DCT-based Hyperspectral Image Classification on Resource-Constrained Platforms

Eung-Joo Lee[1], Yi-Ting Shen[1], Lei Pan[1], Zhu Li[2], and Shuvra S. Bhattacharyya[1]

[1]University of Maryland, ECE Department and UMIACS, College Park, MD 20742, USA
[2]University of Missouri-Kansas City, CSEE Department, Kansas City, MO 64110, USA

***Abstract*—Deep learning based approaches to hyperspectral image analysis have attracted large attention and exhibited high performance in image classification tasks. However, deployment of deep learning based hyperspectral image analysis systems is challenging due to the computational complexity of deep learning and the large amount of data involved in hyperspectral images. To address this problem, this paper introduces a novel framework that integrates deep neural network (DNN) based image analysis by learning the network from discrete cosine transform (DCT) coefficients for hyperspectral image classification. The framework allows designers to derive diverse implementation configurations using a variable number of DCT coefficients for training. These configurations can be used to flexibly trade off classification accuracy and computational cost (e.g., based on specific characteristics of the device being used or based on time-varying operating requirements). Through experiments using a publicly available remote sensing dataset and a resource constrained Android platform, we demonstrate that our proposed approach enables the streamlined deployment of DNN-based hyperspectral image classification.**

I. Introduction

Hyperspectral image (HSI) classification involves mapping each pixel within an HSI into a set of pre-defined classes. The HSI classification problem has been explored in a wide range of applications, such as remote sensing, food processing, and surveillance. The vast amount of spectral information contained in HSIs provides the potential for high accuracy discrimination in challenging classification scenarios.

In recent years, deep learning techniques have been studied actively to address HSI classification problems. Deep learning has shown high accuracy classification performance in this context due to its capability of information extraction from both the spatial and spectral dimensions of HSI frames.

This paper is concerned with efficient deep learning for HSI classification on resource-constrained platforms. Accurate HSI analysis on resource constrained embedded platforms offers flexibility, mobility, and cost effectiveness, and opens up important new applications, such as agile surveillance from unmanned aerial vehicles. However, the high computational requirements of deep neural networks (DNNs) together with the large amounts of data involved in HSI signals make it challenging to deploy deep learning based HSI classification on resource-constrained platforms.

In this paper, we introduce a novel framework for resource-constrained HSI classification that integrates DNN-based and discrete cosine transform (DCT) analysis in a strategic manner. We refer to the framework as the DNN-DCT-integrated HSI Classification (DDHC) framework or just "DDHC" for short. DDHC provides a robust and flexible approach to spectral dimension reduction, which facilitates the optimized mapping of HSI classification onto different types of resource-constrained devices. We demonstrate the effectiveness of our proposed methods in comparison with related methods for HSI analysis through experiments using a publicly available HSI dataset.

The key contributions of this paper are summarized as follows.

- We propose an efficient and configurable framework for processing spectral information during HSI classification. The framework is based on applying DCT analysis across a variable number of spectral bands depending on considerations such as resource availability, real-time response requirements, and accuracy requirements.
- We demonstrate the ability of the proposed framework to achieve efficient and configurable trade-offs between classification performance and computational requirements. In particular, the model complexity of the DNN applied within the framework can be varied strategically for a given resource-constrained deployment.
- We evaluate the memory consumption and processing throughput of the proposed HSI classification framework on a resource-constrained Android device. The results give quantitative insight on the ability of the proposed framework to provide accurate and configurable HSI classification with lean resource usage.

The remainder of the paper is organized as follows. In Section II, we discuss related work on dimensional reduction and HSI classification. In Section III, we present details of the proposed DDHC framework. In Section IV, we develop experimental results that demonstrate the effectiveness of DDHC in achieving high accuracy HSI classification on a resource-constrained Android device. We conclude in Section V.

