CONVOLUTION BASED SPECTRAL PARTITIONING ARCHITECTURE FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Ringo S.W. Chu[*1], *Ho-Cheung Ng*[†], *Xiwei Wang*[‡] *and Wayne Luk*[†]
[*]University College London, [†]Imperial College London, [‡]China Academy of Space Technology

ABSTRACT

Hyperspectral images (HSIs) can distinguish materials with high number of spectral bands, which is widely adopted in remote sensing applications and benefits in high accuracy land cover classifications. However, HSIs processing are tangled with the problem of high dimensionality and limited amount of labelled data. To address these challenges, this paper proposes a deep learning architecture using three dimensional convolutional neural networks with spectral partitioning to perform effective feature extraction. We conduct experiments using Indian Pines and Salinas scenes acquired by NASA Airborne Visible/Infra-Red Imaging Spectrometer. In comparison to prior results, our architecture shows competitive performance for classification results over current methods.

***Index Terms*—** Hyperspectral Imagery, Convolutional Neural Network, Landcover Classification, Remote Sensing, Pattern Recognition

1. INTRODUCTION

Hyperspectral images (HSIs) contain spectrum information for each pixel in the image of a scene, where each spatial pixel is a spectral vector composed of hundreds of contiguous narrow electromagnetic bands reflected or radiated by the detecting materials. HSI classification involves assigning a categorical class label to each unlabelled pixel based on the corresponding spectral and/or spatial feature [1]. With the advent of new hyperspectral remote sensing instruments and their increased temporal resolutions, the amount of high dimensional hyperspectral data is increasing. This results in new practical and theoretical problems due to the high dimensionality where traditional algorithms developed for multi-spectral imagery may no longer be suitable.

Convolutional neural networks (ConvNets/CNN) have shown potential in hyperspectral imagery classification as they use extensive numbers of parameters for feature learning [2]. In spite of the potentiality, high variability spectral signature properties of HSIs complicates the corresponding CNN designs; Time consuming and expensive manual labelling of HSIs has limited the number of training samples. These problems have obstructed and reduced the predictive power of CNN models.

[1]Work performed at Imperial College London.

Existing CNN models for HSI classification are often based on one dimensional or two dimensional CNN architectures. The former set of models are adapted for spectral feature learning [3], while the later sets of models explore local spatial feature learning at each band [4]. These models show deficiency in performing feature extraction on multiple dimensions. Three dimensional CNN is exploited by researchers [5] and yet the network structure is often too sophisticated for deployment on real-time embedded devices such as CPU, FPGA or even GPU. Traditional machine learning methods including Logistic Regression [6] and Kernel-based SVM [7] are proposed by researchers but it has been reported that classification accuracies is inferior and less favourable.

In this paper, we propose a novel 3D CNN architecture that tackles these challenges by *a*) *spectral partitioning within the network to process pixels in each dimension,* and by *b*) *efficient spatial-spectral feature extraction* Essentially, the proposed model first performs a spatial transformation via 2D convolution. The transformed image is partitioned on the spectral level and split into segments for efficient processing. 3D convolution is then applied to each segment. Finally, convoluted segments are concatenated and summarized with fully-connected layers with dropout as regularization to prevent over-fitting. Our architecture is shown to be robust and stable where the model is more accurate compared to BASS Net [3] and 2D-CNN [4]. We also demonstrate and train our architecture using as minimal training samples as possible, which in our case, 20% and 10% of labelled data from Indian Pines and Salinas dataset respectively. All source code can be found in section III.

2. PROPOSED METHODOLOGY

Hyperspectral images are typically represented as a data cube in dimension (x, y, λ), where x and y represent spatial dimensions with space information of pixels, and λ represents the third dimension with a spectral vector that can be used for distinguishing different materials and objects. To improve the classification accuracy and reduce the number of training samples, we propose the use of 3D CNN to perform spatial-spectral feature extraction with spectral partitioning over (x, y, λ) dimensions.

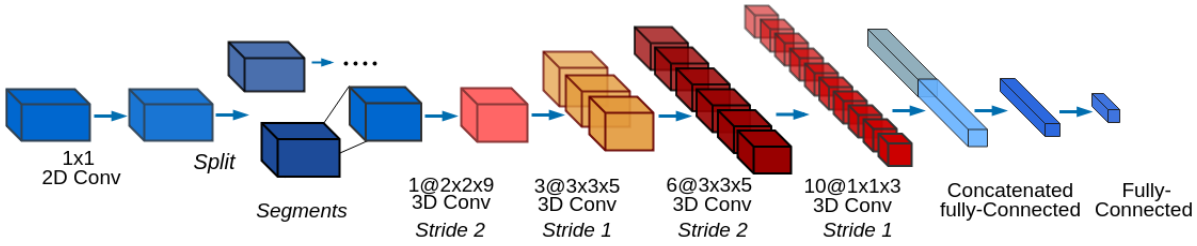

Fig. 1: An illustration of the proposed neural network architecture for HSI classification. The diagram is showing the operation of one of the two segments only. Both segments follow identical 3D convolution operation.

