Band and Quality Selection for Efficient Transmission of Hyperspectral Images

Mohammad Amin Arab
Simon Fraser University
Burnaby, BC, Canada

Kiana Calagari
Simon Fraser University
Burnaby, BC, Canada

Mohamed Hefeeda
Simon Fraser University
Burnaby, BC, Canada

ABSTRACT

Due to recent technological advances in capturing and processing devices, hyperspectral imaging is becoming available for many commercial and military applications such as remote sensing, surveillance, and forest fire detection. Hyperspectral cameras provide rich information, as they capture each pixel along many frequency bands in the spectrum. The large volume of hyperspectral images as well as their high dimensionality make transmitting them over limited-bandwidth channels a challenge. To address this challenge, we present a method to prioritize the transmission of various components of hyperspectral data based on the application needs, the level of details required, and available bandwidth. This is unlike current works that mostly assume offline processing and the availability of all data beforehand. Our method jointly and optimally selects the spectral bands and their qualities to maximize the utility of the transmitted data. It also enables progressive transmission of hyperspectral data, in which approximate results are obtained with small amount of data and can be refined with additional data. This is a desirable feature for large-scale hyperspectral imaging applications. We have implemented the proposed method and compared it against the state-of-the-art in the literature using hyperspectral imaging datasets. Our experimental results show that the proposed method achieves high accuracy, transmits a small fraction of the hyperspectral data, and significantly outperforms the state-of-the-art; up to 35% improvements in accuracy was achieved.

CCS CONCEPTS

• **Computing methodologies** → **Hyperspectral imaging**; **Object identification**; *Neural networks.*

KEYWORDS

Hyperspectral imaging; adaptive data transmission; material identification

ACM Reference Format:
Mohammad Amin Arab, Kiana Calagari, and Mohamed Hefeeda. 2019. Band and Quality Selection for Efficient Transmission of Hyperspectral Images. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3343031.3351047

MM'19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351047

1 INTRODUCTION

Hyperspectral cameras capture a scene in many wavelength bands across the spectrum, providing far more information than regular cameras that operate in the visible light range. For example, when observing a remote object (e.g., a car), signals in the visible light band can show the shape/color of the object, whereas signals in the infrared band can determine the temperature of that object (e.g., whether the car engine is on). Furthermore, signals in other bands can identify the surroundings of that object, e.g., whether the area has vegetation, the moisture level in the soil, and the presence and depth of water nearby.

A simple illustration of hyperspectral imaging is shown in Figure 1, which shows a scene captured in the spatial x, y domain as well as in the spectral λ domain. For each value of λ, the scene is captured in a different frequency band, and thus different information about the scene is revealed. Current, commercially available, hyperspectral cameras can capture more than 200 bands, and thus produce huge amounts of high-dimensional data. The data shown in Fig. (1) is referred to as a hyperspectral data cube. Hyperspectral imaging is useful in many commercial/civilian applications such as agricultural research, land-cover mapping, forest monitoring, and mapping of natural disasters [4], as well as military applications including remote sensing, surveillance, and identification of camouflaged objects [9]. Recent technological advances in capturing and processing devices have made hyperspectral camera systems smaller, more powerful, and cost effective [3]. Thus, their adoption in various applications is expected to accelerate in the future.

In many applications, e.g., forest fire detection [1] and gas leak detection [7] [13] [12], the hyperspectral data is captured from remote sites (e.g., using drones) and needs to be transmitted in a timely manner to a processing station for taking actions. For instance, in forest fire detection, hotspots are locations that have higher temperature and lower humidity and are thus likely to start fires [17]. Timely identification of these spots is crucial in preventing or controlling forest fires, as well as taking precautionary measures such as issuing evacuation alerts. Waiting for drones carrying the hyperspectral cameras to come back with the captured data may take hours and by then the data may be less useful or even obsolete. Data transmission typically occurs over satellite or cellular links, which poses a major challenge, because of the channel dynamics, large data volume, and the complex nature of the hyperspectral data and the way it is used. Specifically, hyperspectral data is typically used to identify/classify material(s) present in a scene. This is possible because hyperspectral cameras capture information across a large portion of the spectrum, and thus each material will likely produce a *spectral signature*, which can be used to identify that material. For example, the presence of soil in a scene results in signal values in certain frequency bands that are quite different from those

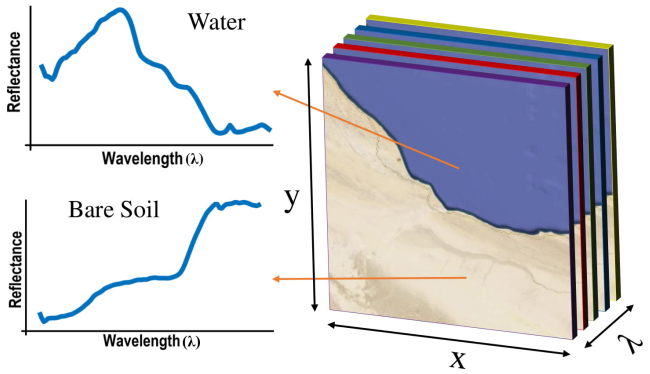


Figure 1: Example of a hyperspectral data cube. The material of each spatial location can be identified using its spectral signature.

produced by the presence of rocks or water. Illustration of spectral signatures is given in the left part of Figure 1.

