Dynamic, Data-Driven Hyperspectral Image Classification on Resource-Constrained Platforms

Lei Pan[1], Rijun Liao[2], Zhu Li[2], and Shuvra S. Bhattacharyya[1]

[1] University of Maryland, ECE Dept. and UMIACS, College Park, MD 20742, USA
{lpan1,ssb}@umd.edu
[2] University of Missouri-Kansas City, CSEE Department, Kansas City, MO 64110, USA
rijun.liao@mail.umkc.edu,lizhu@umkc.edu

1 Introduction

Hyperspectral image processing (HSIP) applications are becoming increasingly common in important application areas such as surveillance [1], medical diagnostic [2], forensics [3], and remote sensing [4]. The utility of HSIP in these fields stems from the high levels of spectral diversity and spectral resolution that hyperspectral images provide compared to conventional image acquisition approaches (e.g., see [5]). An HSIP application of fundamental importance is the problem of *image classification*, which involves mapping each pixel into a set of pre-determined classes.

The capability for data-driven, real-time processing of hyperspectral image classification on resource-constrained platforms opens up the potential for many novel applications. Prior work on HSIP system design has focused on systems with loose resource constraints or without strong emphasis on data-driven adaptivity (e.g., see [6] [7]). In this work, we introduce a novel system for hyperspectral image classification that integrates data-driven adaptivity with real-time, resource-constrained processing.

The rest of the paper is organized as follows. Section II discusses related techniques for scalable hyperspectral image classification and HSIP on resource-constraint. Section III describes scalable hyperspectral image classification design and underlying deep neural network (DNN) architecture. Section IV provides an overview of experiment setup for the proposed system on specific resource-constrained platform. Section V presents results of inference from selected hyperspectral remote sensing datasets with scalable DNN on the testing device. Finally, section VI provides our conclusion on the proposed system and future directions.

2 Related Work

Hyperspectral image (HSI) classification has becoming an important part of HSIP, especially in the area of remote sensing, where different type of land features need to be recognized from a long distance. Therefore classification

accuracy dominates the field of HSI classification study, as seen in [8–10]. Most recent HSIP studies focus on utilizing deep neural network (DNN) approach to enhance classification accuracy, with commonly studied datasets, such as Indian Pines, Pavia Center, Pavia Univerisity, and Salinas. Pixels in these datasets have unique label that belongs to certain classes, such as asphalt, tiles, cornfield etc. Many of the existing works have reported very high accuracy for these types of classification tasks, as seen in [8–11]. On the other hand, achieving high relatively high accuracy while ensuring accepatable throughput has drawn attend for HSIP studies for real-time applciations. There has been various studies on the real-time HSI classification stuides such as [6], which utilized specialized manycore MPPA (Massively Parallel Processor Array) platform, or in [12] the proposed real-time processing system is based on parallelized computation on GPU. In this paper, we present a dynamic, scalable HSI classification system that utilizes convolutional neural network on resource-constrained platforms. To the best of our knowledge, there has not been any prior studies on scalable inference DNN without using separate pre-trained models at run-time.

Previously in [13], the authors have successfully demonstrated the feasibility of processing hyperspectral video stream on an Android device. However, the application determines that the computational complexity of background subtraction algorithm can be efficiently executed with careful software design using dataflow modeling. However, to accommodate classification tasks, more computational intensive algorithms need to be carried out on resource-constrained platforms for real-time.

3 Approach

The proposed system for hyperspectral image classification is designed to adaptively configure system complexity to maximize classification accuracy subject to constraints on real-time performance and energy efficiency. The system is designed using a convolutional neural network (CNN) structure that accepts as input a variable number of hyperspectral input channels $C_1, C_2, \ldots, C_N$, and an integer $n_c \in [1, N]$. The value of n_c gives the number of input channels that is to be used to classify image pixels; the value of n_c can be varied dynamically by the system in which the CNN is embedded. The ordering $C_1, C_2, \ldots, C_N$ is interpreted as a priority list so that the set of channels used for a given classification operation is $\{C_i \mid i \leq n_c\}$.

We refer to our proposed methods as the Variable Band approach for Image Classification (VBIC). Our development of VBIC builds upon LDspectral, which was originally developed as a software tool for design optimization of dynamic, data-driven multispectral image processing systems [14], and has been extended more recently with support for hyperspectral image processing [13]. LDspectral in turn applies Lightweight Dataflow (LD), which is a compact set of application programming interfaces and associated libraries for dataflow-based design and implementation of embedded signal and information processing systems.

