Hyperspectral Image Classification With CapsNet and Markov Random Fields

XUEFENG JIANG*, YUE ZHANG, WENBO LIU, JUNYU GAO, (Member, IEEE), JUNRUI LIU, YANNING ZHANG, (Senior Member, IEEE), AND JIANZHE LIN

1School of Computer Science, Northwestern Polytechnical University, Xi’an 710072, China
2School of Software, Northwestern Polytechnical University, Xi’an 710072, China
3Center for Optical Imagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi’an 710072, China
4Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z2, Canada

Corresponding author: Jianzhe Lin (jianzhelin@gmail.com)

ABSTRACT Hyperspectral image (HSI) classification is one of the most challenging problems in understanding HSI. Convolutional neural network (CNN), with the strong ability to extract features using the hidden layers in the network, has been introduced to solve this problem. However, several fully connected layers are always appended at the end of CNN, which dramatically reduced the efficiency of space utilization and make the classification algorithm hard to converge. Recently, a new network architecture called capsule network (CapsNet) was presented to improve the CNN. It uses groups of neurons as capsules to replace the neurons in traditional neural networks. Since the capsule can provide superior spectral features and spatial information extracted, its performance is better than the most advanced CNN in some fields. Motivated by this idea, a new remote sensing hyperspectral image classification algorithm called Conv-Caps is proposed to make full use of the advantages of both. We integrate spectral and spatial information into the proposed framework and combine Conv-Caps with Markov Random Field (MRF), which uses the graph cut expansion method to solve the classification task. The Caps-MRF method is further proposed. First, select an initial feature extractor, which a CNN without fully connected layers. Then, the initial recognition feature map is put into the newly designed CapsNet to obtain the probability map. Finally, the MRF model is used to calculate the subdivision labels. The presented method is trained with three real HSI datasets and is compared with the latest methods. We find the framework can produce competitive classification performance.

INDEX TERMS Capsule network, deep learning, hyperspectral image classification, Markov random fields.

I. INTRODUCTION

With technology advancing, various types of high-resolution Earth surface images become readily available [1]. Hyperspectral images contain abundant spectral and spatial information [2]. They often consist of hundreds of spectral bands, and are used in mineral exploration, soil testing and environmental monitoring. Hyperspectral image classification aims at automatically assigning a specific semantic label to each pixel according to its spatial-spectral information [3]. In recent years, several classification methods have been developed for HSI classification tasks. New classification methods are mostly based on low-order features or standard features, such as Support Vector Machine (SVM) [4] and Sparse Representation (SR) [5]. However, challenges these methods face are that the marked training samples available are minimal, leading to the Hughes phenomenon and the ignorance of essential features.

Recently, Deep Learning (DL) [6], [7] has become a very active topic in computer vision including image classification [8], data dimensionality reduction [9] and semantic segmentation [10]. Deep learning models can acquire more powerful, abstract, and differentiated functions through deep structure neural networks. Especially, CNN [11] has a strong ability to extract features using the hidden layers in the network, making it more suitable for hyperspectral image classification, and have achieved state-of-the-art results [12]. Unfortunately, the fully connected layers in the CNN model compress the two-dimensional feature map into a one-dimensional feature map, ignoring the information related to the spatial domain, which make the classification algorithm difficult to converge and seriously reduces the classification accuracy [13].
To solve the problems above, we proposed Conv-Caps, which combine the CNN and the CapsNet, with spectral-spatial features to process hyperspectral image classification tasks. With CNN’s powerful capabilities of learning features and the property of equivariance of CapsNet, Conv-Caps has an advanced performance. Otherwise, CapsNet has been successfully applied to many fields, such as tumors classification [14], sound event detection [15], and remote sensing image classification [16]. The main idea of the CapsNet is that vector capsules are utilized to represent internal attributes, classification [16]. The main idea of the CapsNet is that vector capsules are utilized to represent internal attributes, and replace the neuron in the traditional neural network with a set of neurons as a capsule to solve the problem of spatial hierarchies between features effectively. It is useful to identify small sample data to overcome the disadvantage of CNN via the dynamic routing process and vectorize the result. Sabour et al. [17] state that CapsNet exhibits superior performance when dealing with spatial level related issues. However, Luo et al. [18] propose that it is not ideal to apply CapsNet to the HSI data directly. Deng et al. [19] compared the performance of the CapsNet and CNN model under the condition of a similar network structure and found that the capsule network is better than the CNN model in many aspects.

The problem encountered by these methods is that they rarely consider either spectral or spatial features. None of the above models formulate HSI classification into the Bayesian framework that simultaneously considers deep learning models and MRF. To further improve the classification performance, we introduce Markov Random Field into the model as a post-processing method of the Conv-Caps classification map. In the image segmentation task, MRF can encourage adjacent pixels to have the same class label, and it can adequately characterize spatial information via modeling adjacent labels in hyperspectral images. Researches show that MRF can significantly improve the classification accuracy in HSI classification tasks. Based on this, we propose the Caps-MRF model. Specifically, first, the input HSI is deblocked, and the patch is used as the feature input for each pixel of the hyperspectral image. Then, CNN without fully connected layers is used as the initial feature extractor, and a deep network classifier is constructed using CapsNet. Finally, with the classifier’s results, the MRF is introduced to perform smoothing post-processing in the airspace to obtain the final classification output. Experiment results show that our method is more suitable for HSI classification. In summary, the merits of our paper are as follows:

1) To further improve the classification precision, we proposed a novel architecture called Conv-Caps. The proposed method improves CapsNet and enhances the internal connection of the capsule, overcoming the problem of overfitting in classification, and can effectively deal with the classification of hyperspectral scenes.