In this paper, we focus on applications that require transmitting hyperspectral imaging data over limited-bandwidth channels. We present a method to dynamically prioritize the transmission of various components of hyperspectral data based on the available bandwidth in order to maximize the utility of the received (partial) data. Specifically, our method encodes each band of the hyperspectral data into multiple cumulative layers. Then, given a bit budget, the method optimally selects which bands to transmit and the quality (layers) of each band such that the resulting accuracy of processing the received data is maximized. Our method is general and supports any type of processing of hyperspectral data.

We have implemented the proposed method and tested it on multiple hyperspectral datasets. We compared our approach against the state-of-the-art methods in the literature [16], [22], [20], [18], [19], and [15]. Our results show that the proposed method significantly outperforms the state-of-the-art in terms of the achieved accuracy while using the same amount of data; up to 35% in the accuracy can be achieved, especially when the bit budget is small, which is the common case. In addition, the results show that our method efficiently supports gradual transmission of hyperspectral data which is an important feature since many hyperspectral imaging applications target analyzing data from vast remote areas. For example, our results demonstrate that an approximate classification of a scene can be achieved by transferring a very small amount of the hyperspectral data (around 3%). If a finer classification is needed, additional data can be transferred and combined with the data already received. This gradual data transmission and analysis can save substantial communication and processing resources in real hyperspectral applications. Furthermore, because our method jointly and optimally selects bands and their qualities, it can achieve up to 98% reduction in the amount of transmitted data while still achieving very high accuracy.

The rest of this paper is organized as follows. We summarize the related work in Sec. 2. We describe the considered problem in Sec. 3. In Sec. 4, we present our proposed solution. We present our evaluation in Sec. 5, and conclude the paper in Sec. 6.

2 RELATED WORK

In order to address the large volume of data, several works have proposed lossy and lossless compression techniques for hyperspectral image cubes [2, 5, 8, 21]. For example, Zhang and Liu [21] present a lossless compression method based on 3D wavelet coding. Fu et al. [8] present an example of lossy compression method for hyperspectral images, which takes advantage of both the spectral and spatial information. The method first constructs superpixels, which are homogeneous regions that share similar spectral signatures. Different levels of distortion are then assigned to each superpixel using an adaptive coding scheme which optimizes the overall rate-distortion performance. Chen et al. [2] propose a compression method that preserves the quality of important bands, which are specified by target applications. Egho et al. [5] propose an adaptive hyperspectral image compression technique suitable for both lossy and lossless compression in which Karhunen-Loeve Transform (KLT) is used for spectral de-correlation. A comparison of different hyperspectral image compression methods can be found in [11]. Our work in this paper focuses on selecting which bands to transmit and at what qualities, and thus it is orthogonal to compression methods.

Feature engineering methods have been extensively studied for hyperspectral images as a way to select the most important parts of the data. Feature engineering includes feature extraction and feature selection (band selection). In feature extraction, nonlinear transformations are used to extract the most discriminative features. For instance, in [6], the local covariance matrix is used to characterize the correlation among different spectral bands.

The closest works to ours are the ones that present feature/band selection methods, e.g., [15, 16, 18–20, 22]; we will compare our method against all of them in Sec. 5. Feature/band selection methods find the most representative bands from the hyperspectral data cube without transforming them. Thus, they retain the physical meaning of each band. The authors of [16] present an unsupervised band selection method based on manifold ranking, whereas the authors of [22] use fuzzy clustering with particle swarm optimization to select bands. In [19], a clustering-based band selection algorithm is studied. An unsupervised band selection approach is introduced in [15], which obtains the most representative bands using correlation matrix and measurement of block-diagonal structure used in segmenting all bands into subspaces. In [20], a feature selection method is designed based on a memetic algorithm; a memetic algorithm is an extension of genetic algorithms. Finally, a deep convolutional neural network (CNN) has been proposed to select bands [18]. This work first partitions the spectral domain using distance density. It then randomly selects different band combinations according to each partition's distance density and uses the CNN model to test the band combinations.

Note that in our proposed method, band selection is preferred over feature extraction. This is because our approach is designed to provide an approximate estimation of the area first and finer details upon the user's request. Thus, the physical meaning of the bands needs to be retained in order for previously transmitted data to be useful.

3 PROBLEM DESCRIPTION

We consider the problem of optimizing the transmission of remotely-captured hyperspectral imaging data over a dynamic channel to maximize the utility of the received data for complex processing tasks such as material identification and classification. This is an important problem for applications such as teleoperation, remote sensing, surveillance, and controlling unmanned systems. As an example scenario, consider a UAV (Unmanned Aerial Vehicle) equipped with a hyperspectral camera dispatched to explore a remote scene. The camera captures a scene across many frequency bands in the spectrum, and needs to transmit the data to a base station for processing and taking actions such as alerting a human operator about the existence of some objects or steering the UAV to a different location. Our problem is to identify the most important components of the hyperspectral data to transmit from the UAV within a *given bit budget* such that the accuracy of the data processing task at the base station is maximized. This problem is more complex, and more general, than the band selection problem addressed in previous works, e.g., [16, 18, 19], where there is no limitation on the amount of data and thus all bands are available at the base station.

