IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DEEP LEARNING ALGORITHM BASED HYPERSPECTRAL IMAGE CLASSIFICATION

[1]C.Gomathi, [2] P.J.Mercy MCA.,M.Phil.

ABSTRACT

The traditional unsupervised loss function [e.g., mean square error (MSE)] calculates the distance between the predicted value and the original input. However, it is difficult to guarantee the effectiveness of the features only by optimizing the reconstruction error. In order to make the learned features more effective for classification tasks, we optimize the contrastive loss function to make the features from different views of the same sample consistent. This makes the features of the same class aggregate with each other, and the features of different classes are far away from each other. Therefore, the features obtained by optimizing the contrastive loss function of different views could effectively improve the classification accuracy. We use a deep CNN as the base feature extractor. We call this proposed method deep multiview learning.

INTRODUCTION

Hyperspectral image classification is the task of classifying a class label to every pixel in an image that was captured using (hyper)spectral sensors. Image result for hyperspectral image classification

Hyperspectral imaging, like other spectral imaging, collects and processes information from across the electromagnetic spectrum. The goal of hyperspectral imaging is to obtain the spectrum for each pixel in the image of a scene, with the purpose of finding objects, identifying materials, or detecting processes. Hyperspectral image (HSI) classification is a phenomenal mechanism to analyze diversified land cover in remotely sensed hyperspectral images.

The technological progression in optical sensors over the last few decades provides enormous amount of information in terms of attaining requisite spatial, spectral and temporal resolutions. Especially, the generous spectral information comprises of hyperspectral images (HSIs) establishes new application domains and poses new technological challenges in data analysis [1]. With the available high spectral

resolution, subtle objects and materials can be extracted by hyperspectral imaging sensors with very narrow diagnostic spectral bands for the variety of purposes such as detection, urban planning [2], agriculture [3], identification, surveillance [4], and quantification [5, 6]. HSIs allow the characterization of objects of interest (e.g., land cover classes) with unprecedented accuracy, and keep inventories up to date. Improvements in spectral resolution have called for advances in signal processing and exploitation algorithms.

Hyperspectral image is a 3D data cube, which contains two-dimensional spatial information (image feature) and one-dimensional spectral information (spectral-bands). Especially, the spectral bands occupy very fine wavelengths, while the image features such as Land cover features and shape features disclose the disparity and association among adjacent pixels from different directions at a confident wavelength.

LITERATURE SURVEY

In [1], the authors have described the fundamental hurdles of HSI classification that classical machine learning approaches cannot effectively address, as well as the benefits of using deep learning to address these issues. Then, to systematically examine recent achievements in deep learning-based HSI classification, we construct a framework that classifies corresponding works into spectral-feature networks, spatial-feature networks, and spectral-spatial-feature networks. Furthermore, because available training data in the remote sensing sector are typically scarce, and deep network training necessitates a high number of samples, we present several ways for improving classification performance, which might serve as guidance for future research on this subject.

To improve the small sample classification performance of hyperspectral images, a simple but creative classification paradigm based on the morphological attribute profile cube is proposed in [2]. To begin, multiple morphological filters are applied to the hyperspectral image to create morphological attribute profiles. After that, sample features such as morphological attribute profile cubes are extracted. Second, to make full use of the rich spatial-spectral information, the resulting morphological attribute profile cubes are scanned with various scale sliding windows. To finish the classification task, the features from the multi-grained scanning are fed into a deep forest classifier. In this way, the suggested method might improve classification accuracy by utilising a deep network topology.

In [3], the authors have examined the strengths and shortcomings of the most extensively used classifiers in the literature to provide a comprehensive overview of the present state-of-the-art in DL for HSI classification. The authors have present quantifiable findings for each mentioned method using many well-known and extensively utilised HSI scenarios, allowing for a thorough comparison of the methodologies. The research ends with some observations and suggestions for future problems in using DL approaches to HSI classification.