2) Based on Conv-Caps, we introduced the new classification method called Caps-MRF. The proposed method can take full advantage of spectral-spatial properties, combined the CapsNet and MRF, and explained the problem of hyperspectral classification from the perspective of the Bayesian framework. The result is superior to the current state-of-the-art methods on three challenging data sets.

3) We study and analyze the impact of different factors in the proposed architecture to the classification results, such as the patch size, capsule dimension, and network depth. Also, further analysis is carried out to improve the robustness and generalization of the proposed method by enhancing noise and changing the training scale. These can provide valuable guidance for the application of capsule networks in hyperspectral image classification in the future.

The rest of this article is organized as follows. In section II, we describe the background formulations of the capsule and MRF. In Section III, the design procedure is discussed, and the proposed structure is described. In section IV, the experiments on three data sets are illustrated. Finally, the paper ends with section V where we summarized our work.

II. BACKGROUND FORMULATION

A. CapsNets

CapsNet is a recently proposed neural network that may cause a profound impact on deep learning, especially in computer vision. In traditional CNN, a neuron’s input and output are scalars. Different from it, CapsNet’s neurons process the vector. Therefore, Capsule is also called Vector Neuron (VN), and all critical information about the state of features in capsule detection will be encapsulated in the form of a vector.

In this article, CapsNet splits the hyperspectral image into vector capsules \( I_i \) through a convolutional step. Each \( I_i \) is multiplied by a corresponding weight matrix \( W_i \). Finally, a digit capsule \( v_j \) is created by a dynamic routing algorithm. Vector capsule \( v_j \) represents a category of the hyperspectral image, and the modulus of the vector represents the probability of the category. Compared with CNN, capsule networks have some disadvantages, such as being unsuitable for large databases and slower execution. To overcome these shortcomings, Paszke et al. [13] propose a new CNN architecture based on spectral-spatial capsule networks while significantly reduce the network design complexity. Wang et al. [20] expand classification frameworks for HSI based on CapsNet, which introduces the affine transformation matrix. Yin et al. [21] tune a new CapsNet architecture with three convolutional layers and achieve superior performance in HSI classification to the CNN-based methods. Wang et al. [22] resolve the problem that high resolution may increase intra-class difference and inter-class similarity with Caps-TripleGAN.

B. Markov Random Fields

Due to the spatiality of the hyperspectral image, MRF is widely used in the field of hyperspectral applications to model the spatiality [23]. The point of the MRF clustering algorithm is randomly walking in a dense cluster.
Before leaving the cluster, random traversal in dense clusters accesses many nodes. Then in the probability matrix, each value is expanded by a power, which makes it possible to strengthen the fixed points and weaken the loose points.

Let the HSI dataset be \( X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{h \times w \times d} \), where \( h \) is the height, \( w \) is the weight, and \( d \) is the number of spectral bands. \( n = hw \) represents the total number of extracted patches. The set of class labels is defined as \( L = \{1, 2, \ldots, K\} \), \( K \) is the number of classes for the given HSI dataset. Consider \( Y = \{y_1, y_2, \ldots, y_n\} \) as image of class labels, \( Y' = \{y'_1, y'_2, \ldots, y'_n\} \) as the initial probabilistic label prediction for each data point. In the Bayesian framework, the estimation of labels \( Y \) for observations \( Y' \) can be inferred by maximizing a distribution \( P(Y|Y') \):

\[
\hat{Y} = \arg \max_{Y \in L^n} \log P(Y|Y') = \arg \max_{Y \in L^n} \log P(Y'|Y) + \log P(Y),
\]

(1)

where \( P(Y'|Y) \) and \( P(Y) \) represents the likelihood function and the prior probability of category labeling, respectively. The MAP segmentation \( \hat{Y} \) is computed via the \( \alpha \)-Expansion algorithm. The details of the calculation will be given in the next section [24].

III. PROPOSED METHOD

In this section, we will introduce the design principles of network architecture and CapsNet. Then we describe how to compute the segmentation results with an efficient min-cut optimization technique.

A. IMPROVED CAPSULE-BASED CLASSIFIER

Inspired by CNN, we added the CNN architecture to the capsule network, replacing the original fully connected layer with the proposed Conv-Caps. For the problem that CNN cannot consider the spatial hierarchy between elements, CapsNet’s solution is to encode spatial information and calculate the existence probability of the object. In particular, a capsule is a vector containing multiple neurons, each representing various properties of a specific entity that appears in the image. These attributes include many different types of instantiation parameters and help to extract the features of the image, thereby making up for a large amount of information that CNN lost when outputting, effectively preserves the spatial-spectral details of the elements in the HSI data cube.

The proposed architecture is shown in Fig 1, which consists of two dual CNNs, two pooling layers, and a capsule network. Also, the dynamic routing process of the capsule network has been improved.

We use the HSI patch as the input of CapsNet. The first convolution layer uses 100 kernels of size \( 3 \times 3 \) to filter the sample block, the stride is equal to 1, and 100 feature maps are generated. Then, the combined set of features of size \( 9 \times 9 \) will be inputted into the max-pooling layer with kernels of \( 2 \times 2 \). The size of the feature map after pooling is \( 3 \times 3 \). The maps pass through a convolutional layer with 300 kernels of \( 3 \times 3 \), which filters the feature maps into a new feature map of size \( 3 \times 3 \) and enters the next max-pooling layer with the kernel of \( 2 \times 2 \). Now, a data block of size 9 is generated. The next layer is the main capsule layer, the kernel size is \( 3 \times 3 \), the stride is equal to 1, the number of capsules in the output layer is the same as the number of types, and the dimension is 16. And the current capsule is output to which higher-level capsule by dynamic routing.