The captured hyperspectral data is divided into *cubes*. The dimensions of each cube are x, y, λ, where x, y are the spatial dimensions, and λ is the spectral dimension. Cubes are transmitted successively by the camera to the processing station. Once it receives a hyperspectral data cube, the processing station uses this data as input to a pre-trained *deep learning* model to identify materials and objects in the remote scene using their hyperspectral signatures. The data volume in each cube is, however, very large to be transmitted in a timely manner, even after significant compression. To address this problem, we first propose to encode each band in the hyperspectral data cube into multiple qualities, using any scalable coding method that produces *cumulative* quality layers, i.e., the quality progressively increases by adding layers. For concrete discussion, we use a multi-level two-dimensional discrete wavelet transform (2D-DWT) in our solution. Figure 2 illustrates the considered problem for one cube of hyperspectral data. There are N spectral bands in each cube. Each band can be considered as an image of dimensions x, y. Each band is encoded at Q cumulative quality levels. Then, our problem is to optimally and *jointly* select which bands to transmit and the quality of each band given a bit budget C. It is straightforward to show that this band-quality selection problem is NP-Complete, by reducing the multiple-choice knapsack problem to it. The search space for finding the optimal solution is $O(Q^N)$ in the worst case, which is prohibitive as N is in the order of hundreds of bands for current cameras and Q is usually in the range of 2 to 5 quality levels.

It is important to note that the utility of the received bands is typically a non-linear function. This is because, as illustrated in the left part of Figure 1, spectral bands react differently to various materials in the captured scene, creating the so-called spectral signatures for different materials across the spectrum, which indicates that there is correlation among bands. Thus, depending on the processing task to be done on the received bands, the relative importance of individual band varies. In addition, the possibility of using different qualities for each band adds another level of complexity to the problem, compared to prior works [10, 15, 20].

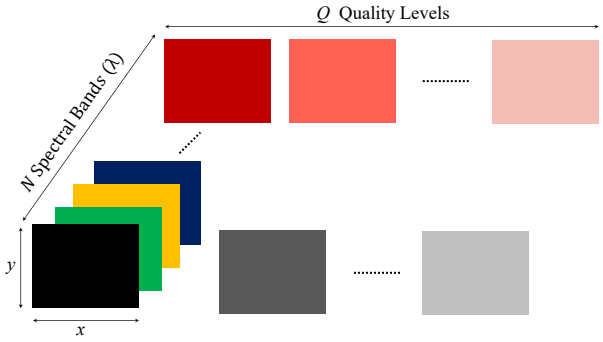


Figure 2: Illustration of the band and quality selection problem for hyperspectral images.

4 PROPOSED SOLUTION

The proposed solution is referred to as BQSA (Band-Quality Selection Algorithm). At high level, BQSA is executed once for each hyperspectral data cube to select bands and their qualities such that a given bit budget for that cube is not exceeded. As discussed in the previous section, jointly selecting bands and their qualities is an NP-Complete problem. We utilize some characteristics of the hyperspectral images to transform this problem into another equivalent problem, which has a substantially smaller search space and thus can be solved in polynomial time (Section 4.1). Then, we design an efficient method to search for the optimal solution in the reduced (polynomial) search space (Section 4.2). Finally, we discuss various optimizations and practical issues (Section 4.3).

4.1 Reducing the Search Space

As discussed in Section 3, the search space is $O(Q^N)$, where Q is the number of quality levels and N is the number of bands, which is prohibitive. To address this, we prioritize the bands and then choose quality levels in a way that substantially reduces this search space. An important feature of our solution is that it retains the physical meaning of each band, that is, it does not mix or transform bands into a different domain where the distinction between bands is lost, as is the case with prior band prioritization methods, e.g., in [10, 14]. This is crucial for utilizing the prioritized bands in achieving classification with different granularities as well as to support various bit budgets.

To prioritize hyperspectral bands, we first transform the 3-D data cube $\boldsymbol{D}_{X\times Y\times N}$ into a 2-D matrix $\boldsymbol{M}_{S\times N}$, where S is the number of pixels in the dataset ($S = X \times Y$). Then, we use PCA (Principal Component Analysis) to extract N principal components from the matrix $\boldsymbol{M}$. Each principal component is a linear combination of bands, which also means that each band has a contribution (weight w_i) to each component i. Further, each principal component i is associated with an *explained variance ratio* (EVR), which is the ratio of the variance of that component to the total variance, and it is denoted by v_i. EVR indicates the relative importance of each principal component. Then, we can define the importance of each band as follows:

$$value_j = w_{j1}\, v_1 + w_{j2}\, v_2 + \cdots + w_{jN}\, v_N, \tag{1}$$

where $value_j$ is the importance of band j, w_{ji} is the weight of band j in the ith principal component, and v_i is the explained variance ratio of the ith principal component. This is an intuitive quantification of the relative value of each band, because it uses the explained variance ratios of principal components (i.e., their relative importance) and the weights of bands which capture the relative contributions of various bands in each principal component. After computing the values of all bands, we rank them in descending order based on their values. We note that the principal component analysis is done offline on training datasets relevant to the task in hand.

