1D-Convolutional Capsule Network for Hyperspectral Image Classification

Haitao Zhang, Lingguo Meng, Xian Wei, Member, IEEE, Xiaoliang Tang, Member, IEEE, Xuan Tang, Member, IEEE, Xingping Wang, Bo Jin, Member, IEEE, and Wei Yao, Member, IEEE

Abstract—Recently, convolutional neural networks (CNNs) have achieved excellent performances in many computer vision tasks. Specifically, for hyperspectral images (HSIs) classification, CNNs often require very complex structure due to the high dimension of HSIs. The complex structure of CNNs results in prohibitive training efforts. Moreover, the common situation in HSIs classification task is the lack of labeled samples, which results in accuracy deterioration of CNNs. In this work, we develop an easy-to-implement capsule network to alleviate the aforementioned problems, i.e., 1D-convolution capsule network (1D-ConvCapsNet). Firstly, 1D-ConvCapsNet separately extracts spatial and spectral information on spatial and spectral domains, which is more lightweight than 3D-convolution due to fewer parameters. Secondly, 1D-ConvCapsNet utilizes the capsule-wise constraint window method to reduce parameter amount and computational complexity of conventional capsule network. Finally, 1D-ConvCapsNet obtains accurate predictions with respect to input samples via dynamic routing. The effectiveness of the 1D-ConvCapsNet is verified by three representative HSI datasets. Experimental results demonstrate that 1D-ConvCapsNet is superior to state-of-the-art methods in both the accuracy and training effort.

Index Terms—Hyperspectral image, deep learning, convolutional neural network, capsule network, spectral-spatial information, PCA whitening, classification.

I. INTRODUCTION

Unlike the colorful image only responding to visible light, HSI contains hundreds of spectral channels (SCs), each of which is an image of the target in a very narrow segment of the electromagnetic spectrum. HSI describes object's surface with abundant spectral-spatial information [1]–[4]. Thus, HSI has become increasingly popular in remote sensing fields, such as ecological science, precision agriculture, and mineral exploration, etc. [2], [5]–[9]. The classification of HSIs is the key to realize the applications above.

In spite of lots of research efforts have been attracted on the aforementioned fields, certain essential characteristics of HSIs make it very challenging for the classification task. Basically, the main challenging characteristics can be summarized as follows.

1) HSI usually consists of several hundreds of SCs, and every pixel has high dimensionality. Hence, HSI has high structural complexity due to the so-called curse of dimensionality (Hughes phenomenon [1]).
2) The quality of HSI is affected by multi-factors, such as weather and illumination. These factors often generate the confusion of HSI, which result in the phenomenon that the same spectrum expresses different objects, or same objects have different spectrum.
3) The number of labeled training samples are often limited because labeling data is expensive and time consuming. Consequently, finite training samples cannot generalize the whole of ground-truth, resulting in degradation of common classifiers.

Researchers are motivated by these challenges to develop more effective methods. Previously, support vector machine (SVM), sparse representation classifier, k-means clustering were employed to classify the HSIs by only using multi-bands spectral information [10]–[12]. In order to capture more useful information, recently, deep learning (DL)-based [13] methods exhibited evident advantage on HSIs classification because of their capacity of feature extraction from low-level to high-level. The method of stacked autoencoder (SAE) for HSIs was proposed in [14]. Another deep learning method was also proposed using deep belief network (DBN) in [15]. These methods extract the global discriminative feature but ignoring the spatial correlation, suffering from the highly computational complexity. Deep feature extraction and classification for HSIs using CNN was introduced by [16], which achieved the state-of-the-art performance due to its ability to extract local spatial relationship.

Generally, CNNs extract features through flexible combination of convolution and pooling layers. This is a simplified process of the comprehension of the brain to visual stimulation from retina to cortex. The flexibility of CNN structure make it suitable for computer vision applications, such as images classification and objects detection [17], [18]. In HSIs classification, CNNs also hold outstanding performance because 3D-convolution [19] can extract spectral-spatial information effectively [16], [20]–[22]. However, this outstanding performance relies heavily on complex network structure due to the characteristics of HSI data. Conventional CNNs suffer from gradient disappearance (GD), spatial information damage and prohibitive training efforts. On the one hand, the training
Recently, DL-based methods have been widely adopted because of their competitive in terms of accuracy and training efforts. The main contributions of this paper are summarized as follows.

1. It proposes a separated spatial and spectral information extraction method, which extracts features on spatial and spectral domains, respectively.
2. It proposes a constraint window method to reduce the complexity of capsule network while holding accuracy.
3. It builds a accurate and efficient capsule network by using our proposed methods, called 1D-ConvCapsNet.

The rest of paper is organized as follows, Section II introduces related works. Section III describes classification strategy of our method. Section IV gives architecture and detailed implementation. Section V presents the results of experiments. Finally, conclusions are given in Section VI.

II. RELATED WORKS

HSIs classification is the fundamental task in remote sensing application. Although HSIs classification has been studied extensively by researchers from multiple perspectives in the past, inherent characteristics of HSIs make it a challenging task. Previously, the SVM is widely used because of its effectiveness and robustness. It projects samples into high-dimensional feature space using kernel-based method to make samples linearly separable. Specifically, [10] employed kernel trick that promoted the separation of samples in a high-dimensional feature space via a nonlinear transformation of a kernel function. With the improvement of spectral resolution, conventional approaches suffer from Hughes phenomenon [1].

In order to deal with this phenomenon, a series of dimensionality reduction (DR) and band selection (BS) approaches were developed by researchers. [27], [28] proposed image low-dimensional representation learning method for reconstructible DR under unsupervised conditions. [29] presented the locality adaptive discriminant analysis method for HSIs classification. [30] proposed the salient BS method via manifold ranking. Recently, DL-based methods have been widely adopted because they are capable of automatically learning features. Compared to conventional methods, DL-based methods have hierarchical structure, which generate high-level features from low-level features via forward propagation. Typically, several DL-based HSIs classification methods were proposed. [14] proposed a spectral-spatial joint information method by combining SAE with PCA. [31] proposed a unsupervised features extraction method which could also be used to extract spectral-spatial information. However, for the above methods, the input data has to be reshaped to vector, which result in the loss of spatial correlation. In order to exploit spatial correlation effectively, CNN has become one of the most important tool in remote sensing because CNN can naturally extract multi-dimensional features. Especially, 3D-convolution [19] can extract features simultaneously in spectral and spatial domains of HSI. Hence, several CNN-based approaches have been proposed. [32] designed a dual branches end-to-end network with skip architecture to learn spectral and spatial features, respectively. [33] proposed a joint features extraction method with two branches, which are devoted to features from the spectral domain and the spatial domain. [34] proposed semi-supervised network with skip connection between the encoder and the decoder in order to solve the problem of limited labeled samples. [35] proposed a HSIs classification method with Markov random fields and CNN from the perspective of unified Bayesian framework. [16] proposed a series of regularized deep feature extraction methods using several convolution and pooling layers, which achieved the state-of-the-art performance.