The output capsule will be calculated by a nonlinear squashing function as follows:

\[
v_j = \text{Squash}(s_j) = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \cdot \frac{s_j}{\| s_j \|},
\]

(2)
where \( v_j \) represents the output of capsule \( i \) after the squashing transformation function, \( s_i \) denotes the total input of capsule \( i \). In this article, we replaced \( \text{Squash}(s_i) \) with \( (s_i)/||s_i|| \) during the iteration, and only squashed it at the end of the output. Improve formula (2) to formula (3).

\[
\text{Squash}(s_i) = \frac{||s_i||}{1 + ||s_i||}
\]

(3)

To be specific, the total input \( s_i \) of capsule \( i \) is obtained by the weighted sum of all the capsules’ output in the previous layer, which is formulated as:

\[
s_i = \sum_i c_{ij}U_{ij},
\]

\[
U_{ij} = W_{ij}I_i,
\]

(4)

where \( I_i \) is the vector of the \( i \)-th input, to ensure \( I_i \) is aligned with the upper-layer capsule \( x_i \), we use the transformation matrix \( W_{ij} \) (the weight of the \( i \)-th vector). \( c_{ij} \) is the routing weight; it represents the contributions of the previous capsules to the upper-layer capsules and is determined by routing softmax. The formula is shown as follows:

\[
c_{ij} = \frac{\exp(b_{ij})}{\sum_i \exp(b_{ik})},
\]

(5)

where \( b_{ij} \) is the log prior probability between the \( i \)-th capsule and the \( j \)-th capsule. \( b_{ij} \) is used to reflect the agreement between \( U_{ij} \) and \( v_j \), and the closer the values of \( U_{ij} \) and \( v_j \) are, the higher the correlation between the two capsules. In order for capsule to be coupled to capsule \( j \) with equal probability, \( b_{ij} \) is initialized to zero. With iterations, repetitively updates \( b_{ij} \) as \( b_{ij} \leftarrow b_{ij} + U_{ij}v_j \), to make \( b_{ij} \) more accurate. Finally, the output layer will replace each of the capsule with its module and perform argument-max (argmax) function to gain the pixel’s label \( Y' \).

In the HSI classification task, the probability of local spectral features existence is represented by the norm of the vector. The separate margin loss \( L_C \) for each capsule \( C \) can be given as:

\[
L_C = T_C \max (0, m^+ - ||\vec{v}_c||)^2 + \lambda (1 - T_C) \max (0, ||\vec{v}_c|| - m^-)^2.
\]

(6)

The total loss function of CapsNet is a weighted sum of margin loss. For an output capsule of \( T_C = 1 \), the margin loss is 0 when the output vector is higher than \( m^+ \), or else it is not 0. Correspondingly, for the output capsule of \( T_C = 0 \), when the output vector is smaller than \( m^- \), the margin loss is 0. Otherwise, it is not 0.

B. PRIORITY OF SMOOTHNESS

After obtaining the probabilistic label prediction \( Y' \) in ConvCaps, we should solve the optimization problem. On this basis, we proposed Caps-MRF. Use smoothness prior to promoting piecewise smooth segmentations. Assume that the conditional class is independent of pixels, and the optimization problem can be simplified as follows:

\[
\hat{Y} = \arg \max_{Y \in L^k} \log P (Y|Y') = \arg \max_{Y \in L^k} \sum_{i=1}^n \log P (y_i'|y_i) + \log P (Y).
\]

(7)

The prior method belonging to the MRF class will smooth the segmentation result by assigning neighboring pixels into the same category. According to the Hammersly–Clifford theorem, the density associated with MRF is the Gibbs’ distribution. Therefore, the previous segmentation model has the following structure:

\[
\hat{Y} = \arg \max_{Y \in L^k} \sum_{i=1}^n \sum_{K=1}^K 1 \{Y_i = k\} \log y_{i'k} + \log \frac{1}{Z} \exp \left( \sum_{i=1}^n \sum_{j \in N(i)} \delta(y_i - y_j) \right),
\]

(8)

where \( Z \) is the normalization constant of density, tunable parameter \( \mu \) controls the smoothness, \( 1 \{\cdot\} \) is an indicator function. \( N(i) \) is the adjacent pixel of pixel \( i \), and \( \delta(y) \) is a unit pulse function, defined as: \( \delta(0) = 1 \) and \( \delta(y) = -1 \) for \( y \neq 0 \). Paired terms \( \delta(y_i - y_j) \) have a higher probability than equal adjacent labels.

C. CAPS-MRF FRAMEWORK FOR HSI CLASSIFICATION

To get a good approximation, we use a series of minimum-cuts computation, map the problem to the suitable graphs. The following is the final classification model:

\[
\hat{Y} = \arg \max_{y \in k^n} \sum_{i=1}^n \sum_{K=1}^K 1 \{y_i = k\} \log Y_{i'k} + \mu \sum_{i=1}^n \sum_{j \in N(i)} \delta(y_i - y_j).
\]

(9)

This model can be regarded as a MRF model. The final step of our proposed method involves spatial context information. The label \( Y' \) represents a Markov Random Field, and the objective function is its energy function, with the first term representing the cost given to different categories of pixels. The probability that a pixel is assigned with a corresponding label and this label depends on different categories of pixels. The second term encourages the labels of neighboring pixels to be close. Thus, it can be approximated by an algorithm based on the \( \alpha \)-expansion [24] graph cut. As a result, a good approximation of \( \hat{Y} \) is produced, which is reasonable and effective from a computational point of view and considering the accuracy of the model used. The detail of the proposed method is to summarize the procedure in Algorithm 1.