II. Related Work

Previous work on dimension reduction for HSI analysis includes Principal Component Analysis (PCA) methods, such as the approach of Rodarmel and Shan [8]. These methods seek to select a small subset of channels that contain most of the information available in the original hyperspectral image. Linear Discriminant Analysis (LDA) has also been studied as a dimensionality reduction method for HSIs (e.g., see [2]).

Another approach involves applying DCT-based information entropy for spectral band selection, and then applying a support vector machine to perform classification [9]. In contrast to this work, we apply the DCT as a preprocessing step on the HSI data rather than as a band selection tool. Moreover, we incorporate significant levels of configurability into the proposed framework with the objective of facilitating resource-efficient implementation.

In our previous work, we developed an adaptive system for real-time hyperspectral video processing on resource-constrained platforms [7]. The system jointly adjusts spectral dimension complexity and the multithreading configuration for efficiently utilizing the available processing resources. Subsequently, Pan et al. developed a scalable DNN-based system for resource-constrained HSI classification. The DNN in this system can be adapted at run-time to include different numbers of spectral bands in HSI analysis. The input spectral bands are selected according to a pre-determined priority list, and the DNN is trained to effectively handle a time-varying subset of bands, which can be tuned based on dynamically changing operating requirements.

The work presented in this paper differs significantly from the works described above. The primary distinguishing aspect of this work is that we apply a channel-wise DCT across a variable number of spectral bands for HSI classification. Additionally, we demonstrate how the proposed DCT-integrated HSI analysis framework leads to a configurable system involving variable numbers of spectral bands, and variable groupings of selected bands into subsets for DCT computation. The resulting configurability enables the optimization of trade-offs among computational cost, memory requirements, and HSI analysis accuracy when mapping HSI processing applications onto resource-constrained platforms.

III. Proposed Method

A key distinguishing aspect of DDHC is that it encodes information in the spectral bands of an HSI in the form of DCT coefficients. The discrete cosine transform (DCT) decomposes an image into a series of harmonic cosine functions, which utilize the real part of the fast Fourier transform. The DCT is known for its ability to isolate components of an image that contain most of the signal energy, and for this reason, the DCT is often used in image compression applications. Typically, most of the image information can be concentrated in a relatively small set of DCT coefficients, and in particular, the information is concentrated in the low frequency components. In DDHC, we exploit these properties of the DCT to achieve frequency separation that enables efficient learning of analysis tasks for HSIs.

At training time, the input to $DDHC$ consists of a labeled dataset involving $X_1, X_2, \ldots, X_m$ hyperspectral images, each consisting of n_r rows of pixels, n_c columns, and n_b spectral bands $\beta = \{B[1], B[2], \ldots, B[n_b]\}$. Two important parameters of DDHC are the number N_s of *subgroups*, and the subset of *selected bands* $\gamma \subset \beta$. The subset γ encapsulates the bands are that chosen for training and inference; all other bands are discarded in the analysis. The configurable subset γ allows the HSI processing system designer to configure processing complexity based on the available inference-time processing resources, and the required inference accuracy.

The total number of selected spectral bands $N_b = |\gamma|$, is partitioned into N_s subsets of adjacent bands, denoted $S[1], S[2], \ldots S[N_s]$, where each of the subsets $S[i]$ is referred to as a *subgroup*. The parameter N_s is restricted to be a divisor of N_b so that each each subgroup contains the same number of bands, which is denoted by N_k ($N_k = N_b/N_s$).

DDHC applies a one-dimensional (1D) DCT along the spectral bands of each pixel in a subgroup. Thus, given a pixel $p = (x_p, y_p)$ and a subgroup $S[i]$, where (x_p, y_p) are the spatial coordinates of the pixel, we derive a DCT representation D_p:

$$D_p[i][j] = \sum_{n=0}^{M-1} I(x_p, y_p, S[i][n+1]) cos[\frac{\pi}{M}(n+\frac{1}{2})(j-1)], \tag{1}$$

for $j = 1, 2, \ldots, M$. Here, $I(x_p, y_p, c)$ denotes the intensity in the enclosing hyperspectral image I at spatial coordinates (x_p, y_p) and spectral coordinate c; $S[i][1], S[i][2], \ldots, S[i][N_k]$ denote the bands contained in $S[i]$ in decreasing order of wavelength (increasing frequency); and $M \in \{1, 2, \ldots, N_k\}$ is a parameter of DDHC that gives the size of the DCT that is applied.