The network architecture for the CNN is shown in Figure 1. The scalable network we propose in this paper is based on the network proposed in [11]. Each block in the network represents either the input/output to the network, or the layers/operations in the scalable network at inference time. More specifically, the input to the network is a three dimensional tensor with size $N \times N \times n_c$, where $N \times N$ is the number of current pixel at (i, j) and its neighboring pixels from the input hyperspectral image, and n_c stands for number of selected spectral bands, equivalent to the number of input channels to the network, selected from the priority list $C_1, C_2, \ldots, C_N$, which is further generated by the band selection algorithm. Output of the network is the label of current pixel at (i, j) to tell which class it belongs to with corresponding probability.

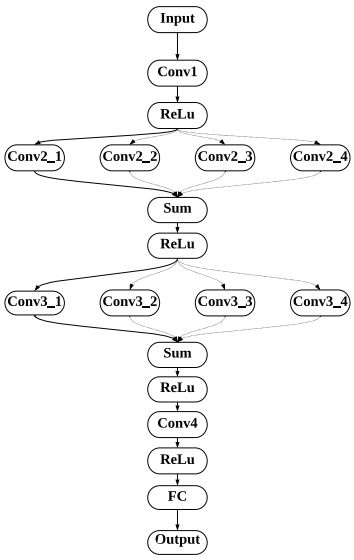

Fig. 1. Example network architecture for scalable CNN at inference time for 4 different number of spectral bands input. Figure was adapted from [11]

The convolutional layers used in the scalable CNN for VBIC are 3-D convolutional layers. The label `Convi_j` means that the convolutional layer at stage i and is the j-th layer at stage i, if applicable. `ReLu` is the rectified linear unit, `Sum` is the summation of results from previous stage, and `FC` stands for the fully connected layer before the final output. The dotted lines in the example figure 1 indicate that the connection between two layers are optional and determined at run-time, depending on the number of input channels, which can be adjusted at run-time.

The scalable aspect of the VBIC is illustrated by reusing existing pre-trained networks for fewer input channels. When n_c is smaller, it indicates that pro-

cessing speed more important than classification accuracy, hence the VBIC will reduce the number of parallel convolutional layers `Convi_j` being used correspondingly to reduce the computational complexity at run-time for the inference. When accuracy of VBIC is given more priority to the processing speed, then n_c will correspondingly increase and the number of number of parallel convolutional layers `Convi_j` will also increase to ensure the maximum achievable accuracy under given n_c. For the example in Fig 1, when $n_c = 30$, then only `Conv2_1` and `Conv3_1` are being connected at run-time for inference. When $n_c = 60$, `Conv2_2` and `Conv3_2` are added to the previously configured graph. When $n_c = 90$, `Conv2_2` and `Conv4_2` are further added to the previously configured graph. Finally, when $n_c = N$, where N is the maximum possible number of spectral bands in the given dataset, all convolutional layers are used at run-time for maximum accuracy.

The dynamic aspect is also the main objective of VBIC. Unlike commonly seen approaches for multi-configuration DNN implementation, VBIC only has one set of parameters for the scalable CNN. At run-time the parallel convolutional layers can be connected or disconnected without affecting other layers' pre-trained weights. This is achieved by training the smallest network with the minimum n_c, and then as n_c increases, more parallel convolutional layers `convi_j` are added to the training without modifying existing weights. The benefit of reusing existing weights is obvious: for resource-constrained platforms, storage space is usually very limited. Storing multiple set of network parameters on these platforms can result in the waste of valuable storage capacity which can be used for storing other important data, such as intermediate results or other features.

As n_c increases from 1 to N, the accuracy can be expected to increase, while the processing complexity (and hence the execution time and energy consumption) also increases. Our novel approach to designing and training the CNN provides systematic optimization of this trade-off between image classification accuracy and processing complexity. Our approach employs the DDDAS paradigm as an integrated part of the design and training process, thereby enabling dynamic-data driven adaptation of the CNN structure based on changing constraints on or priorities among real-time performance, energy consumption and accuracy.

4 Experiments

We demonstrate the proposed VBIC approach through experiments on an Android mobile phone (OnePlus 7 pro), which we use as a platform for prototyping resource-constrained image processing applications. In the experiments, we provide hyperspectral image input to the platform through flash storage. The platform is equipped with an 8-core Qualcomm Snapdragon 855 CPU, 12 GB of RAM, and 256 GB storage.