IV. EXPERIMENTAL RESULT

A. DATA DESCRIPTION AND EXPERIMENTAL SETTINGS

Three public real hyperspectral datasets (see Figure 2, 3, and 4) with different spatial resolutions are used in our
Algorithm 1 Caps-MRF Hyperspectral Image Segmentation

**Input:** HSI patches $X$, smoothness parameter $\mu$, learning rate $\alpha$.

**Output:** output label $\hat{Y}$.

1: Train the Conv-Caps classifier using training samples;
2: Compute probabilistic map $Y'$ by Conv-Caps;
3: Compute the classification labels $\hat{Y}$ using $\alpha$-Expansion ($Y', \mu$).

Experiments to evaluate the proposed method. First is the AVIRIS Indian Pines Data dataset. It contains $145 \times 145$ pixels and 202 spectral bands. There are 16 mutually exclusive ground-truth classes, a total of 10366 mark pixels. Next is the AVIRIS Salinas A Data. It comprises 83 pixels in the Salinas scene, and includes six classes (see Table 2). After the 20 water absorption bands were removed, the available data set was composed of 204 bands. Finally, it is the Pavia Centre (PC). This scene is provided by Prof. Paolo Gamba from the Telecommunications and Remote Sensing Laboratory, Pavia University (Italy). Pavia Centre is a $1096 \times 715$ pixels image. The number of spectral bands is 102 and the geometric resolution is 1.3 meters. There are nine classes included in the ground truth map, and the number of samples for each class is given in Table 3. Different dataset choices can reflect the robustness of the method and are important factors in determining the final outputs and accuracy.

The sampling strategy is one of the important factors that affect the classification results of models, especially when using a limited number of known samples. In this article, the available labeled samples are divided into three subsets, namely training samples, verification samples, and test samples (see Figure 5, 6 and 7). Each different category had the same color as the basic fact category. Randomly select the 10% labeled samples of each data set as the training set to train, and use the performance evaluation of the 10% verification samples to design our network architecture. These samples are also randomly selected. The labeled samples remaining after training is used as a test set to evaluate the network’s capabilities and obtain the final classification results. All methods are numerically compared using the following three classification performance metrics: overall accuracy (OA), average accuracy (AA), and kappa coefficient ($\kappa$). Specifically, OA represents the ratio of category pixels of the correct classification to the total number of categories; AA represents the average accuracy of each category; and $\kappa$ relates to the ratio of errors that are reduced by classification and complete random classification. The larger the value of these three conditions, the better the classification effect.

The hardware is an NVIDIA RTX 2080Ti graphical processing unit (GPU) with 11GiB memory and dual Intel Xeon E5-2678 v3 with 64GiB memory. Regarding our software environment, it is composed of Ubuntu 18.04 x64 as an operating system, NumPy 1.16.4 and pandas 0.24.2, TensorFlow 1.13.1 framework, and Python 3.5 as the programming language, while others are run in Matlab R2014b.
TABLE 1. Classification accuracy (%), OA (%), AA (%) and κ coefficient for Indian Pines data set.

| Class          | Train | Test | Total | SVM       | 1D-CNN   | SS-CNN   | 3D-CNN   | SS-Caps | P-RN | DenseNet | Conv-Caps | Caps-MRF |
|----------------|-------|------|-------|-----------|----------|----------|----------|----------|------|----------|-----------|----------|
| Alfalfa        | 45    | 46   | 91    | 75.76±0.11 | 45.62±1.16 | 87.06±1.90 | 78.14±5.72 | 86.62±0.78 | 86.88±2.18 | 87.25±2.22 | 87.42±3.09 | 87.93±4.43 |
| Corn-no till   | 143   | 148  | 291   | 83.47±5.78 | 90.03±0.75 | 96.01±0.95 | 99.04±0.37 | 98.11±1.71 | 98.74±0.13 | 97.87±0.90 | 98.49±0.99 |
| Corn-min till  | 83    | 80   | 163   | 25.16±4.75 | 35.49±4.66 | 97.39±0.75 | 84.43±1.41 | 96.97±0.25 | 96.26±1.47 | 98.63±0.93 | 99.64±0.78 | 97.94±0.29 |
| Corn           | 24    | 23   | 47    | 68.17±0.42 | 87.53±1.05 | 93.80±1.03 | 93.66±1.16 | 96.75±2.10 | 96.74±1.31 | 98.02±0.97 | 97.97±0.02 | 98.85±0.04 |
| Grass-pasture  | 49    | 48   | 97    | 46.54±1.76 | 63.77±1.34 | 93.64±0.22 | 94.95±0.08 | 92.83±0.42 | 98.70±0.24 | 98.84±0.15 | 99.26±0.06 | 99.89±0.04 |
| Grass-trees    | 73    | 70   | 143   | 80.77±0.74 | 85.69±1.07 | 98.62±0.04 | 88.97±1.08 | 98.19±0.02 | 98.51±0.19 | 98.66±0.07 | 98.13±0.08 | 98.16±0.07 |
| Grass-pasture-mowed | 3   | 25   | 28    | 35.97±1.22 | 83.24±1.47 | 93.42±0.34 | 89.91±1.54 | 94.72±0.68 | 95.80±1.84 | 95.61±1.29 | 91.72±1.52 | 95.57±2.03 |