Band ranking allows us to simplify the band-quality selection problem by imposing some structure on the search process for the optimal solution. Specifically, using band ranking, the band-quality selection problem can be mathematically formulated as follows:

$$\begin{aligned} &\textit{maximize} && \text{Classification Accuracy} \\ &\textit{subject to} && \sum_{i=1}^{N}\sum_{j=1}^{Q} b_{ij}x_{ij} \le C, \\ & && x_{ij} \le x_{(i-1)j}, \quad i = 2,\ldots,N \quad j = 1,\ldots,Q, \\ & && x_{ij} \le x_{i(j-1)}, \quad i = 1,\ldots,N \quad j = 2,\ldots,Q, \\ & && x_{ij} \in \{0,1\}, \quad i = 1,\ldots,N \quad j = 1,\ldots,Q, \end{aligned} \tag{2}$$

where b_{ij} refers to the extra information required to send band i with quality j assuming that the $j-1$ quality of this band is already in hand. The variable x_{ij} takes a binary value indicating whether band i with quality j is selected; $j = 0$ means that the band is not selected. Thus, in order to choose a band with quality J, all x_{ij} with $j \le J$ should be selected. The first constraint ensures better quality for bands with higher ranking. The second constraint allows a band to have a specific quality only if its lower quality versions have already been selected.

By prioritizing the quality of higher ranked bands in the formulation (Eq. (2)), the search space is reduced to $O(N^Q)$, which is substantially smaller than $O(Q^N)$. Recall that N is around 200 bands while Q typically ranges from 2 to 5 quality layers. Although much smaller than the original search space, trying all $O(N^Q)$ band-quality combinations to find the optimal solution is still costly. This is because each combination may involve computing the utility (accuracy) of a deep learning model. We next design an efficient algorithm to quickly explore the search space and find the optimal solution.

4.2 Computing the Optimal Solution

We design a branch-and-bound (B&B) algorithm to find the optimal solution for the band-selection problem. The key idea is to structure the search space as a tree, where each node has an upper bound which describes the best case possible that can occur from the subtree rooted at that node. During the search, if we reach a node that has an upper bound worse than what we already have, the entire subtree under that node is pruned. Such pruning greatly accelerates the search process. In the following, we describe how we form the initial tree, determine the upper bound for each node, and search the tree to find the optimal solution. We note that, like other branch-and-bound algorithms, the focus of our search algorithm is to substantially reduce the *average* computation time for *typical* inputs, where the upper bound helps in pruning many branches quickly. Although unlikely to occur in practice, it is possible to find an input that forces our algorithm to explore every single node in the tree; that is, in the worst-case, the time complexity is $O(N^Q)$.

Forming the search tree: We form a tree with depth of Q levels. The first level (L_1) initially has $N + 1$ branches, each indicating the number of bands it can choose from the highest quality level. We denote this number as Num_1 and it is in the range of 0 to N. Each child node will in turn branch out to indicate the number of bands that can be chosen from the next quality level (Num_2), and so on. Note that the number of branches each node can have in a level depends on the Num of that node and its ancestors. For example, $Num_1 = N$ means all N bands have been selected with highest quality and thus that node can only have one branch with a value of zero ($Num_2 = 0$).

Next, we compute the upper bound for each node, which will be used in pruning tree branches to accelerate the search, without missing the optimal solution. The upper bound depends on the maximum amount of data that can be transmitted C. We first start by pruning the tree branches that do not meet the capacity constraint C. For each level of the tree, we check the feasibility of each branch by calculating the total amount of required data to send the associated bands with the specified qualities and comparing against C.

Figure 3 shows an example of a pruned tree with $Q = 4$ and $N = 200$. In this example, C is sufficient to send up to 20 bands with full quality. After sending 20 bands with their highest quality, the remaining capacity is not enough to send another high quality band; it is sufficient to send 1 more band with quality level 2, or 3 bands with quality level 3. To compute the upper bound for a node in the kth level of the tree, we check all remaining $Q - k$ lower levels under this node. At each level, we choose the maximum number of bands available, and take their maximum band/quality as the upper bound. For example, in Figure 3, at the position of the red node we have enough capacity left to send a maximum of 73 bands with quality level 3, or 185 bands with quality level 4. Based on the tree structure, we are not allowed to choose 73 bands with quality 2, and the remaining 112 bands with quality level 4. However, by choosing this band/quality combination, we are sure that no other combinations will result in a better accuracy since they either have less number of bands, or bands with lower qualities. Therefore, choosing to send 73 bands with quality level 3 and the remaining 112 bands with quality level 4 is a conservative upper bound. Although the upper bound may not even be a feasible solution (as in the example just mentioned), it ensures that we do not miss the optimal solution. The upper bound of the red node will be then set to (13, 2, 73, 54).

Exploring the search tree: With the tree in hand, we start exploring the tree by first checking all nodes in the first level. We evaluate each node by using its upper bound to modify the sample labeled dataset and feed it to our deep learning model. To start the algorithm we need an initial response. Our experiments show that using more bands with low quality is better than using fewer bands with higher quality. So we choose the left most response in the tree. For leaf nodes, we just choose the maximum number available without checking the rest. After reaching a leaf node and having

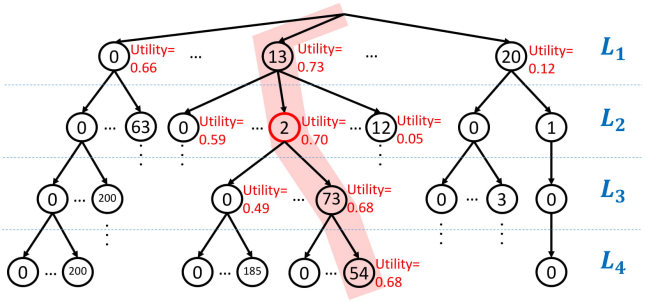


Figure 3: Example tree with 4 quality levels and 200 bands. The value in each node represents the number of bands chosen from that quality level. The highlighted path shows the steps our algorithms takes to reach the optimal solution.

an actual feasible solution at hand, we trace our steps backward. At each step, we compare our current best result with the previously evaluated lower bounds of that level. If our current feasible solution is still better than all other upper bounds in all levels then we have found the optimal solution. If not, we will continue our search down the branches of the node that had the better upper bound. The pink path in Figure 3 shows the process of finding the optimal solution.