In [4], that research suggested a multi-scale dense network (MSDN) for HSI classification that fully utilised varied scale information in the network topology and aggregated scale information across the network. It extracted HSI features in two dimensions, including fine and coarse features. The deep extraction of HSI features was evaluated in the horizontal direction, and the 3-D dense connection structure was employed to aggregate features at different levels. Scale information was examined in the vertical direction, and three-scale feature maps at low, middle, and high levels were constructed using the first layer of the network. For downsampling, the MSDN employed stride convolution and incorporated feature information at various scale levels. Along the diagonal line, the MSDN extracted characteristics. For HSI classification, the network used deep feature extraction reconstruction and multi-scale fusion. On sample HSI datasets, including as the Indian Pines, Pavia University, Salinas, Botswana, and Kennedy Space Center datasets, the MSDN model performed well. It increased the HSI classification training speed and accuracy, as well as the convergence speed, which successfully conserved computer resources and had excellent stability.

In [5], the authors have built on our earlier work by applying the maximum correntropy criterion to the noise and outliers problem, resulting in a more resilient and better generalisation model. As a result, even if some samples are slightly distorted, discriminative characteristics can be extracted. A novel dual-channel architecture based on robust CapsNet is also developed for fusing hyperspectral data with light detection and ranging-derived elevation data for classification.

In [6], the authors have examined unsupervised feature extraction on hyperspectral imagery (HSI) and suggest a unique approach for extracting spectral-spatial features from HSI using autoencoder (AE) networks. Our technique considers data relations, such as input dependency with nearby inputs, which is commonly overlooked by traditional AE-based feature extractors. The loss function of the normal AE is changed in such a way that pixels share common features with nearby pixels. The procedure allows for the creation of smooth compressed images that are represented by the AE's features. For land cover classification, numerical experiments were done using real-world HSI data sets. The findings showed that spectral-spatial characteristics recovered using our method are more discriminative for land cover classification than those extracted using traditional methods.

Using a 3-Dimensional (3D) convolutional autoencoder, an unsupervised spatial-spectral feature learning technique for hyperspectral pictures is proposed in [7] (3D-CAE). To maximise the exploration of spatial-spectral structural information for feature extraction, the proposed 3D-CAE uses only 3D or elementwise operations, such as 3D convolution, 3D pooling, and 3D batch normalisation. A 3D convolutional decoder network is also being developed to reconstruct the input patterns to the proposed 3D-CAE, allowing all of the network's parameters to be taught without using labelled training samples. As a result, effective features can be trained in an unsupervised mode, without the need for pixel label information.

In [8], the proposed generator and discriminator are built on a fully deconvolutional subnetwork and a fully convolutional subnetwork, respectively, to learn upsampling and downsampling techniques adaptively during FE. Furthermore, by exploiting the zero-sum game relationship between the generator and discriminator, an unique min-max cost function is created for training the proposed GAN in an end-to-end fashion without supervision. In addition, the suggested modified GAN substitutes the original Jensen-Shannon divergence with the Wasserstein distance, intending to reduce the instability and difficulty of GAN framework training. The proposed method's usefulness is confirmed by findings from three real-world data sets.

In [9], the authors have allowed for the rapid construction of a large, consistent vocabulary, which aids contrastive unsupervised learning. MoCo's ImageNet classification scores are competitive when using the typical linear procedure. More crucially, MoCo's representations are well-suited to subsequent activities. On PASCAL VOC, COCO, and other datasets, MoCo can outperform its supervised pre-training equivalent in seven detection/segmentation tasks, sometimes by a wide margin. This shows that in many visual tasks, the gap between unsupervised and supervised representation learning has narrowed.

The short sample size challenge of HSI classification is addressed in this research by proposing a deep few-shot learning algorithm in [10]. The suggested method includes three innovative tactics. A deep residual 3-D convolutional neural network is used to extract spectral–spatial information to reduce labelling uncertainty. Second, episodes educate the network to learn a metric space in which samples from the same class are close together and samples from other classes are separated. Finally, in the learned metric space, the testing samples are classified by a closest neighbour classifier.

