TRIPLET CONSTRAINED DEEP FEATURE EXTRACTION FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Fahim Irfan Alam, Jun Zhou*, Alan Wee-Chung Liew*, Jun Jo*, Yongsheng Gao***

*School of Information and Communication Technology, Griffith University, Australia
**School of Engineering, Griffith University, Australia
Emails: fahimirfan.alam@griffithuni.edu.au, {jun.zhou, a.liew, j.jo, yongsheng.gao}@griffith.edu.au

ABSTRACT

Convolutional neural networks (CNNs) have demonstrated significant performance in various visual recognition problems in recent years. Recent research has shown that training multilayer neural networks can extensively improve the performance of hyperspectral image (HSI) classification. In this paper, we apply a triplet constraint property on a 3D CNN. This method directly learns a mapping from images to a Euclidean space in which distances directly correspond to a measure of spectral-spatial similarity. Once this embedding has been established, classification can be implemented with such embeddings as feature vectors. Moreover, we also augment the size of the training samples in different band groups. This produces different yet useful estimation of spectral-spatial characteristics of HSI data and contributes considerably in accurate classification. This method is evaluated on a new dataset and compared with several state-of-the-art models, which shows the promising potential of our method.

Index Terms— Image Classification, Convolutional Neural Network, Hyperspectral image, Triplet constraint.

1. INTRODUCTION

Hyperspectral imaging technology analyses images in contiguous spectral bands over a given spectral range. It enables more accurate and detailed spectral information extraction than is possible with other types of remotely sensed data. The spatial relationships among various spectral responses in a neighborhood can also be explored, which allows development of spectral-spatial models for accurate image classification.

A large number of supervised classification models have been explored recently based on decision trees, random forests, and support vector machines (SVMs) [1, 2]. A random forest [3] constructs several decision trees during training and output the classes of the input HSI pixels by integrating predictions of the individual trees. On the contrary, for problems which are not linearly inseparable, SVMs map data to a kernel-included high-dimensional feature space. They then find an optimal decision hyperplane that can best separate data samples. With limited training data, SVMs have been considered to be an effective model for HSI classification task.

However, different atmospheric conditions, complex light scattering mechanisms and both inter-class & intra-class variability result in an inherent nonlinear procedure for HSI data [4]. The ability of shallow models like random forests and SVMs to handle such nonlinear hyperspectral data is limited whereas deep models are able to extract hierarchical, abstract and invariant features, which are generally more robust to the nonlinear HSI data [4]. As a result, deep models achieve higher classification accuracy than the traditional classifiers in many cases. The use of deep models has also demonstrated considerable success in classifying spectral-spatial features as well [5].

Most of the deep model-based classification methods employ 1-D deep learning architectures, equipped with fully connected layers. As a result, the number of trainable parameters to be estimated is extensively large which is an undesirable settings for remote sensing image classification since the training samples are often limited [6]. Moreover, important structure information residing in HSI data are substantially lost due to the 1-D networks and vector-based feature alignment process since the data has an inherent 2-D structure in the spatial domain.

Yu *et al.* [7] proposed a CNN architecture which uses a convolutional kernel to extract spectral features along the spectral dimension and used normalization layers and a global average pooling layer to obtain features in the spatial domain. On the other hand, 3D-CNN can learn the signal changes in both spatial and spectral dimensions of local spectral images. Therefore, it can extract significant discriminative information for classification and exploit powerful structural characteristics for hyperspectral data. Recently, this model was adopted by Chen *et al.* [5] for feature extraction and classification of hyperspectral images based on three-dimensional data across all the bands. This approach combines both spectral and spatial information and later used regularization to improve performance. Similar work has been proposed to extract spectral-spatial features from pixel or pixel-pairs using

deep CNN [8].

However, without abundant training samples, a deep network in general, faces the problem of "overfitting" which means the representation capability may not be sufficient to perform well on test data. It is therefore very important to increase the size of the training samples in order to handle this overfitting issue. For remote sensing applications, the available labeled samples can be utilized by interpolating those to produce new virtual samples which can represent a powerful estimation of spectral-spatial characteristics of the HSI data.