Hay-windrowed   | 48    | 43   | 91    | 76.43±2.67 | 83.27±0.68 | 96.53±0.79 | 95.14±0.20 | 99.08±0.21 | 98.64±0.73 | 99.41±0.22 | 99.58±0.02 | 100±0.00 |
| Oats           | 2     | 18   | 20    | 69.27±3.01 | 67.08±0.85 | 91.04±0.98 | 96.36±1.15 | 97.92±0.13 | 99.43±0.07 | 99.19±0.41 | 99.52±0.06 | 99.89±0.10 |
| Soybean-no till| 98    | 97   | 195   | 94.83±2.46 | 80.67±2.17 | 94.92±0.99 | 94.14±1.47 | 99.59±0.13 | 99.09±0.32 | 99.46±0.21 | 99.01±0.90 | 99.84±0.08 |
| Soybean-min till| 246  | 245  | 491   | 42.50±3.45 | 76.17±1.24 | 97.62±0.66 | 98.58±0.89 | 98.54±0.08 | 98.79±1.67 | 97.70±0.22 | 98.56±0.04 | 98.94±0.09 |
| Soybean-clean  | 60    | 53   | 113   | 13.74±3.32 | 89.22±2.71 | 90.88±2.15 | 95.93±1.48 | 99.31±0.30 | 98.70±0.48 | 98.65±0.39 | 99.11±0.81 | 99.87±0.05 |
| Wheat          | 21    | 18   | 39    | 93.15±0.50 | 76.05±4.46 | 96.15±1.79 | 94.15±1.34 | 99.57±0.28 | 99.39±0.16 | 99.65±0.17 | 99.86±0.22 | 100±0.00 |
| Woods          | 127   | 118  | 245   | 94.61±1.05 | 75.50±2.43 | 83.34±0.50 | 98.64±1.26 | 97.86±1.55 | 96.53±1.45 | 98.34±0.69 | 97.90±0.88 | 98.54±0.29 |
| Buildings-Grass-Trees-Drives | 39  | 38   | 77    | 46.40±0.73 | 83.65±5.07 | 97.59±0.81 | 95.87±0.61 | 98.88±0.75 | 98.24±0.84 | 98.30±1.49 | 99.06±0.82 | 99.17±0.01 |
| Stone-Steel-Towers | 10  | 8    | 18    | 0±0.00   | 90.42±5.13 | 97.82±0.93 | 98.74±0.03 | 98.38±0.42 | 97.65±2.93 | 98.56±1.25 | 98.43±1.08 | 99.15±0.19 |

B. PARAMETER ANALYSIS

In this experiment, some parameters need to be set in advance. For the Conv-Caps method, through the test comparison, we set the network depth to 6. The first convolutional layer has a kernel size of 3 and a number of 100. The second convolutional layer has a kernel size of 3. And the patch size set to 9, strides are set to 1. The parameters $m^+$, $m^-$, and $\lambda$ in the loss function were set to 0.9, 0.1, and 0.5, respectively. For the proposed Caps-MRF, the smoothness parameter $\mu$ is set to 5.0. The training epoch was set to 100 with a learning rate of 0.001. In addition to these parameters above, several other important parameters may also affect the classification performance. In this section, we tested three parameters, including patch size, the effect of different network depths, and...
TABLE 3. Classification accuracy (%), OA (%), AA (%) and $\kappa$ coefficient for Pavia Centre data set.

| Class                | Train | Test | Total | SVM  | 1D-CNN | SS-CNN | 3D-CNN | SS-Caps | P-RN | DenseNet | Conv-Caps | Caps-MRF |
|----------------------|-------|------|-------|------|--------|--------|--------|---------|------|----------|-----------|----------|
| Water                | 82    | 742  | 824   | 77.88±9.21 | 87.70±1.20 | 96.54±0.21 | 98.41±1.83 | 98.51±1.04 | 97.86±1.50 | 98.11±0.62 | 98.64±0.11 | 99.97±0.01 |
| Trees till           | 82    | 738  | 820   | 78.35±5.65 | 88.56±3.64 | 97.21±0.05 | 97.05±1.23 | 99.11±0.54 | 98.84±0.96 | 98.56±0.64 | 98.70±0.10 | 99.37±0.02 |
| Asphalt till         | 82    | 734  | 816   | 56.78±7.43 | 89.39±0.39 | 98.28±0.04 | 98.58±1.12 | 99.43±0.49 | 99.36±0.26 | 99.07±0.28 | 99.15±0.08 | 99.71±0.03 |
| Self-Blocking Bricks | 81    | 727  | 808   | 78.35±0.82 | 90.55±4.23 | 98.35±0.05 | 98.15±0.38 | 98.89±0.33 | 98.61±0.66 | 99.31±0.13 | 99.01±0.08 | 99.74±0.05 |
| Bitumen              | 81    | 727  | 808   | 81.45±7.13 | 89.47±0.25 | 97.68±0.51 | 97.33±1.06 | 98.51±0.53 | 98.87±1.04 | 99.45±0.23 | 99.19±0.07 | 99.57±0.32 |
| Tiles                | 126   | 1134 | 1260  | 77.60±3.61 | 71.27±11.43 | 97.24±0.05 | 97.09±1.76 | 98.90±0.09 | 98.85±1.11 | 98.58±0.79 | 99.11±0.12 | 99.51±0.28 |
| Shadows              | 48    | 428  | 476   | 82.06±1.66 | 88.84±0.03 | 97.79±0.16 | 97.15±2.55 | 98.78±0.04 | 98.37±0.99 | 97.70±1.84 | 98.49±0.41 | 99.56±0.06 |
| Meadows              | 82    | 742  | 824   | 77.10±0.09 | 89.71±2.14 | 98.00±3.71 | 98.93±0.74 | 98.89±0.68 | 98.13±1.74 | 98.05±1.66 | 98.51±0.98 | 99.28±0.51 |
| Bare Soil            | 82    | 738  | 820   | 79.26±2.80 | 90.05±0.07 | 97.67±1.47 | 98.48±0.67 | 99.64±0.20 | 98.76±0.91 | 98.59±0.08 | 99.52±0.03 | 99.73±0.04 |