2.1. Three Dimensional Convolutional Neural Networks

As mentioned, 3D CNN is capable of capturing features from both spatial and spectral dimensions when compared to traditional 1D and 2D CNNs. The value of a neuron at a given position (x, y, z) is denoted as:

$$v_{ij}^{xyz} = \sigma\left(\sum_{m}\sum_{h=1}^{H_i-1}\sum_{w=1}^{W_i-1}\sum_{r=1}^{R_i-1} k_{ijm}^{hwr} v_{(i-1)m}^{(x+h)(y+w)(z+r)} + b_{ij}\right) \quad (1)$$

where i indicates the current layer, m indexes the feature map in the $(i-1)$th layer connected to the current feature map. H_i and W_i represent the height and width of the kernel, R_i represents the depth of the kernel towards the spectral dimension, k_{ijm}^{hwr} is the value of (x, y, z) on feature m, b_{ij} is the bias term of feature j on layer i and σ indicates the activation function.

2.2. Proposed Architecture

Our architecture takes an input of a small image cube from hyperspectral data. The small cube is taken from 24 neighbours of its central pixel with patch size $= 5$, where the central pixel is required for classification (Fig. 2). In other words, the input image cube has size $5 \times 5 \times N_b$, where N_b is the spectral size of the hyperspectral image. From our prior experiment, a patch size $= 5$ contributes to the finest result in classification when compared to patch size $= 3$ or 7.

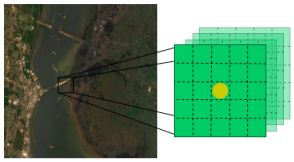

Fig. 2: Cubic data extract from a HSI as input to proposed CNN architecture.

This cubic data is then fed to the proposed CNN architecture as briefly illustrated in Figure 1. The architecture begins with one 2D convolution layer using a 1×1 filters for linear transformation of each input spectral band. *Spectral partitioning* is then performed on the convoluted image, where the image is split into two segments with a non-overlapping spectral axis. Each segments are followed by a stack of four 3D convolutions sequentially with different kernel size. Operations of each segments are identical, kernel weights and biases of the stack of convolutions are shared across the segments to reduce total number of parameters.

In the first 3D CNN layer, it has one convolution filter of size $2 \times 2 \times 9$ with sub-sampling stride 2 on λ. Consider the data cube contains redundant information, the purpose of this layer is to learn the low-level features and reduce the high-dimensionality. The second layer has three filters with size $3 \times 3 \times 5$ and stride 1. This layer is designed for learning local mid-level features of HSI. The third layer has five $3 \times 3 \times 5$ kernels with stride 2 to further reduce the dimensionality on the spectral level. The final 3D-CNN layer consists of 10 $1 \times 1 \times 3$ kernels with stride 1, which aims at extracting high-level spectral features.

The convoluted segments are concatenated and fed to two Fully-Connected layers (FC); The first one has 120 neuron units and the second one has number of units equal to the number of categorical classes of the dataset. Note that Rectify Linear Units (ReLu) are employed as activation functions on every layer except the final layer, where Softmax function is applied for classification. The final label for the pixel is given by the $\arg\max$ from the Softmax function.

2.3. Optimisation, Learning, Training and Inference

With limited dataset provided in the training stage, overfitting can become a serious problem where classification performance is good on the training set, but low on the testing set. Therefore, the training is regularised by applying dropout at fully-connected layer, with ratio set to $p = 0.5$.

We carry out the training by minimizing the Cross-Entropy object function using Adaptive Moment Optimization gradient descent (Generally known as Adam). The adam is set to have initial learning rate 5×10^{-4}. The batch size is set to have 50 for each training iteration and require 650 epochs for convergence.

The segments described in Section 2.2 are placed sequentially during the training stages; During the inference stage, segments can be adjusted to align in sequential, parallel or pipeline, and can be customized with regards to runtime constraints. In this paper, we demonstrate our model by placing segments in parallel.

Table 1: Comparison of proposed network versus other methods in classification accuracy (%) on Indian Pines dataset.

