remote Sensing*. He is a Fellow of SPIE. He was the recipient of the National Research Council Senior Research Associateship Award from 2002 to 2003 sponsored by the U.S. Army Soldier and Biological Chemical Command, Edgewood Chemical and Biological Center, Aberdeen Proving Ground, Maryland.