CNN has become a powerful tool for HSIs classification task. However, compared to ordinary images classification task, CNN-based methods have more complex structures due to the complexity of HSI data, which require exorbitant efforts during the training phase. Firstly, with the structure deepening, the gradient is gradually lost during the propagation process, resulting in the slow convergence rate. [36] introduced the rectified linear unit (ReLU) as the activation function in order to alleviate the GD. Based on this, [17] designed AlexNet with ReLU activation function and won the annual ImageNet competition. Secondly, CNNs often use pooling to control scale of networks, which inevitably damage the spatial information due to down sampling. To deal with this, [37] proposed a global pooling method, called dynamic k-max pooling, which keeps top-k values during the pooling operation. [38] presented the architecture with overlap pooling for HSIs classification by using different combinations of max pooling and mean pooling. Thirdly, CNN-based methods are unable to detect pose information of the objects because convolution filters can only represent the activity related to features. It means that CNNs are invariant for spatial transformation of objects. Hence, CNNs are incapable of modeling relative relationship between objects. Some data augmentation methods have been adopted in order to make CNNs more robust with respect to spatial transformation and prevent overfitting under the limited training samples [16], [39].

In order to overcome the shortcomings of traditional CNN, [40] proposed the conception of capsule, which can encode instantiation parameters of entity (an object or object part). Capsule is a collection of neurons, which describes the pose
information and existence probability of an entity. Hence, the capsule carries more information about properties of the entity than the conventional scalar neuron of CNN. In order to effectively use information stored in the capsule neurons, [23] proposed dynamic routing between capsules and designed a novel network, called CapsNet. Therein, a capsule neuron is organized into a vector, whose length and orientation respectively represent existence probability and properties of an entity. Dynamic routing is used for communication between capsules by adjusting the coupling coefficient between predictive vector and high-layer capsule. The coupling indicates that entity in the image should be paid attention to rather than directly encoding it. Therefore, the capsule-based network is more expressive and explanatory than the conventional CNN. Specifically, by taking advantages of capsules, capsule-based networks exhibit high precision, fast convergence, strong noise immunity and generalization.

However, CapsNet has massive training parameters, which result in high storage pressure and slow training speed. Parameter redundancy makes CapsNet difficult to directly work on large images. In this paper, 1D-ConvCapsNet was proposed, which is an easy-to-implement method for HSIs classification task. The details of 1D-ConvCapsNet will be described in Section IV.

III. CLASSIFICATION STRATEGY

A. CNN-based model

The CNN-based model is popular in HSIs analysis and processing because the convolution can easily extract features on multi-dimension. As shown in Fig. 1, a classic CNN model mainly contains two modules, features extraction module (FEM) and classifier module (CM). Generally, FEM uses combination of the convolution and the pooling layers to extract high-level features, and CM uses several full connection layers as a classifier. In CNN, the convolution layer is the most important component, which is related to two aspects. One aspect is the statistical properties of images [41], that means features learned at a region can be applied to others. This fact allows convolution filters to detect the same features at all position of images. Another aspect is the finding of neuroscience [42], which reveals that cells within receptive fields of vision system are sensitive to visual stimulus and have strong responses to interested features. Additionally, visual cells are mainly composed of two types, s-cells and c-cells. The s-cells have an intensively response to their preferences, which functionally correspond to the convolution layer. The c-cells are able to concentrate multiple s-cells to achieve large receptive fields and resist distortion, which functionally correspond to the pooling layer. Therefore, the classic CNN model can be regarded as the oversimplified simulations of the visual system.

Concretely, convolution filters implement the aforementioned ideas in a manner that locally connect and share parameters. Taking 3D-convolution as an example, the convolution operation of one filter at position \((x, y, z)\) in layer \(l\) can be defined as follows:

\[
\alpha_{x,y,z} = \max(0, x).
\]

Where \(\alpha_{x,y,z}\) is the activity of neuron at position \((x, y, z)\) in a feature map of layer \(l\). The \(m\) is index of the feature maps generated by the previous layer. Constant \(\alpha\), \(\beta\) and \(\gamma\) represent the spatial size of convolution filter. In HSIs classification task, \(\alpha\) and \(\beta\) correspond to the spatial domain, and \(\gamma\) corresponds to the spectral domain. The \(w_{l,m}^{i,j,k}\) is weight parameter at \((i, j, k)\) of convolution filter corresponding to the \(m\)-th feature map in layer \(l\). The \(b(l)\) is the bias parameter of convolution filter in layer \(l\). All \(w_{l,m}^{i,j,k}\) and \(b(l)\) are trained by back propagation (BP) algorithm.

Function \(\sigma(\cdot)\) is defined as the nonlinear activation function. It introduces the nonlinearity into neural network (NN) for enhancing performance. Function ReLU is widely used because of its advantages of simplicity, rapidity and avoiding GD. It is given by the following equation:

\[
\sigma(x) = \max(0,x).
\]

The pooling layer is located behind the convolution layer, which provides a larger receptive field and a degree of transform invariance through down sampling. The max-pooling operation is defined as follows:

\[
o_{x,y,z} = \max(R_{x,y,z}V)
\]

where the \(V\) is defined as the input feature maps generated by previous convolution layer, and the \(R_{x,y,z}\) is defined as a operator that extracts patch from \(V\) at position \((x, y, z)\). The \(o_{x,y,z}\) is the maximum in the patch.

B. Capsule-based model

Different from CNN-based model, the basic unit of the capsule-based network is capsule neuron, which consists of several scalars. In CapsNet, capsule neuron first describes an entity in the form of a vector, which holds the existence probability and properties of an entity. Specifically, properties are expressed as instantiation parameters, i.e., pose (position, size, orientation), deformation, texture, etc. Then, a viewpoint-invariant representation can be obtained by multiplying viewpoint matrix and capsule. This idea stems from computer graphics, which can be understood as the inverse rendering process. The rendering process is to give an abstract representation and instantiation parameters of the entity, and then get
the image by using the render function. Capsule is in an opposite way, it acquires approximately abstract representation of entity via viewpoint matrix and instantiation parameters. This process has the viewpoint invariance, meaning that whether the direction of observation changes, the abstract representation of entity can be obtained by using the same viewpoint matrix.