OA(%) | 78.30±1.69 | 87.06±1.43 | 97.28±0.96 | 98.16±1.21 | 98.75±0.13 | 98.64±0.21 | 99.14±0.27 | 99.11±0.06 | 99.84±0.03 | 97.00±0.00 | 96.50±0.00 | 97.50±0.00 |

AA(%) | 76.83±7.86 | 84.93±1.70 | 95.83±2.18 | 93.17±3.27 | 93.81±0.20 | 93.96±0.71 | 93.36±0.97 | 92.66±2.10 | 94.65±3.78 | 95.00±0.00 | 95.50±0.00 | 96.00±0.00 |

$\kappa (\times 100)$ | 66.85±3.45 | 85.91±1.69 | 96.24±1.27 | 95.60±0.54 | 94.93±0.25 | 94.54±0.61 | 94.07±1.92 | 94.67±0.32 | 95.02±0.13 | 95.50±0.00 | 96.00±0.00 | 96.50±0.00 |

FIGURE 4. PC datasets: (a) pseudo-color image of the PC dataset; (b) ground truth samples of the PC dataset.

and the size of capsules in CapsNet, to evaluate the impact of parameter changes on classification performance. In the following experiments, we use Indian Pines datasets and uniformly select 10% of the training samples when changing other parameters. Other data are used to test.

1) ANALYSIS OF PATCH SIZE
The size of the hyperspectral patch determines how many features the model can capture, and therefore affects the classification results—the larger the patch, the better the discriminative features. Nevertheless, when the size becomes larger to a certain extent, the classification performance may deteriorate. In this section, we will fix other parameters and test the effect of different patch sizes $k = \{7, 9, 11, 13\}$ on classification performance.

It can be seen from Figure 8 that for the IP data set, OA first improves as the patch size increases, but then becomes worse. As we can see, the more tremendous sample patch results are better because it can contain more spatial information to optimize performance. However, too large a sample patch will also bring other problems, such as too much calculation time. For a proper trade-off, we choose the patch size of 9 as the default setting.

FIGURE 5. Spatial distribution of training, validation and test samples for the IP dataset: (a) training samples for fitting CNN and CAPs, as well as SVM classifiers; (b) validation samples for evaluating SVM, CNN, and CAPs; and (c) test samples for predicting SVM, CNN, and CAPs.

FIGURE 6. Spatial distribution of training, validation and test samples for the SA dataset: (a) training samples for fitting CNN and CAPs, as well as SVM classifiers; (b) validation samples for evaluating SVM, CNN, and CAPs; and (c) test samples for predicting SVM, CNN, and CAPs.
2) EFFECT OF DIFFERENT NETWORK DEPTHS

In the framework of Caps-MRF, different layer depths will significantly affect the training effect. We determine the most suitable network depth by increasing or decreasing non-linear layers. According to the information given in the literature, the deeper level of deep learning model, the worse of HSI classification results [31]. So we train and test the proposed method on five networks with depth 5, 7, 9, and 11.

Figure 9 shows the experimental results. As can be seen from the table, our experiment verifies that increasing network depth does not necessarily lead to better classification results. May be due to the gradient vanishing problem in the deeper neural network.

We set the network depth to 7 for the best performance. In other experiments, it is used as the default setting for network depth.

3) THE DIMENSION OF THE CAPSULE

One of the critical components of CapsNet is the capsule. In the proposed model, the activity of neurons in the capsule represents various attributes of remote sensing scene images. There are two layers of capsules, PrimaryCaps and DigiCaps, which are lower-level capsules representing small entities and higher-level capsules representing more complex entities, respectively. Considering the size of the capsules in CapsNet, the smaller size of the capsules will weaken the capsules’ performance, leading to confusion about similar scene categories in the context of the image. High-capsules may contain more noise or extra information [32]. Too high or too low will hurt the classification results, so we set a set of values ((6,12), (8,16), (10,20), (12,24)) to study. The effect of capsule size on the model. The experimental results are shown in Figure 10. It can be seen that the value (8, 16) has the best performance, so we use it for the proposed model.

C. CLASSIFICATION RESULT

In this section, we report the classification results of the proposed method and other classical methods. Select the parameters as described above, and Table 1-Table 3 shows the classification results of the Indian Pines, SalinasA, and PaviaU datasets using different methods. It can be seen that for IP datasets, Caps-MRF has the highest OA, AA, and K. At the same time, the proposed Conv-Caps are 0.20%
higher than SS-CNN in OA. For SA datasets, OA of Caps-MRF is better, which indicates that the superiority of our method lies in less sample and higher accuracy. For PC data sets, MRF optimization improves Conv-Caps’ OA by 0.73%. Compared with other methods, the classification accuracy of the method based on the capsule network is higher.

Finally, we evaluated the accuracy of classification from a visual perspective. As shown in Figure 11, 12, and 13, we can find out how different classification methods affect the classification results. It can be seen from the picture that the classification error is mainly caused by some fragment structure or intra-class variation. The classification graph of each model or classifier of the SA dataset is very satisfactory. Because the SA dataset is a relatively simple data set, all models have obtained fairly good results and accuracy, including the SVM classifier.