Spectral-spatial features can be further exploited to design important constraints for the training of deep models. Instead of using additional steps such as regularization as in [5] to improve classification performance, spectral-spatial characteristics of the available samples can be compared in the feature space to measure similarities between samples. In this regard, minimizing differences between samples belonging to the same class and maximizing differences between samples belonging to difference classes through learning by 3D CNN can build powerful embedding as feature vectors. This embedding is expected to provide useful cues for the subsequent classification of our HSI data. This is the motivation of this work.

In our method, we supply the samples in a batch and learn a Euclidean embedding for the samples using a 3D CNN network. To implement the idea, we extended the concept of "triplet constraint" presented in [9] for remote sensing images. We train our network in such a way that the squared distances in the embedding space correspond directly to the similarity between the samples. Those samples of the same class should have smaller distances and samples of different classes have larger distances. Our 3D-CNN computes the loss of the model by separating the positive pair of samples from the negative sample by a distance margin.

The sample generation is further improved by employing data augmentation to produce more effective feature embeddings. In this regard, we treat hyperspectral images as spectral groups consisting of the image spanning over a few spectral bands instead of all the bands across the spectra as in [5]. We apply sample fusion between pairs of available training samples of the same class and transformation operations to produce virtual samples from these spectral groups. These augmented samples will be more accurate estimations of local spectral-spatial structure description of the data and hence, can contribute in producing better feature embeddings. Once we establish such embeddings, we map those into classes by adding a classification loss and perform a final classification on our data. Our framework is illustrated in Fig. 1.

2. METHODOLOGY

In this section, we explain the feature embedding process [9] that is extended for remote sensing image classification followed by a brief explanation of the 3D CNN used for training the feature embedding and classification accordingly.

2.1. Construction of Triplet Constraint

In this method, we intend to construct an embedding $f(s)$ from an image sample s into a feature space in a way to measure the squared distance between all samples such that the distance between samples belonging to the same class is small and distance between samples of difference classes is large. We design a loss function with an end-to-end learning of the whole classification system. The motivation is that the loss encourages the samples of the same class to be projected onto a single point in the embedded space. Moreover, the loss also tries to enforce a margin between each pair of samples from the same class to all other samples. In this way, a manifold is formed containing the samples for one class and at the same time, enforce the discrimination to other classes.

2.1.1. Triplet Loss

A key concept in our model is the triplet [9]. In each triplet, two samples belong to the same class (positive samples) and one sample belongs to a different class (negative sample). One of the positive samples is termed as "anchor" sample to which the distance will be compared. In the embedding process, an image sample s is embedded into a d-dimensional Euclidean space. Given an image sample s_i^a (anchor), $i \in 1, \cdots, M$ where i is the index of the triplet and M is the number of all possible triplets of samples, we enforce a relationship so it is closer to all other samples s_i^p (positive) of the same class than it is to any other sample s_i^n (negative) of a different class along the spectral channel $\lambda \in B$, where B is the number of bands in the spectral channel. Formally, this relationship on a triplet $f(s_i^a)_\lambda, f(s_i^p)_\lambda, f(s_i^n)_\lambda \in T$ is defined as:

$$||f(s_i^a)_\lambda - f(s_i^p)_\lambda||_2^2 + \alpha < ||f(s_i^a)_\lambda - f(s_i^n)_\lambda||_2^2 \quad (1)$$

where α is a margin that is enforced between the positive and negative pair of samples and T is the set of all possible triplets of samples in the training set. Therefore, the triplet loss to be minimized is calculated as:

$$L_t = \sum_i^M (||f(s_i^a)_\lambda - f(s_i^p)_\lambda||_2^2 - ||f(s_i^a)_\lambda - f(s_i^n)_\lambda||_2^2 + \alpha) \quad (2)$$

In this way, many triplets will be generated which may fulfil the constraint in Eq. (1). However, using all possible triples will cause slower convergence and not every triplet may contribute to classification. Hence, it is important to select effective triplets that can improve the classification performance.