Note that evaluating the deep learning model is an expensive task which takes time in the order of seconds to execute. Thus, in order to avoid having to compute the same selection multiple times, each time a selection is evaluated, the result is saved for future use.

4.3 Remarks and Optimizations

Band Bundling: To further reduce the running time of the band-quality selection algorithm, we propose grouping multiple neighbouring bands together in a *bundle*. This is not expected to significantly impact the accuracy of the produced solution, because in practice the spectral signature does not change abruptly across neighboring bands (see Figure 1). We evaluate the impact of the bundle size on the accuracy and processing time in Section 5.2.

Deployment of BQSA: The proposed algorithm is executed by the receiving side for each hyperspectral data cube; the capturing camera (sending side) is not assumed to do any processing other than encoding the hyperspectral images into multiple quality levels. The receiving side has a trained deep learning model for the tasks to be performed on the hyperspectral data. It also has a labeled sample dataset. During runtime, the available bit budget is estimated by the sending side and sent to the receiving side. Using this budget and the sample dataset, the algorithm determines the optimal bands to request and their qualities. Once these bands arrive, they are used as input to the deep learning model, and based on the results, various actions can be taken according to the semantic of the target application. For example, more bands/qualities from the same data cube can be requested to improve the accuracy or achieve finer-grained classification. Then, another hyperspectral data cube is processed, and so on.

One possible optimization on the above operation is as follows. Since spectral signatures of different materials do not change, prior hyperspectral datasets can be used to pre-compute search trees for common use cases. For example, search trees for multiple bandwidth values expected to be observed during the flight of the UAV carrying the hyperspectral camera can be pre-computed using datasets of various materials that might be present in the explored remote scene. The resulted trees can be pre-loaded on the UAVs so that the now much simpler search process can be done on them to choose bands and their quality to transmit.

5 EVALUATION

We implemented the proposed solution and compared it against the state-of-the-art band selection approaches using diverse hyperspectral imaging datasets.

5.1 Datasets and Experimental Setup

Datasets: We use four hyperspectral imaging datasets; most of them were used in previous works which enables us to conduct fair comparisons. The datasets were captured from diverse environments, have different resolutions, and cover different spectral ranges. Each dataset is a hyperspectral cube in the form x, y, λ, where x, y are the spatial dimensions and λ is the spectral dimension. Each x, y pixel has a label indicating the actual material of the physical location corresponding to that pixel, which provides ground truth for our evaluation. The details of the four datasets are as follows:

- *Kennedy Space Center (KSC).* This dataset was acquired by the NASA Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) on top of the Kennedy Space Center in Florida. It has 512×614 pixels and 224 bands in the 400–2500 nm spectral range. The data contains 13 upland and wetland classes, e.g., marshes, swamps, water, and scrubs.
- *Indian Pines (IN).* This dataset was acquired by the NASA AVIRIS from North-western Indiana. It has 145×145 pixels and 224 bands in the 400–2500 nm spectral range. It contains 16 vegetation classes including: alfalfa; different kinds of corns, grass, and soybeans; windrowed hay; oat; wheat; and woods. It also includes two mixed classes of building/grass/tree and stone/steel/towers.
- *Pavia University (PU).* This dataset was gathered by the Reflective Optics System Imaging Spectrometer from Pavia University in Northern Italy. It has a size of 610×340 pixels and 103 bands in the 430–860 nm spectral range. There are nine labeled classes in this dataset: asphalt, meadows, gravel, trees, painted metal sheets, bare soil, bitumen, self-blocking bricks, and shadows.
- *Salina Scene (SA).* This dataset was also captured by the NASA AVARIS. It has 512×217 pixels and 204 bands in the 400–2500 nm spectral range. It contains 16 classes of vegetables including different kinds of broccoli, lettuce, corn, and celery; vineyard fields including untrained, vertical trellis, developed soil vineyard, and untrained grapes; bare soils; fallow; and stubble.

Deep Learning with Multiple Quality Levels. As discussed in Sec. 3, our problem formulation and solution are general. To provide a concrete case study on using our solution, we describe it in the context of classification of hyperspectral data using deep learning. Specifically, we consider the state-of-the-art Spectral-Spatial Residual Network (SSRN) architecture in [23]. We define the loss as Eq.(3)

in which the difference between predicted $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L]$ and ground-truth $\boldsymbol{y} = [y_1, y_2, \ldots, y_L]$ one-hot label vectors and L is the number of classes. For comparison purposes, we report overall accuracy.

$$J(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sum_{i=1}^{L} y_i (\log \sum_{j=1}^{L} e^{\hat{y}_j} - \hat{y}_i) \quad (3)$$