The transformation specified by Eq. 1 can be viewed as a channel-wise DCT on the hyperspectral image. This transformation results in an $N_s \times M$ matrix representation D_p for each HSI pixel p, which we refer to as the *subgroup-DCT matrix* of p.

The number of subgroups N_s determines the number of distinct DC components that are used in DDHC to encapsulate HSI information. As N_s increases, the number of DC components increases, but the amount of detail (additional DCT coefficients) that accompanies each component decreases. Therefore, we empirically select a setting for N_s in an effort to maximize the overall HSI analysis accuracy. More systematic approaches for determining N_s represent a useful area for future work. In our experiments, we use the empirically determined setting $N_s = 4$.

DDHC is designed to use a convolutional neural network (CNN) structure, which accepts as input a variable number of spectral channels that are encoded with DCT coefficients using Eq. 1. We refer to this CNN structure as the *DDHC Network*. An inference operation in the DDHC Network takes as input the subgroup-DCT matrices for a 7×7 patch of HSI pixels, and delivers as output a classification result for the pixel in the center of the patch. The classification result is an element of a set of pre-defined classification labels. The inference input can be viewed as a 4-dimensional tensor with size $7 \times 7 \times N_s \times M$. Figure 1 illustrates the process of deriving the subgroup-DCT matrices in DDHC. The derived matrices are used as inputs to the DDHC Network at inference time.

In the design of the DDHC Network, we use as a base structure the *multi-scale 3D DCNN* network proposed by He

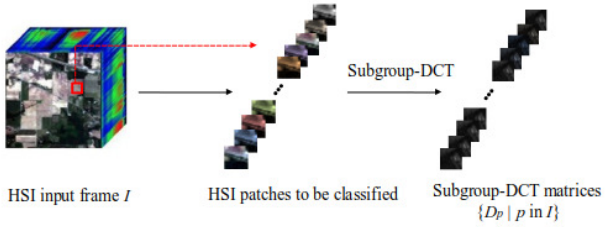

Figure 1. Derivation of subgroup-DCT matrices in DDHC.

et al. [6]. However, our use of this structure differs significantly from He's use in that we apply DCT coefficients as input to the network rather than directly applying HSI pixel values. Also, in our application of multi-scale 3D DCNN, we modify the network by using different numbers of kernels per layer. More specifically, we introduce a parameter N_c, which controls the number of kernels in each convolutional layer of the network. Higher values of N_c result in more complex networks that in general provide higher accuracy, but at the expense of higher computational and memory costs. The value of N_c is configured by the HSI processing system designer based on the available processing resources, and the version of the DDHC Network to be deployed is then trained for this setting of N_c.

The structure of the DDHC Network, which is inherited from the multi-scale 3D DCNN, is illustrated in Fig. 2. The network consists of 5 levels, where the first 4 levels contain 1, 4, 4, and 1 convolutional layers each, respectively, and the last level consists of a single fully connected layer.

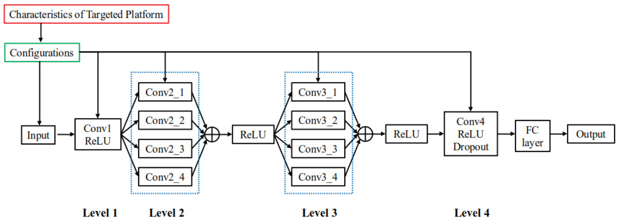

Figure 2. The structure of the DDHC Network.