2.1.2. Selecting Triplets

In this paper, we use small mini-batches to supply the samples which should be a meaningful representation of the

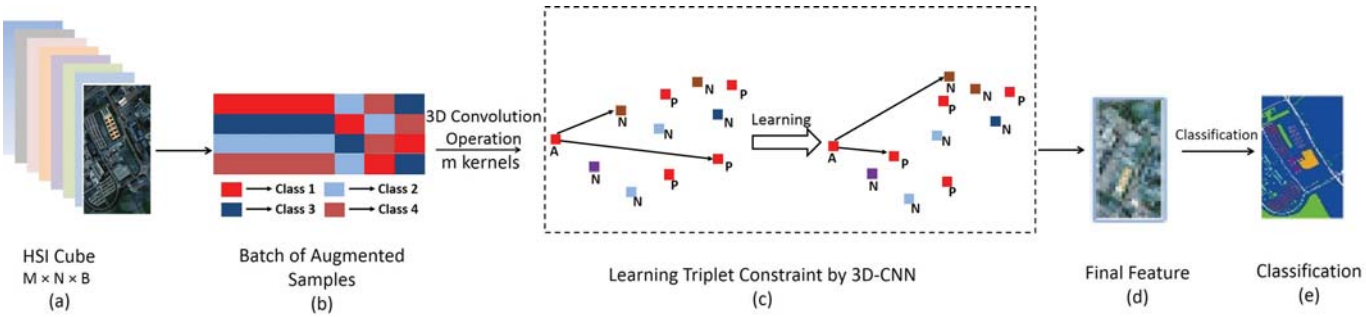

Fig. 1. Proposed Architecture. (a) The original Hyperspectral cube with B bands. (b) Batch of augmented samples: each mini-batch contains positive samples from one class and negative samples from different classes. (c) 3D-CNN learns feature embedding with triplet loss. (d) Resulting final spectral-spatial features produced by 3D-CNN. (e) Final classification map.

positive anchor distances. In this regard, we select a minimal number of samples of any one class in each mini-batch with randomly sampled negative samples added accordingly. We select the positive samples such that satisfy $\mathbf{argmax}_{s_i^p}||f(s_i^a)_\lambda - f(s_i^p)_\lambda||_2^2$ and the negative samples with $\mathbf{argmin}_{s_i^n}||f(s_i^a)_\lambda - f(s_i^n)_\lambda||_2^2$.

However, in practice, it is not computationally feasible to obtain argmax and argmin across the entire training set. Therefore, we divide our training set into several subsets and save our 3D CNN network in every n step during the training of those subsets to mark network checkpoints. Then, we generate triplets in every n step using the most recent network checkpoint and compute the armax and argmin on the subset accordingly. In case of selecting the positive samples, we consider all anchor-positive pairs in a mini-batch instead of selective anchor-positive pairs by utilizing such local structure information of the data in order to generalize well. The sample generation can be further improved by training data augmentation whose details and advantages will be described in Section 3.

In case of selecting negative samples, we consider the samples which are spectrally close to the positive samples since it will introduce challenges to the training process. The local minima caused during the training of the model while selecting the hardest negative samples can result in a collapsed model $(f(s) = 0)$ [9]. Therefore, we select the the negative samples as follows:

$$||f(s_i^a)_\lambda - f(s_i^p)_\lambda||_2^2 < ||f(s_i^a)_\lambda - f(s_i^n)_\lambda||_2^2 \qquad (3)$$

In this way, we consider the negative samples which are further away from the anchor than the positive samples but the squared distance is still close to the anchor-positive distance. Since these negative samples lie within α, we are considering samples whose spectral properties are close to the positive samples.

2.2. 3D-CNN for Feature Representation & Classification

In this method, we employ 3D-CNN to learn the feature embedding based on the triplet constraint and generate effective spectral-spatial structure representation of the data. Repeated convolution operations by 3D kernels on the spectral channel produce multiple feature maps along the spectral dimension. In this process, the number of feature maps in the previous layer will be multiplied by the number of kernels in the current layer which will produce as many feature maps as the output of the l-th convolution layer. Therefore, 3D convolution can preserve the spectral information of the input data.

During the 3D CNN training, all the connections/weights are being updated by using Stochastic Gradient Descent (SGD). We randomly initialize the model parameters and the triplet loss L_t is accordingly used to iteratively update the weights during the SGD iterations. The feature embeddings are learned by the end of the training and can be subsequently mapped to classes by adding an additional fully connected layer with logistic regression (LR) as a classifier on top of the network to generate the required classification results. By using soft-max, LR triggers the output units to 1 in order to represent the results as a set of conditional probabilities. For the given input *X*, the probability that the input belongs to class *c* over a total of *K* classes is estimated as:

$$P(y_c|X, W) = \frac{e^{X_c W_i}}{\sum_K e^{X_K W_K}} \qquad (4)$$

where W are the weights of the LR layer. We then calculate the classification loss (L_C) by computing the cross-entropy as

$$L_C = -\sum_{c=1}^{K} y_{o,c} log(P_{o,c}) \qquad (5)$$

where o is the computed probability by Eq.4. This loss is added to the triplet loss that we calculated earlier in Eq. (2). This mapping to classes is important to evaluate the performance of our 3D CNN on a testing set. Therefore, the total loss that is being minimized is defined as $L = L_t + L_C$.