PROPOSED METHODOLOGY

The paramount challenge for HSI classification is the curse of dimensionality which is also termed as Hughes phenomenon. To confront with this difficulty, feature extraction methods are used to reduce the dimensionality by selecting the prominent features. In unsupervised methods, the algorithm or method automatically groups pixels with similar spectral characteristics (means, standard deviations, etc.) into unique clusters according to some statistically determined criteria. Further, unsupervised classification methods do not require any prior knowledge to train the data. The familiar unsupervised methods are principal component analysis (PCA) and independent component analysis (ICA).

A. Principal component analysis

It is the most widely used technique for dimensionality reduction. In comparative sense, appreciable reduction in the number of variables is possible while retaining most of the information contained by the original dataset. The substantial correlation between the hyperspectral bands is the basis for PCA. The analysis attempts to eliminate the correlation between the bands and further determines the optimum linear combination of the original bands accounting for the variation of pixel values in an image .

The mathematical principle of PCA relies upon the eigen value decomposition of covariance matrix of HSI bands. The pixels of hyperspectral data are arranged as a vector having its size same as the number of bands. X i = x 1 x 2 . . … x N T , where N is the number of HS bands. The mean of all the pixel vectors is calculated as:

$$m = 1\ M \sum i = 1\ M\ x\ 1\ x\ 2 \ldots x\ N\ i\ T$$

where M = p ⋆ q is the number of pixel vectors for a HS image of "p" rows and "q" columns. The covariance matrix is determined as:

$$C = 1\ M \sum i = 1\ M\ X\ i - m\ X\ i - m\ T$$

The covariance matrix can also be written as:

$$C = ADA\ T$$

D is the diagonal matrix composed of eigen values λ 1 . … λ N of C and A is the orthogonal matrix with the corresponding eigen vectors (each of size N) as columns. The linear transformation y i = A T X i , i = 1 , 2 . … M , is adapted to achieve the modified pixel vectors which are the PCA transformed bands of original images. The first K rows of the matrix A T are selected such that, the rows are the eigen vectors corresponding to the eigen values arranged in a descending order. The selected K rows are multiplied with the pixel vector X i to yield the PCA bands composed of most of the information contained in the HS bands.

In hypespectral data, most of the elements are covered by the sensors with high spectral resolution which cannot be well described by the second order characteristics. Hence, PCA is not an effective tool for HS image classification since it deals with only second-order statistics.