Table I summarizes key parameters of the convolutional layers in the DDHC Network. Here, $\kappa_h, \kappa_w, \kappa_d$ represent the height, width, and depth (HWD) of the associated kernels, respectively. Similarly, $\sigma_h, \sigma_w, \sigma_d$ represent the HWD of the kernel stride, and ρ_h, ρ_w, ρ_d represent the HWD of the padding that is applied.

When DDHC is used with all available spectral bands ($N_b = n_b$), we use HWD values for the kernel size, stride, and padding as defined in the original multi-scale 3D DCNN network. When $N_b < n_b$, we reduce the depth of selected kernels using the parameters $\{\delta_i\}$ shown in Table I. This reduction in kernel depth is needed to properly match the associated convolutional layers to the dimension of their inputs. Similarly, we introduce the parameter Σ for the stride depth of the first convolutional layer and parameters $\{z_i\}$ for the padding depth of selected convolutional layers. These parameters are also introduced to properly adjust dimensions based on changes to the number of network input channels N_b.

Table I
KEY PARAMETERS OF THE CONVOLUTIONAL LAYERS IN THE DDHC NETWORK.

Kernel Name	No. of Kernels	Kernel Size κ_h, κ_w, κ_d	Kernel Stride $\Delta(\sigma_h, \sigma_w, \sigma_d)$	Padding ρ_h, ρ_w, ρ_d
Conv1	N_c	3, 3, δ_1	1, 1, Σ	0, 0, z_1
Conv2_1	N_c	1, 1, 1	1, 1, 1	0, 0, 0
Conv2_2		1, 1, δ_2		0, 0, z_2
Conv2_3		1, 1, δ_3		0, 0, z_3
Conv2_4		1, 1, δ_4		0, 0, z_4
Conv3_1	N_c	1, 1, 1	1, 1, 1	0, 0, 0
Conv3_2		1, 1, δ_2		0, 0, z_2
Conv3_3		1, 1, δ_3		0, 0, z_3
Conv3_4		1, 1, δ_4		0, 0, z_4
Conv4	N_c	2, 2, δ_5	1, 1, 1	0, 0, 0

Fig. 3 illustrates the process used in DDHC to classify an HSI pixel p. As detailed previously in this section, the process starts by deriving the subgroup-DCT matrix D_p. Then the DDHC Network is applied to D_p to derive a classification label for p. Key stages of the DDHC Network include the set of convolutional layers, the fully connected layer, and the output classification layer, as illustrated in Fig. 3.

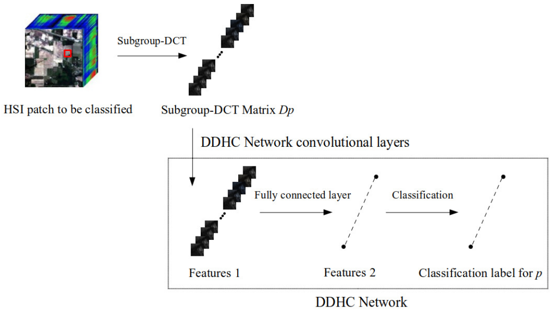

Figure 3. An illustration of the process used in DDHC to classify an HSI pixel.

DDHC enables the definition of multiple candidate networks by varying the set of selected bands γ, and the number of kernels per layer N_c (see Table I). Prior to training, the HSI processing system designer defines the set of candidate networks $G = \{g_1, g_2, \ldots, g_n\}$, which are trained separately to optimize based on the specific settings for the γ and N_c. By varying $N_b = |\gamma|$ and N_c, the designer can exercise control over the complexity/accuracy trade-offs encompassed by G. Any approach can be used to select the band-subset γ from among the full set β of available bands. In this sense, DDHC is compatible with a wide variety of band-subset selection approaches that can be found in the literature. In our experiments, we apply the Uniform Band Selection approach [4] to determine the band subset for each g_i for a given, designer-specified value of N_b.