We evaluate our VBIC-based Android implementation using a number of commonly used datasets in remote sensing: Indian Pines and Pavia University. The Indian Pines dataset [15] has 145×145 pixels and 224 spectral bands with

wavelength ranging from 400 - 2500 nanometers(nm). The pixels are categorized into 16 classes. The Pavia University dataset has 610×610 pixels classified into 9 classes, and 103 spectral bands with spectral coverage of 430 - 860 nanometers (nm).

The VBIC network is trained in PyTorch. The network training is divided into 4 stages: Firstly the networks are trained with $n_c = 30$ and the first 30 bands with highest priority from the priority list $C_1, C_2, \ldots, C_N$ are selected. Only `Conv2_1` and `Conv3_1` are used for this stage of training. Secondly, n_c is increased to 60 and the first 60 spectral bands are selected from the priority list. `Conv2_2` and `Conv3_2` are added to the network training and the previously trained weights are fixed during backward propagation. Thirdly, n_c is further adjusted to 90 and `Conv2_3` and `Conv3_3` are again added to the network training. Same as previous stage, all existing weights are fixed during the new stage of training. Finally, n_c is set to be N to enable all spectral bands being involved during final training stage, and layers `Conv2_4` and `Conv3_4` are added to the network graph from stage three for training without changing previously trained weights for all other layers.

All networks are trained with input 3-D patch size of $7 \times 7 \times n_c$, batch size of 40, and weight decay of 0.01 for all the layers. The training algorithm used for all networks is AdaGrad [16], same as the algorithm used in baseline network [11]. The datasets are divided into training set and testing set. Training dataset set contains 80% of pixels from each class, and the rest 20% of pixels are used for inference. More details on the experiments and present results that demonstrate the utility of VBIC in enabling dynamic data-driven processing of hyperspectral image streams on resource-constrained platforms are shown in the section 5.

5 Results

In this section we present the experiment results for VBIC with the Pavia Univerisity dataset and Indian Pines dataset.

Table 1. Model size and overall accuracy comparison for VBIC with different number of spectral bands input under two datasets

	Dataset			
	Pavia University		Indian Pines	
n_c	Model size	Overall accuracy	Model size	Overall accuracy
30	18,042	88.01%	27,009	76.98%
60	45,210	93.64%	72,097	92.83%
90	73,402	95.58%	118,209	95.66%
Maximum	89,306	95.55%	284,897	96.98%

Table 1 shows the model size and overall accuracy for the VBIC under four different values for $n_c = \{30, 60, 90, \text{Maximum}\}$. The model size increases as the number of input channels increase to the VBIC. As expected, VBIC exhibits highest accuracy when using the maximum number of spectral bands. Although the overall accuracy for $n_c = 30$ is below 90% for both datasets, it is worth noting that the model sizes are significantly smaller than model sizes with larger n_c. This helps demonstrated the effectiveness of the trade-off between model size and overall accuracy when the scalable CNN method is employed in VBIC, while also justify the usage of fewer number of spectral bands on resource-constrained platforms when the memory usage is the primary concern for the user's use case. Figure 3 illustrates the model size and overall accuracy for different n_c under two datasets used.

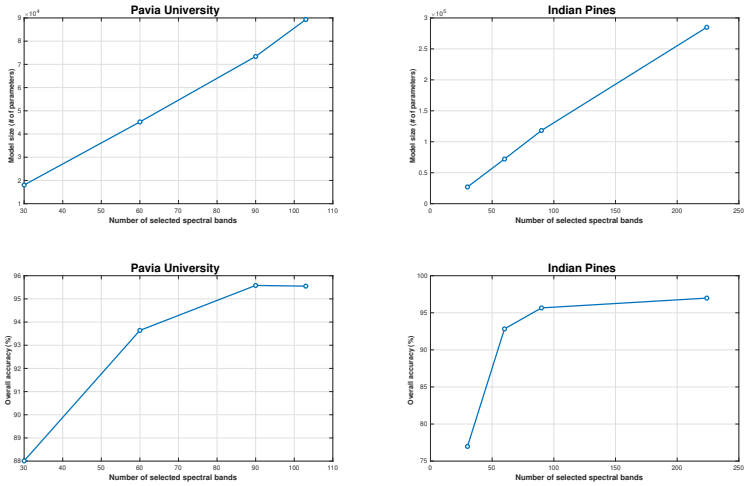

Fig. 2. Model size and overall accuracy comparison for VBIC under different number of n_c with Pavia University of Indian Pines datasets

Another important aspect for the inference of HSI classification is the throughput of the system, which is particularly important for a dynamic system like VBIC. To demonstrate its scalability at run-time and advantages when using different number of spectral bands n_c, we present the inference speed in Table 2

Inference of VBIC under each n_c is repeated 20 times for the same dataset. The average inference throughput and its standard deviation are recorded in Table 2.