Dynamic routing [23] is the most important part of the capsule-based network. It determines the coupling coefficient by measuring the agreement between capsules, allowing them to be dynamically connected. This mechanism make child capsules more inclined to send messages to the parent capsule with large coupling, and child capsules also receive the feedback from parent that indicates which entity in the image should be paid attention to. Intuitively, the execution of dynamic routing is shown in Fig. 2, where child capsule is denoted by \( u^{(l)}_i \in \mathbb{R}^{d^{(l)}} \) and parent capsule is denoted by \( u^{(l+1)}_j \in \mathbb{R}^{d^{(l+1)}} \). The viewpoint-invariant representation is denoted by \( \hat{u}^{(l+1)}_j \in \mathbb{R}^{d^{(l+1)}} \), which is also called the prediction vector. Firstly, prediction vector \( \hat{u}^{(l+1)}_j \) is obtained by multiplying \( W^{(l+1)}_{k,ji} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}} \) by \( u^{(l)}_i \), and log prior \( t^{(l+1)}_{k,ji} \) is initialized to zero. Secondly, \( u^{(l+1)}_{j+1} \) is equal to weighted sum of all \( \hat{u}^{(l+1)}_{j|i} \) through the squashing nonlinear activation function. Thirdly, the log prior \( t^{(l+1)}_{k,ji} \) is updated by the accumulation of the scalar product between \( \hat{u}^{(l+1)}_{j|i} \) and \( u^{(l+1)}_j \). By iterating the second and the third steps, the coupling coefficient can be allocated to achieve dynamic connection between capsules. Similar to CNN, the viewpoint matrix \( W^{(l+1)}_{k,ji} \) can be learned by BP algorithm.

In the HSI’s classification task, this mechanism also can be used, but the key is how to express entity with capsule. In this work, we define a part of the SCs as an entity. Capsule can encode the existence probability and instantiation parameters of the entity using the length and direction of the vector, respectively. As shown in Fig. 3 each type of ground-truth has its own characteristics in the representation of capsules. Hence, these characteristics can be interpreted by dynamic routing during program execution.
Fig. 4. The architecture of our proposed method. It consist of four layers, SpatialConv layer is used to extract spatial information; PrimaryCaps layer is used to extract spectral information and form capsule units; 1D-ConvCaps layer is used to reduce parameters; ClassCaps layer is used to represent part-whole relationship.

B. PrimaryCaps Layer

This layer is the first of capsule layers, represented by \( \mathcal{L}^{(2)} \), the input of which is \( \mathbf{O}^{(1)} \). Its goal is to extract spectral information from \( \mathbf{O}^{(1)} \) and form capsule units. Firstly, the PrimaryCaps layer applies \( \mathcal{K}^{(2)} \) 1D-convolution filters with size \( f^{(2)} \) to \( \mathbf{O}^{(1)} \) to obtain \( \mathbf{K}^{(2)} \) feature maps. Secondly, these feature maps are stacked into a tensor with size \( c^{(2)} \times a^{(2)} \times d^{(2)} \), where \( a^{(2)} \) is the number of capsule arrays, and \( d^{(2)} \) is the dimension of every capsule.

Compared to the 3D-convolution, our proposed spectral-spatial information extraction method is more efficient because this separated method can reduce parameter redundancy. Compared to the scalar neuron of CNN-based model, capsule is more expressive because additional information is stored. Therefore, the capsule network has great advantages in noise immunity, convergence speed and generalization ability. From this layer, the basic unit of network is capsule neuron, which connects to 1D-ConvCaps layer through constraint window.

C. 1D-ConvCaps Layer

This layer is used to reduce the number of parameters, represented by \( \mathcal{L}^{(3)} \). In CapsNet, full connection is used directly between adjacent capsule layers, resulting in parameters redundancy. Hence, CapsNet has high storage pressure and time cost due to numerous parameters. Herein, we propose the constraint window method for capsule network, which utilizes a local strategy to reduce parameters redundancy of the network. Constraint window realizes the goal through local connection and sharing parameters.

This layer constructs \( \mathcal{K}^{(3)} \) constraint windows on input \( \mathbf{O}^{(2)} \) for generating \( \mathcal{K}^{(3)} \) output capsule arrays, denoted by \( \mathbf{O}^{(3)} \in \mathbb{R}^{c^{(3)} \times a^{(3)} \times d^{(3)}} \), where \( a^{(3)} = \mathcal{K}^{(3)} \). The size of each window is \( f^{(3)} \times a^{(2)} \times c^{(2)} \), where \( f^{(3)} \) is the artificially specified size. The \( q \)-th window has a viewpoint tensor \( \mathbf{T}^{(3)}_{q} \in \mathbb{R}^{d^{(3)} \times f^{(3)} \times a^{(2)} \times d^{(2)}} \), which does not depend on a fixed spatial location and shares parameters with other child capsules. For the parent capsule \( \mathbf{u}^{(3)}_{q,k} \in \mathbf{O}^{(3)} \) at the \( k \)-th position of the \( q \)-th capsule array, it is equal to the product of \( \mathbf{T}^{(3)}_{q} \) and \( \mathbf{C}^{(2)}_{k} \in \mathbb{R}^{f^{(3)} \times a^{(2)} \times d^{(2)}} \). Formally, \( \mathbf{u}^{(3)}_{q,k} \) expressed by following equation:

\[
\mathbf{u}^{(3)}_{q,k} = \mathbf{T}^{(3)}_{q} \times \mathbf{C}^{(2)}_{k} + \mathbf{b}^{(3)}_{q}
\]

where \( \mathbf{C}^{(2)}_{k} \) is child capsules of \( \mathbf{O}^{(2)} \) covered by constraint window at position \( k \) and \( \mathbf{b}^{(3)}_{q} \) is learned bias for parent capsule. Moreover, \( \mathbf{T}^{(3)}_{q} \) can be decomposed into \( f^{(3)} \times a^{(2)} \) viewpoint matrices \( \mathbf{W}^{(3)}_{q,i,j} \in \mathbb{R}^{f^{(3)} \times d^{(2)}} \), where \( i \) and \( j \) are indices of \( a^{(2)} \) and \( f^{(3)} \) respectively. Hence, \( \mathbf{u}^{(3)}_{q,k} \) is the sum of product of viewpoint matrix and corresponding child capsule. It is formulated as follows:

\[
\mathbf{u}^{(3)}_{q,k} = \sum_{i=1}^{a^{(2)}} \sum_{j=1}^{f^{(3)}} \mathbf{W}^{(3)}_{q,i,j} \times \mathbf{c}^{(2)}_{i,(k-1)s+j} + \mathbf{b}^{(3)}_{q}
\]

where \( s \) is the stride of constraint window move to next position. Similar to conventional convolution, the viewpoint matrix can be learned by BP algorithm.