The training dataset $X_1, X_2, \ldots, X_m$ is applied separately to each network g_i. The system designer can then evaluate the accuracy, execution time, and memory costs of the resulting trained versions of all candidate networks. This evaluation can be used to determine the best DDHC configuration for a given hardware platform that is being considered for deploying the HSI system.

IV. Experiments

We evaluate the proposed DDHC framework using the Indian Pines dataset [3]. This dataset contains 145×145 pixels with 224 spectral bands. As suggested in [1], we discard the water absorption bands, which results in a modified dataset consisting of 200 bands. Each pixel in the dataset is categorized into 16 classes. We use 80% of the pixels for training, and the remaining 20% for testing. The split is performed so that for each class, 80% of the pixels are in the training set and the remaining 20% are in the testing set. Among the pixels in the training dataset, we use 10% for validation, which involves optimizing hyperparameters related to the loss function and learning rate. For training, we use a batch size of 40, learning rate of 0.01, weight decay of 0.01, and epoch of 150. AdaGrad is used for the optimizer, and dropout is used for regularization. Additionally, flip augmentation is adopted to increase the amount of available training data.

Except where specified otherwise, we repeat each experiment five times and report the average values across the five trials. As a resource-constrained platform on which to evaluate DDHC, we use an Android mobile phone (OnePlus 7 Pro), which is equipped with an 8-core Qualcomm, 256 GB storage, and 12 GB of RAM. The proposed methods are compared with other two other HSI classification methods: HSI classification using Uniform Band Selection (UBS) [4] (without incorporation of DCT analysis), and HSI classification based on band selection using the Correlation Matrix (CM) [5]. For UBS and CM, we train the multi-scale 3D DCNN network separately to optimize its performance in terms of the band subsets derived by the respective methods. We use the PyTorch (Version 1.0) machine learning library for network implementation and training.

Table II lists parameter settings that we use in our experiments. Each row in the table corresponds to a different set of experiments for comparing DDHC, UBS, and CM. The first column provides different settings for N_b, the number of selected bands. The remaining columns provide values for configuring selected kernel depth, stride depth, and padding depth settings.

Table II
Parameter settings used in experiments.

N_b	(δ_1, δ_2, δ_3, δ_4, δ_5)	Σ	(z_1, z_2, z_3, z_4)
200	$(11, 3, 5, 11, 3)$	3	$(0, 1, 2, 5)$
64	$(11, 3, 5, 11, 3)$	3	$(0, 1, 2, 5)$
32	$(11, 3, 5, 7, 3)$	3	$(0, 1, 2, 3)$
16	$(11, 3, 3, 3, 3)$	2	$(1, 1, 1, 1)$
8	$(5, 3, 3, 3, 3)$	2	$(1, 1, 1, 1)$
4	$(3, 1, 1, 1, 2)$	1	$(0, 0, 0, 0)$

Table III presents experimental results comparing UBS, CM and DDHC in terms of classification accuracy. The rows labeled UBS, CM, and DDHC give the measured accuracy for the corresponding methods. The rows labeled N_{par} and N_{mac} in this table give the number of model parameters and the number of multiply-accumulate (MAC) operations, respectively, in the DNN configurations that are applied. The N_{mac} values are tabulated in terms of thousands of MACs (e.g., for $N_c = 16$ and $N_b = 200$ we have about 22.5 million MAC operations). The N_{par} and N_{mac} values help to assess the complexity of the DNN models in terms of computational cost and memory requirements. Results in Table III are tabulated for $N_c \in \{4, 8, 16\}$.

The results in Table III show that the DDHC consistently provides better classification accuracy compared to UBS and CM for all settings of N_b such that $N_b \geq 16$ (i.e., except for settings in which the number of selected bands is extremely small). We anticipate that the superior accuracy performance of DDHC can be attributed to its novel use of the DCT, and the ability of the DCT to encapsulate most of the available signal information in a small set of low-frequency DCT coefficients. Additionally, Table IV summarizes the compression ratio in each configuration; the configuration having the largest number of parameters is used as the baseline to calculate each compression ratio.