3. EXPERIMENTS

In this section, we present the experimental results on real-world hyperspectral remote sensing images. Then we analyse the performance of the proposed method in comparison with several alternatives.

3.1. Dataset

For better evaluation of our proposed method, we collected AVIRIS images from the USGS database[1] from different location in the region of north America. The geographical locations had considerable impact on the entire dataset as the atmospheric effects varied significantly which produces different surface features and hence introduced more challenges in classification. We used a total of 250 images from which we generated training and testing samples. The spatial resolutions of the 145×145 images range from 2.4 to 18 meters per pixels. To fit into our proposed supervised training framework, we performed a pixelwise manual labeling on the images and created a training set containing six classes, including road, water, building, grass, tree and soil.

3.2. Addition of Virtual Samples

In this paper, we can simulate the variance in the spectral responses of remote sensing scenes in order to generate a virtual sample by multiplying the original data responses from two real samples with a random factor and then adding a Gaussian noise. The new virtual sample is assigned with the same class label as the real samples since the hyperspectral characteristics of the new fused virtual sample shall be sitting between the real samples which belong to the same class. We spectrally divide the original image into several images consisting of smaller number of spectral bands and generate virtual samples within those spectral groups. Hence, they give us multiple spectral information of the same sample from different wavelengths which produces different yet useful estimation of spectral-spatial characteristics of the new samples. We further augment the training samples by transforming each sample pixel and its 5×5 neighbours using rotation ($90°$, $180°$, $270°$) and flipping operations within each band group to produce additional images. From these transformed images, we select limited number of samples for training.

3.3. Design of the CNNs

For the CNN used in our method, we used $9 \times 9 \times 9$ convolution kernels. We adopted five convolution layers and three pooling layers with 2×2 pooling kernel in each layer. ReLU layers were used as well and cut off the features that were less than 0. For the logistic regression, the learning rate was set to 0.05 and the number of epochs was 500. The distance margin α was set to 0.2.

[1] https://earthexplorer.usgs.gov/

3.4. Results and Comparison

We used triplet constraint as an important feature embedding to be learned during the training of 3D CNN for classification. The main purpose of using this constraint is to improve the classification performance of 3D CNN instead of adopting additional post-processing stages. Therefore, we evaluated the effectiveness of our method with two recent 3D CNN based HSI classification methods. The first baseline method 3D-CNN [10] employed a standard 3D CNN and the second baseline method 3D-CNN-LR [5] used L2 regularization and dropout in the training process to improve the classification results. We also compared our method with "MTMF" [2], an SVM classifier with a Gaussian kernel. To make fair comparisons, we randomly selected 15% samples for training and the rest of the available samples for testing.

Table 1. Comparison of performance from different methods

Class	3D-CNN [10]	3D-CNN-LR [5]	MTMF [2]	Proposed Method
Road	73.70	78.64	63.40	82.15
Water	87.71	90.15	80.45	94.48
Building	81.79	84.06	70.65	88.47
Grass	84.15	87.11	73.36	90.54
Tree	83.49	85.50	73.90	88.15
Soil	80.05	81.05	67.66	85.91
OA(%)	79.95	82.48	69.35	**86.16**
AA(%)	81.82	84.42	71.57	**88.28**

In Table 1, we see that our proposed method achieves better classification accuracy compared to other methods. We can draw two significant conclusions from these results. Firstly, the use of "triplet constraint" provide useful feature embedding for the classifier. Secondly, 3D-CNN is able to learn effective spectral-spatial features as it produces significantly better classification than shallow models like SVM.

As mentioned in section 2.1.2, we consider all anchor-positive pairs during training. To validate this option, we also compared with hard anchor-positive pairs in which case we selected positive samples that are relatively close to the anchor samples than other positive samples. Fig. 3.4 illustrates the comparison which shows that the training losses for both settings are converging with the increasing iteration. But the testing loss of hard anchor-positive pair setting keeps increasing after a pivoting point at 2000 iterations while that of all anchor-positive pair setting keeps decreasing. Therefore, it can be deduced that selecting hard samples does not cover all kinds of data distributions and on the other hand, all anchor-positive samples avoid this problem by resulting in a more generalized outcome over the testing set.