D. ClassCaps Layer

This layer is a dense capsule layer, which represents the part-whole relationship of capsules, denoted by \( \mathcal{L}^{(4)} \). Its output \( \mathbf{O}^{(4)} \in \mathbb{R}^{n \times d^{(4)}} \) is obtained by using dynamic routing on \( \mathbf{O}^{(3)} \). Every capsule \( \mathbf{u}^{(4)}_{k} \in \mathbf{O}^{(4)} \) represents a class of ground truth in HSI, where the length of vector is the probability of sample belonging to the corresponding class. Hence, the capsule \( \mathbf{u}^{(4)}_{k} \) is also referred as the activation capsule. In order to obtain the activation capsule from \( \mathcal{L}^{(3)} \), firstly, dynamic routing needs to
calculate the weighted sum of prediction vectors by following equation:

\[ s_k^{(4)} = \sum_{i=1}^{g} \sum_{j=1}^{n} t_{i,j,k}^{(4)} \times u_{k,i,j}^{(4)}. \]  

(6)

Therein, prediction vector \( u_{k,i,j}^{(4)} \) is equal to product of the child capsule \( u_{i,j}^{(3)} \) and the corresponding viewpoint matrix \( W_{i,j,k}^{(4)} \), and \( t_{i,j,k}^{(4)} \) is the coupling coefficient between \( u_{k,i,j}^{(4)} \) and \( u_{k}^{(4)} \). Formally, \( t_{i,j,k}^{(4)} \) is computed by a function of routing softmax as follows:

\[ t_{i,j,k}^{(4)} = \frac{\exp(t_{i,j,k}^{(4)})}{\sum_{g=1}^{n} \exp(t_{i,j,g}^{(4)})} \]

(7)

where \( t_{i,j,k}^{(4)} \) is the log prior of \( u_{k,i,j}^{(4)} \) coupling to \( u_{k}^{(4)} \) and initial value is zero. Secondly, capsule \( u_{k}^{(4)} \) is computed by a nonlinear activation function, called squashing:

\[ u_{k}^{(4)} = \frac{||s_{k}^{(4)}||^2 s_{k}^{(4)}}{1 + ||s_{k}^{(4)}||^2 ||s_{k}^{(4)}||}. \]

(8)

Finally, the agreement between \( u_{k}^{(4)} \) and \( u_{k}^{(4)} \) is measured by the simple scalar product \( \langle u_{k,i,j}^{(4)}, u_{k}^{(4)} \rangle \). A large scalar product means the increasing of coupling coefficient between capsules. By iterating the above process and accumulate \( t_{i,j,k}^{(4)} \), the coupling coefficient \( t_{i,j,k}^{(4)} \) can be obtained rapidly. This not only transmits information between capsules, but also connects parts to the whole by assigning coupling coefficients.

E. Loss Function

In training phase, a patch \( P_{x,y} \) is input into the network and \( n \) activity vectors are obtained by forward propagation. Before the back propagation, the network needs to measure the gap between \( O^{(4)} \) and label via loss function. Herein, we use the margin loss as the global loss function. It can be defined as follows:

\[ L = \sum_{k=1}^{n} (T_k \max(0, r^+ - ||u_k^{(4)}||)^2 \\
+ \lambda(1 - T_k) \max(0, ||u_k^{(4)}|| - r^-)^2) \]

(9)

where \( T_k \) is an indicator function. It can be defined by

\[ T_k = \begin{cases} 1, & \text{if class } k \text{ is present in sample} \\ 0, & \text{otherwise.} \end{cases} \]

(10)

Herein, the indicator function \( T_k \) is used to indicate which part (addends) of \( L \) is active. The first part works when \( T_k = 1 \). Otherwise, the second part works when \( T_k = 0 \). In order to avoid the maximum or collapse loss, the loss function \( L \) introduces the concept of boundary, forcing the length of the activity vector \( ||u_k^{(4)}|| \) falls into small interval. The boundary parameters \( r^+ \) and \( r^- \) are upper boundary and lower boundary, respectively. Additionally, the regularization parameter \( \lambda \) is used to shrink the influence of activity vector when the corresponding class does not exist in sample.

V. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our proposed method on three representative HSI datasets. Firstly, we introduce three hyperspectral datasets, which are used to verify our proposed 1D-ConvCapsNet. Secondly, we provide the hyper-parameters setup, experimental environment and contrastive methods. Finally, we compare experimental results and give relative discussions.

![Image 324x77 to 551x253](a) (b)

Fig. 5. The real image (a) and ground truths map (b) of Indian Pines dataset.

| Label          | Class | Training | Validation | Test   |
|----------------|-------|----------|------------|--------|
| 1 Alfalfa      | 9     | 4        | 3          |        |
| 2 Corn-no      | 285   | 142      | 1,001      |        |
| 3 Corn-min     | 166   | 83       | 581        |        |
| 4 Corn         | 47    | 23       | 167        |        |
| 5 Grass/pasture| 96    | 48       | 339        |        |
| 6 Grass/trees  | 146   | 73       | 511        |        |
| 7 Grass/pasture-mo | 5   | 2       | 21         |        |
| 8 Hay-win      | 95    | 47       | 336        |        |
| 9 Oats         | 4     | 2        | 14         |        |
| 10 Soy-no      | 194   | 97       | 681        |        |
| 11 Soy-min     | 491   | 245      | 1,719      |        |
| 12 Soy-cle     | 118   | 59       | 416        |        |
| 13 Wheat       | 41    | 20       | 144        |        |
| 14 Woods       | 253   | 126      | 886        |        |
| 15 BGTD        | 77    | 38       | 271        |        |
| 16 SST         | 18    | 9        | 66         |        |
| Total          | 1     | 2,045    | 1,018      | 7,186  |

![Table I](Image 535 to 551x621)

TABLE I

The Sample Numbers of 20% Training Set, 10% Verification Set and 70% Test Set on Indian Pines Dataset

![Image 324x535 to 551x621](a) (b)

Fig. 6. The real image (a) and ground truths map (b) of University of Pavia.
of Pavia (UP) and Salinas (SA), respectively. IP has uneven distribution of sample numbers and low spatial resolution, UP has small spectral channels, and SA has large data volume. These characteristics are used to verify that whether 1D-ConvCapsNet can achieve the desired performance under a variety of conditions.

1) Indian Pines: IP scene is the subset of Big Indian Pines (BIP) scene, which is generated by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor at Northwestern Indiana, USA, in 1992. It consists of several agricultural crops, such as corn, oat and wheat. The IP contains 145 × 145 pixels with a spatial resolution of 20m, including 16 classes of interested ground truths. Each pixel consists of 220 SCs, which cover wavelengths from 400 to 2500nm. The real image and label map are shown in Fig. 5. The numbers of each ground truth in training, validation and test sets are recorded in Table I. We can see that there are a few samples of some ground truths, i.e., Alfalfa, Grass/pasture-mo and Oats.

2) University of Pavia: UP scene is generated by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor in the city of Pavia, Italy, during a flight campaign over Pavia in 2001. It is a typical image of the city, including many building materials, such as bricks, asphalt and metal sheets. The UP contains 610 × 340 pixels with a spatial resolution of 1.3m, including 9 classes of interested ground truths. Each pixel consists of 103 SCs, which cover wavelengths from 430 to 860nm. The real image and label map are shown in Fig. 6. The numbers of each ground truth in training, validation and test sets are recorded in Table II.