Fig. 4 and Fig. 5 illustrate data from Table III in a visual form. Fig. 4 illustrates trade-offs between classification accuracy and N_{mac} for different (N_b, N_c) settings of the three methods UBS, CM, and DDHC. Similarly, Fig. 5 illustrates trade-offs between classification accuracy and N_{par}. The vertical axis in each of the figures shows the percentage representation of the error rate, which is $1-\alpha$, where α denotes the classification accuracy. Fig. 4 and Fig. 5 illustrate that DDHC provides better ranges of accuracy-complexity trade-offs compared to the baseline methods.

Table III
Experimental results comparing UBS, CM, and DDHC.

N_b	200	64	32	16	8	4
N_c	16					
N_{par}	284897	84705	39137	18657	13441	9057
N_{mac}	22482	6246	2307	792	503	169
UBS	92.92	92.39	88.32	81.38	**73.59**	54.29
CM	92.92	92.2	85.59	81.87	71.55	**67.1**
DDHC	**93.29**	**93.81**	**91.82**	**85.07**	66.57	58.27
N_c	8					
N_{par}	139129	39033	16761	7289	4681	3513
N_{mac}	6383	1774	670	244	143	51
UBS	92.25	87.88	84.04	72.38	59.67	49.04
CM	92.25	86.93	79.64	71.05	**63.3**	**55.93**
DDHC	**93.08**	**90.95**	**85.13**	**75.71**	60.14	52.72
N_c	4					
N_{par}	68741	18693	7685	3141	1837	1509
N_{mac}	1977	550	214	84	44	17
UBS	87.44	83.14	74.63	60.33	49.61	39.39
CM	87.44	80.36	73.5	59.5	**49.64**	46.16
DDHC	**89.11**	**83.83**	**79.04**	**60.89**	48.23	**46.53**

To demonstrate its use on a resource-constrained platform, we implement DDHC on the Android device described earlier in this section. Table V and Table VI summarize results of

Table IV
Compression ratio in each configuration.

N_b N_c	200	64	32	16	8	4
16	1	3.36	7.28	15.27	21.20	31.46
8	2.05	7.30	17.00	39.09	60.86	81.10
4	4.14	15.24	37.07	90.70	155.08	188.80

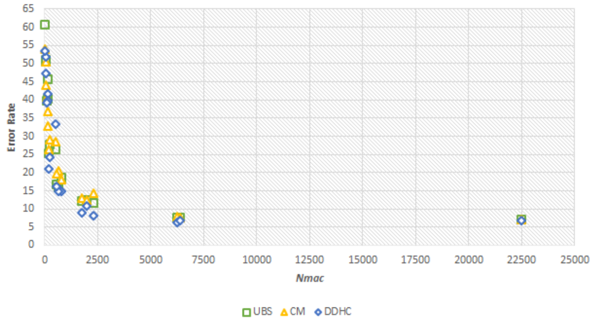

Figure 4. Trade-offs between classification accuracy and N_{mac}.

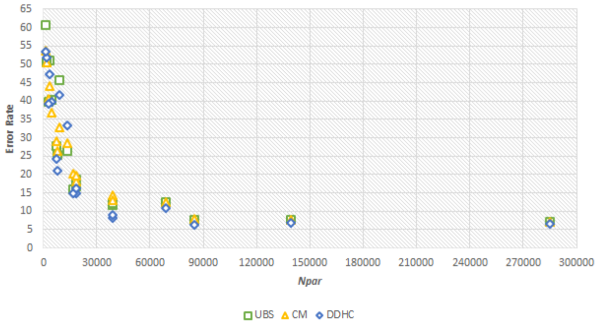

Figure 5. Trade-offs between classification accuracy and N_{par}.

our experimentation with DDHC on the targeted Android device. The measurements reported here are summarized over 50 trials; the average values and standard deviations are reported in the columns labeled "Avg." and "Std.", respectively. Throughput is measured in terms of Pixel Classifications Per Second (PCPS). The results shown in Table V and Table VI help to validate the utility of DDHC for resource-constrained deployment, and to quantify important operational trade-offs on a commercial, off-the-shelf resource-constrained platform.