Table 2. Inference Throughput comparison for VBIC with different number of spectral bands input under two datasets

	Dataset			
	Pavia University		Indian Pines	
n_c	Throughput (it/s)	Standard deviation (it/s)	Throughput (it/s)	Standard deviation (it/s)
30				
60				
90				
Maximum				

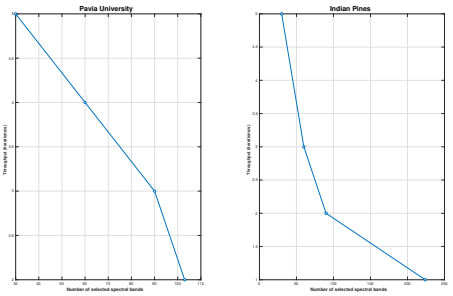

Fig. 3. Throughput comparison for VBIC under different number of n_c with Pavia University of Indian Pines datasets

6 Conclusion

In this paper, we have developed new scalable CNN inference methods VBIC for hyperspectral image classification on resource-constrained platforms. Using these methods, we have prototyped an novel system for convolutional neural network inference on embedded systems, conducted experiments using the proposed systems, and demonstrated the dynamicness and adaptiveness of proposed systems. Future directions include more compact network design without adding/removing network layers at inference time by adjusting number of filters in each convolutional layer, exploration of embedded device GPU and dedicated artificial intelligence accelerator e.g. the DSP on Qualcomm mobile CPUs and Qualcomm Hexagon tensor accelerator to further improve the system throughput, and incorporate VBIC into LDspectral framework completely to further expand the practicality of our proposed system.

References

1. P. W. Yuen and M. Richardson, "An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition," *The Imaging Science Journal*, vol. 58, no. 5, pp. 241–253, 2010.

2. J. Freeman *et al.*, "Multispectral and hyperspectral imaging: applications for medical and surgical diagnostics," in *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, 1997, pp. 700–701.
3. G. Edelman, E. Gaston, T. Van Leeuwen, P. Cullen, and M. Aalders, "Hyperspectral imaging for non-contact analysis of forensic traces," *Forensic science international*, vol. 223, no. 1-3, pp. 28–39, 2012.
4. S. B. Serpico and L. Bruzzone, "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 7, pp. 1360–1367, 2001.
5. J. Patrick, R. Brant, and E. Blasch, "Hyperspectral imagery throughput and fusion evaluation over compression and interpolation," in *Proceedings of the International Conference on Information Fusion*, 2008, pp. 1–8.
6. D. Madroñal, R. Lazcano, R. Salvador, H. Fabelo, S. Ortega, G. M. Callico, E. Juarez, and C. Sanz, "SVM-based real-time hyperspectral image classifier on a manycore architecture," *Journal of Systems Architecture*, vol. 80, pp. 30–40, 2017.
7. S. M. Chai, A. Gentile, W. E. Lugo-Beauchamp, J. Fonseca, J. L. Cruz-Rivera, and D. S. Wills, "Focal-plane processing architectures for real-time hyperspectral image processing," *Applied Optics*, vol. 39, no. 5, pp. 835–849, 2000.
8. Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, and G. Bai, "Hsi-cnn: a novel convolution neural network for hyperspectral image," in *2018 International Conference on Audio, Language and Image Processing (ICALIP)*. IEEE, 2018, pp. 464–469.
9. A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-d deep learning approach for remote sensing image classification," *IEEE Transactions on geoscience and remote sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.
10. Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network," *Remote Sensing*, vol. 9, no. 1, p. 67, 2017.
11. M. He, B. Li, and H. Chen, "Multi-scale 3d deep convolutional neural network for hyperspectral image classification," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3904–3908.
12. Z. Wu, Q. Wang, A. Plaza, J. Li, L. Sun, and Z. Wei, "Real-time implementation of the sparse multinomial logistic regression for hyperspectral image classification on gpus," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 7, pp. 1456–1460, 2015.
13. 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*, 2019.
14. H. Li, K. Sudusinghe, Y. Liu, J. Yoon, M. van der Schaar, E. Blasch, and S. S. Bhattacharyya, "Dynamic, data-driven processing of multispectral video streams," *IEEE Aerospace & Electronic Systems Magazine*, vol. 32, no. 7, pp. 50–57, 2017.
15. 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," *Purdue University Research Repository*, vol. 10, p. R7RX991C, 2015.
16. J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of machine learning research*, vol. 12, no. Jul, pp. 2121–2159, 2011.