3) Salinas: SA scene is generated by AVIRIS sensor at Salinas valley, California, USA, in 1992. Like IP, it also consists of several agriculture related fields, such as vegetables, bare soils, and vineyard fields. The SA contains 512 × 217 pixels with a spatial resolution of 3.7m, including 16 classes of interested ground truths. Each pixel consists of 224 SCs, which cover wavelengths from 400 to 2500nm. The real image and label map are shown in Fig. 7. The numbers of each ground truth in training, validation and test sets are recorded in Table III.

4) Data Preprocessing: The performance of classifiers is affected by using raw data due to the high correlations between SCs. Hence, it is necessary to preprocess the data before going to the training stage. In our method, the correlation between SCs is eliminated by using PCA-Whitening.

B. Experimetal Setup

1) Hyperparameters: The optimal structure of 1D-ConvCapsNet is determined by repeated experiments, which is shown in Table IV. Adam optimizer is used to train 50 epochs with 0.01 learning rate and 3 routing iterations without regularization and normalization. The batch size is set to 64 for the IP and 256 for the rest, which is related to the number of samples. The Hyperparameters $r^+$, $r^-$ and $\lambda$ in the loss function are set to 0.9, 0.1 and 0.5, respectively. During the experiments, 20% patches of each class are randomly selected as the training set, 10% patches as the validation set and the rest patches as the test set. The accuracy on the validation set

A. Hyperspectral Datasets and Preprocessing

Three representative hyperspectral datasets in HSIs classification are selected, which are Indian Pines (IP), University
is recorded during every epoch, the highest of which is regarded as optimal parameters to verify the performance on test set.

2) Experimental Environment: All experiments are conducted under the same environment. The software platform consists of Intel Core i7-7820HK processor (four core/eight threading) with 8M L3-cache, 16GB DDR4 memory with 2800Mhz serial speed, Nvidia GeForce GTX 1070 GPU with 8GB DDR5 video memory and 1TB HDD with 7200 RPM. The operating system, Keras 2.1.1 based on TensorFlow-gpu 1.3.0 consists of Intel Core i7-7820HK processor (four core/eight threading) with 8M L3-cache, 16GB DDR4 memory with 2800Mhz serial speed, Nvidia GeForce GTX 1070 GPU with 8GB DDR5 video memory and 1TB HDD with 7200 RPM.

3) Comparative methods: In experiment 1, four well-known HSIs classification methods are selected as comparative methods. They are SVM with radial basis function (RBF-SVM) [47], MLP with four hidden layers, semi-supervised convolutional neural network (Se-2D-CNN) [34] and 3D-convolutional neural network (3D-CNN) [16]. Note that RBF-SVM and MLP focus on spectral information, Se-2D-CNN focuses on spatial information, and 3D-CNN and 1D-ConvCapsNet consider spectral-spatial information for HSIs classification. In Experiment 2, CNN model and CapsNet [23] are selected as comparative methods to compare the gaps between different training methods. The structure of CNN model is similar to 1D-ConvCapsNet except the neuron type.

In order to quantify the accuracy of classifiers, overall accuracy (OA), average classification accuracy (AA) and kappa coefficient (κ) serve as evaluation metrics. In order to obtain stable results, we conduct 20 experiments and take the median of OA as convincing result. The training time is also recorded to evaluate the time cost in the training stage. Since the training time is affected by the system utilization rate and other factors, the minimal training time in 20 experiments is recorded as the result.

C. Results of experiment and Discussion

1) Experiments 1: This experiment validate the performances of the proposed and comparative methods on IP, UP and SA. In the RBF-SVM, the parameters γ and C are set to 0.005 and 100, respectively. In the MLP, the number of neurons with ReLU activation function and 50% dropout probability for each layer is 2048, 4096, 2048 and 2048, respectively, which uses Adam optimizer to train 500 epochs with 0.01 learning rate. In the Se-2D-CNN, the default settings are used. In the 3D-CNN, due to the extremely large model scale, it cannot be directly run in our environment. In order to make 3D-CNN work, the number of filters in every layer is reduced to 32, and the rest of settings are unchanged. Table VIII records the accuracy and the evaluation metrics of all methods on IP, UP and SA, where each row represents the classification accuracy and each column represents each method.

From the results recorded in Table VIII, we can observe that the accuracy of the proposed method is superior to the RBF-SVM, MLP, Se-2D-CNN and 3D-CNN on three datasets.
Compared to 3D-CNN, the accuracy of 1D-ConvCapsNet on IP has great improvement, but on UP and SA it is not so obvious. This is because the proposed method does not adjust structure and hyperparameters for UP and SA. Generally, the accuracy of 1D-ConvCapsNet and 3D-CNN are superior to the rest of comparative methods. This result is expected because RBF-SVM and MLP only focus on spectral information, and Se-2D-CNN only focus on spatial information. Our proposed method and 3D-CNN can extract information from spectral and spatial domains, leading to higher accuracy. However, 1D-ConvCapsNet only trains 50 epochs, which means that our method converges faster than 3D-CNN and has lower training cost. The training time of 1D-ConvCapsNet on three datasets is about 2% of 3D-CNN. Moreover, 1D-ConvCapsNet can achieve outstanding performance on classes with small number of samples, i.e., Alfalfa, Grass/pasture-mo, Oats and SST of IP. We can see from the Table V all comparative methods perform unsatisfactorily on those classes, because these methods cannot generalize the entire ground truths using very few training samples. Especially for Se-2D-CNN, those classes do not provide enough texture information, resulting in poor accuracy performance. The performance of 1D-ConvCapsNet on those classes is much better than comparative methods, which benefit from powerful representation and interpretation capabilities of the capsule network.

In order to intuitively represent the performance of different methods, all samples are input into the trained classifiers to obtain classification map for each method, which are shown in Fig. 8-10. It can be seen that the classification maps of SVM and MLP contain a lot of noise. This is because the quality of spectral information is affected by spatial resolution and imaging conditions, resulting in the phenomenon that the same
This experiment, which extracts spectral-spatial information by using our proposed methods. The structure of CNN-based model is identical to 1D-ConvCapsNet except the neuron type, which is denoted by c-CNN. In other words, all units of c-CNN are standard scalar neuron rather than vector neuron of capsule network. Since the original CapsNet operates on 2D-data, the 2D-convolution of CapsNet is modified to 1D-convolution after spatial information is extracted by using the SpatialConv layer. Table VIII-X records the accuracy and evaluation metrics of three methods on IP, UP and SA, where each row represents the classification accuracy and each column represents one training method. Fig. 11-13 illustrate the classification map of each method.