Table V
Results on measured throughput (PCPS) from our experimentation with DDHC on the targeted Android device.

N_c	16		8		4	
N_b	Avg.	Std.	Avg.	Std.	Avg.	Std.
200	2.69	0.05	4.02	0.07	8.07	0.10
64	3.00	0.02	6.22	0.06	11.44	0.35
32	4.73	0.03	10.77	0.29	13.11	0.21
16	8.15	0.10	11.89	0.22	20.38	0.13
8	10.16	0.40	18.83	0.13	34.83	0.52
4	15.85	0.10	33.96	0.28	48.58	0.57

Table VI
Results on peak memory consumption (kilobytes) from our experimentation with DDHC on the targeted Android device.

N_c	16		8		4	
N_b	Avg.	Std.	Avg.	Std.	Avg.	Std.
200	378.0	122.5	324.9	17.4	282.9	16
64	384.1	54.1	274.9	13.4	227.3	7.5
32	274.9	9.9	220.2	8.6	196.4	0.7
16	213.3	11.1	192.7	1.8	193.2	1.3
8	190.9	1.6	190.4	1.4	190.6	1.1
4	190.1	1.3	190.3	1.1	190.2	1.1

V. Conclusion

This paper has introduced a novel framework for resource-constrained hyperspectral image (HSI) classification. The framework integrates deep-neural-network-based and discrete cosine transform (DCT) analysis in a strategic manner. The framework provides a robust and flexible approach to spectral dimension reduction, which enhances HSI classification accuracy and facilitates the optimized mapping of HSI classification onto resource-constrained devices. The effectiveness of the proposed methods has been demonstrated through experiments using a relevant HSI dataset. Interesting directions for future work include exploring systematic approaches for decomposing the set of selected spectral bands into subgroups for DCT analysis.

Acknowledgments

This research was supported in part by the Air Force Office of Scientific Research under the DDIP Program.

References

[1] N. Audebert, Bertrand Le Saux, and S. Lefevre. Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geoscience and Remote Sensing Magazine*, 7(2):159–173, 2019.

[2] T. V. Bandos, L. Bruzzone, and G. Camps-Valls. Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3):862–873, 2009.

[3] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe. 220 band AVIRIS hyperspectral image data set: June 12, 1992 Indian Pine Test Site 3, 2015.

[4] C.-I. Chang and S. Wang. Constrained band selection for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1575–1585, 2006.

[5] R. Hang, Z. Li, Q. Liu, and S. Bhattacharyya. PriNET: A prior driven spectral super-resolution network. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, page 6 pages in online proceedings, London, United Kingdom, July 2020.

[6] M. He, B. Li, and H. Chen. Multi-scale 3D deep convolutional neural network for hyperspectral image classification. In *Proceedings of the International Conference on Image Processing*, pages 3904–3908, 2017.

[7] H. Li, L. Pan, E. J. Lee, Z. Li, M. J. Hoffman, A. Vodacek, and S. S. Bhattacharyya. Hyperspectral video processing on resource-constrained platforms. In *Proceedings of the Workshop on Hyperspectral Image and Signal Processing*, pages 1–5, 2019.

[8] C. Rodarmel and J. Shan. Principal component analysis for hyperspectral image classification. *Surveying and Land Information Systems*, 62(2):115–222, 2002.

[9] S. S. Sawant and P. Manoharan. Unsupervised band selection based on weighted information entropy and 3D discrete cosine transform for hyperspectral image classification. *International Journal of Remote Sensing*, 41(10):3948–3969, 2020.