DEEP MULTIVIEW LEARNING FOR HSI CLASSIFICATION

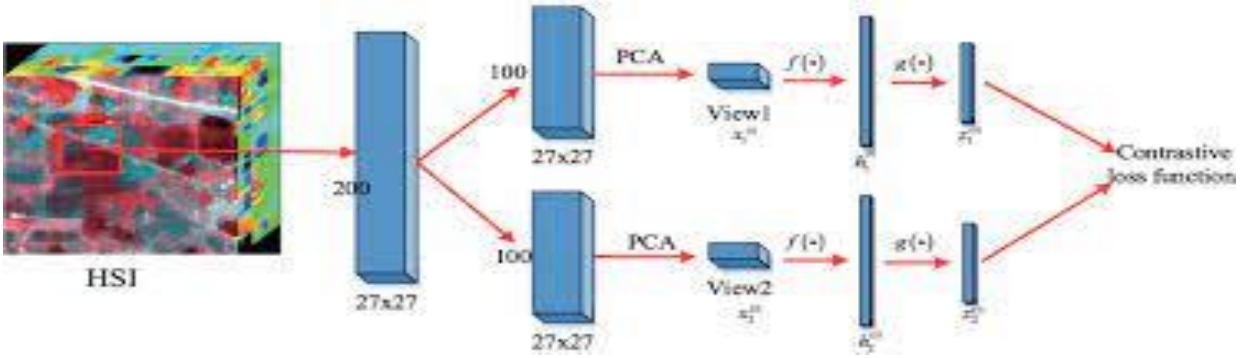

Fig. 1. Pipeline of the deep multiview learning for HIS

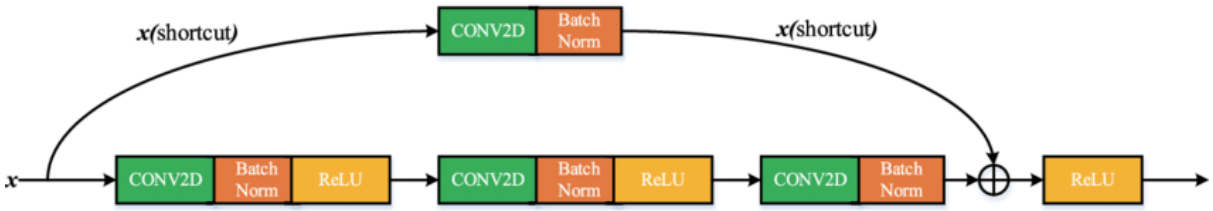

Fig. 2. Illustration of a standard residual block. CONV2D denotes a convolutional layer, BatchNorm denotes a batch normalization layer, and ReLU denotes a ReLU layer.high-level feature vector. A multilayer perceptron with two fully connected layers g(·) is used to transform the latent features h(i) 1 and h(i) 2 into z (i) 1 and z (i) 2 . Then, we define the contrastive loss on z (i) 1 and z (i) 2 rather than h(i) 1 and h(i) 2 . In the training procedure, a minibatch of N samples is randomly selected as the training samples of one parameter update. The contrastive prediction task is defined on pairs of samples derived from the minibatch, resulting in 2N views. In 2N views, two views from the same sample are taken as a positive pair, and two views from different samples are taken as negative pairs. The contrastive loss is defined denotes the cosine similarity between two vectors zi and z j . In a minibatch, the total loss is computed across all positive pairs. Our goal is to learn representations that capture information shared between multiple sensory views without human supervision. This loss is defined according to the similarity between views, which means that it does not need any human supervision information. In other words, it is an unsupervised method. More importantly, this loss ensures that the network learns to extract view-invariant features, which is a useful representation of the samples. When the number of views is more than 3, the features of different views are combined in pairs. The contrastive loss is calculated respectively for the features of two combined views. Then, the sum of the contrastive loss calculated from different combined views is calculated as the final loss function.

B. Deep Residual Network

The f (·) that extracts representation vectors from views could be various network architecture. Recent studies [41], [42] reveal that the classification performance benefits from bigger models. Residual learning has become a common method to improve the accuracy in natural image recognition and HSI classification [43], [44]. Therefore, a variant of Resnet50 [41] is used as the network that extracts representation vectors from views. Deep residual learning could make training deep network easier. Thus, it has been widely used in a variety of classification tasks. As shown in Fig. 2, the core idea of deep residual learning is to introduce a shortcut connection, which directly skips one or more layers. The deep residual network is based on residual block. As shown in Fig. 2, there are three convolutional layers in a standard residual block. Each convolution layer is followed by a batch normalization layer (BatchNorm) and a ReLU layer. The original input is then added with the output of the last convolutional layer as the output of a residual block, which is a shortcut operation. The output of a residual block is activated by a ReLU layer as the input of the later residual block. Note that the dimension of the input data may be different from the output dimension of the last convolutional layer of the residual block. When the dimension of the input and the last convolutional layer of the residual block is different, a 1 × 1

convolutional layer followed by a batch normalization layer is applied to the input to conduct a resample operation in order to ensure consistent data dimensions. The original Resnet50 consists of one convolutional layer, 16 residual blocks, two pooling layers, and one classification

TABLE I

DETAILS OF DEEP RESIDUAL NETWORK USED AS THE BASE FEATURE EXTRACTOR

	Stage 1	**MaxPool**	**Stage 2**	**Stage 3**	**Stage 4**	**Stage 5**	**AvgPool**
Parameters	**3*3,64**	**2*2 Max pooling**	**[1*1,64; 3*3,64; 1*1,256] * 3**	**[1*1,128; 3*3,128; 1*1,512] *4**	**[1*1,256; 3*3,256; 1*1,102] *6**	**[1*1,512; 3*3,512; 1*1,2048] *3**	**Global average pooling**