From the result recorded in Table VIII-X we can observe that c-CNN has significantly higher accuracy than SVM, MLP and Se-2D-CNN due to the use of our proposed spectral-spatial information extraction method. CapsNet and 1D-ConvCapsNet are superior to c-CNN overall, because the advantages of capsules in expressiveness and interpretability. It should be noted that the c-CNN achieve this result with 500 training epochs to, while the CapsNet and 1D-ConvCapsNet with 50 training epochs. 1D-ConvCapsNet is superior to CapsNet in accuracy because the 1D-ConvCaps layer can learn which spectrum expresses different objects or same objects have different spectrum. Therefore, classifiers which only focus on spectral information have high error rates. The classification map of Se-2D-CNN contains less noise than that of SVM and MLP. However, Se-2D-CNN does not perform well on the edges of classes and ground truths with similar textures, because these samples provide insufficient spatial information for discrimination. 3D-CNN and 1D-ConvCapsNet achieve higher accuracy and less noise than other comparative methods, because they can act on both spatial information and spectral information. This result is consistent with Table VII.

2) Experiments 2: This experiment is used to compare gaps between different training methods. A CNN-based model and CapsNet are selected as the comparative methods in this experiment, which extract spectral-spatial information by using our proposed methods. The structure of CNN-based model is identical to 1D-ConvCapsNet except the neuron type, which is denoted by c-CNN. In other words, all units of c-CNN are standard scalar neuron rather than vector neuron of capsule network. Since the original CapsNet operates on 2D-data, the 2D-convolution of CapsNet is modified to 1D-convolution after spatial information is extracted by using the SpatialConv layer. Table VIII-X records the accuracy and evaluation metrics of three methods on IP, UP and SA, where each row represents the classification accuracy and each column represents one training method. Fig. 11-13 illustrate the classification map of each method.

From the result recorded in Table VIII-X we can observe that c-CNN has significantly higher accuracy than SVM, MLP and Se-2D-CNN due to the use of our proposed spectral-spatial information extraction method. CapsNet and 1D-ConvCapsNet are superior to c-CNN overall, because the advantages of capsules in expressiveness and interpretability. It should be noted that the c-CNN achieve this result with 500 training epochs to, while the CapsNet and 1D-ConvCapsNet with 50 training epochs. 1D-ConvCapsNet is superior to CapsNet in accuracy because the 1D-ConvCaps layer can learn which spectrum expresses different objects or same objects have different spectrum. Therefore, classifiers which only focus on spectral information have high error rates. The classification map of Se-2D-CNN contains less noise than that of SVM and MLP. However, Se-2D-CNN does not perform well on the edges of classes and ground truths with similar textures, because these samples provide insufficient spatial information for discrimination. 3D-CNN and 1D-ConvCapsNet achieve higher accuracy and less noise than other comparative methods, because they can act on both spatial information and spectral information. This result is consistent with Table VII.

2) Experiments 2: This experiment is used to compare gaps between different training methods. A CNN-based model and CapsNet are selected as the comparative methods in this experiment, which extract spectral-spatial information by using our proposed methods. The structure of CNN-based model is identical to 1D-ConvCapsNet except the neuron type, which is denoted by c-CNN. In other words, all units of c-CNN are standard scalar neuron rather than vector neuron of capsule network. Since the original CapsNet operates on 2D-data, the 2D-convolution of CapsNet is modified to 1D-convolution after spatial information is extracted by using the SpatialConv layer. Table VIII-X records the accuracy and evaluation metrics of three methods on IP, UP and SA, where each row represents the classification accuracy and each column represents one training method. Fig. 11-13 illustrate the classification map of each method.

From the result recorded in Table VIII-X we can observe that c-CNN has significantly higher accuracy than SVM, MLP and Se-2D-CNN due to the use of our proposed spectral-spatial information extraction method. CapsNet and 1D-ConvCapsNet are superior to c-CNN overall, because the advantages of capsules in expressiveness and interpretability. It should be noted that the c-CNN achieve this result with 500 training epochs to, while the CapsNet and 1D-ConvCapsNet with 50 training epochs. 1D-ConvCapsNet is superior to CapsNet in accuracy because the 1D-ConvCaps layer can learn which spectrum expresses different objects or same objects have different spectrum. Therefore, classifiers which only focus on spectral information have high error rates. The classification map of Se-2D-CNN contains less noise than that of SVM and MLP. However, Se-2D-CNN does not perform well on the edges of classes and ground truths with similar textures, because these samples provide insufficient spatial information for discrimination. 3D-CNN and 1D-ConvCapsNet achieve higher accuracy and less noise than other comparative methods, because they can act on both spatial information and spectral information. This result is consistent with Table VII.

2) Experiments 2: This experiment is used to compare gaps between different training methods. A CNN-based model and CapsNet are selected as the comparative methods in this experiment, which extract spectral-spatial information by using our proposed methods. The structure of CNN-based model is identical to 1D-ConvCapsNet except the neuron type, which is denoted by c-CNN. In other words, all units of c-CNN are standard scalar neuron rather than vector neuron of capsule network. Since the original CapsNet operates on 2D-data, the 2D-convolution of CapsNet is modified to 1D-convolution after spatial information is extracted by using the SpatialConv layer. Table VIII-X records the accuracy and evaluation metrics of three methods on IP, UP and SA, where each row represents the classification accuracy and each column represents one training method. Fig. 11-13 illustrate the classification map of each method.

From the result recorded in Table VIII-X we can observe that c-CNN has significantly higher accuracy than SVM, MLP and Se-2D-CNN due to the use of our proposed spectral-spatial information extraction method. CapsNet and 1D-ConvCapsNet are superior to c-CNN overall, because the advantages of capsules in expressiveness and interpretability. It should be noted that the c-CNN achieve this result with 500 training epochs to, while the CapsNet and 1D-ConvCapsNet with 50 training epochs. 1D-ConvCapsNet is superior to CapsNet in accuracy because the 1D-ConvCaps layer can learn which spectrum expresses different objects or same objects have different spectrum. Therefore, classifiers which only focus on spectral information have high error rates. The classification map of Se-2D-CNN contains less noise than that of SVM and MLP. However, Se-2D-CNN does not perform well on the edges of classes and ground truths with similar textures, because these samples provide insufficient spatial information for discrimination. 3D-CNN and 1D-ConvCapsNet achieve higher accuracy and less noise than other comparative methods, because they can act on both spatial information and spectral information. This result is consistent with Table VII.

2) Experiments 2: This experiment is used to compare gaps between different training methods. A CNN-based model and CapsNet are selected as the comparative methods in this experiment, which extract spectral-spatial information by using our proposed methods. The structure of CNN-based model is identical to 1D-ConvCapsNet except the neuron type, which is denoted by c-CNN. In other words, all units of c-CNN are standard scalar neuron rather than vector neuron of capsule network. Since the original CapsNet operates on 2D-data, the 2D-convolution of CapsNet is modified to 1D-convolution after spatial information is extracted by using the SpatialConv layer. Table VIII-X records the accuracy and evaluation metrics of three methods on IP, UP and SA, where each row represents the classification accuracy and each column represents one training method. Fig. 11-13 illustrate the classification map of each method.