Class	*BASS Net*	*Conv2D*	*Our Model*
Corn-notil	94.77	91.97	98.23
Corn-mintill	95.02	95.98	97.58
Corn	94.94	98.88	96.07
Grass-pasture	99.17	99.72	99.17
Grass-trees	99.63	99.08	99.27
Hay-windrowed	100.00	100.00	100.00
Soybean-notil	96.29	98.77	98.77
Soybean-mintil	97.88	98.26	96.36
Soybean-clean	97.52	90.54	98.20
Woods	97.68	98.63	99.05
BGTD	98.28	94.48	97.93
OA	97.38±0.2	97.00±0.1	97.98±0.2
AA	97.23±0.2	96.76±0.1	98.92±0.2

3. EXPERIMENTS & RESULTS

3.1. Dataset and Preprocessing

The Indian Pines Scene and Salinas Scene datasets, which were acquired by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over Northwestern Indiana and Salinas-Valley, California, are used in this experiment. Indian Pines scene provides 224 spectral channels in the wavelength ranges from 0.4 to 2.5μm. 200 channels remain after discarding the water absorption regions. We remove some classes in this dataset due to insufficient samples. Similar to Indians Pines scene, Salinas scene also consists of 224 spectral bands. 20 channels are discarded for correction and only 204 channels remain for the experiment.

We randomly select 20%, 5% and 75% from each class as our training data, validation data and testing data respectively for Indian Pines dataset. Similarly, we randomly select 10%, 5% and 85% as the training, validation and testing set for Salinas dataset. All pixel values are calibrated through normalization via the following transformation $z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$. where $x = (x_1 ... x_n)$ represents all pixel values of an HSI.

3.2. Classification Result

We implement our model using TensorFlow, and Scikit-learn and it is trained on a single NVIDIA GeForce GTX 1080Ti GPU. Source code can be viewed on Github as an open source project [2].

We compare the performance of our approach to two other recent techniques: BASS Net [3] and 2D-CNN [4] to obtain the overall accuracy (OA) and the average per-class accuracy (AA) using the same dataset. Accuracy values are calculated based on the average of five classification runs. Table 1 and

[2] https://github.com/custom-computing-ic/SpecPatConv3D-Network

Table 2: Comparison of proposed network versus other methods in classification accuracy (%) on Salinas dataset.

Class	*BASS Net*	*Conv2D*	*Our Model*
Brocoli green weeds 1	100.00	99.78	100.00
Brocoli green newline weeds 2	99.72	100.00	99.82
Fallow	100.00	100.00	100.00
Fallow rough plow	99.84	100.00	100.00
Fallow smooth	99.83	98.91	99.50
Stubble	100.00	100.00	100.00
Celery	99.69	99.56	99.91
Grapes untrained	97.02	91.37	94.20
Soil vinyard develop	100.00	99.9	99.60
Corn senesced green weeds	96.78	98.01	97.46
Lettuce romaine 4wk	95.89	97.37	97.47
Lettuce romaine 5wk	98.34	100.00	99.59
Lettuce romaine 6wk	100.00	100.00	100.00
Lettuce romaine 7wk	99.26	99.47	99.37
Vinyard untrained	84.31	87.96	96.52
Vinyard vertical trellis	99.13	99.07	99.32
OA	96.84±0.1	96.27±0.1	98.73±0.1
AA	98.12±0.1	98.21±0.1	98.92±0.1

Table 2 summarise the results on OA and AA for different approaches on Indian Pines and Salinas datasets.

Our model obtains the best results compared to other models, with 98.0% and 98.7% in overall accuracy on Indian Pines scene and Salinas scene respectively. This demonstrates the superiority of the proposed 3D CNN for hyperspectral classification, where the corresponding classification maps of the proposed model clearly distinguish the boundaries between different classes (Figure 3 and Figure 4).

Finally, the changes in accuracy against the epochs for the training, validation and testing data is shown in Figure 5 for both datasets. Intuitively, the network converges after 100 epochs and reaches to an optimal point with a stable rate after 600 epochs for the training data. After the convergence, the testing and validation accuracy are closed to training accuracy. This demonstrates the robustness of our model where the over-fitting problem is diminished and the features on HSI are extracted and learned by the proposed model.