SSRN consists of a spectral feature learning section, concatenated with a spatial feature learning section, followed by pooling and fully connected layers. However, this model requires the availability of the full data cube in its full quality as input. In our experiments, a high accuracy was achieved by SSRN framework when tested on data where all bands are available with full quality. Its accuracy, however, substantially dropped to around 15% when given all bands but with reduced quality or given a subset of bands with the original quality. We modify the training process to allow the model to process subsets of the bands and with different quality levels. We generate a representative training dataset that results in good accuracy as follows. We first acquire a labeled hyperspectral dataset that can be used for training. For each data point in the training set, we choose Q random values, $[L_1, L_2, \ldots, L_Q]$ such that $(L_1 + L_2 + \cdots + L_Q) \leq N$. These values indicate the number of bands chosen from each quality level, and can thus represent a sample selection. If the sum is smaller than N, some bands will be discarded. We then modify the data point according to that selection by changing the quality of each band to the respective selected quality. If a band is not selected, it is removed. The modified point is then placed in the modified dataset. We repeat this process 100 times for each data point in the training set, resulting a modified dataset that is 100 times the original training data size.

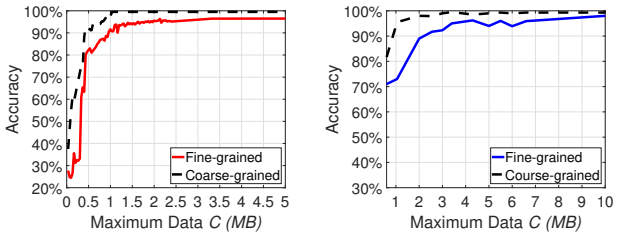
We divide each dataset into three parts: training, validation, and testing. Each of the first two parts is 10% of the dataset, and both are used to train and validate the CNN model. The third part has 80% of the dataset and is used in evaluating the accuracy of the proposed method and comparing it against others in the literature. We use a 4-level 2D-DWT decomposition to generate 5 different quality levels for each band. We then pre-process the data by normalizing each band to a mean value of zero and a unit variance.

To evaluate the scalability and adaptability of our solution, we define two levels of classification granularities. Specifically, for each of the PU and KSC datasets, we create 2 versions of the data. The first version uses the original labels of the datasets, with PU and KSC having 9 and 13 classes, respectively. We refer to this version as the fine-grained classification. In the second version, we merge similar classes together, resulting in a coarse-grained classification. For example, in the PU dataset, we merge the Asphalt and Bitumen classes together, and we also merge the Meadows and Trees classes, resulting in a total of 7 classes: Asphalt/Bitumen, Vegetation (Meadows/Trees), Gravel, Metal Sheets, Bare Soil, Bricks, Shadows. Similarly, the coarse-grained version of the KSC dataset has 8 classes instead of 13.

In the following subsections, we present representative samples of our results.

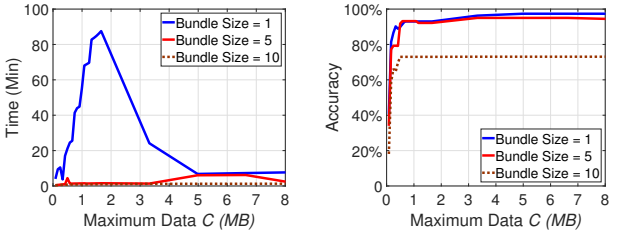
5.2 Performance of our Method

We analyze the performance of our method from different perspectives. First, we use our method to select bands and their qualities



(a) PU (b) KSC

Figure 4: Accuracy of the proposed method for different levels of granularities.



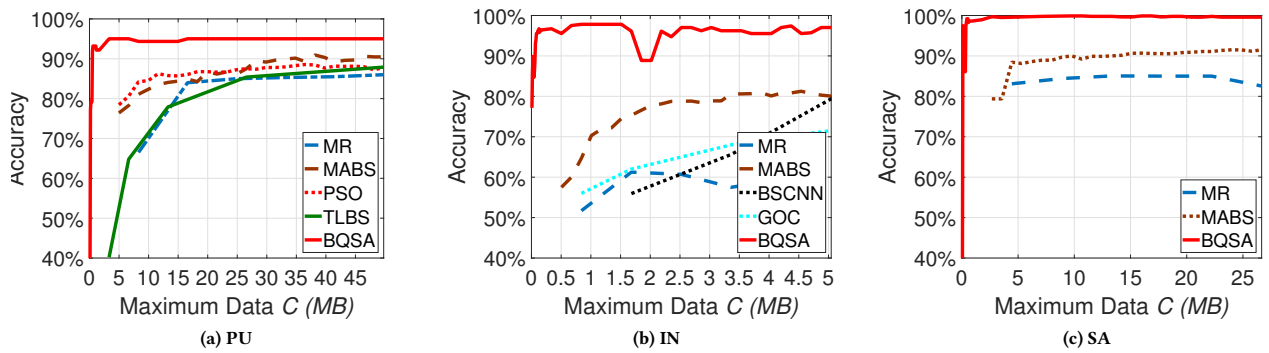
(a) Execution time (b) Classification accuracy

Figure 5: Effect of bundle size on accuracy and execution time.

and measure the accuracy it achieves given different maximum bit budget C. The accuracy is computed by comparing the class produced by our method for each pixel against the ground-truth class defined in the label of that pixel.