From the result recorded in Table VIII-X we can observe that c-CNN has significantly higher accuracy than SVM, MLP and Se-2D-CNN due to the use of our proposed spectral-spatial information extraction method. CapsNet and 1D-ConvCapsNet are superior to c-CNN overall, because the advantages of capsules in expressiveness and interpretability. It should be noted that the c-CNN achieve this result with 500 training epochs to, while the CapsNet and 1D-ConvCapsNet with 50 training epochs. 1D-ConvCapsNet is superior to CapsNet in accuracy because the 1D-ConvCaps layer can learn which spectrum expresses different objects or same objects have different spectrum. Therefore, classifiers which only focus on spectral information have high error rates. The classification map of Se-2D-CNN contains less noise than that of SVM and MLP. However, Se-2D-CNN does not perform well on the edges of classes and ground truths with similar textures, because these samples provide insufficient spatial information for discrimination. 3D-CNN and 1D-ConvCapsNet achieve higher accuracy and less noise than other comparative methods, because they can act on both spatial information and spectral information. This result is consistent with Table VII.

2) Experiments 2: This experiment is used to compare gaps between different training methods. A CNN-based model and CapsNet are selected as the comparative methods in this experiment, which extract spectral-spatial information by using our proposed methods. The structure of CNN-based model is identical to 1D-ConvCapsNet except the neuron type, which is denoted by c-CNN. In other words, all units of c-CNN are standard scalar neuron rather than vector neuron of capsule network. Since the original CapsNet operates on 2D-data, the 2D-convolution of CapsNet is modified to 1D-convolution after spatial information is extracted by using the SpatialConv layer. Table VIII-X records the accuracy and evaluation metrics of three methods on IP, UP and SA, where each row represents the classification accuracy and each column represents one training method. Fig. 11-13 illustrate the classification map of each method.

From the result recorded in Table VIII-X we can observe that c-CNN has significantly higher accuracy than SVM, MLP and Se-2D-CNN due to the use of our proposed spectral-spatial information extraction method. CapsNet and 1D-ConvCapsNet are superior to c-CNN overall, because the advantages of capsules in expressiveness and interpretability. It should be noted that the c-CNN achieve this result with 500 training epochs to, while the CapsNet and 1D-ConvCapsNet with 50 training epochs. 1D-ConvCapsNet is superior to CapsNet in accuracy because the 1D-ConvCaps layer can learn which
primary capsules are more important and combine them into complex entities.

As can be seen from Fig. 11-13 classification maps obtained by CapsNet and 1D-ConvCapsNet have fewer noises than that of the c-CNN due to the high accuracy of the capsule network. The c-CNN has a lot of noises on Corn-no and Soy-no of IP, Bare Soil and Self-b Bricks of UP, and Grapes-un and Vinyard-un of SA. The 1D-ConvCapsNet have almost no errors in those classes and its classification maps are closer to label maps.

Fig. 14 illustrates the convergence speed of three methods during the training stage, where the horizontal and vertical axes represent the number of epochs and the accuracy of verification, respectively. It can be seen that the two capsule-based methods converge faster than the c-CNN, and the accuracy of verification is higher than the c-CNN. 1D-ConvCapsNet converges slightly slower than CapsNet, but their accuracy of verification is similar. This is because our proposed method needs to express complex entities before connecting parts to the whole. Table XI records the training time and the number of parameters for CapsNet and 1D-ConvCapsNet, which represent the time cost and the storage cost, respectively. Compared to the CapsNet, 1D-ConvCapsNet is faster and lighter with the same accuracy, which means fewer efforts are required in the training stage. As can be seen in Table XI 1D-ConvCapsNet is about 4%-7% of CapsNet in terms of the training time and the number of parameters on three datasets. This is because of the local strategy of 1D-ConvCapsNet by using local connection and sharing parameters on PrimaryCaps layer.

In order to verify the performance of 1D-ConvCapsNet on small samples, 10% and 5% of samples are randomly selected as the training sets in each class. The proportion of verification set is still 10%, and the remaining pixels are the test set. The hyperparameters of 1D-ConvCapsNet are unchanged. The OA of 1D-ConvCapsNet on the small training sets is recorded in Table XII. We can see that 1D-ConvCapsNet can maintain high accuracy with 10% training samples, and the performance by using 5% training samples is still acceptable.

The above experimental results show that 1D-ConvCapsNet has high accuracy and low training cost. Compared to the comparative methods, our proposed method is highly competitive. The accuracy of 1D-ConvCapsNet can reach the level of 3D-CNN which is state-of-the-art method introduced by [16]. However, 1D-ConvCapsNet is much better than 3D-CNN in training speed and the number of parameters. Compared to CapsNet, 1D-ConvCapsNet greatly reduces the number of parameters and guarantees the accuracy. Our methods save time and storage cost, and extend the application of capsule network.

VI. CONCLUSION

In this work, we proposed a fast and accurate capsule network for HSIs classification task, called 1D-ConvCapsNet. Firstly, 1D-ConvCapsNet separately extracts features on spatial and spectral domains. Compared to 3D-CNN, our separate feature extraction method is lightweight and fast due to fewer
parameters. Secondly, 1D-ConvCapsNet uses local strategy to reduce the scale of capsule network. Finally, 1D-ConvCapsNet obtains predictions by using dynamic routing. It is expected to have fewer parameters than several state-of-the-art methods and ensure high precision.

The effectiveness of 1D-ConvCapsNet has been validated on three representative datasets in the HSIs classification field. Experimental results showed that 1D-ConvCapsNet is very competitive with the comparison algorithm. The accuracy of 1D-ConvCapsNet achieved the level of state-of-the-art methods, but with much lower training time and hardware requirements. Compared to CapsNet, 1D-ConvCapsNet is about 4%-7% of CapsNet in terms of training time and the number of parameters on three datasets. 1D-ConvCapsNet also achieved outstanding performance on small samples due to powerful representation and interpretation capabilities of capsule network. In the future work, we expect to extend the 1D-ConvCapsNet to more efficient way for solving HSIs classification problem, such developing more effective regularizations and the tensor constraint.

REFERENCES

[1] D. Landgrebe, “Hyperspectral image data analysis,” IEEE Signal Process. Mag., vol. 19, no. 1, pp. 17-28, Jan. 2002.

[2] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, “Hyperspectral remote sensing data analysis and future challenges,” IEEE Geosci. Remote Sens. Mag., vol. 1, no. 2, pp. 6-36, Jun. 2013.

[3] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, “Advances in hyperspectral image classification: Earth monitoring with statistical learning methods,” IEEE Signal Process. Mag., vol. 31, no. 1, pp. 45-54, Jan. 2014.

[4] P. Ghamisi et al., “Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art,” IEEE Geosci. Remote Sens. Mag., vol. 5, no. 4, pp. 37-78, Dec. 2017.

[5] A. Chihyam and H. Z. Shafri, “A review on hyperspectral remote sensing for homogeneous and heterogeneous forest biodiversity assessment,” Int. J. Remote Sens., vol. 31, no. 7, pp. 1837-1856, 2010.