In addition, the comparison of Figure 11h and Figure 11i proves that the use of MRF can significantly improve classification accuracy. Compared with the results of other methods, the performance of 3D-CNN is second only to
Caps-MRF. The main reason is that it involves joint spectral-spatial features, which is better than the single spectral or spatial features.

For the PC dataset, by comparing the real ground reference with the classification map, we found that the classification results produced by the SVM model are very noisy. The reason may be that only the spectral information contained in the HSI data is considered. And yet, SS-Caps and Caps-MRF obtain a smoother appearance in the visualization results by combining spectral and spatial features, the results further illustrate the advantages of the spectral-spatial method. At last, it can be concluded that the proposed method enhances the generalization ability of the network. We can study spectral-spatial features by more effectively considering the spatial position, spectral features, and possible transformations of spectral-spatial features. Caps-Net can use higher levels of abstraction to characterize HSI data. In order to define the class boundaries more accurately, the basic information of the corresponding spectral space characteristics and feature data conversion is supplemented as a set of instantiated parameters.

We also compared the number of parameters and FLOPs of several different methods. Taking the IP data set as an example, the result is shown in Figure 14. The abscissa represents the parameter of networks, the ordinate represents accuracy, and the radius of the circle represents FLOPs. It can be seen that compared with other networks, our method obtains better results with fewer parameters.

**D. FURTHER EXPLANATION**

In order to further evaluate the robustness and generalization ability of the proposed method, the classification results
obtained by the proposed method are compared with the results under different training set sizes, and the noise has been added to study the adaptability of the method. The IP data set is still used as an instance, with the same parameters as shown above.

1) DISCUSSION OF TRAINING SIZE
One of the critical factors in determining the performance of a model or classifier is the number of training samples. Without a large number of training samples available, deep learning models may be difficult to extract effective features. For HSI data, there are not enough training samples to choose from, so it is crucial to determine the number of training samples about HSI classification.

As shown in Figure 15, we randomly selected 5%, 10%, 15%, and 20% from the Indian Pine dataset as training samples. Use SVM, 1D-CNN, 3D-CNN, SS-Caps, and the proposed method. Compared with 3D-CNN and SS-Caps, the OA obtained by Caps-MRF increased by 2.75% and 0.44%, respectively. By increasing the training sample size, the classification accuracy will also increase until it reaches a stable level. The above experimental results further prove the superiority of the method under the small training sample size.

2) DISCUSSION OF NOISE LEVEL
Noise robustness experiments were conducted using noise-contaminated HSI, and the classification performance of the four classical methods used in the previous section and the proposed method at different noise levels had been compared. Taking the IP data set as an example, a group of Gaussian noises with a mean of 0 and different variances ($\sigma$) are added for classification experiments. The value of $\sigma$ is (0.15, 0.3, 0.45). The parameter settings of all methods are the same as above. It can be seen from the figure that the increase in noise has an impact on the classification performance of all methods. Among them, SVM is the most sensitive to noise and has the fastest performance degradation, possibly because it can only extract shallow features. However, the method using the spectral-spatial characteristics shows excellent classification performance.

According to Figure 16, compared with other methods, the method based on the capsule network shows superior performance under different noise levels. As the noise variance $\sigma$ increases, the OA of Caps-MRF decreases from 98.87% to 97.69%, which is much better than other methods. Experiments also show that even under boisterous conditions, our Caps-MRF still performs well.

3) USE SPATIALLY DISJOINT SAMPLES
Since the random abstract strategy will affect the reliability of the classifier results [33], in order to accurately measure the true generalization ability of the model, we use a strictly spatially separated training and test set for experiments. Also take the IP data set (available from the GRSS DASE website (http://dase.grss-ieee.org) as an example, using the method proposed in the previous section for comparison. As shown in Figure 17, it can be seen that our method has more
advantages than other similar methods. At the same time, there are huge performance differences between different methods.

The reason why the performance of the spectral space method is significantly affected may be due to the high intra-class similarity and inter-class variability of HSI, and the presence of some noise. That is, the spatial correlation between the training and testing sets will reduce the quality of the conclusions obtained by the spectral space method.

V. CONCLUSION

In this article, a new, improved capsule network called the Conv-Caps was investigated for HSI classification. Furthermore, based on Conv-Caps, the MRF framework was introduced, and a new method called Caps-MRF was presented. Correctly, Caps-MRF can better use spectral and spectral-spatial features in HSI data, and MRF helps maintain labelling consistency in smaller neighbourhoods. Experimental results on three HSIs (Indian Pines, Pavia Center and Salinas A) illustrate the proposed framework has a better performance than traditional methods, especially in the case of a limited number of training samples.

The proposed methods explored the potential of the capsule network for hyperspectral classification. The equivariance of the capsule network effectively guarantees the retention of valid information, and the MRF spatial restriction helps achieve more accurate model convergence. In future work, we will further consider improving CapsNet in combination with other post-classification methods. Since the capsule can better extract spectral features, it will play a significant role in HSI image processing.