Two sample results from the PU and KSU datasets are shown in Figure 4. The results show that using our band-quality selection method, only a small portion of the data is required to achieve high accuracy even for fine-grained classification. For example, for a 97% accuracy for the PU dataset, Figure 4a, transmission of only 5MB of data is enough for identifying all individual classes in the dataset (the fine-grained curve in the figure). Given that the total data size is 68MB, our method can achieve almost 90% reduction in the amount of transmitted data while still achieving very high accuracy. Figure 4a also shows that the coarse-grained classification can be achieved with transmitting only 2MB out of the 68MB dataset, that is, with less than 3% of the total data, our method can identify 7 different classes with an accuracy of 98%. Furthermore, notice that our method enables gradual transmission of hyperspectral data, which is quite beneficial especially when capturing/exploring large areas. For example, a coarse-grained classification can be performed initially. Then, if a finer classification is needed, *additional* data can be transfered and combined with the data already received to produce the fine-grained classification. For the PU datasets, for example, additional 5MB of data is needed beyond the 2MB used for the initial coarse-grained classification.

We next analyze the effect of band bundling (as described in Sec. 4.3) on the accuracy and running time of our method. We show



(a) PU (b) IN (c) SA

Figure 6: Comparison between state-of-the-art methods and our method (BSQA) for three different datasets.

the results for the PU dataset in Figure 5 for bundles of sizes 1, 5, and 10 and for different maximum amount of data values C. Figure 5a shows the time to construct the branch and bound search tree and compute the optimal solution for a given maximum amount of data. This time was measured on a commodity workstation with GeForce GTX 980 GPU. As shown in the figure, using individual bands (bundle size of 1) may require a long time to find the optimal solution, especially for the data amounts in the range of 1–3 MB. Outside this range, the data size is either too small which makes the search tree very small, or the data is large enough which makes the algorithm quickly prune many branches. Bundles of size 10 bands run very fast, but produce lower accuracy as shown in Figure 5b. Bundles of size 5 bands achieve comparable accuracy to individual bands while requiring much smaller executing times. Thus, we use bundles of 5 bands in the rest of our experiments.

5.3 Comparison against State-of-the-Art

We compare the proposed method, which is referred to as BQSA, against the closest and most recent works in the literature for feature/band selection for hyperspectral images, which are MR [16], BSCNN [18], MABS [20], PSO [22], TLBS[15], and GOC [19]. These methods are briefly described in Sec. 2

We implemented some of these methods as close as we could to the descriptions in their corresponding papers. However, due to the lack of some details and parameter values, the performance of such methods was much lower than what is reported in their papers. To be conservative, we compared the performance of our method against the *best* results reported in the papers of the other methods on the datasets used in those papers. Specifically, for each method, we check the dataset used and find the achieved accuracy for various data amounts C from the figures in the paper. Some papers used number of bands; which we converted to data values given the size of each band in the dataset. Some of the previous methods were evaluated on a subset of the four hyperspectral datasets and/or on smaller ranges of data. These methods are not shown in some figures, and/or their curves do not span the whole range.

Three samples of our results are shown in Figure 6 for the PU, IN, and SA datasets. The figure clearly shows that our method outperforms all others for all datasets and for all data ranges. The accuracy improvements are substantial, especially for the smaller values of the available data, which are the most important and practical ranges for transmitting data over wireless channels. For example, for the PU dataset, the best accuracy produced by any method in the literature for $C = 5$ MB is less than 80%, while our method produces an accuracy of about 95%. For the IN dataset, the gain is even higher, where our method achieves an accuracy of more than 95% for a very small value of $C = 0.5$ MB, while other methods result in an accuracy less than 60%.

6 CONCLUSIONS

Hyperspectral imaging facilitates remote exploration of a scene by capturing the spectrum of each pixel. However, the high dimensionality and huge data size of hyperspectral images impose many challenges on transferring and processing such data. In this work, we proposed a method to select and transmit the most important parts of the hyperspectral data that maximize the utility of the received data. Our method jointly optimizes the number of selected bands and their qualities while taking into account previously sent information. We compared our method against the closest ones in the literature using four labeled hyperspectral imaging datasets. Our results show that the proposed method can achieve approximate classification with high accuracy using a small amount of data. For example, in one of our datasets, an accuracy of 98% for coarse-grained classification is achieved with less than 3% of the hyperspectral data transferred. In addition, our method significantly outperforms the state-of-the-art, especially in practical cases where the available bandwidth to transmit hyperspectral data is limited. For example, for one of our datasets, the best classification accuracy produced by any method in the literature was less than 60% when the maximum allowed data to transmit was 0.5 MB, while our method achieved an accuracy of more than 95% for the same amount of data.

ACKNOWLEDGMENT

This work is supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

REFERENCES

[1] Robert Allison, Joshua Johnston, Gregory Craig, and Sion Jennings. 2016. Airborne optical and thermal remote sensing for wildfire detection and monitoring. *Sensors* 16, 8 (2016), 1310.

[2] Hao Chen, Ye Zhang, Junping Zhang, and Yushi Chen. 2010. A BOI-preserving-based compression method for hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing* 48, 11 (2010), 3913–3923.

[3] Linsen Chen, Cc Dong, Du Chen, Han Li, Yuanyuan Zhao, Xun Cao, and Zhan Ma. 2017. Mobile Multispectral Video Streaming. In *Proc. of the Thematic Workshops of the ACM Multimedia Conference (MM'17)*. ACM, 126–134.

[4] Valerie C Coffey. 2012. Multispectral imaging moves into the mainstream. *Optics and Photonics News* 23, 4 (2012), 18–24.