[6] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, “Advances in spectral-spatial classification of hyperspectral images,” Proc. IEEE, vol. 101, no. 3, pp. 652-675, Mar. 2013.

[7] X. Zhang, Y. Sun, K. Shang, L. Zhang, and S. Wang, “Crop classification based on feature set construction and object-oriented approach using hyperspectral images,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 9, pp. 4117-4128, Sep. 2016.

[8] K. Manjunath, S. Ray, and D. Vyas, “Identification of indices for accurate estimation of anthocyanin and carotenoids in different species of flowers using hyperspectral data,” Remote Sens. Lett., vol. 7, no. 10, pp. 1004-1013, 2016.

[9] A. J. Brown, B. Sutter, and S. Dunagan, “The MARTE VNIR imaging spectrometer experiment: Design and analysis,” Astrobiology, vol. 8, no. 5, pp. 1001-1011, 2008.

[10] G. Camps-Valls and L. Bruzzone, “Kernel-based methods for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 43, no. 6, pp. 1351-1362, Jun. 2004.

[11] Y. Chen, N. M. Nasrabadi, and T. D. Tran, “Hyperspectral image classification using dictionary-based sparse representation,” IEEE Trans. Geosci. Remote Sens., vol. 50, no. 10, pp. 3973-3985, Oct. 2011.

[12] J. Haut, M. Paoletti, J. Plaza, and A. Plaza, “Cloud implementation of the K-means algorithm for hyperspectral image analysis,” J. Supercomput., vol. 73, no. 1, pp. 514-529, 2017.

[13] Y. LeCun, Y. Bengio, and G. E. Hinton, “Deep learning,” Nature, vol. 521, pp. 436-444, May 2015.

[14] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, “Deep learning-based classification of hyperspectral data,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 7, no. 6, pp. 2094-2107, Jun. 2014.

[15] Y. Chen, X. Zhao, and X. Jia, “Spectral-spatial classification of hyperspectral data based on deep belief network,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 8, no. 6, pp. 2381-2392, Jun. 2015.

[16] 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 Trans. Geosci. Remote Sens., vol. 54, no. 10, pp. 6232-6251, Oct. 2016.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097-1105.

[18] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504-507, Jul. 2006.

[19] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 221231, Jan. 2013.

[20] Y. Li, H. Zhang, and Q. Shen, “Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network,” Remote Sens., vol. 9, no. 1, pp. 67, 2017.

[21] A. Ben Hamida, A. Benoit, P. Lambert and C. Ben Amar, “3-D Deep Learning Approach for Remote Sensing Image Classification,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 8, pp. 4420-4434, Aug. 2018.

[22] M. He, B. Li and H. Chen, “Multi-scale 3D deep convolutional neural networks for hyperspectral image classification,” in Proc. IEEE Int. Conf. Image Process., 2017, pp. 3904-3908.

[23] S. Sabour, N. Frost, and G. E. Hinton, “Dynamic routing between capsules,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 3859-3869.

[24] R. LaLonde, U. Bagci. (2018). “Capsules for Object Segmentation.” [Online]. Available: [http://arxiv.org/abs/1804.04243]]

[25] G. E. Hinton, S. Sabour, and N. Frost. “Matrix capsules with EM routing,” in Proc. Int. Conf. Learn. Represent., 2018. [Online]. Available: https://openreview.net/forum?id=HJWLGWRb

[26] A. Mobiny, and H. V. Nguyen. (2018). “Fast CapsNet for Lung Cancer Screening.” [Online]. Available: http://arxiv.org/abs/1806.07416

[27] X. Wei et al., “Reconstructible Nonlinear Dimensionality Reduction via Joint Dictionary Learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 1, pp. 175-189, Jan. 2019.

[28] X. Wei, H. Shen, and M. Kleinsteuber, “Trace Quotient Meets Sparsity: A Method for Learning Low Dimensional Image Representations,” in
[29] Q. Wang, Z. Meng, and X. Li, “Locality adaptive discriminant analysis for spectral-spatial classification of hyperspectral images,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 11, pp. 2077-2081, Nov. 2017.

[30] Q. Wang, J. Lin, and Y. Yuan, “Salient band selection for hyperspectral image classification via manifold ranking,” IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 6, pp. 1279-1289, Jun. 2016.

[31] C. Tao, H. Pan, Y. Li, and Z. Zou, “Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification,” IEEE Geosci. Remote Sens. Lett., vol. 12, no. 12, pp. 2438-2442, Dec. 2015.

[32] X. Ma, A. Fu, J. Wang, H. Wang and B. Yin, “Hyperspectral Image Classification Based on Deep Deconvolution Network With Skip Architecture,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 8, pp. 4781-4791, Aug. 2018.

[33] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, “Learning and transferring deep joint spectral-spatial features for hyperspectral classification,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 8, pp. 4729-4742, Aug. 2017.

[34] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, “A semi-supervised convolutional neural network for hyperspectral image classification,” Remote Sens. Lett., vol. 8, no. 9, pp. 839-848, Sep. 2017.

[35] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, “Hyperspectral image classification with Markov random fields and a convolutional neural network,” IEEE Trans. Image Process., vol. 27, no. 5, pp. 2354-2367, May 2018.

[36] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in Proc. 14th Int. Conf. Artif. Intell. Statist., 2011, pp. 315-323.

[37] N. Kalchbrenner, E. Gregor, and P. Blunsom, (2018). “A Convolutional Neural Network for Modelling Sentences.” [Online]. Available: https://arxiv.org/abs/1404.2188

[38] H. Gao, S. Lin, C. Li, and Y. Yang, “Application of Hyperspectral Image Classification Based on Overlap Poolings,” Neural Process Lett., pp. 1-20, Jun. 2018.

[39] J. Acquarelli, E. Marchiori, L. M. C. Buydens, T. N. Tran, and T. van Laarhoven, “Spectral-spatial classification of hyperspectral images: Three tricks and a new supervised learning setting,” Remote Sens., vol. 10, no. 7, pp. 1156, Jul 2018.

[40] G. E. Hinton, A. Krizhevsky, and S. D. Wang, “Transforming auto-encoders,” in Proc. Int. Conf. Artif. Neural Netw., 2011, pp. 4451.

[41] D. J. Field, “Wavelets, vision and the statistics of natural scenes,” Phil. Trans. Roy. Soc. London A, Math., Phys. Eng. Sci., vol. 357, no. 1760, pp. 2527-2542, 1999.

[42] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” The Journal of physiology, vol. 160, no. 1, pp. 106-154, Jan. 1962.

[43] B. Waske, S. van der Linden, J. Benediktsson, A. Rabe, and P. Hostert, “Sensitivity of support vector machines to random feature selection in classification of hyperspectral data,” IEEE Trans. Geosci. Remote Sens., vol. 48, no. 7, pp. 2880-2889, Jul. 2010.