REFERENCES

[1] 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.
[2] J. M. Bioncas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. 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] Q. Wang, S. Liu, J. Chanussot, and X. Li, “Scene classification with recurrent attention of VHR remote sensing images,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 2, pp. 1155–1167, Feb. 2019.
[4] 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.
[5] Y. Chen, N. M. Nasrabadi, and T. D. Tran, “Hyperspectral image classification using dictionary-based sparse representation,” IEEE Trans. Geosci. Remote Sens., vol. 49, no. 10, pp. 3973–3985, Oct. 2011.
[6] 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.
[7] H. Gao, Y. Yang, D. Yao, and C. Li, “Hyperspectral image classification with pre-activation residual attention network,” IEEE Access, vol. 7, pp. 176587–176599, 2019.
[8] L. Zhang, Z. Chen, J. Wang, and Z. Huang, “Rocket image classification based on deep convolutional neural network,” in Proc. 10th Int. Conf. Comput. Commun., Circuits Syst. (ICCCAS), Dec. 2018, pp. 383–386.
[9] Z. Zhou, J. Mo, and Y. Shi, “Data imputation and dimensionality reduction using deep learning in industrial data,” in Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC), Dec. 2017, pp. 2329–2333.
[10] S. Turan and G. Bilgin, “Semantic nuclei segmentation with deep learning on breast pathology images,” in Proc. Sci. Meeting Elect.-Electron. Biomed. Eng. Comput. Sci. (EBBT), Apr. 2019, pp. 1–4.
[11] 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.
[12] Q. Wang, X. He, and X. Li, “Locality and structure regularized low rank representation for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 2, pp. 911–923, Feb. 2019.
[13] M. E. Paololetti, J. Mario Hau, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, and F. Pla, “Capsule networks for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 4, pp. 2145–2160, Apr. 2019.
[14] P. Afshar, K. N. Plataniotis, and A. Mohammad, “Capsule networks for brain tumor classification based on MRI images and coarse tumor boundaries,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2019, pp. 1368–1372.
[15] F. Vesperini, L. Gabrielli, E. Principi, and S. Squartini, “Polyphonic sound event detection by using capsule neural network,” IEEE J. Sel. Topics Audio, Lang. Process. (JSTALP), Jan. 2020, pp. 1–11.
[16] T. Tian, X. Liu, and L. Wang, “Remote sensing scene classification based on res-capsnet,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Jul. 2019, pp. 525–528.
[17] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 3856–3866.
[18] Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, and G. Bai, “HSI-CNN: A novel convolution neural network for hyperspectral image,” in Proc. Int. Conf. Audio, Lang. Image Process. (ICALIP), Jul. 2018, pp. 464–469.
[19] F. Deng, S. Pu, X. Chen, Y. Shi, T. Yuan, and S. Pu, “Hyperspectral image classification with capsule network using limited training samples,” Sensors, vol. 18, no. 9, p. 3153, Sep. 2018.
[20] W.-Y. Wang, H.-C. Li, L. Pan, G. Yang, and Q. Du, “Hyperspectral image classification based on capsule network,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Jul. 2018, pp. 3571–3574.
[21] J. Yin, S. Li, H. Zha, and X. Luo, “Hyperspectral image classification using CapsNet with well-initialized shallow layers,” IEEE Geosci. Remote Sens. Lett., vol. 16, no. 7, pp. 1095–1099, Jul. 2019.
[22] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, “Caps- TripleGAN: GAN-assisted CapsNet for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 9, pp. 7232–7245, Sep. 2019.
[23] M. Khodadadzadeh, R. Rajabi, and H. Ghassemian, “Combination of region-based and pixel-based hyperspectral image classification using erosion technique and MRF model,” in Proc. 18th Iranian Conf. Electr. Eng., May 2010, pp. 294–299.
[24] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[25] 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.
[26] A. Boulch, N. Audebert, and D. Dubuqu, “Autoencoder for the visualisation of images hyperspectrales,” in Proc. 25th Colloque Gretsi, 2017.
[27] B. Liu, X. Yu, F. 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.
[28] 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.
[29] M. Paololetti, J. Hau, R. Fernandez-Beltran, J. Plaza, A. Plaza, and F. Pla, “Deep pyramidal residual networks for spectral–spatial hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 2, pp. 740–754, Feb. 2019.
[30] M. Paololetti, J. Hau, J. Plaza, and A. Plaza, “Deep&Dense convolutional neural network for hyperspectral image classification,” Remote Sens., vol. 10, no. 9, p. 1454, Sep. 2018.
[31] 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.
[32] S. Li, X. Zhu, Y. Liu, and J. Bao, “Adaptive spatial-spectral feature learning for hyperspectral image classification,” IEEE Access, vol. 7, pp. 61534–61547, 2019.

[33] 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.

XUEFENG JIANG is currently with the School of Computer Science, Northwestern Polytechnical University, Xi’an, Shaanxi, China. His current research interests include computer vision and e-learning.

YUE ZHANG is currently pursuing the degree with the Software School, Northwestern Polytechnical University, Xi’an, Shanxi, China. She was a Silver-Medalist of one separate seasons of the International Collegiate Programming Contest. Her research interests include deep learning, remote sensing image processing and analysis, and hyperspectral image classification.

WENBO LIU is currently pursuing the degree with the Computer School, Northwestern Polytechnical University, Xi’an, Shanxi, China. He was a Bronze-Medalist and a Silver-Medalist of two separate seasons of the International Collegiate Programming Contest. His current research interests include hyperspectral image(HSI) segmentation and HSI band selection.

JUNYU GAO (Member, IEEE) received the B.E. degree in computer science and technology from the Northwestern Polytechnical University, Xi’an, Shaanxi, China, in 2015, where he is currently pursuing the Ph.D. degree with the Center for Optical Imagery Analysis and Learning. His research interests include computer vision and pattern recognition.

JUNRUI LIU is currently with the School of Computer Science, Northwestern Polytechnical University, Xi’an, China. Her current research interests include computer vision and software engineering.

YANNING ZHANG (Senior Member, IEEE) is currently with the School of Computer Science, Northwestern Polytechnical University, Xi’an, Shaanxi, China. Her current research interests include computer vision and software engineering.

JIANZHE LIN is currently with the Department of Electrical and Computer Engineering, The University of British Columbia.

* * *