Algorithm 1 Minibatch Training Procedure

Require: Batch size N, deep residual network f (·), fully connected network g(·), data augmentation operation τ

for sampled minibatch {x}N i=1 **do**

for i in {1,..., N} **do**

Draw two augmentation operations τ1 ∼ τ , τ2 ∼ τ

The first view

x2k−1 = τ1(**x**(i) 1)

h2k−1 = f (**x**2k−1)

z2k−1 = g(**h**2k−1))

The second view

x2k = τ1(**x**(i) 2)

h2k = f (**x**2k)

z2k = g(**h**2k))

end for

for i in {1,..., 2N}, j in {1,..., 2N} **do**

si,j = z i z j ||zi || ||z j ||

end for

calculate the loss $\zeta_{i,j} = -\log \exp(s_{i,j}) \, 2N \, k=1 \, 1[k=i] \exp(s_{i,k})$

calculate the total loss $\zeta = 1 \, 2N \sum 2N \, i=1 \; \zeta_{2i-1,2i} + \zeta_{2i,2i-1}$

update networks $f(\cdot)$ and $g(\cdot)$ to minimize ζ

end for

layer (fully connected layer). The purpose of the training network is not to classify but to learn the representations of views. Therefore, the Resnet50 without classification layer is used as the base feature extractor $f(\cdot)$. As shown in Fig. 3, the Resnet50 $f(\cdot)$ actually consists of 49 convolutional layers, $1 + 3 \times (3 + 4 + 6 + 3) = 49$. The details of the deep residual network used as the base feature extractor $f(\cdot)$ are shown in Table I. Note that the output of the deep residual network is a 2048 vector. Subsequently, a multilayer perceptron with two fully connected layers $g(\cdot)$ is applied to the output vector of the Resnet50 $f(\cdot)$ to reduce the dimensions of output features. In fact, the network used to extract features includes 49 convolutional layers and two fully connected layers.

C. Training and Testing Procedure

The contrastive loss is defined on the outputs of the multilayer perceptron. More specifically, the pseudocode for a training minibatch procedure is given in Algorithm 1. Data augmentation is a common technique that can effectively improve the generalization ability of a model and has been widely used in supervised deep learning. However, data augmentation has not been used in the contrastive prediction task. Consequently, two data augmentations (random cropping and random Gaussian blur) are used to improve the robustness of network training. In the testing procedure, the deep residual network trained on a specific HSI is used as a feature extractor. Then, all samples of this specific HSI pass through the deep residual network to output the corresponding feature vectors. So far, conventional machine learning methods could be applied to the extracted features to complete the classification task. Here, an SVM classifier and an RF classifier are used.

EXPERIMENTAL RESULTS AND ANALYSIS

The proposed method is implemented by the PyTorch library. The results are generated on a PC equipped with an Intel Core i7-9750H with 2.6 GHz and an Nvidia GeForce RTX 2070M. The PC's memory is 16G.

A. Data Sets

To demonstrate the effectiveness of the proposed method, the University of Pavia data set, the Indiana Pines data set, the Salinas data set, and the Houston data set are used to conduct classification experiments. In the feature learning procedure, 50% unlabeled samples are used as the training data, and the remaining 50% samples are used as the testing data. In each data set, five labeled samples per class are randomly selected as the training samples for the supervised classifier in the classification procedure.

The University of Pavia data set is acquired by the ROSIS sensor during a flight campaign over Pavia, Northern Italy. It has 103 spectral bands coverage from 0.43 to 0.86 μm and a geometric resolution of 1.3 m. The image size is 610 × 340 pixels. In this data set, 42 776 pixels with nine classes are labeled. Labels, the number of labeled training samples, and the number of testing samples are listed in Table II. The second data set is the Indiana Pines data set. This data set is gathered by Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines test site in Northwestern Indiana and consists of 145 × 145 pixels and 224 spectral reflectance bands in the wavelength range 0.4–2.5 μm; 24 bands covering the region of water absorption are removed, resulting in 200 bands for classification. This scene contains two-third agriculture and one-third forest or other natural perennial vegetation; 10 249 pixels with 16 classes are labeled. Labels, the number of labeled training samples, and the number of testing samples are listed in Table III.