Fig. 3 illustrates the results on our dataset generated by the proposed method. The first column is the ground truth. The second column is the classification map generated by the 3D-CNN after including triplet constraint loss. The third column

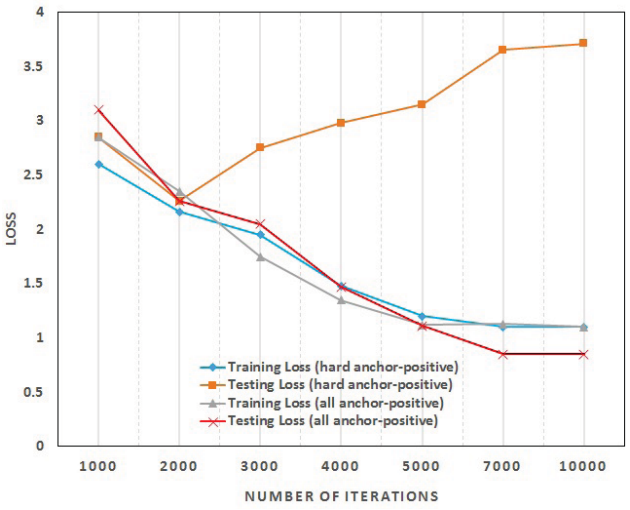

Fig. 2. Comparison of training and test losses between all anchor-positive and hard anchor-positive pairs.

is a binary map that shows the effect of corresponding misclassification obtained by comparing with the ground truth. The white pixels indicate the misclassified pixels.

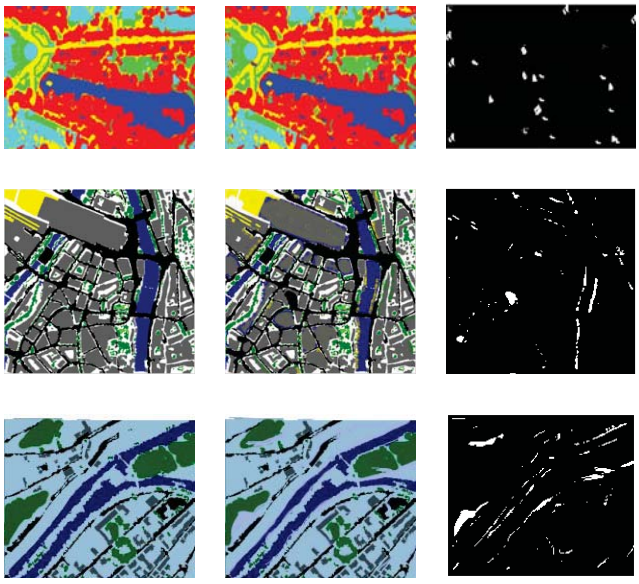

Fig. 3. (a) Ground truth; (b) output of the our method; (c) difference map against ground truth.

4. CONCLUSION

We presented a 3D-CNN-based hyperspectral image classification method. We adopted the triplet constraint property and extended it to build a useful feature embedding for remote sensing images and further used it for classification task. We also significantly increased training samples from smaller spectral groups and effectively used those in constructing triplet samples. Comparison with several state-of-art methods shows the potential of using triplet constraint in deep learning based classification framework.

5. REFERENCES

[1] L. Ballanti, L. Blesius, E. Hines, and B. Kruse, "Tree species classification using hyperspectral imagery: A comparison of two classifiers," *Remote Sensing*, vol. 8, pp. 445, 2016.

[2] I. Dpido and A. Plaza, "Unmixing prior to supervised classification of urban hyperspectral images," in *2011 Joint Urban Remote Sensing Event*, 2011, pp. 97–100.

[3] J. Ham, Yangchi Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.

[4] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 10, pp. 1537–1541, 2016.

[5] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "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, 2016.

[6] Q. Lu, Y. Ma, and G. Xia, "Active learning for training sample selection in remote sensing image classification using spatial information," *Remote Sensing Letters*, vol. 8, no. 12, pp. 1210–1219, 2017.

[7] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88 – 98, 2017.

[8] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844–853, 2017.

[9] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[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, 2017.