[5] Chafik Egho and Tanya Vladimirova. 2014. Adaptive hyperspectral image compression using the KLT and integer KLT algorithms. In *In Proc. of the NASA/ESA Conference on Adaptive Hardware and Systems (AHS'14)*. IEEE, 112–119.

[6] Leyuan Fang, Nanjun He, Shutao Li, Antonio J Plaza, and Javier Plaza. 2018. A New Spatial-Spectral Feature Extraction Method for Hyperspectral Images Using Local Covariance Matrix Representation. *IEEE Transactions on Geoscience and Remote Sensing* (2018).

[7] C. Floridia, F. C. Salgado, J. B. Rosolem, F. R. Bassan, J. P. V. Fracarolli, R. S. Penze, and L. M. Pereira. 2016. Methane leak detection and spectral analysis by using only optical time domain reflectrometry in semidistributed remote optical sensors. In *2016 IEEE SENSORS*. 1–3. https://doi.org/10.1109/ICSENS.2016.7808684

[8] Wei Fu, Shutao Li, Leyuan Fang, and Jón Atli Benediktsson. 2017. Adaptive spectral–spatial compression of hyperspectral image with sparse representation. *IEEE Transactions on Geoscience and Remote Sensing* 55, 2 (2017), 671–682.

[9] Vinay Kumar and JK Ghosh. 2017. Camouflage Detection Using MWIR Hyperspectral Images. *Journal of the Indian Society of Remote Sensing* 45, 1 (2017), 139–145.

[10] M. Lahlimi, M. A. Kerroum, and Y. Fakhri. 2017. Band selection with the Bhattacharyya distance based on the Gaussian mixture model for hyperspectral image classification. In *2017 International Conference on Electrical and Information Technologies (ICEIT)*. 1–6. https://doi.org/10.1109/EITech.2017.8255282

[11] Astha Puri, Ershad Sharifahmadian, and Shahram Latifi. 2014. A comparison of hyperspectral image compression methods. *International Journal of Computer and Electrical Engineering* 6, 6 (2014), 493.

[12] D. M. Tratt, K. N. Buckland, E. R. Keim, and P. D. Johnson. 2016. Urban-industrial emissions monitoring with airborne longwave-infrared hyperspectral imaging. In *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. 1–5. https://doi.org/10.1109/WHISPERS.2016.8071711

[13] Philippe Lagueux Alexandre ValliÃÍres AndrÃÏ Villemaire Jean Giroux Vincent Farley, Martin Chamberland. 2007. Chemical agent detection and identification with a hyperspectral imaging infrared sensor. *Proc.SPIE* 6661, 6661 – 6661 – 9. https://doi.org/10.1117/12.736731

[14] J. Wang, X. Wang, K. Zhang, K. Madani, and C. Sabourin. 2018. Morphological Band Selection for Hyperspectral Imagery. *IEEE Geoscience and Remote Sensing Letters* 15, 8 (Aug 2018), 1259–1263. https://doi.org/10.1109/LGRS.2018.2830795

[15] J. Wang, K. Zhang, P. Wang, K. Madani, and C. Sabourin. 2017. Unsupervised Band Selection Using Block-Diagonal Sparsity for Hyperspectral Image Classification. *IEEE Geoscience and Remote Sensing Letters* 14, 11 (Nov 2017), 2062–2066. https://doi.org/10.1109/LGRS.2017.2751082

[16] Qi Wang, Jianzhe Lin, and Yuan Yuan. 2016. Salient band selection for hyperspectral image classification via manifold ranking. *IEEE transactions on neural networks and learning systems* 27, 6 (2016), 1279–1289.

[17] Arif Kurnia Wijayanto, Octo Sani, Nadia Dwi Kartika, and Yeni Herdiyeni. 2017. Classification Model for Forest Fire Hotspot Occurrences Prediction Using ANFIS Algorithm. In *IOP Conference Series: Earth and Environmental Science*, Vol. 54. IOP Publishing, 012059.

[18] Ying Zhan, Dan Hu, Haihua Xing, and Xianchuan Yu. 2017. Hyperspectral Band Selection Based on Deep Convolutional Neural Network and Distance Density. *IEEE Geoscience and Remote Sensing Letters* 14, 12 (2017), 2365–2369.

[19] F. Zhang, Q. Wang, and X. Li. 2017. Hyperspectral image band selection via global optimal clustering. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 1–4. https://doi.org/10.1109/IGARSS.2017.8126818

[20] Gong Li Zhang, Ma and Liu. 2017. Memetic algorithm based feature selection for hyperspectral images classification. *IEEE Congress on Evolutionary Computation* (2017).

[21] Jing Zhang and Guizhong Liu. 2007. Hyperspectral images lossless compression by a novel three-dimensional wavelet coding. In *Proc. of the ACM Multimedia Conference (MM'07)*. ACM, 759–762.

[22] Ma Zhang and Gong. 2017. Unsupervised Hyperspectral Band Selection by Fuzzy Clustering with Particle Swarm Optimization. *IEEE Geoscience and Remote Sensing Letters* 14, 5 (2017), 773–777.

[23] Zilong Zhong, Jonathan Li, Zhiming Luo, and Michael Chapman. 2018. Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework. *IEEE Transactions on Geoscience and Remote Sensing* 56, 2 (2018), 847–858.