4. CONCLUSION

This paper proposes an architecture for HSI classification, with spectral partitioning to reduce dimensionality and spatial-spectral features extracted by 3D CNN. Experiments show that our method outperforms comparable methods regarding classification accuracy while using a fewer amount of training

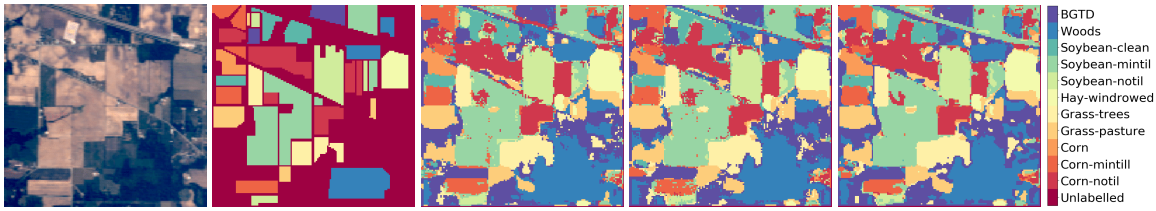

Fig. 3: Indian Pines dataset: True colour composite, Ground truth image, Classification map of BASS-Net, Conv2D and Our method. (Left to right)

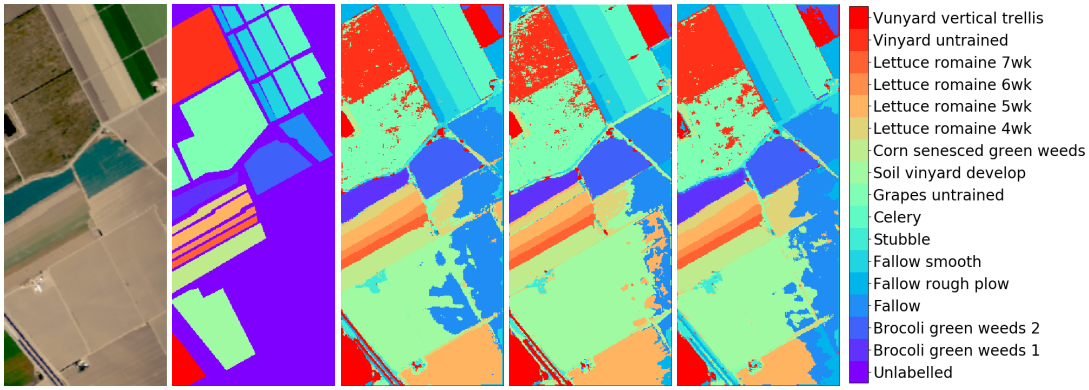

Fig. 4: Salinas dataset: True colour image, Ground truth image, Classification map of BASS-Net, Conv2D and Our method. (Left to right)

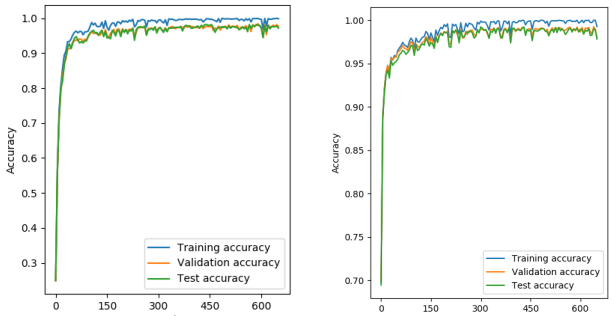

Fig. 5: The changes in accuracy against the epochs for (a) Indian Pines dataset, (b) Salinas dataset.

data.

Currently, the proposed architecture consists of multiple layers of 3D CNN, which is still very computationally intensive. Future work will include complexity reduction of the model. For example, extending the network with other types of layers such as depth-wise convolution and migrating the design onto embedded devices such as FPGA to perform real-time processing.

5. ACKNOWLEDGEMEMT

The authors are grateful for the support by Intel, United Kingdom EPSRC (grant numbers EP/I012036/1, EP/L00058X/1, EP/L016796/1, EP/N031768/1), European Union Horizon 2020 Research and the Lee Family Scholarship.

6. REFERENCES

[1] S. P. Sabale and C. R. Jadhav, "Hyperspectral Image Classification Methods in Remote Sensing - A Review," in *2015 International Conference on Computing Communication Control and Automation*, 2015, pp. 679–683.

[2] Y. Luo et al., "HSI-CNN: A Novel Convolution Neural Network for Hyperspectral Image," in *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, 2018, pp. 464–469.

[3] A. Santara et al., "BASS Net: Band-Adaptive Spectral-Spatial Feature Learning Neural Network for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5293–5301, Sep. 2017.

[4] Y. Chen et al., "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, Oct 2016.

[5] Z. Tian et al., "Hyperspectral Classification Via Spatial Context Exploration with Multi-Scale CNN," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018, pp. 2563–2566.

[6] J. M. Haut et al., "Cloud Implementation of Logistic Regression for Hyperspectral Image Classification," in *17th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE)*, 2017, pp. 1030–1041.

[7] J. Leng et al., "Cube-CNN-SVM: A Novel Hyperspectral Image Classification Method," *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1027–1034, 2016.