TABLE II

Labels, The Number Of Labeled Training Samples, And The Number Of Testing Samples For The University Of Pavia Data Set

No	Class	Training	Testing
1	Asphalt	5	6631
2	Meadows	5	18649
3	Gravel	5	2099
4	Trees	5	3064
5	Sheets	5	1345
6	Bare Soil	5	5029
7	Bitumen	5	1330
8	Bricks	5	3682
9	Shadows	5	947
Total		45	42776

CONCLUSION

Deep learning-based approaches for HSI categorization have recently received a lot of attention. A deep-learning classifier, on the other hand, is known for requiring hundreds or thousands of labelled examples to train. As a result, for researchers, training models to learn usable representations of HSIs in an unsupervised manner is the Holy Grail. We suggested the deep multiview learning method for HSI classification in this study. The proposed method could greatly improve classification accuracy by training the network to learn view-invariant features, especially in the case of small samples. Furthermore, in the HSI field, we first investigate the use of a deep residual network with 51 layers. Experiments show that employing larger models is necessary. The improvement of classification accuracy is based on the idea of sacrificing training time, despite the fact that this strategy has achieved great classification performance. The proposed method has a disadvantage in that the training procedure takes a long time. We only built two views in order to make the process easier. We will create more views in the future to boost categorization

performance even further. Finally, the suggested technique is simple to integrate with currently available supervised classifiers. We only put the SVM and RF classifiers to the test. We plan to test more classifiers in the future.

REFERENCES

1. S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi and J. A. Benediktsson, "Deep Learning for Hyperspectral Image Classification: An Overview," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 9, pp. 6690-6709, Sept. 2019, doi: 10.1109/TGRS.2019.2907932.

2. B. Liu et al., "Morphological Attribute Profile Cube and Deep Random Forest for Small Sample Classification of Hyperspectral Image," in IEEE Access, vol. 8, pp. 117096-117108, 2020, doi: 10.1109/ACCESS.2020.3004968.

3. M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," ISPRS J. Photogramm. Remote Sens., vol. 158, pp. 279–317, Dec. 2019.

4. C. Zhang, G. Li and S. Du, "Multi-Scale Dense Networks for Hyperspectral Remote Sensing Image Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 11, pp. 9201-9222, Nov. 2019, doi: 10.1109/TGRS.2019.2925615.

5. H. -C. Li, W. -Y. Wang, L. Pan, W. Li, Q. Du and R. Tao, "Robust Capsule Network Based on Maximum Correntropy Criterion for Hyperspectral Image Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 738-751, 2020, doi: 10.1109/JSTARS.2020.2968930.

6. S. Koda, F. Melgani and R. Nishii, "Unsupervised Spectral–Spatial Feature Extraction With Generalized Autoencoder for Hyperspectral Imagery," in IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 3, pp. 469-473, March 2020, doi: 10.1109/LGRS.2019.2921225.

7. S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li and Q. Du, "Unsupervised Spatial–Spectral Feature Learning by 3D Convolutional Autoencoder for Hyperspectral Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 9, pp. 6808-6820, Sept. 2019, doi: 10.1109/TGRS.2019.2908756.

8. M. Zhang, M. Gong, Y. Mao, J. Li and Y. Wu, "Unsupervised Feature Extraction in Hyperspectral Images Based on Wasserstein Generative Adversarial Network," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 5, pp. 2669-2688, May 2019, doi: 10.1109/TGRS.2018.2876123.

9. K. He, H. Fan, Y. Wu, S. Xie and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9726-9735, doi: 10.1109/CVPR42600.2020.00975.

10. B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan and R. Wang, "Deep Few-Shot Learning for Hyperspectral Image Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 4, pp. 2290-2304, April 2019, doi: 10.1109/TGRS.2018.2872830.